Peer-Reviewed Technical Communication

A Sensor Web Architecture for Integrating Smart Oceanographic Sensors Into the Semantic Sensor Web

Joaquín del Río, *Member, IEEE*, Daniel Mihai Toma, Enoc Martínez, Thomas C. O'Reilly, Eric Delory,
Jay S. Pearlman, *Life Fellow, IEEE*, Christoph Waldmann, and Simon Jirka

***Abstract*—Effective ocean and coastal data management are needed to manage marine ecosystem health. Past ocean and coastal data management systems were often very specific to a particular application and region, but this focused approach often lacks real-time data and sharing/interoperating capability. The challenge for the ocean observing community is to devise standards and practices that enable integration of data from sensors across devices, manufacturers, users, and domains to enable new types of applications and services that facilitate much more comprehensive understanding and analyses of marine ecosystem. A given kind of sensor may be deployed on various platforms such as floats, gliders or moorings, and thus must be integrated with different operation, and data management systems. Simplifying the integration process in existing or newly established observing systems would benefit system operators and is important for the broader application of diverse sensors. This paper describes a geospatial "sensor web" architecture developed by the "NeXOS" project for ocean and coastal data management, based on the concepts of spatial data infrastructure and the Sensor Web Enablement framework of the Open Geospatial Consortium. This approach reduces the effort to propagate data from deployed sensors to users. To support the realization of the proposed Next generation Ocean Sensors (NeXOS) architecture, hardware and software specifications for a Smart Electronic Interface for Sensors and Instruments (SEISI) are described. SEISI specifies small lower-power electronics, minimal operating system, and standards-basedresearch software to enable web-based sharing, discovery, exchange, and processing of sensor observations as well as operation of sensor devices. An experimental scenario is presented in which sensor data from a low-power glider with low-bandwidth intermittent satellite communications is integrated into the geospatial sensor web using the NeXOS architecture.**

***Index Terms*—Interoperability, metadata, Sensor Model Language (sensorML), standards.**

NOMENCLATURE

XML	XML is a markup language that defines a set of rules for encoding documents in a format that is both human readable and machine readable.
EXI	Binary XML format for exchange of data on a computer network.
FP7	Seventh Framework Programme, European union research and development funding program.
GEOSS	GEOSS is being built by the Group on Earth Observations on the basis of a 10-Year Implementation Plan running from 2005 to 2015.
Interoperability	A characteristic of a system, whose interfaces are completely understood, to work with other systems, present or future, in either implementation or access, without any restrictions.
JSON	A light-weight data interchange format.
Metadata	Data that provides information about other data.
NeXOS	FP7 project lead by plocan.eu.
OGC	A standards organization for geospatial information systems.
PUCK	This standard defines a protocol for RS232 and Ethernet connected sensors. PUCK addresses installation and configuration challenges for sensors by defining a standard sensor protocol to store and automatically retrieve metadata and other information from the sensor device itself.
SEISI	SEISI is a set of software and hardware components defining an interoperable architecture to propagate data from sensors, through platforms to the web.
SensorML	SensorML provides standard models and an XML encoding for describing sensors and measurement processes. It can be used to describe a wide range of sensors, including both dynamic and stationary platforms and both *in situ* and remote sensors.
Sensor Alert Service (SAS)/SES	The SES is an enhancement of the SAS. Both are used to provide and publish/subscribe based access to sensor data and measurements.

Manuscript received July 23, 2017; accepted October 24, 2017. Date of publication November 27, 2017; date of current version October 11, 2018. This work was supported in part by the NeXOS Project. NeXOS is a collaborative project funded by the European Commission Seventh Framework Programme, under the call OCEAN-2013.2—The Ocean of Tomorrow 2013—Innovative multifunctional sensors for *in situ* monitoring of marine environment and related maritime activities, under Grant 614102, and it is composed of 21 partners including SMEs, companies, and scientific organizations from six European countries; visit: www.nexosproject.eu. This work was also supported in part by the Fixed Point Open Ocean Observatory (FixO3) Network Project from the European Union's Seventh Programme for research, technological development, and demonstration under Grant 312463. FixO3 seeks to integrate European open ocean fixed point observatories and to improve access to these key installations for the broader community. The proposal has 29 partners drawn from academia, research institutions, and SMEs; visit: http://www.fixo3.eu/. *(Corresponding author: Joaquín del Río.)*

Guest Editor: O. Zielinski.

J. del Río, D. M. Toma, and E. Martínez are with the Department Electronics, SARTI Research Group, Universitat Politècnica de Catalunya, Vilanova i la Geltru 08800, Spain (e-mail: joaquin.del.rio@upc.edu; daniel.mihai.toma@upc.edu).

T. C. O'Reilly is with the Monterey Bay Aquarium Research Institute, Road Moss Landing, CA 95039-9644 USA (e-mail: oreilly@mbari.org).

E. Delory is with the M Oceanic Platform of the Canary Islands, Telde 35200, Spain (e-mail: eric.delory@plocan.eu).

J. S. Pearlman is with the IEEE France Section, Paris 75116, France (e-mail: jay.pearlman@jandfent.com).

C. Waldmann is with the MARUM—Center for Marine Environmental Sciences, University of Bremen, Bremen 28359, Germany (e-mail: waldmann@marum.de).

S. Jirka is with the 52°North Initiative for Geospatial Open Source Software GmbH, Muenster 48155, Germany (e-mail: jirka@52north.org).

Digital Object Identifier 10.1109/JOE.2017.2768178

SOS	This standard defines a web service interface, which allows querying observations, sensor metadata, as well as representations of observed features.
SPS	The OpenGIS SPS Interface Standard defines interfaces for queries that provide information about the capabilities of a sensor and how to task the sensor
SQL	It is a special-purpose programming language designed for managing data held in a relational database management system.
SWE	The OGC's SWE standards enable developers to make all types of sensors, transducers, and sensor data repositories discoverable, accessible, and useable via the web.

I. INTRODUCTION

SENSORS collect data for oceanographic research and societal benefit. They should be able to seamlessly contribute data to large Earth observing initiatives (regional and global) that make data discoverable and accessible to large and diverse user communities. Such initiatives include The Copernicus Marine Environment Monitoring Service (CMEMS), the Global Earth Observation System of Systems (GEOSS), and the Web of Things, to name a few. However, there are many systems and sensor types deployed in practice that rely on diverse communication protocols, where each protocol defines many communication layers from the physical/hardware interface up to command and data formats. Connecting disparate devices into a network typically requires specialized "driver" software that can translate these protocols between the individual sensors and the platforms on which they are installed [1]. The platforms typically require extensive manual configuration to match the driver software and other operational details of each network port to a specific sensor. Some data protocols, such NMEA, can facilitate such configuration, but in most cases a specific configuration still has to be done manually.

Oceanographic sensors are usually developed by small companies, and lack standard protocol for configuration, operation, data acquisition, and data formats. RS 232 and RS 485 serial communications are the dominant physical layer protocols (although increasingly displaced by Ethernet), but in general each manufacturer defines distinct syntax and command sets as well as data formats for the sensors it produces [1], [4]. These sensors are often integrated into an observing system or sensor network, which provides a software infrastructure for functions such as data acquisition, data logging, and data transfer via hard-wired or wireless telemetry links. Driver software must be written, and must be properly configured when the sensor has been physically installed into a communication port on the observing system. Thus sensor integration can be a time consuming, expensive, and challenging task. Moreover an efficient data access approach for users is needed. As a comprehensive solution, the development of an architectural concept and implementation to integrate sensors and their data into the Internet is a core objective of the Next generation Ocean Sensors (NeXOS) project funded by the European Seventh Framework Programme. This paper introduces the NeXOS sensor web architecture (see Fig. 1) as an example of how interoperability standards help to facilitate infrastructure for sharing oceanographic observation data, and the integration of sensor data into applications [4].

Sensor Web Enablement (SWE) standards specify how users interact with sensors and their data through interfaces and formats defined and maintained by the Open Geospatial Consortium (OGC). The user needs just a web browser to access data from a sensor web. There is no need for custom operating system-dependent applications to access raw sensor data. The way to access and manage sensor data is succinctly described in a set of standards that have been developed by the community under the OGC umbrella. In the following sections, more details about the standards used in our proposal will be described.

Standardizing the installation process of a new sensor helps to reduce operating costs of the observatory. In some cases, a sensor must be integrated with marine observation platforms such as oceanographic gliders or buoys at sea under extreme environmental conditions. Standardizing and streamlining installation and operating processes can dramatically reduce costs, as well as the risk of failures due to manual errors [5]. Standard protocols and formats also facilitate interoperability, maintenance, and replacement of observatory sensors and maintain traceability of the data they generate [2]. A key idea of our proposal is that the information about the sensor resides physically "inside" the sensor, i.e., sensor metadata is stored within the device itself, and marine platforms can interrogate the sensor to download this information to identify, configure, and operate the device. Our proposed system uses OGC PUCK protocol as a standard way to access this metadata, and more details about it are found below.

