Project	MONOCLE H2020 (grant 776480)	Start / Duration	1 February 2018/ 48 Months
Dissemination	PUBLIC	Nature	REPORT
Date	20/11/2018	Version	1.1

MONOCLE

Multiscale Observation Networks for Optical monitoring of Coastal waters, Lakes and Estuaries

Deliverable 5.2

System architecture and standards report

Project Description

Funded by EU H2020 MONOCLE **creates sustainable *in situ* observation solutions for Earth Observation (EO) of optical water quality in inland and transitional waters.** MONOCLE develops essential research and technology to lower the cost of acquisition, maintenance, and regular deployment of *in situ* sensors related to optical water quality. The MONOCLE sensor system includes handheld devices, smartphone applications, and piloted and autonomous drones, as well as automated observation systems for e.g. buoys and shipborne operation. The sensors are networked to establish interactive links between operational Earth Observation (EO) and essential environmental monitoring in inland and transitional water bodies, which are particularly vulnerable to environmental change.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 776480

Project	MONOCLE H2020 (grant 776480)	Start / Duration	1 February 2018/ 48 Months
Dissemination	PUBLIC	Nature	REPORT
Date	20/11/2018	Version	1.1

Deliverable Contributors:	Name	Organisation	Role / Title
Deliverable Leader	Kathrin Poser	WI	WP5 Lead
Contributing Author(s)	Oliver Clements	PML	Task 5.2 lead
Reviewer(s)	Stefan Simis	PML	Coordinator
Final review and approval	Stefan Simis	PML	

Document History:

Release	Date	Reason for Change	Status	Distribution
0.1		Initial document	Draft	Internal
0.2		Document for review	Draft	Internal
1.0	31/07/2018	First release	Completed	EC
1.1	20/11/2018	Digital Object Identifier added	Released	Public

To cite this document:

Poser, K., Clements, O. (2018). D5.2 System architecture and standards report. *Deliverable report of project H2020 MONOCLE (grant 776480)*, doi: 10.5281/zenodo.1492271

TABLE OF CONTENTS

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 776480

List of abbreviations

Abbreviation	Meaning
API	Application Programming Interface
CAT	Catalogue Service
JSON	Javascript Object Notification
OGC	Open Geospatial Consortium
O&M	Observation and Measurements
SAS	Sensor Alert Service
SensorML	Sensor Markup Language
SES	Sensor Event Service
SOS	Sensor Observation Service
SPS	Sensor Planning Service
SWE	Sensor Web Enablement
WCS	Web Coverage Service
WFS	Web Feature Service
WMS	Web Map Service
WNS	Web Notification Service
XML	Extensible Markup Language

1. Executive Summary

This document describes the proposed architecture of the MONOCLE data collection, processing, analysis and delivery system at the start of system and sensor development activities. The MONOCLE system encompasses a number of different sensors deployed from a range of platforms delivering water quality data to a common system, allowing users to easily access and interact with the sensors and collected data. The architecture specifies the subsystems of the MONOCLE system, their interactions and the communication standards that will be used.

2. Scope

This report is mainly intended for internal use but may also be of use in communicating the MONOCLE sensor and data connectivity principles to closely related projects. A basic understanding of the importance of interoperable data interfaces in versatile, scalable data networks is assumed while the scope of each of the proposed standards is also introduced below.

This document reflects the state of planning at the start of the MONOCLE project, and the chosen solutions are therefore still expected to evolve. At this point, a functional description of the system and interfaces is needed rather than details on the implementation. The latter will follow once the newly developed sensors and data interpretation techniques reach sufficient maturity. This overview may therefore be considered as a living document. The final architecture and interface descriptions will be part of D5.4 “System user and developer handbook”.

3. Introduction

The aim of MONOCLE is to implement enabling technologies for the deployment, management and maintenance of in situ sensors and sensor networks, to ultimately reduce uncertainties associated with the retrieval of water quality information from satellite observations. MONOCLE will introduce new sensor technological development across a range of innovative platforms, combining high-end reference sensors in a spatially sparse configuration with a complementary network of low cost sensors including smartphones and unmanned aerial vehicles (UAV or drones).

