Development of an assesSment tOol for predicting the dynamic riSk of drowning on bEAcheS

- *SOSeas* -

System Architecture of the SOSeas Service

15th November 2020

CITICAN

FUNDACION INSTITUTO DE HIDRÁULICA AMBIENTAL DE CANTABRIA

UC UNIVERSIDAD DE CANTABRIA

Index

List of Acronyms

API	Application Programming Interface
ANN	Artificial Neural Networks
C3S	Copernicus Climate Change Service
CMEMS	Copernicus Marine Environment Monitoring Service
DIAS	Data Information Access Service
DSS	Decision Support System
ETL	Extract, Transform & Load
GFS	Global Forecast System
M2M	Machine to machine communication
NOAA	National Oceanic and Atmospheric Administration
OGC	Open Geospatial Consortium
PWA	Progressive Web App
SDGs	Sustainable Development Goals
SOBRASA	The Brazilian Life-Saving Society
TDS	THREDDS Data server
UX	User Experience
UI	User Interface
WEkEO	We knowledge Earth Observation
WCS	Web Coverage Service
WMS	Web Map Service

List of Figures

1 INFRASTRUCTURE

Operational Systems require having a robust and reliable infrastructure. Figure 1 shows the main sections of the current IHCantabria System Infrastructure, highlighting the virtual infrastructure that holds the development and production environments. The development or preproduction Servers provide the required environment to design and test the software developments, whereas the Production environment releases, under a versioning control system, mature versions of the Systems.

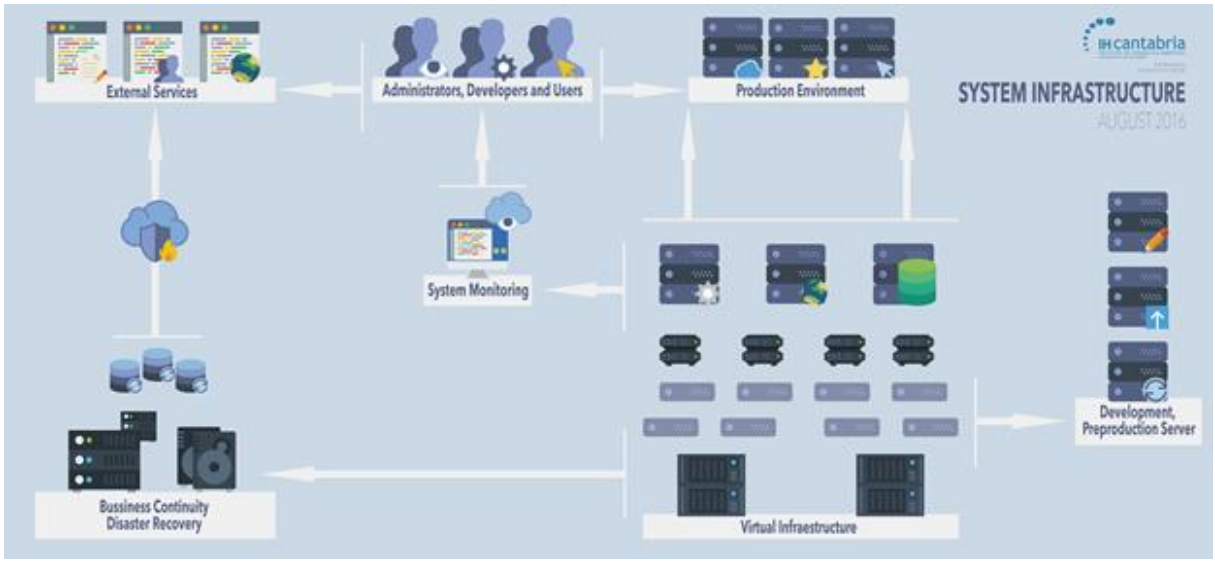

Figure 1. System Infrastructure

Virtualization is a proven software technology that makes it possible to run multiple operating systems and applications on the same server at the same time. Virtual machines act like a real computer but the software executed on these machines is separated from the underlying hardware resources. Virtualization can increase IT agility, flexibility, and scalability while creating significant cost savings. Workloads get deployed faster, performance and availability increases, and operations become automated.