PUCK and other OGC SWE standards [6] form the basis of the NeXOS sensor web architecture are introduced in this paper. The remainder of this paper is structured as follows. In Section II, the motivation and requirements underlying the sensor web architecture are described. In Section III, the main principles and components of the proposed architecture are presented. In Sections IV, V, and VI, description of how this architecture has been implemented, deployed, and validated. Finally, in Section VII, a conclusion and an outlook are provided.

II. MOTIVATION AND REQUIREMENT ANALYSIS

At present, much oceanographic and coastal data are of limited value as they are locked in proprietary systems and in "vertical" device- and usage-specific applications. A great opportunity lies in being able to create "horizontal" independent applications and services that use sensor data aggregations across devices and domains. Such applications and services can focus on the needs and interests of users, such as data from devices belonging to a particular user or of particular geographic/temporal interest to the user. Although some observing systems fulfill some of these goals, they could be made more efficient and lower cost through use of standard protocols.

A basic requirement of such systems is the capability to discover sensors of interest in a user's proximity, e.g., identifying nearby clusters of hydrophones in a coastal area and visualizing marine mammal detections to the user. To fulfill this requirement, web-enabled ocean sensor systems must be designed to support data interoperability and at least some level of commonality in data taxonomies, ontologies, naming, and metadata assignment and processing.

For efficiency, a system should work with a minimal set of interoperable data and metadata formats that work across devices and domains, allowing applications to discover and access sensors of interest, interpret their observations, and trigger sensors or actuators where appropriate. The metadata should provide basics and context such as calibration history, sensor ID/name, command set for configuration and control, engineering units, time synchronization, location, calibration methods, maintenance periods, and other parameters, all of which should be automatically accessible independent of any human interaction. Regardless of implementation, clients need to be able to query sensors to obtain real-time observations and to query historical sensor values, e.g., by name and time. The sensors should be easily integrated into observing systems, which call for standards for connectors, power supplies, data formats, protocols, and handling procedures. For remote observing systems, sensors should be able to check themselves autonomously for

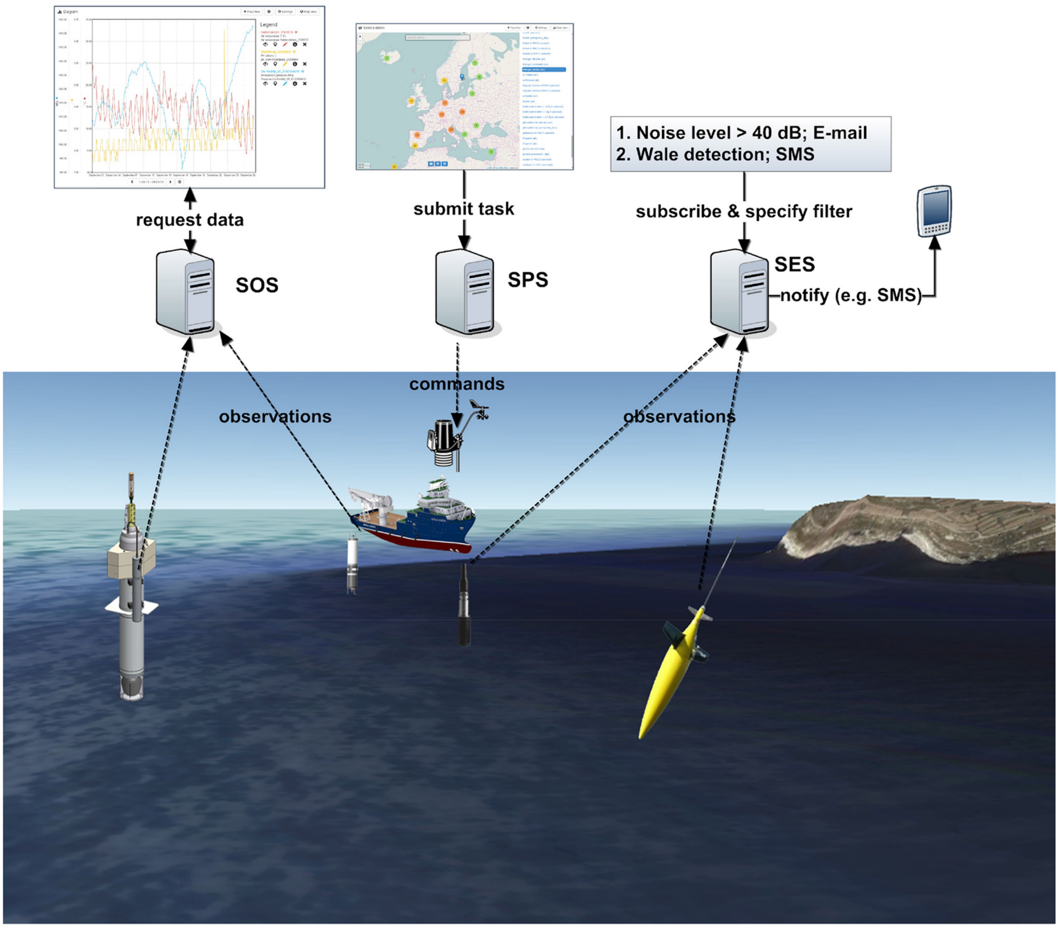


Fig. 1. Sensor web architecture for the NeXOS project. Data coming from sensors are available on the web. Data are propagated from sensors to the web using standard protocols such as OGC SOS. Users also interact with sensors and receive event notifications through standardized mechanisms.

possible malfunctions to report back to the operations center. These are exactly the objectives of the NeXOS sensor web architecture presented in this paper. For the design of the sensor web architecture several functional requirements were identified.

1) "Pull" access to observation data by a client, i.e., following a request–response pattern.
2) Push delivery of observation data to clients: data are injected without a request.
3) Visualization of the collected observation data for all sensors, accounting for sensor specificities.
4) Automatic conversion of sensor readings into appropriate format to enable data access via the web.

In addition to these sensor and data access needs, the following requirements help to improve cost effectiveness.

1) Reusability: The components and implementations of the sensor web architecture shall be as generic as possible and shall follow international standards. Thus, data providers shall be able to reuse the resulting architecture and software in multiple application contexts beyond this project.
2) Interoperability: Through the use of international standards, the integration of interoperable sensor data into applications shall require less effort. As soon as new sensor data sets are available in the sensor web infrastructure, all clients compatible with these standards will be able to access the data immediately. When needed, standard security protocols can be also applied to sensor actuation and data access.
3) Open Source: For each component of the sensor web infrastructure at least one open source implementation shall be provided. This will allow data providers to rely on free implementations. Furthermore, the open source license of the developed components will ensure that users of the software are not bound to a single vendor.

Many of the components needed to build such an architecture are available; the question is how to define a convincing demonstration case, where all aspects that might come up in other application cases are covered. At first, this is a very complex problem and it is tempting to stray into cumbersome premature generalizations that may hinder adoption and implementation. Probably the best and the fastest way forward is to first define a minimal usable subset of guidelines and specifications for the ocean and coastal observing community, with room for refinement as experience in building and operating those systems is gained. Therefore for NeXOS we propose a framework that builds on the concepts and prototypes developed for the OGC Architecture Implementation Pilot [7] and the OGC Ocean Science Interoperability Experiment (http://www.opengeospatial.org/projects/initiatives/oceansie).

III. NEXOS SENSOR WEB ARCHITECTURE

The vision of providing oceanographic and coastal data for geospatial sensor web services is not new and there are many initiatives working in that direction. Examples of sensor web initiatives with the goal to make oceanographic and coastal information available for smart appli-

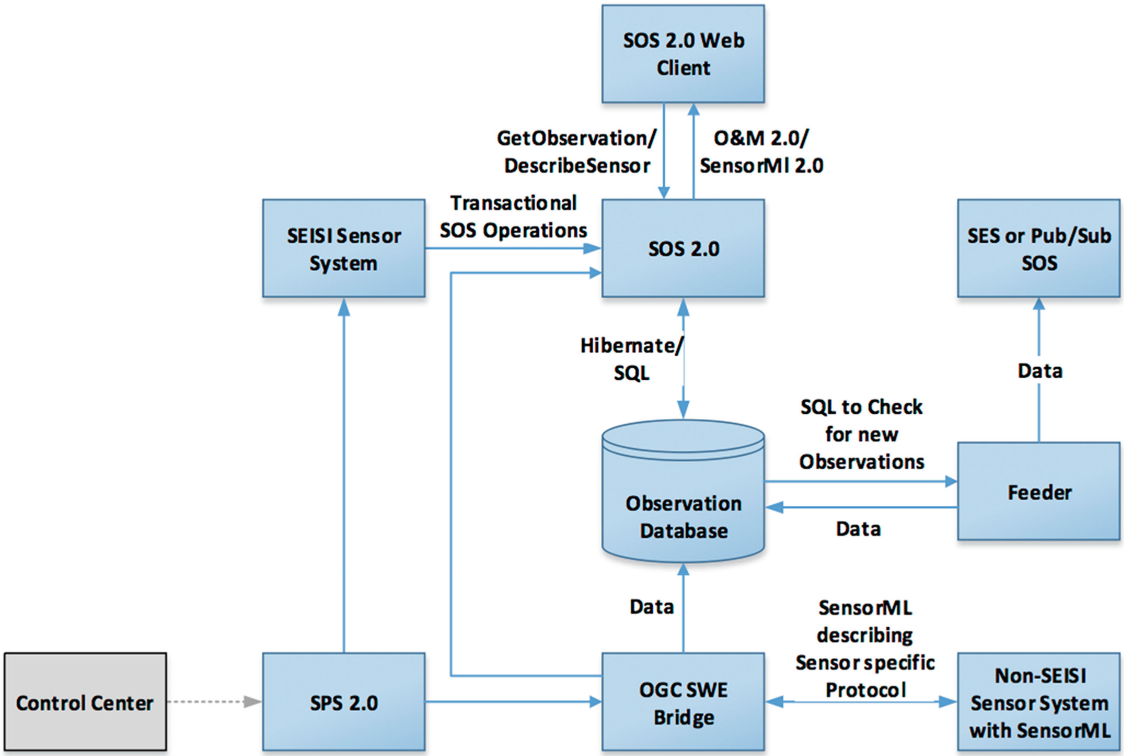


Fig. 2. Overview of the components for the Ocean and Coastal Sensor Web Architecture.