MONOCLE will collect sensor data from multiple sources:

- In-situ data collected by:
  - non-expert participants (e.g. citizen scientists)
  - automated instruments (e.g. on buoys, ships)
  - manually operated instruments (e.g. hand-held or submerged sensors)
- Satellite data of inland, transitional and coastal water bodies
- Image data collected with Remotely Piloted Aircraft Systems (RPAS)

The challenge is to build a system which allows combining all these data, make them easily accessible and allow integrating them into new and existing data analysis and processing, regardless of the location of the source data or the consumer system. To reach this aim, the MONOCLE architecture will make use of open standards as much as possible, mainly those of the Open Geospatial Consortium (OGC), and clearly define its design principles and requirements for sensors to be added to the network.

4. System overview

The MONOCLE system will provide a framework for collecting, storing, processing, integrating, visualising and distributing data collected by a range of different sensors. An overview of the envisaged system and its architecture concept is presented in Figure 1.

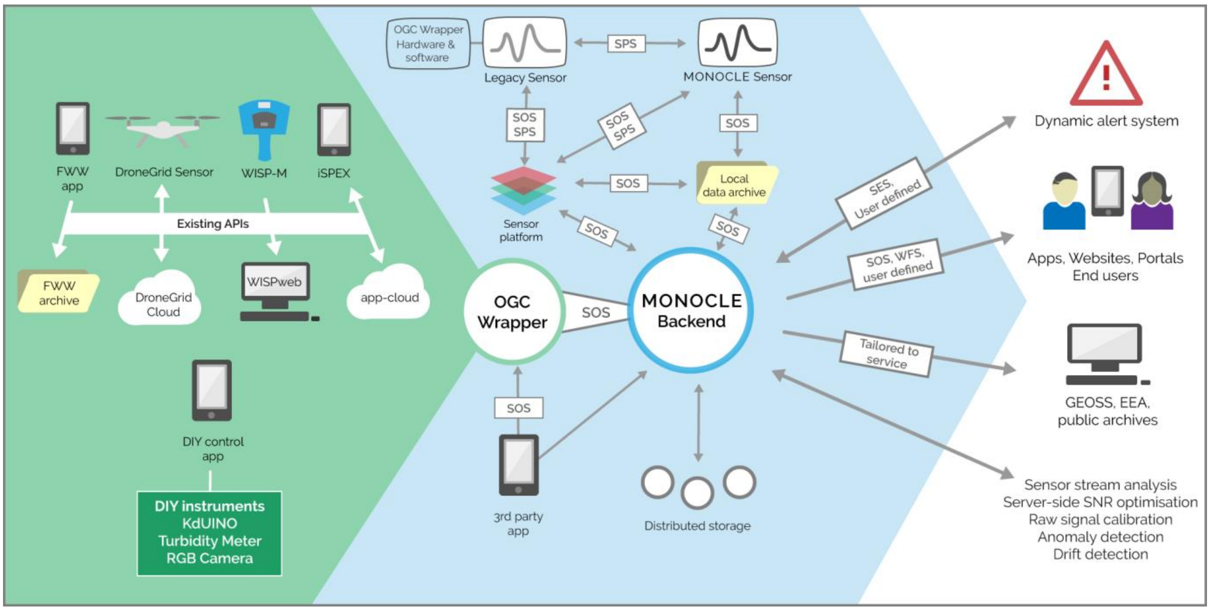

Figure 1 Overview of proposed MONOCLE architecture

The core of the MONOCLE system is the MONOCLE data backend. The backend will store, or dynamically collect, data from the different sensor systems. It will provide different paths of interaction for user applications such as via a visualisation frontend, or via machine-to-machine interfaces. The different sensors developed or improved within MONOCLE will provide data to the backend via well-defined interfaces. For pre-exiting sensors, wrappers will be implemented to allow them to connect to the MONOCLE system.

Individual parts of the system are already in place at several project partners, as well as external stakeholders, whether they be data producers or consumers or end-users (Figure 1). It is important to note that the MONOCLE data infrastructure extends, rather than recreates, existing solutions, by providing both forward and backward compatibility. Backward compatibility is achieved by developing wrappers around existing interfaces to make these compliant with MONOCLE (meta)data standards, including a foreseen hardware/software OGC interface for relevant legacy environmental sensors. Forward compatibility is ensured by building on standards for data and service interoperability. Thus, to realise the overall MONOCLE system, the existing systems have to be extended, adjusted or wrapped to conform to the MONOCLE architecture requirements. New developments within MONOCLE will be built on these requirements from the start.

System requirements

Functional system requirements

Functional requirements include how the system should deal with input data, which operations it performs and how it makes data available to the users.