Main components of the virtualized infrastructure for the SOSeas Service are showed in Figure 2. The infrastructure is composed of numerical analysis and processing components, orange elements, storage and management, blue elements, and user interfaces, purple elements.

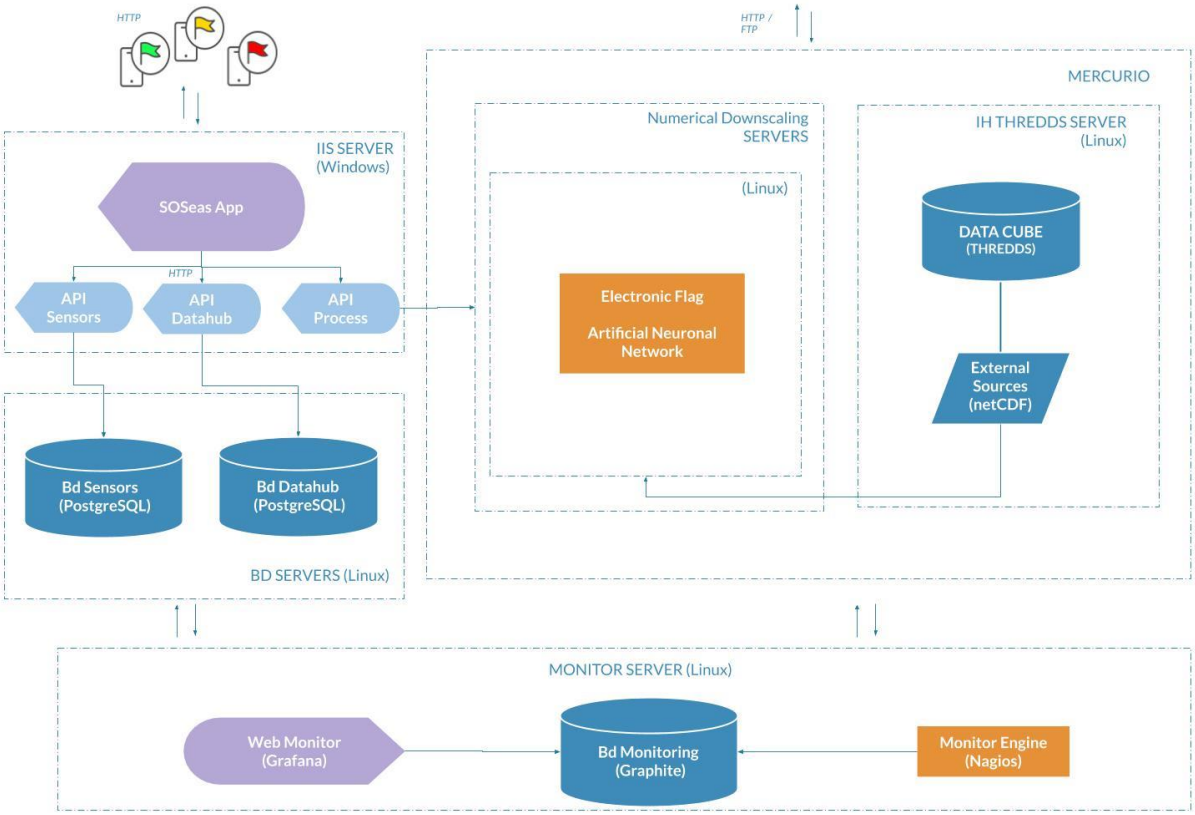

Figure 2. Main components of the Infrastructure

Numerical analysis and processing components:

- Electronic Flags ANN Process – Artificial Neuronal Network processes for tailored information products.
- Monitor Engine (Nagios) – Monitoring of the Infrastructure, numerical modelling. (inputs and outputs) and processes.