cations and services have and are being developed in different international projects such as: CMEMS, the GEOSS, and the Web of Things, to name a few. Examples of projects that include the oceanographic and coastal monitoring domain include SCHeMA (http://www.schema-ocean.eu), SenseOCEAN (http://www.senseocean.eu), EmodNET, SeaDataNet, IMOS, IOOS, and the OGC working group to extend the current SWE standards toward the Internet of Things (www.opengeospatial.org/projects/groups/sweiotswg). In the NeXOS architecture presented here there are three new main components: 1) the Smart Electronic Interface for Sensors and Instruments (SEISI hardware and firmware), which facilitates the integration of oceanographic sensors on various platforms such as floats, gliders cabled observatories, vessels of opportunity or moorings; 2) OGC SWE Bridge, which executes on a deployed platform to provide data to a shore-based OGC Sensor Observation Service (SOS); and 3) several web service implementations, based primarily on OGC-SWE specifications.

As shown in Fig. 2, these interfaces use the OGC SOS for accessing the measured data, the OGC Sensor Planning Service (SPS) for controlling sensor parameters, and an approach for subscribing to push-based sensor data streams.

A detailed description of the architecture shown in Fig. 2 follows. The starting point is always a system composed of a sensor or a set of sensors connected to a platform. As shown in the figure, the sensor can be either a "SEISI system" or "non-SEISI system." In either case, the sensor is accompanied by a Sensor Model Language (SensorML) file that describes the sensor.

A "SEISI system" indicates a PUCK enabled sensor that can transfer its data through Transactional SOS (T-SOS) Operations (such as InsertSensor or InsertObservation) directly to an SOS server if a connection to the Internet is available. The T-SOS operations can be executed by the sensor itself (if it is designed with this capability), or through a specific electronic board called SEISI Hardware. These transactional operations are described in more detail in Fig. 7, and the objectives are to declare a new sensor and populate sensor data into the SOS server. On the other hand, a "non-SEISI system," refers to a PUCK enabled or non-PUCK enabled sensor unable to communicate directly with an SOS server. Instead the non-SEISI's data is first transferred to a software component called the OGC SWE Bridge, which then eventually relays the data to a T-SOS.

The user can discover, visualize, and retrieve data from the SOS with "SOS web client" software.

Secondary components such as the "SPS" module will allow the user to communicate with an Internet-connected SEISI directly, or through the "OGC SWE Bridge" with a non-SEISI system or with SEISIs not directly connected to the Internet. Also, thanks to a "feeder" software component a Sensor Event Service (SES) that will inform the user about events related to the data, such as when new data are available.

The "SEISI and non-SEISI systems" in Fig. 2 are software with associated physical sensors plus cables plus the host controller of the platform; the rest are software components that are executed on a server computer.

The fundamental functionality of the architecture is the interoperable access to sensor data. Thus, an essential component is a web service interface for pull-based access to observation data (i.e., following a request–response pattern). Within the NeXOS architecture, this interface is realized through the OGC SOS 2.0 standard [8]. This interface defines standard operations for requesting sensor measurements and metadata as well as for publishing newly acquired data. It relies on two further specifications: ISO/OGC Observations and Measurements (O&M) 2.0 [9] is used for modeling and encoding the measured observation data, and OGC SensorML 2.0 [10] standard is applied to the metadata associated with the observations and corresponding sensors. With regard to the SOS, special emphasis is put on the need for easy integration into existing IT infrastructures (e.g., configuration mechanisms that can be flexibly coupled with existing observation databases).

The SOS interface offers a typical request-response pattern for accessing observation data. This means that users/clients submit a request for a specific set of observation data (e.g., for specific parameters, locations, time periods, etc.) and the service returns this requested data set as a response. However, in addition, many users need to receive new observations as soon as they are measured. Such a delivery pattern cannot be efficiently implemented by the request-response pattern which would require continual polling by the user. Thus a web service for push delivery of observation data is required. For this purpose the OGC SES discussion paper [11] and the current activities of the OGC Publish/Subscribe Standards Working Group provide a valuable foundation. These specifications have been evaluated within this work to define a push-based data subscription/delivery service interface.

A further important functionality is the configuration and control of sensor systems. This includes simple settings such as the sampling rate of a temperature sensor but also complex tasks such as the track planning for an autonomous underwater glider sensor platform. While these configure and control operations involve diverse protocols, there is a need for a common to define and submit such commands. To address this need, the proposed NeXOS sensor web architecture includes a tasking component based on the OGC SPS 2.0 [12]. This interface allows users to determine which tasks can be executed by a sensor and which parameters can be changed, to manage and submit tasks as well as to request information on how the resulting data can be accessed from other services. To allow users to explore and visually analyze available observation data, this architecture also includes a data viewer for visualizing the collected observation data for all oceanographic sensors. This is achieved by an SOS 2.0 client capable of providing the user with an overview of the data available from SOS servers and visualizing these data sets as maps and graphs. To allow the adjustment to specific use cases and individual requirements, this client shall be designed to allow flexible customization. Finally, the transfer of observation data from the sensor into the sensor web infrastructure and the submission of commands from components such as the SPS to the devices is addressed. These operations rely on ISO/OGC O&M 2.0 [9] and the OGC SensorML 2.0 [10] standard.

After this overview of the core components of the NeXOS sensor web architecture, the following section provides some insight how this architecture concept is implemented.

IV. IMPLEMENTATION

During the first two years of the NeXOS project, the overall architecture as well as outlines of the component specifications were developed. Based on that design work a first iteration of prototypes has been conducted, resulting in several new or enhanced implementations. Of these implementations, four will be presented in this section in more detail: the SOS, a JavaScript sensor web client, the SEISI, and the OGC SWE Bridge. These components are available as open source software.

A. SOS Server

The NeXOS SOS implementation is based on the 52°North SOS 4.0 development [13]. To ensure easy integration of the SOS server into existing data management systems, the SOS has been implemented completely independent of a specific database management system or data model. To achieve this independency the 52°N Hibernate abstraction framework that links existing databases with an SOS server is used [http://hibernate.org/]. This layer hides specific aspects (e.g., Structured Query Language (SQL) "dialects") of different database management systems from SOS clients. This is achieved through mappings that describe which element of the core SOS data model corresponds to which table/column of a specific database. As a result, the abstraction layer acts as a translator between different database models. An overview of this approach to allow the flexible integration of database management systems as well as data models into the SOS implementation can be seen below in Fig. 3, where users access data through the SOS client, while sensor systems populate sensor data into the SOS server or into the database. As noted above, Hibernate software links data stored in existing databases with an SOS server, allowing an SOS client to discover and access such data. These three components (SOS client, SOS server with Hibernate, and the database) can operate on separate distributed machines. Normally, the SOS server and database will be executed on a server computer with an accessible URL over the Internet. The SOS client can be opened in any browser on any Internet connected computer to access data.

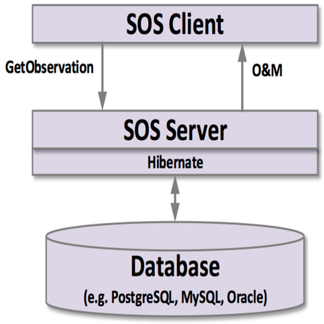


Fig. 3. Typical Deployment Scenario for SOS Servers.

B. SOS Client With Visualizer

To explore the data available through the SOS, it is necessary to provide the users with tools to discover and visualize the available observation data sets. For this purpose NeXOS uses the 52°North JavaScript client. Together with other projects (especially the FP7 project GEOWOW) and stakeholders from other domains (especially hydrology and air quality) a first version of this client was developed. Fig. 4 displays the client's abilities to visualize time series data.

