# JERICO-S3 DELIVERABLE

Joint European Research Infrastructure for Coastal Observatories

**Science, Services, Sustainability**

| | |
|---|---|
| **DELIVERABLE #, WP# and full title** | JERICO-S3 D7.2 - WP7 - Suite of Sensor-Web components for the end-to-end integration of the JIIM into operable infrastructures |
| **5 Key words** | Interoperable instrument module, testing, coastal observation |
| **Lead beneficiary** | Ifremer |
| **Lead Author** | Laurent Delauney |
| **Co-authors** | Enoc Martínez, Simon Jirka, Christian Autermann, Jan Speckamp, Jérôme Blandin |
| **Contributors** | |
| **Final version date/ Submission date** (dd.mm.yyyy) | 16.05.2024 |

## Nature: D (Demonstrator)

(R = Report, P = Prototype, D = Demonstrator, O = Other)

## Dissemination level : Public

PU = Public, PP = Restricted to other programme participants (including the Commission Services), RE = Restricted to a group specified by the consortium (including the Commission Services), CO = Confidential, only for members of the consortium (including the Commission Services)

**GRANT N°:** 871153
**PROJECT ACRONYME :** JERICO-S3
**PROJECT NAME :** Joint European Research Infrastructure for Coastal Observatories - Science, services, sustainability
**COORDINATOR :** Laurent DELAUNEY - Ifremer, France - jerico@ifremer.fr

# DOCUMENT TECHNICAL DESCRIPTION

| Document ID | JERICO-S3-WP7-D7.2-16052024_V1.2 |
|---|---|

| REVISION HISTORY | | | |
|---|---|---|---|
| **Revision** | **Date** | **Modification** | **Author** |
| V1.0 | 01/02/2024 | Creation of the document | L. GODIVEAU |
| V1.1 | 02/05/2024 | Draft | All |
| V1.2 | 16/05/2024 | Minor revisions | E. Delory |
| | | | |

| APPROVALS | | | | |
|---|---|---|---|---|
| | **Name** | **Organisation** | **Date** | **Visa** |
| **Coordinator** | Delauney Laurent | Ifremer | 19/07/2024 | X |
| **WP Leaders** | Delory Eric | PLOCAN | 16/05/2024 | X |
| **WP Leaders** | Simone Marini | CNR | 19/07/2024 | X |

| Diffusion list | | | |
|---|---|---|---|
| Consortium beneficiaries | Third parties | Associated Partners | other |
| X | X | | |

# TABLE OF CONTENT

# 1. SUMMARY

The coastal EGIM (cEGIM) is a module designed for the JERICO community to ease and foster the observation of coastal parameters in an interoperable fashion. A prototype was built in order to demonstrate its concept through a demonstration that took place at a Coast-HF site off Luc-sur-Mer, part of the English Channel Pilot Super Site. Special attention was paid to ensuring end-to-end compatibility with Sensor Web Enablement and Internet of Things standards. The server side of the demonstrator was built using the 52°North SensorThings API module. The client application enabling the Web-based visualisation of the published sensor data used a dedicated instance of the 52°North Helgoland Sensor Web Viewer. An algal bloom detection algorithm was integrated in the COSTOF2 data-logger of the cEGIM through the use of an embedded Odroid single board computer. The proprietary data stream generated by the COSTOF2 was converted into a format adhering to the SensorThing API standard.

# 2. INTRODUCTION

The aim of task 7.2 was to design, build and test an interoperable instrument module (cEGIM) that would host a core set of observing instruments, support interfacing with existing JERICO observing facilities and be compatible with open standards for sensor and data interoperability. The implementation and demonstration of the cEGIM are reported in D7.3, D7.4, D7.7 and D7.9. This report shows how the cEGIM was implemented with a view to ensuring end-to-end compatibility with Sensor Web Enablement and Internet of Things standards, as specified in the JERICO-S3 project description.

# 3. MAIN REPORT

## 3.1. Sensor Web components

The main Sensor Web Components that were used for building the demonstrator for the end-to-end integration of the JIIM were the SensorThings API module of the 52°North Sensor Web Server as well as the 52°North Helgoland Sensor Web Viewer.