Integration of heterogeneous data

The different sensors deliver different types of data (point measurements, images and time series) and with varying spatial coverage, physical measurements and sensing principles. The MONOCLE system needs to be able to receive and process all these data and deliver products based on heterogenous sources to its users. The data should be self-describing.

Federated access to distributed data

The different data sources are not only heterogeneous in format, content and measurement frequency, but can also be hosted at different locations (sensor providers). From the user perspective, the distributed system should be transparent - the user should not need to be aware where the data are stored, but should instead be presented with a consistent logical dataset.

Near real-time data availability

For operational management of system components and the most versatile use of the data, sensor results should be available in near real-time, i.e. within seconds to minutes from acquisition, given sufficient network connectivity. Consequently, data acquisition, processing and quality control should be fully automated.

Variable data access

The system should allow different options for data access and data provision. A web visualisation system should allow users to find and view data from the different sensors. An alerting system should allow users to specify conditions under which they want to receive alert notifications. Machine-to-machine interfaces should allow various external applications to access and retrieve the MONOCLE data. Data accessibility controls need to be implemented per use type (e.g. scientific vs commercial), user group, and individual user.

Discoverability

The MONOCLE sensors and their observations should be discoverable also for users who are not aware of the MONOCLE project, i.e. they should be registered in data repositories and advertise their capabilities. Discoverability of data should be managed at the user, use type, user group and dataset level, so that inaccessible data (e.g. under embargo) may be discoverable even if they are not fully accessible, as determined by the data owner.

Sensor interaction

Interaction with sensors should be bidirectional. Through the MONOCLE system it should be possible for sensor operators to address individual sensors for planning, scheduling, and tasking. Synchronization of measurements between sensors is a major requirement.

Non-functional system requirements

Non-functional requirements define how the system should work and by that capture the constraints on the system design or implementation. For the MONOCLE system, the following non-functional requirements have been identified:

Extensibility

The system must be capable of easily connecting to new data sources. Also, the input sources may evolve over time. Therefore, an architectural layer that is capable of ingesting a dynamic number of information sources needs to be provided.

Interoperability

On a technical level, it should be easy to connect new data sources. It should be possible to connect to data sources via common well-defined interfaces such as those of the Open Geospatial Consortium.

Modularity

The system should follow a modular design: each of its components should operate, to the extent possible, independently of the other system components. Where communication between the modules is necessary, it should be conducted exclusively through well-defined interfaces.

Scalability

The system's performance must not unduly degrade as the number or frequency of observations increases. The system must be able to process all input data in near real-time.

Robustness

The system should be robust to external factors. The system shall endeavour to remain operational and provide high availability even in the event of external factors, such as a workstation or server crash, unexpected pauses in transmissions, electrical problems, or sensor failure.

Security

The system must provide access to the data and services based on the data providers' access policies. Different levels of access can be provided to different users.

Sub-systems

MONOCLE Backend

The MONOCLE Backend is the core of the MONOCLE architecture and should be seen as a collection of data harvesting, processing, archiving and quality control mechanisms using common software libraries. In addition, the backend includes a collection of algorithms for event triggering and anomaly detection, which may be sensor specific. The backend will further contain a number of resources to aid curation of data sources, such as information on sensor calibration, access constraints and data licensing. The backend will incorporate the data processing routines developed in WP6.

MONOCLE Sensor

A MONOCLE sensor is a sensor that is designed to integrate into the service based communication system of MONOCLE. This will allow the sensor to talk directly to the MONOCLE back-end. Where bespoke systems are already in place for sensors that will be further developed in MONOCLE, such as the WISPcloud or DroneGrid Cloud, a wrapper will be built that will act as a translation engine, converting standardised requests into the bespoke data retrieval requests for the individual systems.

Any sensor can become a MONOCLE-compatible sensor, either by including sensor support in the legacy sensor interface developed in WP2 (see next item), or by implementing a basic set of OGC compliant services in the sensor communication interface. MONOCLE encourages sensor developers to contact the consortium to include sensors in compatibility testing.

MONOCL OGC Wrapper

A wrapper that provides Open Geospatial Consortium standard service interface will be created that will act as an intermediary between existing data collection systems. The wrapper will also act as a service interface for existing legacy sensors, effectively upgrading them to the same capabilities as a MONCLE Sensor. The wrapper will have both a software component (Python libraries) and a hardware component (low-cost microcontroller).