C. Smart Electronic Interface for Sensors and Instruments

To ensure the easy integration of the NeXOS sensors into previously introduced SOS implementation, the SEISI has been implemented [14]. *SEISI* refers to specifications for a set of hardware and software components. As an example, an instrument (e.g., CTD) connected to a platform controller such as a buoy, vehicle, etc., is SEISI-compliant if it implements certain components that appear in Fig. 5. SEISI devices provide standard services for data access, data push, and sensor configuration based on the existing standard OGC SWE specifications for OGC SOS [8] and SPS [12]. SEISI devices may be deployed on platforms with a direct TCP/IP connection to the Internet, such as cabled observatories, ships or buoys with an Internet connection. A compliant SEISI device includes an RS232 serial interface as well as Ethernet, so a SEISI may also be deployed on platforms with limited-bandwidth intermittent non-TCP/IP links such as Iridium satellite. In the latter case, the NeXOS SWE Bridge component is also present on the platform, providing the connection between the at-sea sensors and Internet web services across the low-bandwidth link.

The links between at-sea sensors and Internet-based NeXOS sensor web components are shown in Fig. 5. These components are based on OGC SWE protocols to provide sensor detection, identification, configuration, and data acquisition.

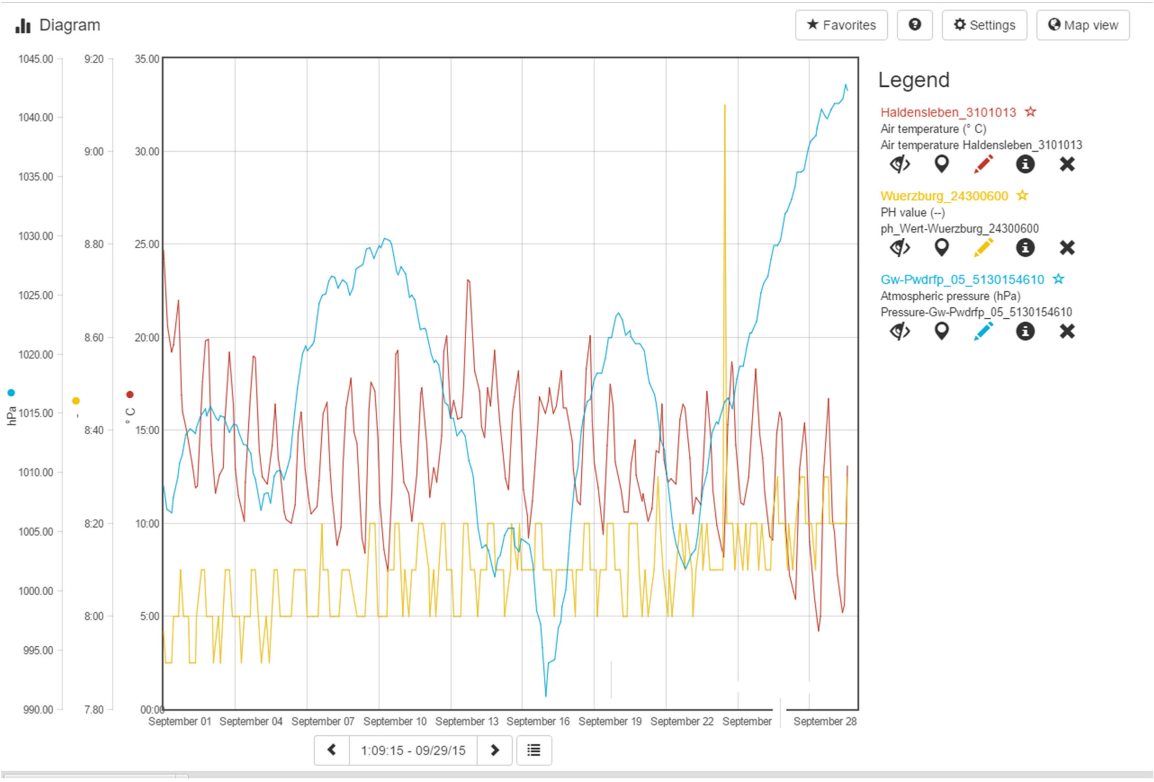


Fig. 4. Screenshot of sensor web client developed in the JavaScript language.

In the SEISI architecture shown in Fig. 5, we see four distinct OGC standard protocols namely: 1) SOS; 2) T-SOS; 3) SPS; and 4) PUCK protocol (which can be implemented across RS232 or Ethernet). In addition to these standards, a proprietary protocol service is implemented to cover any other sensor functionalities, which may not be fulfilled by the available standards. In the rest of this section, we specify in greater detail the functional services that enable interoperability among the different parts of the system.

1) SEISI Data Format: OGC SWE architecture sets specific data accessibility requirements. In architectures based on web services, data exchange is typically accompanied by a description of the transferred content by means of semantic representation languages, of which the eXtensible Markup Language (XML) is probably the most common. Nevertheless, the size of XML messages is often too large for the limited bandwidth of oceanographic platforms such as gliders or profilers. Furthermore, the text nature of XML representation complicates the parsing of messages by CPU-limited devices, compared to binary formats. For these reasons, the working group of the World Wide Web Consortium (W3C) [15] has proposed the Efficient XML Interchange (EXI) format, sometimes referred to as "binary XML" [16], which makes it possible even for very constrained devices to natively process and generate messages using an open data format compatible with XML. Therefore, we adopt EXI's schema-less encoding and processing for all OGC SOS operations of the SEISI system (i.e., *GetCapabilities, DescribeSensor, GetObservation, InsertSensor*, and *InsertResult*). Moreover, EXI is employed for SEISI metadata provided by the PUCK protocol [17] and encoded in SensorML 2.0. Further details about EXI and schema-less processing can be found in [18].

2) SEISI SOS Web Service: The SOS web service implemented for SEISI provides standard access to sensor data by clients when the device is connected to a TCP/IP network as illustrated in Fig. 6, and is sufficiently lightweight for deployment on CPU-limited devices. The lightweight SOS only implements the core operations of OGC SOS Specification v2.0 (i.e., *GetCapabilities, DescribeSensor*, and *GetObservation*).

The *GetCapabilities* operation provides access to metadata and detailed information about the available capabilities of the SOS. By using HTTP, GET, or POST request, the service capabilities can be retrieved from the SEISI encoded as an XML or EXI response. The capabilities response contains metadata about the service, such as information about the interface, the unique sensor identifiers, the observation offering of the SEISI and a list of one or more quantities observed by this offering.

The *DescribeSensor* operation provides the sensor metadata in SensorML format. The sensor description contains information about the sensor capabilities and characteristics, details such as calibration data and available communication protocols, interfaces, and data formats of the SEISI. The SensorML also contains a standardized description of the sensor's command protocol, i.e., it specifies commands that should be issued to configure and operate the device. In our NeXOS SEISI implementation, the SensorML is physically stored within SEISI sensors and retrieved through OGC PUCK protocol (see below).

The *GetObservation* operation provides access to the observations made by the sensors. On request the SOS returns the SEISI system observations expressed in OGC O&M standard format, using EXI binary encoding.

3) SEISI T-SOS Web Client: The T-SOS operations implemented for SEISI provides standard data push service as shown in Fig. 7, i.e., the SEISI can push its data and metadata to an external SOS. Like other NeXOS implementations, the T-SOS client is sufficiently lightweight to be deployed on CPU-limited devices. The T-SOS implements the *InsertSensor, InsertResultTemplate*, and *InsertResult* operations of OGC SOS Specification v2.0.