The server side of the demonstrator is built using the 52°North SensorThings API (STA) implementation which is available as part of the 52°North Sensor Web Server (https://github.com/52North/sensorweb-server-sta). This implementation is available as open source software under the GNU General Public License, Version 2.0. As part of the JERICO-S3 project, the main implementation work that had to be performed on this component comprised the customisation of the data publication functionality (for use by interoperability layer linking it to the Costof2). Furthermore, during the deployment a series of minor adjustments were identified in order to improve interoperability and robustness.

For building the client application that enables the Web-based visualisation of the published sensor data, a dedicated instance of the 52°North Helgoland Sensor Web Viewer was created and published. This open source component is available via the Apache License Version 2.0 (https://github.com/52North/helgoland-toolbox). This client framework is based upon a generic toolbox framework which offers reusable components that can be used to build advanced web-based sensor web client applications. As part of the JERICO-S3 project, the main focus of the development work was on creating an improved support for the OGC SensorThings API data model and interface specification. While the Helgoland viewer was previously based on the internal 52°North Helgoland API, an abstraction layer has been integrated that allows linking the Helgoland Viewer also to other types of APIs such as the OGC SensorThings API.

Furthermore, towards the end of the JERICO-S3 project, 52°North is investigating how further relevant emerging API standards such as the OGC Connected Systems API specification may be used as a data source for the Helgoland Sensor Web Viewer. Finally, a dedicated version of the viewer was deployed that is specifically customised to the JERICO-S3 project.

Figure 1 shows the map view of the Helgoland Sensor Web Viewer. This view is mainly used for exploring available sensor stations as well as the data (time series) measured at these stations. As soon as the user identifies a time series that shall be visualised, it can be added to the diagram view.

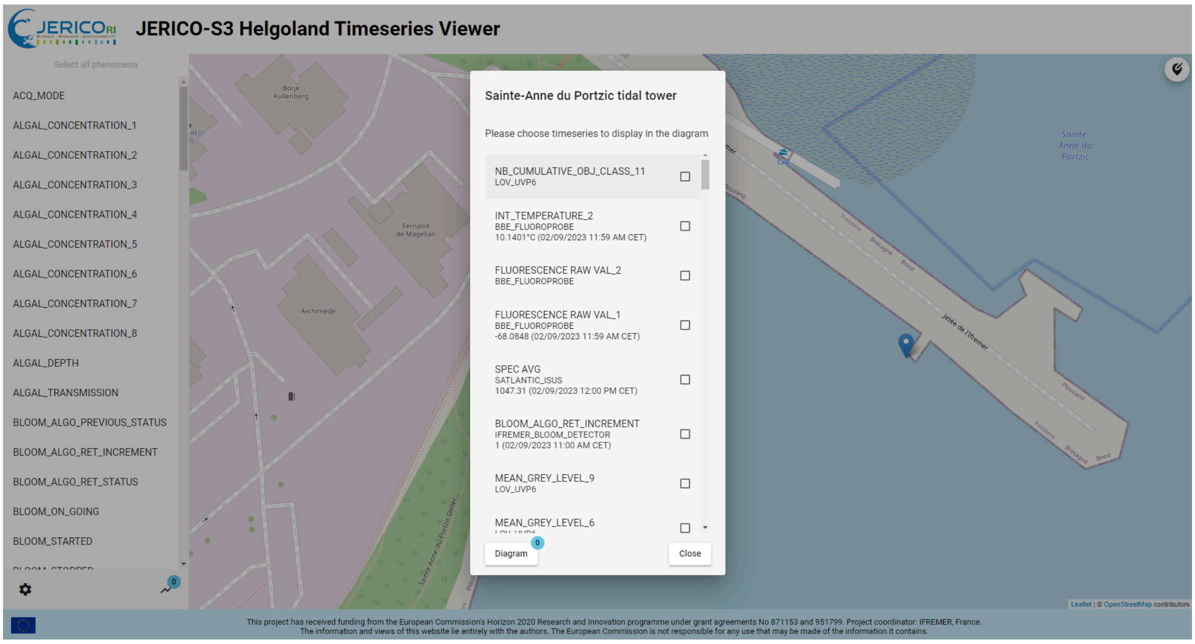

Figure 1: Map view of the 52°North Helgoland Sensor Web Viewer

The diagram view of the Helgoland Sensor Web Viewer is shown in Fig.2. It is capable of visualising the data of selected time series in a diagram. This is complemented by a range of functionality to adjust the styling of the displayed time series, to retrieve further metadata about the time series (e.g. dates of first and last measurements as well as the main characteristics of the time series), and to share the current view with other users.