Legacy Sensor

In the MONCLE architecture a legacy sensor is any sensor that provides monitoring capabilities but does not have a direct service interface. These sensors will be integrated into the MONCLE system using the MONOCLE OGC Wrapper.

User Applications

A variety of user applications will be produced throughout the MONOCLE project. The applications will interface with the MONOCLE back-end through the provided OGC services. The applications will act as the entry point to the MONOCLE system for all types of users. An example of a user application that will be provided is a web based GIS system that will be able to act as a search, visualisation, analysis and download facility.

Interfaces

Existing Application Programming Interface (API) <-> MONOCLE Backend

The interface for the connection between existing APIs and the MONOCLE backend will require the APIs to either be wrapped in or augmented with OGC service interfaces. The minimum requirement is for the sensor data to be provided in OGC formats (such as O&M, WCS – see next section). Ideally, bidirectional communication should be implemented to allow for sensor planning and triggering, with sensor status/start/stop commands supported at minimum.

Legacy Sensor <-> MONOCLE Backend

This interface deals with the communication from legacy or pre-existing sensors and the MONOCLE Backend, accommodating a large number of commercially successful sensors used in environmental monitoring and scientific research today which do not use standard protocols. To facilitate standardised communication, hardware and software wrappers will be created that act as a translation engine, converting the data streams of the sensors into standardised systems, such as OGC Sensor Observation Service outlined in the Interfaces section of this report.

Sensors <-> Sensor Platform

The MONOCLE architecture envisages the need for communication between an individual sensor and the wider sensor platform or network it is a part of. This two way communication will be used for

- Providing data to a sensor platform controller for ingestion in a MONOCLE Backend
- Requesting a calibration value from another sensor in the network
- Sharing time and positioning data between sensors
- Providing error or alert information to the sensor platform, or triggering new events
- Synchronization of sensors operating on a single platform

MONOCLE Sensor <-> MONOCLE Backend

Sensors created during the project will be built to utilise standard data interfaces outlined in the next section. The communication will be bi-directional because data will flow from the individual sensor to the MONOCLE Backend as well as from the backend to the sensor, for instance to aid in self-calibration.

MONOCLE Backend -> User Applications

This interface deals with the transfer of data from the MONOCLE Backend to the various user applications that will be created/supported during the MONOCLE project. The sub-system will require the ability for the user application to request specific data for a specific location and time. To cater for these requirements it is proposed that the OGC SOS and WFS standard interfaces be used. These standards are discussed in the protocols section of this report.

MONOCLE Backend -> Public Archives

For the submission of MONOCLE data to public archives, in principle the same interfaces can be used as for the interaction with user applications. However, if the archives have their own data ingestion procedures that are not covered by the OGC services, wrappers or custom connectors will be implemented where necessary.

Protocols

Sensor web enablement (SWE)

The major part of the protocols to be implemented in MONOCLE will be from the OGC Sensor Web Enablement suite of standards. An overview of these standards is given in Figure 2. On the left side of the figure are the standards that deal with service provision whereas the standards on the right hand side deal with data encoding.

OGC Sensor Observation Service

The OGC Sensor Observation Service (SOS) provides a standardised interface for querying and transmitting data collected by a particular sensor in a particular spatial and temporal location. The standard also provides a mechanism to explore what sensors are available from a given service. The communication interfaces can act in two directions, i.e. data can be pushed from a sensor to a service and from a service to an end user or consumer.

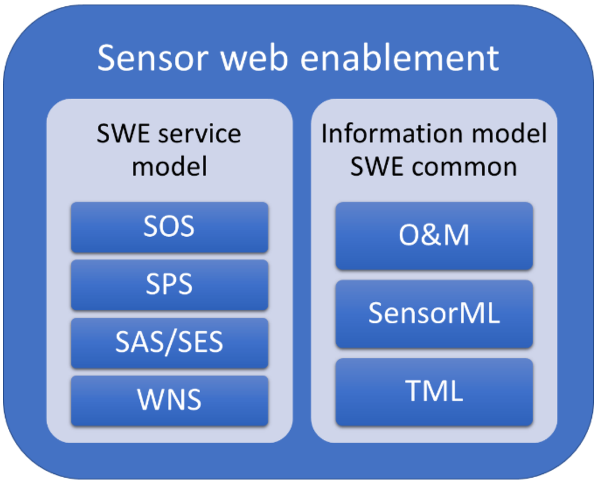

Figure 2: The Sensor Web Enablement suite of standards

As with most OGC standard interfaces SOS is based on the exchange of standardised XML document fragments using the HTTP protocol. The standard supports both HTTP GET and POST methods. For POST the XML is provided as the request payload. For GET requests the XML fragment is split out into a set of Key Value Pairs (KVP). The service will always respond with an XML fragment following the structure defined within the standard. The methods provided by the standard are summarised in Figure 3, showing how a consumer and a data producer will utilise the standard.