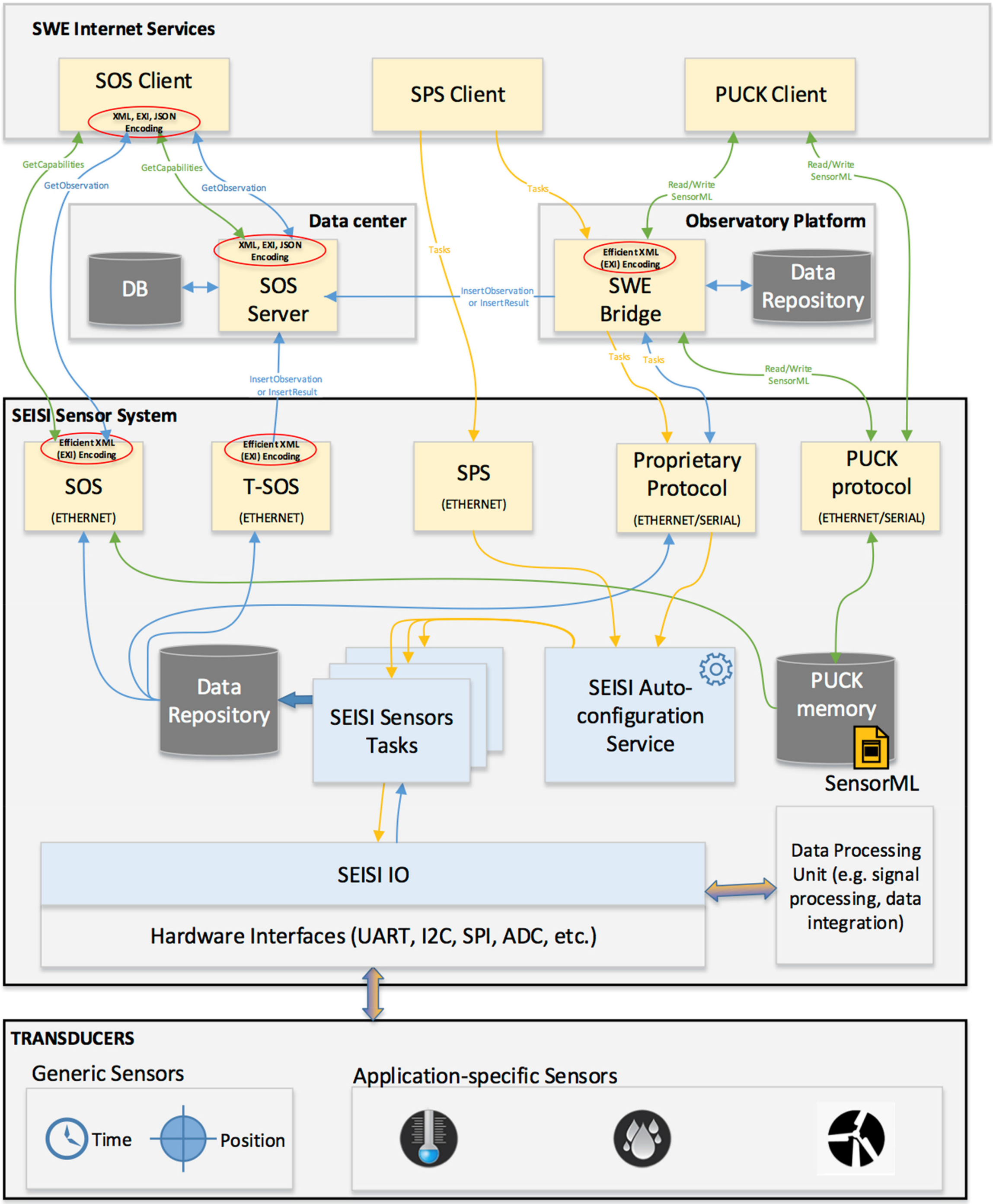


Fig. 5. SEISI system architecture and links with other components.

The *InsertSensor* operation publishes the SEISI description to an SOS server. The publish request can be sent by the SEISI encoded as either an XML or EXI request. The *InsertSensor* request contains metadata about the SEISI, such as information about the interface, the unique sensor identifiers, observation offerings of the SEISI, and a list of one or more quantities observed by this offering.

The *InsertResultTemplate* operation is used by the SEISI system to insert a "result template" (encoded as XML or EXI) to an SOS server; the template describes the structure of subsequent observations uploaded by SEISI to the SOS through *InsertResult* operations (below).

The SEISI uploads raw data to the SOS through the SOS *InsertResult* operation, passing the data in the structure specified by the previous *InsertResultTemplate* operation. The actual data values are expressed in OGC Observations and Measurement format and may be binary EXI encoded.

4) SEISI PUCK Protocol: OGC adopted the PUCK protocol in 2012 as a new member of the SWE framework of specifications [17]. Briefly, PUCK defines a set of standard commands that enable platform software to identify and retrieve metadata information about the sensor from the device itself.

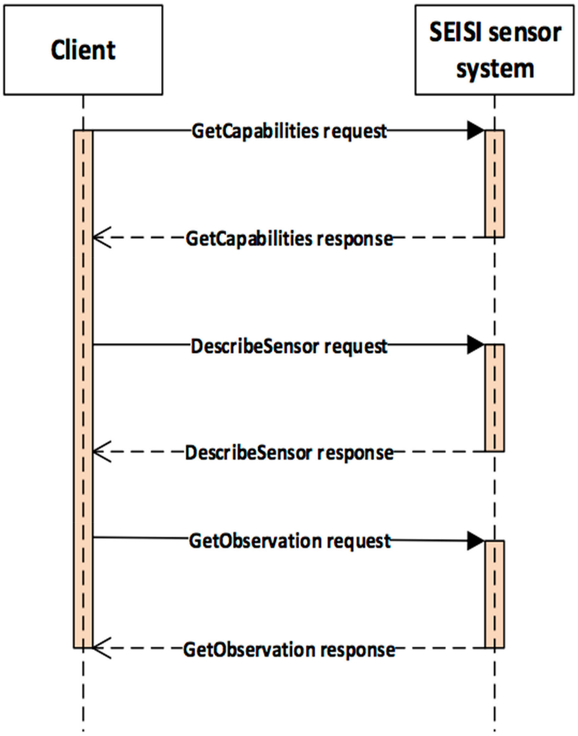


Fig. 6. SOS data access service procedure.

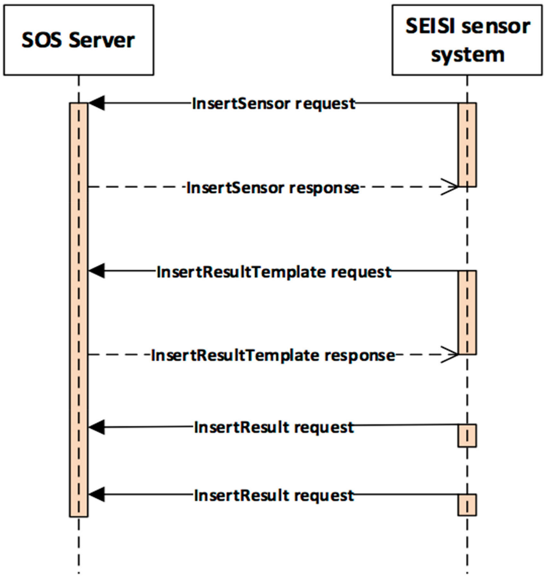


Fig. 7. T-SOS data push service procedure.

OGC PUCK does not itself fully implement interoperability, but rather provides the lower tier in the hierarchy of SWE standards that achieve this goal. OGC PUCK establishes a protocol to retrieve metadata directly from a sensor. Several manufacturers of the Smart Ocean Sensor Consortium in the USA have already implemented the PUCK protocol on their sensors. European NeXOS partners developing new sensors are implementing this protocol in their devices as part of the SEISI specification, including TRIOS (optical sensors), SMID (hydrophones), and NIVA (carbon cycle), while NeXOS platform manufacturers including Alseamar (SeaExplorer glider), TeledyneWebResearch (Slocum glider), CMR (Sail Buoy), and NKE (Provor float) use PUCK protocol on the platform side to retrieve metadata from attached PUCK-compliant sensors. PUCK provides a formatted electronic datasheet, which contains the information needed to identify the sensor model and manufacturer, as well as a universally unique identifier for each device. A PUCK-enabled sensor may also carry an additional payload that can include a SensorML document. PUCK protocol was originally defined for sensors with an RS232 interface, and the latest revision extends the protocol to Ethernet; this “IP PUCK” protocol uses the Zeroconf standard (http://www.zeroconf.org/) to enable easy installation and discovery of sensors in an IP network. SEISI systems use PUCK and its payload to enable sensor detection, identification, and configuration as illustrated in Fig. 8. The SensorML includes a description of the sensor’s manufacturer-specific command protocol and data format [19], i.e., it describes nonstandard protocols and formats in a standard way.

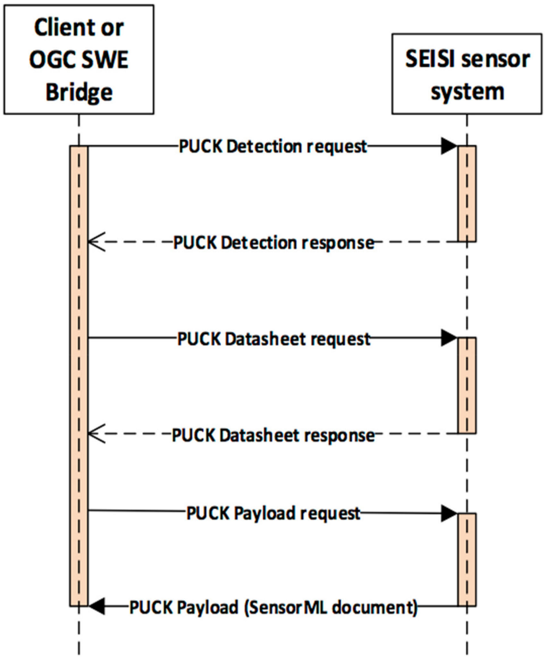


Fig. 8. PUCK sensor detection, identification, and configuration services procedure.

The platform host can automatically retrieve and use this sensor protocol descriptor from the device when it is installed, and thus can operate the sensor without *a priori* knowledge or sensor-specific “driver” software on the host. The generic SWE Bridge plays this role in the NeXOS architecture. This “plug-and-play” implementation significantly reduces the effort needed to integrate new sensors. Building on previous work [2], NeXOS specifies the standardized description of sensor protocols and has developed tools to support the creation of those descriptors [20]. Having this standardized language for the description of sensor protocols will facilitate the process of integrating sensors into marine observing systems, moving toward plug and play oceanic sensor systems [21].

Although sensors implementing PUCK protocol can store their metadata within the sensor, the NeXOS architecture is also compatible with sensors that do not implement PUCK. In such cases, the SensorML document with all the sensor metadata can be stored in a specific location on the platform file system and the SWE Bridge can be configured to read the sensor metadata from either a PUCK-enabled sensor or from the repository (for non-PUCK enabled sensors).