Storage and management:

- DATA CUBE (THREDDS) – Manages all the NetCDFs.
- Sensors BD – Manages the information collected from *in situ* sensors.
- Datahub BD – Manages all the metadata of the products stored in NetCDF format.
- Monitoring BD – Manages all the data collected by the Monitor Engine.
- Sensors API – Interoperability access to the data stored at the Sensors relational database.
- Data hub API – Interoperability access to the data stored at the Datahub relational database.
- Process API - Interoperability access to calculate the Electronic Flags

User Interfaces (UIs):

- SOSeas Service – UI of the SOSeas Service.
- Web Monitor – UI of the Monitoring System.

2 TECHNOLOGIES

The Infrastructure of the SOSeas Service is based on a cloud-based architecture. The architecture required to run the service is composed by two main sections: back-end and front-end. Figure 3 shows a graphical representation of the workflow between Front-end and Back-end sections for any standard Web application.

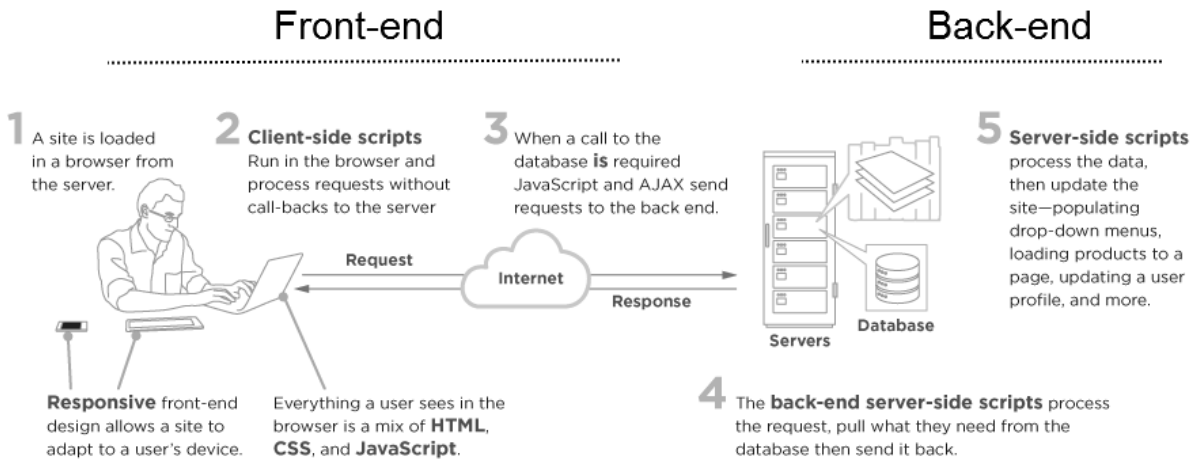

Figure 3. Front-end and back-end sections

The front-end includes the user interface and the client's computer system used for accessing the Geospatial information through interoperability protocols. In order to provide access through any mobile device a Progressive Web App has been specifically designed and developed. Section **¡Error! No se encuentra el origen de la referencia.** "**¡Error! No se en cuentra el origen de la referencia.**", describes in detail the User Interface and User Experience of the app developed.

On the other hand, the back-end includes servers, data storage system, virtual machines, backup system, processing system, monitoring system and the required software to provide interoperability protocols to provide access to the data. Back-end is in charge of gathering, perform the analytical and numerical modelling, manage data and provide interoperability protocols.

The main technologies that are used by the SOSeas Serive are Javascript (front-end) and python (back-end), see Figure 4.