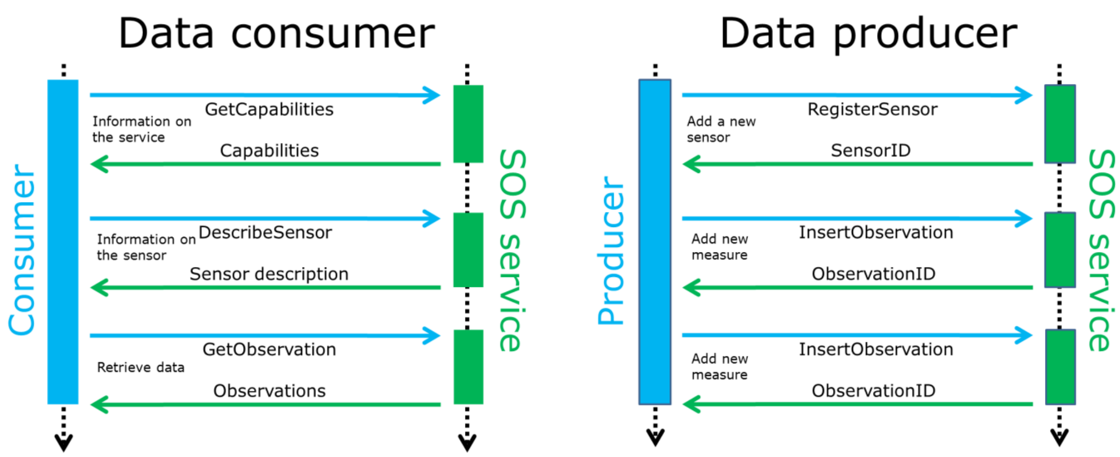

Figure 3 Typical SOS interaction for Consumer and Producer from istsos.org licensed under CC BY 4.0

The SOS will be used as the primary communication interface between existing data processing and storage systems and the primary MONOCLE backend. The communication will be facilitated by an OGC wrapper that will act as a translation engine, converting the standardised queries from SOS into the specific data requests needed for bespoke systems. New sensors that are created during the project will have SOS built in allowing them to deposit sensor readings in the MONOCLE backend directly. The SOS will also be used to provide data from the MONOCLE backend to user applications. The standard allows very specific data flow, limiting unnecessary bandwidth usage.

OGC Sensor Planning Service

The OpenGIS® Sensor Planning Service Interface Standard (SPS) defines interfaces for queries that provide information about the capabilities of a sensor and how to task the sensor. The standard is designed to support queries that have the following purposes: to determine the feasibility of a sensor planning request; to submit and reserve/commit such a request; to inquire about the status of such a request; to update or cancel such a request; and to request information about other OGC Web services that provide access to the data collected by the requested task. OGC SPS will be used for intra sensor network communication. An example of the use of SPS would be a smart sensor that requires a calibration value querying whether a sensor is available to provide the needed measurement. The smart sensor will then be able to request the calibration value, get information as to how to retrieve it and finally use that value to calibrate its own output.

OGC Sensor Event Service, OGC Sensor Alert Service and OGC Web Notification Service

These proposed standards by the OGC propose to fill in some gaps in the OGC Sensor Web Enablement suite of services. These specifications have not yet reached the status of OGC standard, so developments will be followed during the MONOCLE project to see if they provide suitable mechanisms for automated data alerts and event reporting. Custom systems that align with the SES, SAS and WNS standards will be implemented if no existing implementations of the specifications are available.

The SES, SAS and WNS specifications, where available, will be utilised in the creation of automated alerts to user applications. This could take the form of an alert to sensor network managers that there is an issue that needs fixing or an alert to a user system that a particular sensor has a reading that has triggered some predefined threshold.