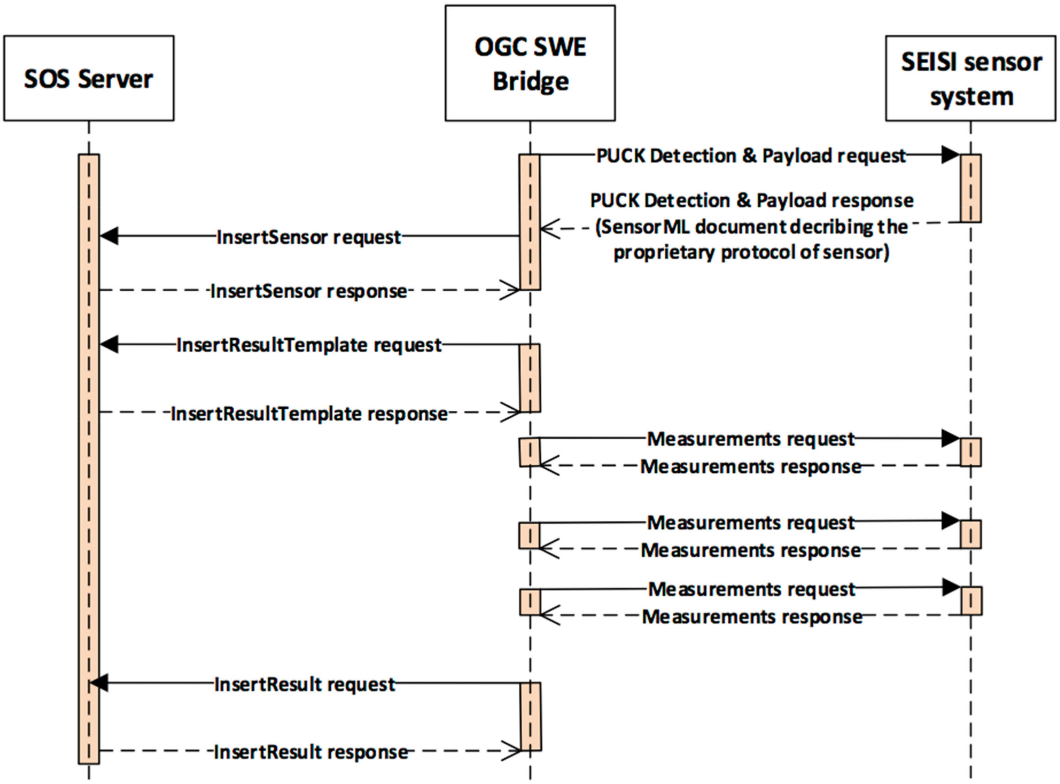


Fig. 9. Procedures of the OGC SWE Bridge.

5) SEISI System Proprietary Protocols: Although OGC SWE standards provide many building blocks for an ocean observing system, the SWE web service components assume TCP/IP links. In reality many ocean sensors are deployed on power-constrained host platforms that have intermittent low-bandwidth non-TCP/IP links to shore. Power is a precious commodity on gliders, profilers, AUVs, and other platforms, and they rely on satellite links such as Iridium short-burst messaging to communicate with shore. Therefore, NeXOS and SEISI utilize proprietary protocols as needed for sensor detection, identification, configuration, and data acquisition on these platforms. The SEISI has been implemented with a proprietary set of commands based on Standard Commands for Programmable Instrumentation (SCPI) [22], so even though the commands are "proprietary" they are standard-based. SCPI provides software-level syntax and commands for operating sensors over any transport, including Ethernet and RS232. Using the standardized description of the SCPI syntax and commands inside the SensorML document, the OGC SWE Bridge software component can automatically retrieve and use this sensor protocol descriptor when the device is installed on platforms such as gliders or profilers to operate the sensor.

6) OGC SWE Bridge: Underwater gliders and profilers are becoming important assets of ocean observing systems that provide sampling capability in regions where high spatial resolution is required, offering economical platforms for interdisciplinary ocean observations, However, most of these platform only support the RS232 physical/electrical interface for sensors, and low-bandwidth intermittent satellite communication links with shore-based control and data management systems. To host SEISI systems on these platforms, the sensor detection, identification, configuration, and measuring operations and services need to be sufficiently lightweight and adapted to platform interfaces and resource constraints. The OGC SWE Bridge is a NeXOS software component that executes on the platform, performing the following functions:

1) retrieves the sensor's SensorML from the device through PUCK protocol;
2) based on the sensor's protocol descriptor found in the SensorML configures the sensor and acquires data from it;
3) transfers the sensor metadata to shore through the low-bandwidth/intermittent link;
4) once on shore the metadata and data are transferred to an SOS.

V. DATA TRANSFER EVALUATION

While cable- and ship-based observing platforms may have nearly unlimited power available for communication links, the power resources available to autonomous mobile and underwater platforms and their communications links—acoustic or RF—are usually highly constrained. Thus very efficient data transfer mechanisms and formats must be utilized on these systems. We demonstrate that the EXI "binary XML" W3C standard is very efficient in several situations, through our analysis of the following specific use cases.

1) TCP/IP-connected observatory platforms, such as cabled observatories or "ferrybox" systems, e.g., integration of the smart hydrophone developed by the NeXOS Project [25].
2) Non-TCP/IP-connected observatory platforms, such as gliders and profilers, where we developed tests for the integration of a CTD sensor based on the SEISI systems.

A. *TCP/IP-Connected Observatory Platforms*

In TCP/IP-connected observatory platforms, the communication between the deployed sensor platform and upper layers such as the SOS Server and the SOS client occurs directly over TCP/IP. To evaluate the efficiency of the proposed implementation of the SOS and T-SOS in the smart hydrophone, we compare the size of response/request

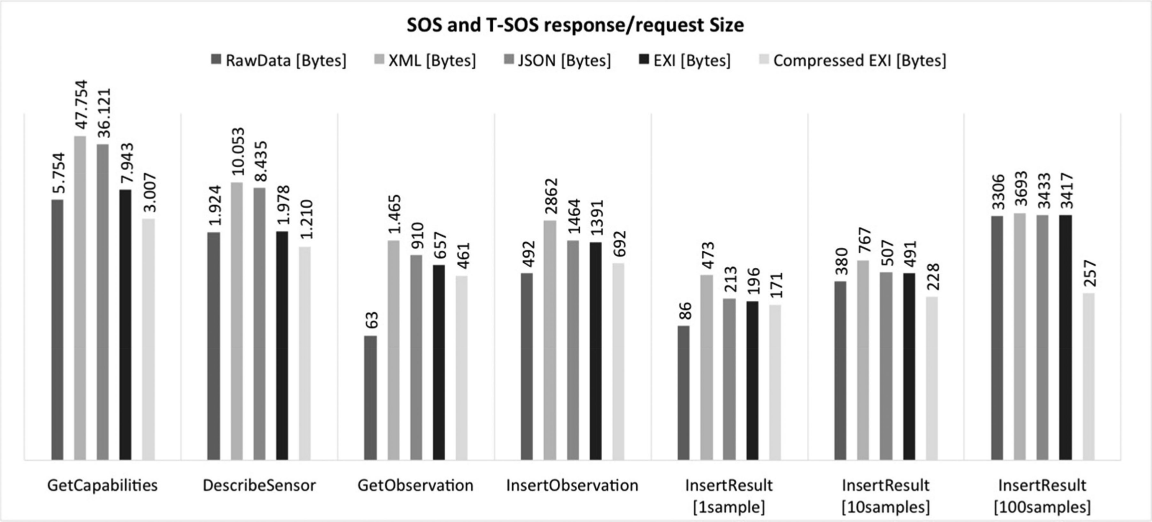


Fig. 10. Response/request size evaluation for the SOS and T-SOS operations.