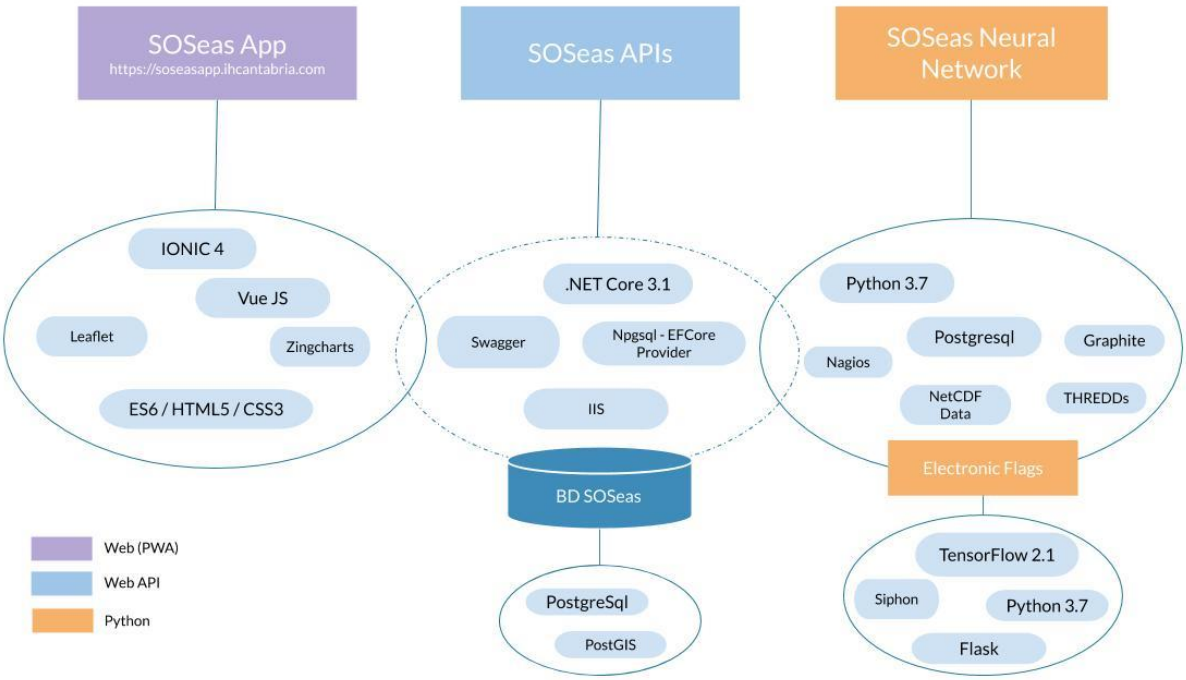

Figure 4. Front-end (Web) and back-end Technologies (API & Python)

The technologies are listed and described:

- *IONIC* (https://ionicframework.com/) is a complete open-source SDK for hybrid mobile app development. Ionic provides tools and services for developing hybrid mobile, desktop, and Progressive Web Apps based on modern web development technologies and practices.
- *Vue JS* (https://vuejs.org/) is an open-source JavaScript framework for building user interfaces and is perfectly capable of powering sophisticated Single-Page Applications when used in combination with modern tooling and supporting libraries.
- *Leaflet* (https://leafletjs.com/) is an open-source JavaScript library used to build interactive web mapping applications. It works efficiently across all major desktop and mobile platforms, can be extended with lots of plugins, has an easy to use and well-documented API and a simple, readable source code.
- *Zingchart* (https://www.zingchart.com/) is one of the most feature-rich, fully customizable JavaScript charting library available. ZingChart is built with vanilla JavaScript, but it has integrations for including charts in projects built in Angular, Backbone, Ember, jQuery, and React. Zingchart is a simple JavaScript library for building responsive charts and dashboards.