OGC SWE Common Data Model

The Sensor Web Enablement (SWE) Common Data Model Encoding Standard defines low level data models for exchanging sensor related data between nodes of the OGC Sensor Web Enablement (SWE) framework. These models allow applications and/or servers to structure, encode and transmit sensor datasets in a self-describing and semantically enabled way. Both the Observations & Measurements and the SensorML standards are based on the SWE common model

OGC Observations & Measurements

The Observations & Measurements (O&M) standard consists of models and XML Schema for encoding observations and measurements from a sensor, both archived and real-time. O&M is the typical format to encode responses to the GetObservation request of an SOS. In the O&M definition, an observation is an act that results in the estimation of the value of a feature property, and involves application of a specified procedure, such as a sensor, instrument, algorithm or process chain. The procedure may be applied in-situ, remotely, or ex-situ with respect to the sampling location. O&M defines a core set of properties for an observation:

- feature of interest

- observed property
- result
- procedure – the instrument, algorithm or process used (described using SensorML)
- phenomenon time – the real-world time associated with the result
- result time – the time when the result was generated
- valid time – the period during which the result may be used

An observation can be a single value (such a concentration of chlorophyll-a in specific location at a specific time or a spatial coverage (such as an image taken by a drone that covers a certain spatial extend at a given time).

OGC Sensor Markup Language

The OGC Sensor Markup Language (SensorML) provides models for describing sensor systems and processes associated with sensor observations: it provides information needed for discovery of sensors, location of sensor observations, processing of low-level sensor observations, and listing of taskable properties, as well as supporting on-demand processing of sensor observations. A process can be, for example, a measurement procedure conducted. In SensorML, a sensor is defined as a process which is capable of observing a phenomenon and returning an observed value. This process can be characterised by a number of properties including inputs, outputs, parameters, and process methods. Metadata of a process include its identification and classification, as well as characteristics such as the temporal availability or its spatial description. SensorML is used in SWE services (SOS, SPS) to describe their associated sensors.

While the standard encoding of SensorML is in XML as with most OGC standards, there are also JSON encoding rules available.

SensorThings API

The OGC SensorThings API is a standard specification for providing an open and unified way to interconnect Internet of Things devices, data and applications over the Web. The SensorThings API is an open standard, builds on Web protocols and the OGC Sensor Web Enablement standards, and applies an easy-to-use REST-like style. The result is to provide a uniform way to expose the full potential of the Internet of Things. Although promising for relatively simple environmental sensors, the standard is not yet mature and as such will be evaluated throughout the lifetime of MONOCLE to assess the possible uses and benefits of the standard (Liang et al, 2016).

Further OGC services

OGC Web Feature Service

The WFS standard makes geographic feature data available through a highly configurable interface. By default, the data returned by a WFS is in Geography Markup Language (GML) which is written as XML. However, emerging versions of the standard also support the JavaScript Object Notation (JSON) which is a preferred data format for developers and data consuming applications. Government agencies, private organisations and academic institutes use this standard to publish vector geospatial datasets in a way that makes it easier for receiving organisations to compile new maps or conduct analysis on the supplied data.

WFS provides a standard interface for requesting vector geospatial data consisting of geographic features and their properties. The benefit of this is that WFS clients can request source data from multiple WFS servers, and then render the data for display on the client or process the data further as part of a workflow. The standard guarantees that the data can be handled consistently with other data using a common geospatial coordinate reference system. Feature properties encoded using common data types such as text strings, date and time can also be handled consistently.

The WFS will be used as a mechanism to request data from the MONOCLE backend. The interface will be utilised by user applications.

OGC Web Coverage Service

The OGC Web Coverage Service standard defined an interface for querying, sub-setting and downloading raster data. This can be used for the automatic download of validation earth observation datasets. As with other OGC standard interfaces the operations are provided through the HTTP protocol. The service can be queried to establish the available data through the GetCapabilities operation. This will return a list of available coverages/datasets as well as some metadata describing the spatial and temporal bounds of the data. More dataset specific metadata can be requested through the DescribeCoverage operation. This returns all available metadata about the dataset. Data can finally be downloaded using the GetCoverage operation. This operation allows the data to be subsetted both in space and time such that only the specific data that is required need to be transferred.

OGC Web Map Service

The Web Map Service standard provides an interface for requesting geo-registered map images via the web. A WMS request defines the geographic layer(s) and area of interest to be processed. The response to the request is one or more geo-registered map images (returned as JPEG, PNG, etc) that can be displayed in a browser application. As the response is not actual data, but a visual map representation, in MONOCLE this standard will only be used for user applications and not for internal communication in the MONOCLE system.