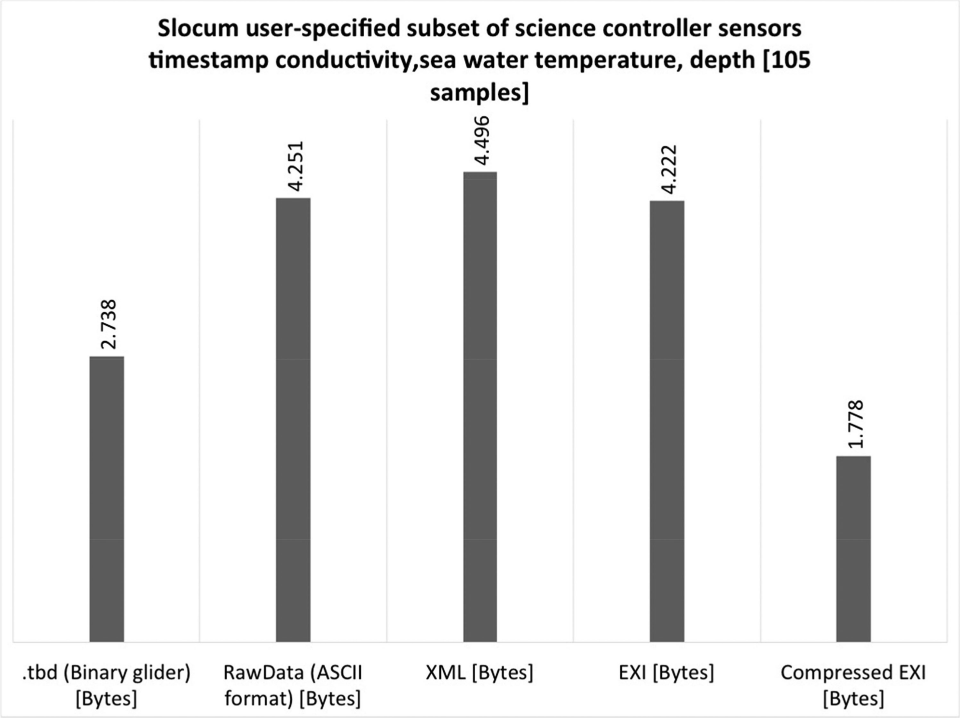


Fig. 11. Request size evaluation for the OGC SWE Bridge operations.

messages, containing sound pressure level (SPL) measurements and the timestamp, encoded in three different formats; XML, JSON, and EXI. According to Fig. 10, EXI and compressed EXI generate the smallest response/request since the messages are encoded as a bit-packed EXI body stream. In addition, when EXI compression is used, the response and request are at least 50% smaller in comparison with other encoding types.

B. Non-TCP/IP-Connected Observatory Platforms

In non-TCP/IP-connected observatory platforms, the communication between the SEISI systems and web services as the SOS Server is accomplished through the OGC SWE Bridge service running on the platform, e.g., on a Slocum glider. In such scenario where the SWE Bridge cannot communicate directly to the SOS Server, the SWE Bridge generates files with the T-SOS actions. These files will be transmitted by the platform controller to a host on shore where a simple software proxy will inject these files into the SOS Server. These files contain T-SOS actions such as InsertResult with corresponding sensor measurements. To evaluate the efficiency of the proposed implementation of the OGC SWE Bridge on the Slocum glider, we compared the size of messages containing a timestamp, conductivity, sea water temperature, and depth measurements encoded in three different formats; Slocum proprietary binary format and SWE Bridge output files encoded in XML and EXI. As illustrated in Fig. 11, Slocum binary format and compressed EXI generate the smallest request, and EXI compression is at least 35% smaller than the Slocum binary format.

VI. EXPECTED BENEFITS FROM THE USE OF SUCH STANDARDS

We believe that standards are an indispensable element to manage the design, construction and operation of complex systems. In coming years, we will see the emergence of even more sophisticated observational tools such as more capable AUVs, that already today urgently require standardized methods of payload sensor integration. However the present diversity of platforms, vehicles, moorings, cabled infrastructure, etc., seem to preclude a unified approach. In this paper, we have described a concept that is cross cutting and can be implemented with diverse sensors on different platforms. The benefit of such an approach is obvious: cost efficiency, technical ease, higher flexibility in regard to integrating new sensors, easy replacement of sensors, or components and so on. A general acceptance will only come if the benefits are demonstrated as part of use case studies and practical experience. NeXOS is a project that precisely aims at those demonstrations. The authors of this paper are convinced that the next generation of observatory and platform designers will push sensor manufactures to use these types of standards [23].

Some real scenarios and examples from the NeXOS project that demonstrate the cost efficiency of the proposed architecture are listed as follows.

1) The TRIOS company (Berman, Germany) www.trios.de) invested just two engineering days to upgrade one optical sensor to be OGC PUCK-enabled, and just several hours were needed to generate its proper SensorML file. The controller at the TRIOS Buoy with a Linux S.O. was executing the SWE Bridge and injecting real time data into the sensor web.
2) The OGC SWE Bridge provides a "universal sensor driver" that can operate any device whose protocol can be described with SensorML. SensorLab and Plocan integrated the following sensors into a Wave Glider ASV using this approach: SMID A1 for noise measurement SPL (125 and 63Hz, 10, 50 i 90%), TRIOS O1 [26], [27] to measure Polycyclic Aromatic Hydrocarbon, Tryptophan, and Chromophoric Dissolved Organic Matter, and the SensorLab pH sensor (http://www.sensorlab.es/). The software integration took less than a week to inject real data.
3) Alseamar integrated the A1 Nexos Hydrophone into the SeaExplorer using the same approach. First, the SWE Bridge code was adapted to the SeaExplorer controller; this work was challenging since SWE Bridge execution must be coordinated with other tasks on the SeaExplorer controller. This effort paidoff, since integration of the NeXOS hydrophone, or other new PUCK enabled sensors doesn't require new software drivers.

VII. CONCLUSION AND OUTLOOK

This paper has introduced the NeXOS sensor web architecture, which is a work in progress. This architecture is able to fulfill the central requirement to provide an interoperable exchange of oceanographic sensor data to facilitate a smooth integration the sensor into the web, taking into account the full chain: sensor–platform–communication link–web interface–end user. While first components are already available, the architecture will be continuously developed and enhanced in the next few years. Based on currently ongoing evaluation activities of the first available implementations and further emerging requirements, the NeXOS sensor web components will be advanced to a comprehensive suite of tools for sharing oceanographic observation data in an interoperable manner.

Besides making the underlying components available as open source software, NeXOS will also contribute to the advancement of the relevant conceptual foundations as well as to relevant international spatial data infrastructure and sensor web standards. As these standards are usually designed in a domain independent manner, NeXOS guidance on how to apply them in the field of oceanography will strengthen interoperability as well as the acceptance among relevant stakeholders. Toward this end, NeXOS consortium partners are now working to define sensor web profiles and templates for marine applications, building on developments from European and international initiatives, e.g., ESONET-EMSO, Seadatanet, GROOM, JERICO, FixO3, and Oceansites [24].

ACKNOWLEDGMENT

The authors would like to thank to the EMRP Meteomet2 joint research project, JRP (ENV58 MeteoMet2): The EMRP is jointly funded by the EMRP participating countries within EURAMET and the European Union. The Monterey Bay Aquarium Institute has been an important partner in this work. The authors would also like to thank the David and Lucile Packard Foundation for their support.

REFERENCES

[1] C. Waldmann, "The German contribution to ESONET—Integrating activities for setting up an interoperable ocean observation system in Europe," in *Proc. OCEANS EUROPE*, 2007, vol. 1–3, pp. 738–741.

[2] J. del Río *et al.*, "Standards-based plug & work for instruments in ocean observing systems," *IEEE J. Ocean. Eng.*, vol. 39, no. 3, pp. 430–443, Jul. 2014, doi: 10.1109/JOE.2013.2273277.

[3] European Commission Inspire, Annex III Sec. 14&15, Jul. 25, 2011. [Online]. Available: http://inspire.jrc.ec.europa.eu/

[4] D. M. Toma *et al.*, "Smart sensors for interoperable smart ocean environment," in *Proc. IEEE Oceans*, Santander, Spain, 2011, doi: 10.1109/Oceans-Spain.2011.6003654.

[5] T. O'Reilly *et al.*, "Instrument interface standards for interoperable ocean sensor networks," in *Proc. IEEE Oceans*, Bremen, Germany, 2009, doi: 10.1109/OCEANSE.2009.5278251.

[6] A. Bröring *et al.*, "New generation sensor web enablement," *Sensors*, vol. 11, no. 3, pp. 2652–2699, 2011.

[7] G. Percivall, "GEO Task AR-07-02 architecture implementation pilot," presented at the *GEO Archit. Data Committee Meeting*, Wayland, MA, USA, Sep. 2007.

[8] A. Bröring, C. Stasch, and J. Echterhoff, *OGC Implementation Specification: Sensor Observation Service (SOS) 2.0 (12-006)*, Wayland, MA, USA: Open Geospatial Consortium, Inc., 2012.

[9] *ISO TC 211, ISO 19156:2011—Geographic Information—Observations and Measurements*, Int. Standard ISO, Geneva, Switzerland, 2011.

[10] M. Botts and A. Robin, *OGC Implementation Specification: Sensor Model Language (SensorML) 2.0.0 (12-000)*, J. Del Río and E. Delory Eds. Wayland, MA, USA: Open Geospatial Consortium, Inc., 2014.

[11] J. Echterhoff and T. Everding, *OGC Discussion Paper: OGC Sensor Event Service (SES) 0.3.0 (08-133)*. Wayland, MA, USA: Open Geospatial Consortium, Inc., 2008.

[12] S. Ingo and J. Echterhoff, *OGC Implementation Specification: Sensor Planning Service (SPS) 2.0.0 (09-000)*. Wayland, MA, USA: Open Geospatial Consortium, Inc., 2011.

[13] S. Jirka, D. M. Toma, J. del Río, and E. Delory, "A sensor web architecture for sharing oceanographic sensor data," in *Proc. IEEE Sensor Syst. Changing Ocean*, Oct.13–17, 2014, pp. 1–4.