- *ES6/HTML5/CSS3* JavaScript enables interactive web pages and thus is an essential part of web applications. The vast majority of websites use it, and all major web browsers have a dedicated JavaScript engine to execute it. Alongside HTML and CSS, JavaScript is one of the three core technologies of the World Wide Web.
- *NetCDF* (https://www.unidata.ucar.edu/software/netcdf/) is a Java library and self-describing, machine-independent data formats that support the creation, access, and sharing of array-oriented scientific data.
- *PostgreSQL* (https://www.postgresql.org/) is an open source relational database management system.
- *Npgsql* (https://www.npgsql.org/) is an open source ADO.NET Data Provider for PostgreSQL, it allows programs written in C# to access the PostgreSQL database server.
- *Python* (https://www.python.org/) is an interpreted, high-level, general-purpose programming language.
- *Flask* (https://flask.palletsprojects.com/en/1.1.x/) extensible web microframework for building web applications with Python.
- *Siphon* (https://unidata.github.io/siphon/latest/index.html) is a collection of Python utilities for downloading data from remote data services. Much of Siphon's current functionality focuses on access to data hosted on a THREDDS Data Server. It also provides clients to a variety of simple web services.
- *THREDDS* Data Server (https://www.unidata.ucar.edu/software/thredds/current/tds/) is a web server that provides metadata and data access for scientific datasets, using a variety of remote data access protocols.
- .NET Core (https://github.com/dotnet/core) is a free and open-source, managed computer software framework primarily developed by Microsoft.
- *Swagger* (https://swagger.io/) is an open-source software framework backed by a large ecosystem of tools that helps developers design, build, document, and consume RESTful web services.
- *TensorFlow is an open source platform for machine learning. (https://www.tensorflow.org/)*
- *Grafana* (https://grafana.com/) is an open source analytics & monitoring solution for databases.
- *Nagios* (https://www.nagios.org/) is an open source server monitoring software.
- *Graphite* (https://graphiteapp.org/) is an open source tool that monitors and graphs numeric time-series data such as such as the performance of computer systems.

3 APIs

Application Programming Interfaces, commonly known by its acronym “APIs”, are software elements that act as an abstraction layer, with a set of rules and specifications, which can be used by other software.

The SOSeas Service makes use of three APIs: one API specifically designed for the dynamic analysis of forecasting flags, named API process SOSeas, and other two APIs that manage the access to the in-situ and forecasting data products required: Sensors API and Data Hub API. The following subsection describe each of them.

3.1 API PROCESS SOSEAS

The flag prediction is provided via the API Process SOSeas, which is in charge of the ANN computations and makes use of TensorFlow and Flask technologies.

Figure 5 shows the UI that introduces the API Process and the required inputs to calculate the forecasting flag.
https://apiprocess.ihcantabria.com/

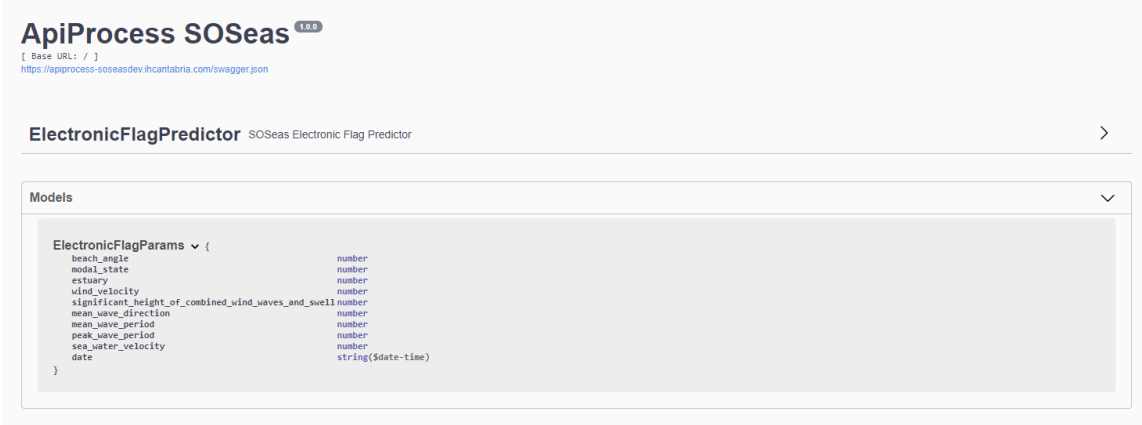

Figure 5. API Process of the SOSeas Service

3.2 SENSORS API

The Sensors API provides access to *in situ* observations, such as buoys or anemometers. Therefore, through the Sensors API any software developer or technician could access to the real-time data from in situ sensors. Figure 6 shows the UI to introduce the Sensors API capabilities (methods) to any developer or technician.