OGC Catalogue Service

OGC Catalogue Services (CAT) support the ability to publish and search collections of descriptive information (metadata records) for geospatial data, services, and related information. Metadata in catalogues represent resource characteristics that can be queried and presented for evaluation and further processing by both humans and software. Catalogue services are required to support the discovery and binding to registered information resources within an information community. In the development of the CAT services, the focus has clearly been on the WMS, WFS and WCS family of services. When building a sensor discovery solution, the specific characteristics of sensors and sensor networks have to be taken into account (Jirka, 2010). This concerns especially the highly dynamic structure of sensor networks (e.g. mobility of sensors, continuous addition of new sensors, defective sensors and removal of sensors) which makes it difficult to rely on conventional OGC Catalogues that have been developed to handle more static data sources.

An OGC discussion paper was published to propose a Sensor Instance Registry (SIR) service, a web service interface for managing the metadata and status information of sensors. The service also included automatically harvesting sensor metadata, transforming the collected metadata sets into a data model compatible to OGC Catalogues and to push harvested metadata into OGC Catalogue instances. However, this service has not made it to an OGC standard and is currently not implemented. Therefore, it still needs to be established how best to publish MONOCLE sensors and data in catalogues.

Implementation matrix

Table 1 gives on overview of which OGC standards are currently expected to be implemented at the interfaces of MONOCLE subsystems.

Table 1. Implementation matrix of standards within MONOCLE

	MONOCLE Sensor	MONOCLE Backend	Legacy Sensor	MONOCLE OGC Wrapper	User Application
MONOCLE Sensor	SOS/SPS/SES	SOS	-	-	-
MONOCLE Backend	SOS	-	-	SOS	SOS/WFS/WMS/WCS
Legacy Sensor	-	-	-	Existing communication API or SOS	-
MONOCLE OGC Wrapper	-	SOS	Existing communication API or SOS	-	SOS
User Application	-	SOS/WFS/WMS/WCS	-	-	-

MONOCLE sensors status

Table 2 gives an overview of the MONOCLE sensors and their currently available standardized interfaces. All sensor providers have committed to improving the interoperability of their sensors, most have identified this as a priority development area (see D2.1 “Report on analysis of the requirements for MONOCLE sensors including projection of cost-savings and stakeholder feedback”).

Table 2. List of MONOCLE sensors and systems

System	Developer	Platform	Currently available standardized interfaces
HSP1	Peak Design Ltd	Fixed or moving platform (buoy, ship, pole)	-
CLAM	PML	Fixed or moving platform (buoy, ship, pole)	-
Sun tracking radiance platform	PML	Ship	-

WISPStation	Water Insight	Fixed or moving platform (buoy, ship, pole)	Custom-made REST interface
Prosumer RPAS drone systems	VITO / Sitemark	Drone	WCS, WMS
iSPEX	University of Leiden/DDQ	Smartphone app	JSON/REST interface via Parse server
KdUino	CSIC	Smartphone app	-
FreshWater Watch	Earthwatch	Smartphone app	JSON data stream

5. Exploitation and Dissemination

While this deliverable is mainly intended for internal use to serve as guidance for the implementation of the MONOCLE system across the project partners, the report is made publicly available to help align activities with similar research projects and sensor manufacturers. The final architecture and interface descriptions, which will be part of D5.4 "System user and developer handbook", will inform users at all levels (data consumers, data integrators, sensor providers) on how to use and interact with the MONOCLE system. Any stakeholders interested in using or reproducing part of this report, or looking at ways to implement the data standards described here within the context of environmental water quality monitoring, are encouraged to contact MONOCLE through the project website (monocle-h2020.eu) or contact the authors directly.