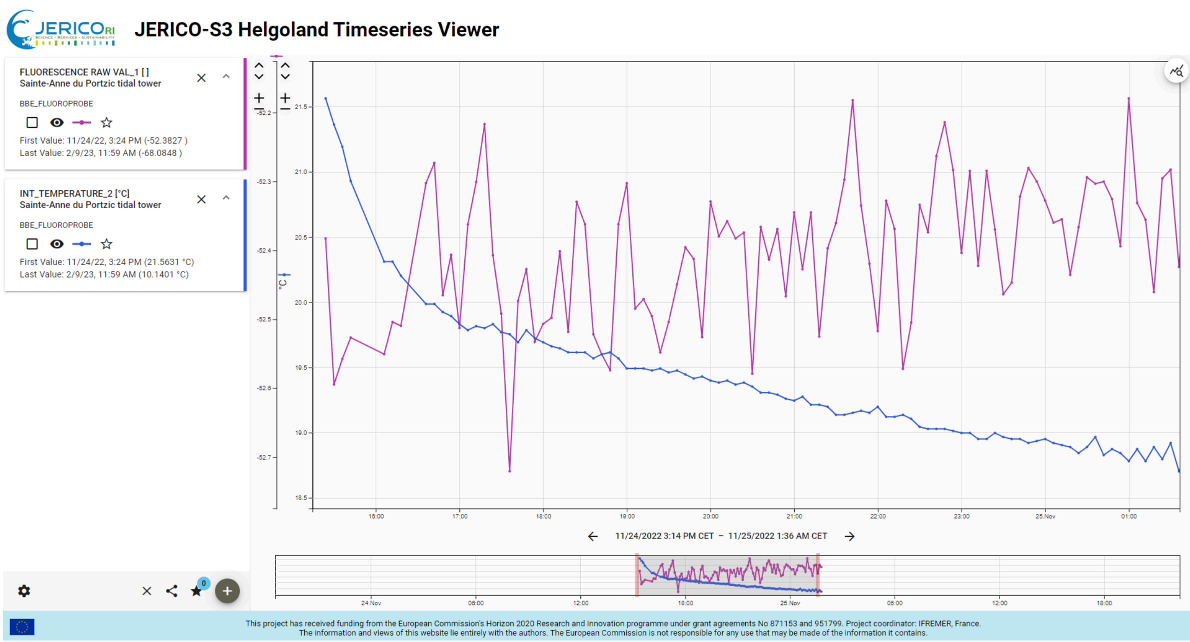


Figure 2: Diagram view of the 52°North Helgoland Sensor Web Viewer

## 3.2. integration of the Algal Bloom Detection algorithm into the Costof2

Although the Costof2 is an ultra-low power data-logger, very efficient for long deployments, it doesn't have the resources to host computationally-demanding algorithms. To enhance its computing capabilities, an Odroid single-board computer was employed and communication between the Costof2 and the algorithm was facilitated through a Python3 Flask API. This Flask API managed the seamless exchange of requests and results, ensuring robustness in the integration process.

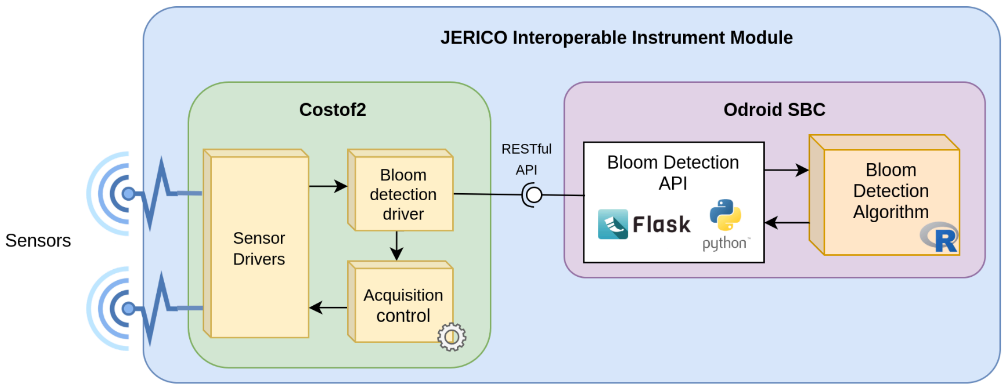


Figure 3: Detail of the components contained within the JERICO Interoperable Instrument Module (JIIM).