Figure 6.Web Interface of the Sensors API

As an example, Figure 7 shows the result, JSON format, of a query to the "LatestData" method collected by an anemometer.

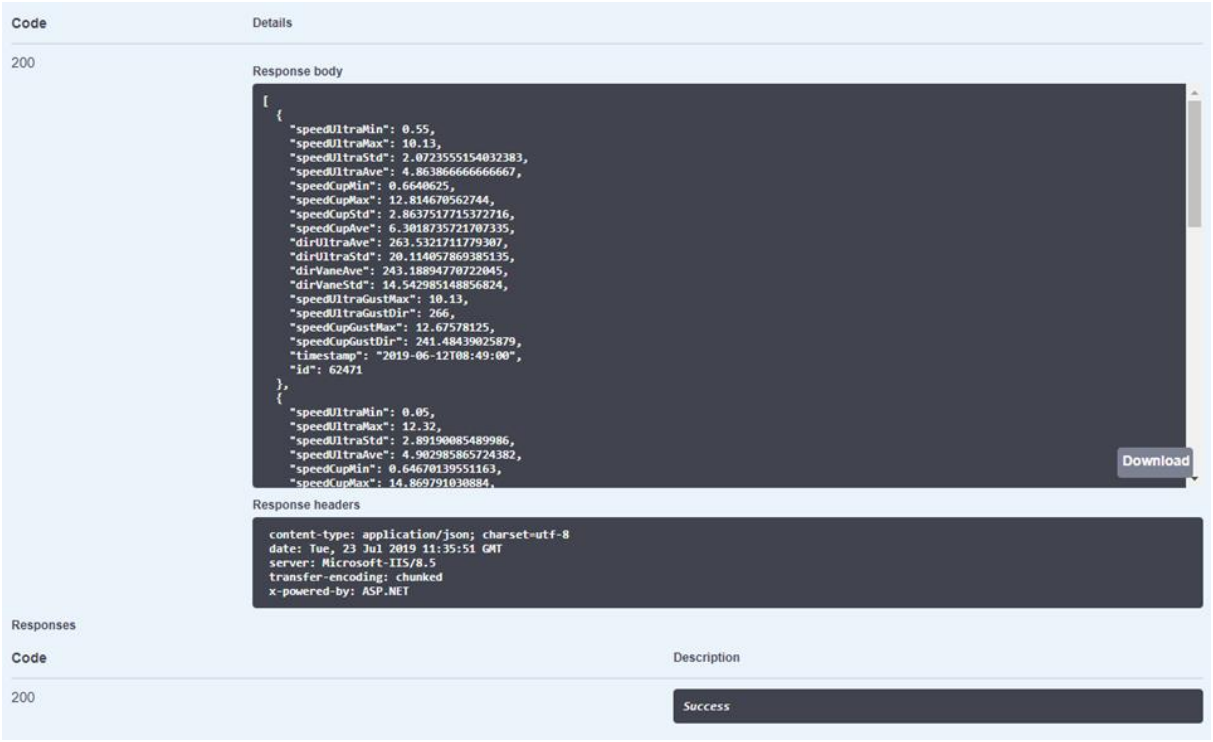

Figure 7. Query from the Sensors API

The Sensors API is based on a Relational database management system, PostgreSQL, which is in charge of the management of observations obtained from the sensors (buoys, gauges, anemometers).

3.3 DATA HUB API

Data hub API facilitates the search and access to metadata of the operational metocean products, which are stored in NetCDF format. The technology implemented to facilitate interoperable access to files stored in NetCDF format is THREDDS Data Server (TDS), which provides geospatial standard protocols established by the Open Geospatial Consortium (OGC) such as WMS, WCS in addition to NetCDF Markup Language (NCML), NetcdfSubset, among others.