6. Future activities/recommendations

Implementing the software infrastructure for the MONOCLE system is a major activity in the coming project period. The next steps include:

- *Detailed implementation guidelines*: While this report gives an overview of the standards and specifications to be implemented by the different system components, a more detailed design specification of requirements will be developed that specifies the required and desired functionalities to the level of specific requests, and encodings to be implemented for subsystem. A tutorial will be made available to the sensor developers to guide the implementation on their side. This is planned to take place by month 8 (Oct 2018).
- *Implementation of the backend:* The data storage system and receiving interfaces are expected to be running in demonstration (non-operational) mode by month 12 (Jan 2019), i.e. a working concept for any and all sensor systems that have the required connectivity, and data models for sensors that first require further interface development. (PML)

- *Implementation of interfaces on the sensors/platforms:* The specified interfaces should be implemented on all sensors and platforms by the time they will be deployed in the validation campaigns starting from the second campaign (month 17, June 2019). The first campaign in August 2018 will focus on the individual sensors rather than their software infrastructure. The responsibility of implementing the respective interfaces lies with the sensor/platform providers. In addition PML will implement an OGC wrapper for legacy sensors. The timing depends on the deployment in the validation campaigns; for most sensors, this will be validation campaign 2 in M15.
- After the first test runs with these initial implementations, systems and interfaces will be updated and problems fixed. By month 20 (Sep 2019), the first near real-time data streams should be available from mature test sites providing a reference framework for other sensors/platforms and development in WP5 and on-the-fly signal analysis algorithm testing in WP6.

References

Baumann, P. (2012) OGC© WCS 2.0 Interface Standard- Core Version 2.0.1. OGC © Interface Standard. URL: https://portal.opengeospatial.org/files/09-110r4

Botts, M. & Robin, A. (2014) OGC © SensorML: Model and XML Encoding Standard Version 2.0. OGC© Encoding Standard. URL: https://portal.opengeospatial.org/files/?artifact_id=55939.

Bröring, A., Stasch, C. & Echterhoff, J. (2012) OGC © Sensor Observation Service Interface Standard Version 2.0. OpenGIS © Implementation Standard. URL: https://portal.opengeospatial.org/files/?artifact_id=47599.

Cox, S. (2013) OOGC Abstract Specification Geographic information â€” Observations and measurements Version 2.0. OGC © Standard: Abstract Specification. URL: http://www.opengis.net/doc/is/om/2.0.

De la Beaujardiere, J. (2006) OpenGIS© Web Map Server Implementation Specification Version 1.3.0. OpenGIS Implementation Specification. URL: http://portal.opengeospatial.org/files/?artifact_id=14416.

Echterhoff, J. & Everding, T. (2008) OpenGIS © Sensor Event Service Interface Specification (proposed) Version 0.3.0. OpenGIS © Discussion Paper. URL: https://portal.opengeospatial.org/files/?artifact_id=29576.

Jirka, S., Nüst, D., Schulte, J. & Houbie, F. (2010) Integrating the OGC Sensor Web Enablement Framework into the OGC Catalogue. ISPRS Archives Volume XXXVIII-4/W13: Proceedings of WebMGS 2010: 1st International Workshop on Pervasive Web Mapping, Geoprocessing and Services. 26.-27. August 2010. Como, Italy. URL: http://www.isprs.org/proceedings/XXXVIII/4-W13/ID_13.pdf.

Liang, S., Huang, C.-Y. & Khalafbeigi, T. (2016) OGC SensorThings API Part 1: Sensing. Version 1.0. OGC© Implementation Standard. URL: http://docs.opengeospatial.org/is/15-078r6/15-078r6.html.

Nebert, D., Voges, U. & Bigagli, L. (2016) OGC© Catalogue Services 3.0 - General Model Version 3.0. OGC© Implementation Standard. URL: http://docs.opengeospatial.org/is/12-168r6/12-168r6.html.

Robin, A. (2011) OGC© SWE Common Data Model Encoding Standard Version 2.0.0. OGC© Encoding Standard. URL: http://portal.opengeospatial.org/files/?artifact_id=41157.

Robin, A. (2018) JSON Encoding Rules SWE Common / SensorML Version 1.0. OGC© Best Practice. URL: http://docs.opengeospatial.org/bp/17-011r2/17-011r2.html.

Simonis, I. & Echterhoff, J. (2006) Draft OpenGIS © Web Notification Service Implementation Specification Verion 0.0.9. OpenGIS Best Practices Paper. URL: portal.opengeospatial.org/files/?artifact_id=18776.

Simonis, I. & Echterhoff, J. (2007) OGC © Sensor Alert Service Implementation Specification Version 0.9.0. Candidate OpenGIS© Interface Standard. URL: http://portal.opengeospatial.org/files/?artifact_id=24780&version=1&format=pdf.

Simonis, I. & Echterhoff, J. (2011) OGC© Sensor Planning Service Implementation Standard Version 2.0. OpenGIS Implementation Standard. URL: http://www.opengis.net/doc/IS/SPS/2.0.