The integration setup, depicted in figure 3, involves the use of an Odroid single-board computer (SBC) to host and provide the necessary computational power for the Algal Bloom Detection Algorithm. The Costof2 serves as the data logger, responsible for collecting and storing environmental data. The communication bridge between the Costof2 and the algorithm is established through a Python3 Flask API.

The execution Workflow is initiated by Costof2, gathering environmental data from the sensors. Afterwards, it triggers the execution of the Algal Bloom Detection Algorithm on the Odroid via the API. The algorithm results are sent back to the Costof2 via the Flask API for storage and further analysis. If an algal bloom is detected, the acquisition control routine of the Costof2 is adjusted to the new situation.

The Bloom Detection API source code is freely available at the following Bitbucket repository under MIT licence: [https://bitbucket.org/Enoc_M/cegim-bloom-detection-api](https://bitbucket.org/Enoc_M/cegim-bloom-detection-api)

## 3.3. Connection with the Costof2 to the SensorThings API service

The Costof2, acting as the datalogger, collects environmental data and transmits it to the shore station in a proprietary binary format to maximise transmission efficiency. The data is received by a "binary decoder" software, which converts the binary format into structured JSON data, as shown in figure 4.

Once the data is stored in JSON-formatted files, it is then passed to the "Costof2-to-SensorThings" converter software. This converter serves as an interoperability layer designed to transform proprietary data streams from the Costof2 into a format that adheres to the SensorThings API standard. The Costof2-to-SensorThings converter ensures that the transformed data aligns with the structure and conventions specified by the SensorThings API standard, a standardised web service protocol developed by the Open Geospatial Consortium for the interoperable exchange of sensor data and metadata.

The standardised data is submitted to the SensorThings API service, where it is archived. This service ensures the storage, retrieval, and querying of the environmental data, providing a standardised interface for applications to interact with the collected information. On top of the SensorThings API service there is the Helgoland data viewer, which ensures user-friendly data visualisation. Additionally, advanced users can also query the SensorThings API programmatically using their own scripts.

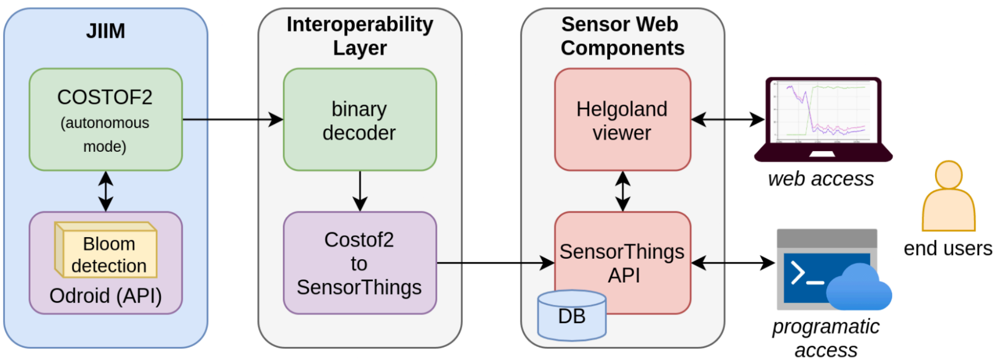


Figure 4: Interoperability workflow, the JIIM (cEGIM) data is injected into Sensor Web components using an interoperability layer at the shore station.

The Costof2-to-SensorThings source code is publicly available under MIT licence at the following git repository : https://bitbucket.org/Enoc_M/costof2sensorthings

# 4. CONCLUSION

Thanks to the work described above, the cEGIM could be fully demonstrated, not as another single module providing its proprietary data, but smoothly integrating into the suite of standard Open Geospatial Consortium Sensor Web Enablement tools. All developments are documented and available through the URLs provided in the core text of this report.