[14] D. M. Toma, J. del Río, S. Jirka, E. Delory, and J. Pearlman, "Smart electronic interface for web enabled ocean sensor systems," in *Proc. IEEE Sensor Syst. Changing Ocean*, Oct. 13–17, 2014, pp. 1–4.

[15] World Wide Web Consortium, "Efficient XML interchange working group," [Online]. Available: http://www.w3.org/XML/EXI/

[16] J. Schneider and T. Kamiya, "Efficient XML Interchange (EXI) Format 1.0," [Online]. Available: http://www.w3.org/TR/exi/

[17] T. O'Reilly, *OGC® PUCK Protocol Standard*, OGC Candidate Encoding Standard 09-127r2, Open Geospatial Consortium, Wayland, MA, USA, 2014. [Online]. Available: http://www.opengeospatial.org/standards/puck

[18] A. P. Castellani, N. Bui, P. Casari, M. Rossi, Z. Shelby, and M. Zorzi, "Architecture and protocols for the Internet of Things: A case study," in *Proc. 8th IEEE Int. Conf. Pervasive Comput. Commun. Workshops*, 2010, pp. 678–683.

[19] A. Bröring, S. Below, and T. Foerster, "Declarative sensor interface descriptors for the sensor web," in *Proc. 1st Int. Workshop Pervasive Web Mapping, Geoprocess. Services*, Como, Italy, 2010, vol. 38, pp. 26–27.

[20] A. Bröring, F. Bache, T. Bartoschek, and C. P. J. M. van Elzakker, "The SID Creator: A Visual Approach for Integrating Sensors with the Sensor Web," in *Advancing Geoinformation Science for a Changing World*, vol. 1, S. Geertmann, W. Reinhardt, and F. Toppen, Eds. Berlin, Germany: Springer-Verlag, 2011, pp. 143–162.

[21] A. Bröring, P. Maué, K. Janowicz, D. Nüst, and C. Malewski, "Semantically-enabled sensor plug & play for the sensor web," *Sensors*, vol. 11, pp. 7568–7605, 2011.

[22] SCPI Specification Volume 1-4. [Online]. Available: http://www.ivifoundation.org/specifications/default.aspx

[23] C. Waldman, J. del Río, D. Toma, T. O'Reilly, and J. Pearlman, "Intelligent sensors—Why they are so important for future ocean observing systems," in *Proc. IEEE Sensor Syst. Changing Ocean*, Oct. 13–17, 2014, pp. 1–5.

[24] S. Jirka and J. del Río, "SWE-based Observation Data Delivery from the Instrument to the User - Sensor Web Technology in the NeXOS Project," in *Proc. 19th EGU Gen. Assem.*, 2017, vol. 2, p. 7769.

[25] E. Delory, D. Toma, J. del Río, P. Ruiz, L. Corradino, P. Brault, and F. Fiquet, "Developing a new generation of passive acoustics sensors for ocean observing systems," in *Proc. IEEE Sensor Syst. Changing Ocean*, doi: 10.1109/SSCO.2014.7000383.

[26] J. Pearlman and O. Zielinski, "A new generation of optical systems for ocean monitoring - matrix fluorescence for multifunctional ocean sensing," *Sea Technol.*, vol 2, pp. 30–33, Feb. 2017.

[27] J. Wollschläger, D. Voß, O. Zielinski, and W. Petersen, "*In situ* observations of biological and environmental parameters by means of optics—development of next-generation ocean sensors with special focus on an integrating cavity approach," *IEEE J. Ocean. Eng.*, vol. 41, no. 4, pp. 753–762, Oct. 2016, doi: 10.1109/JOE.2016.2557466.

Joaquín del Río (S'07–M'09) was born in Catalonia, Spain, in 1976. He received the B.S., M.S., and Ph.D. degrees in telecommunication engineering and electronic engineering from the Technical University of Catalonia, Catalonia, Spain, in 1999, 2002, and 2011, respectively.

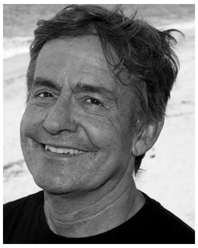
Since 2001, he has been a Professor in the Electronic Engineering Department, Universitat Politècnica de Catalunya, Barcelona, Spain. He is a member of the research group "Remote Acquisition Systems And Data Processing (SARTI)," UPC. His research interests include electronic instrumentation, interoperability in marine sensor networks, and wireless sensor networks.

Daniel Mihai Toma received the M.Sc. degree in electrical engineering from the Technical University "Gheorghe Asachi," Iasi, Romania, in 2008 and the Ph.D. degree in electronic engineering from the Universitat Politècnica de Catalunya (UPC), Barcelona, Spain, in 2012.

He is currently a member of the research group "Remote Acquisition Systems and Data Processing (SARTI)," UPC. His current research interests include electronic instrumentation, marine instrumentation, wireless ad hoc networks, interoperability in sensor networks, and synchronization and scheduling.

Enoc Martínez received the B.S. and M.S. degrees in telecommunication engineering and electronic engineering from the Technical University of Catalonia (UPC), Barcelona, Spain, in 2012 and 2015, respectively, where he is currently working toward the Ph.D. degree at the Department of Electronic Engineering, "Remote Acquisition Systems and Data Processing (SARTI)" research group, UPC.

His current research interests include electronic instrumentation, sensor interoperability, sensor web technologies, and passive acoustic monitoring.



Thomas O'Reilly received the B.S. degree in geology and the M.S. degree in geochemistry.

He is currently a Software Engineer at the Monterey Bay Aquarium Research Institute (MBARI), Moss Landing, CA, USA, where he develops applications for sensors, sensor networks, and autonomous vehicles. He is also with marine scientists to characterize the chemistry, biology, and geology of the ocean and the sea floor. He led a team at MBARI to develop prototype "PUCK" protocol for self-identifying sensors, and the protocol was later refined and adopted as a formal standard by the Open Geospatial Consortium. Before working at MBARI, he helped to develop a thermal infrared spectrometer that characterized the surface of Mars from an orbiting spacecraft.

Eric Delory has worked in environmental science and bioengineering in positions ranging from applied research to management, in Europe and Asia. He started his career developing diverse portable hardware solutions and signal processing algorithms for biomedical and environmental sensing applications. Applications ranged from human heart rate telemetry technologies to developing systems for the assessment of noise impacts on mammal sensory systems. In the 1990s, he was with the National University of Singapore, where he was a member of the founding team of the Acoustic Research Laboratory, working on new signal processing and pattern recognition methods, in particular wavelets and neural networks. Back to Europe, he cofounded the Laboratory of Applied Bioacoustics at the Technical University of Catalonia, where he kept working in signal processing, sensing science, and technologies. His interest extended to multidisciplinary observing systems, including sensor, and data interoperability, in particular working with the IEEE on the initial efforts of standardization of the Global Earth Observation System of Systems, for which he became a Liaison of the European Seas Observatory Network ESONET, Instigator of the EMSO infrastructure. Since 2010, he has been leading the Observatory of the Oceanic Platform of the Canary Islands, mainly doing system Research and Development in optics, acoustics, and interoperability. He coordinates NeXOS, a 21-partner European collaborative project developing compact cost-effective sensors, optical and acoustic, for the monitoring of ocean variables. He also recently engaged in European integration initiatives. He has coordinated the transnational access activities of the European Fixed Open-Ocean Observatories Consortium FixO^3 and also currently coleads the new technologies work package in AtlantOS, Kiel, Germany.

Jay S. Pearlman (M’84–SM’02–F’08–LF’15) is currently a Professor (Adjunct) in the Department of Electrical and Computer Engineering, University of Colorado, Denver, CO, USA, and the Director of FourBridges, Liberty Township, OH, USA. For the last decade, he has been working with the IEEE France on research and development actions. His background includes sensors, remote sensing, and information systems. He continues this work with activities in sensor innovation and applications including interests in interoperability and facilitating cross-discipline research. He is also addressing information infrastructure though his work with the Research Data Alliance and EarthCube. He has more than 80 publications and 25 U.S. and international patents.

Christoph Waldmann received the Intermediate Diploma degree from the University of Munich, Munich, Germany, in 1977, and the Diploma and Ph.D. degrees from the University of Kiel, Kiel, Germany, in 1981 and 1985, respectively, all in physics.

He is currently with the Center for Marine Environmental Sciences, MARUM, University of Bremen, Bremen, Germany. His research interests include the development of new methods of efficient, longterm observation of processes in the sea using a range of techniques from hydroacoustic methods and instruments to optical methods for measuring basic parameters in the water column from mobile and stationary observation platforms.

Simon Jirka is currently a Project Manager and Community Leader at the Sensor Web Group, Open Source initiative 52°North, Muenster, Germany. His activities are focused on Spatial Data Infrastructures (SDI) as well as Sensor Web architectures. Besides his contribution to several national and European projects such as COLABIS, NeXOS, and ODIP, he is also involved in several INSPIRE related activities. Furthermore, he contributes to the Sensor Web Enablement initiative and different domain working groups (e.g., hydrology) of the Open Geospatial Consortium. Through his daily work, comprising research activities as well as commercial consulting and software development projects He has a broad range of experience in SDI and Sensor Web related topics.