The meteocean information is stored as a “Product”. Products are obtained from different “Sources”, data providers such as CMEMS, NOAA, AEMET, etc. One metocean Product could host several “Variables”, which are characterized depending on the “Type of variable” and refer to a specific “Moment”, such as past conditions or short term forecast, among others. Finally, several “Clients” make use of the metocean Products, Clients are applications such as SOSeas, TRL Plus, MSP Platform, etc.. All these elements are accessible though the Data hub API, see Figure 8.

Figure 8. User Interface Data hub API

Figure 9 shows the result, in JSON format, of a query about the list of products stored in the data hub. Among the results, the characteristics or metadata of the products are listed: spatial, temporal resolution, URL access to the THREDDs catalogue, etc.

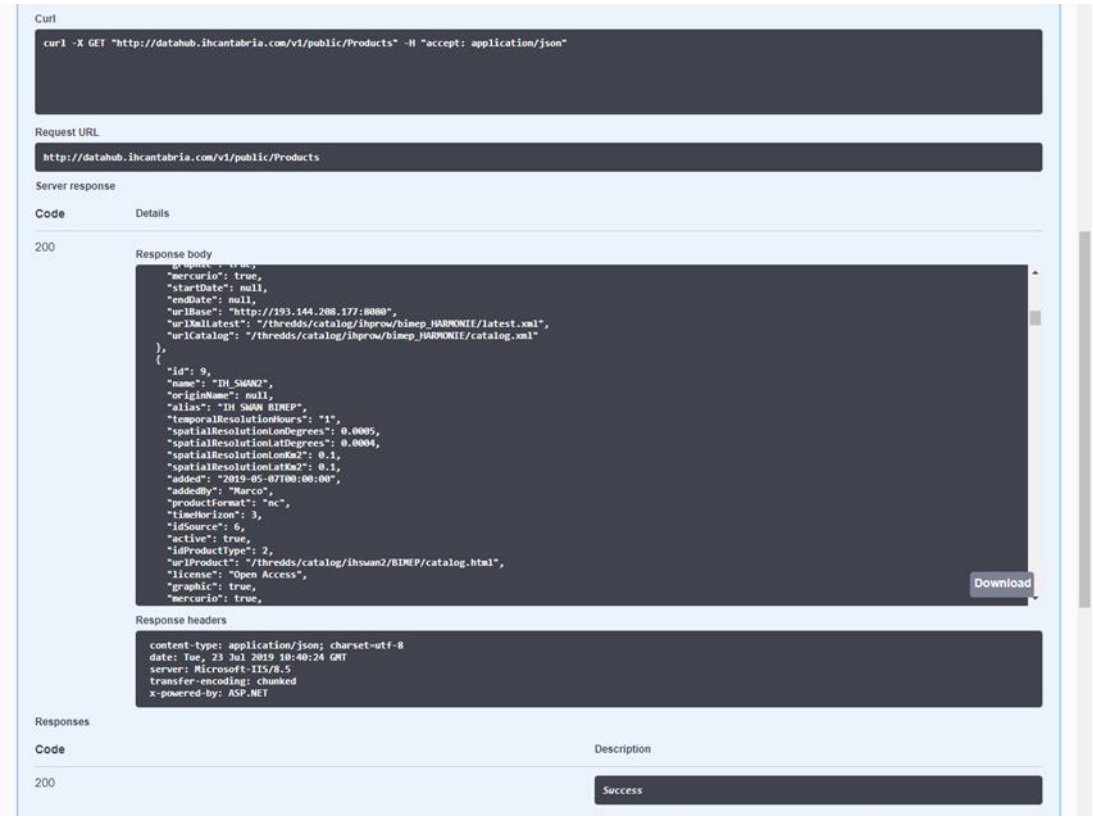

Figure 9. Query from the Data hub API

4 MONITORING SYSTEM

The infrastructure is being monitored 24 hours a day, 7 days a week, 365 days a year, ensuring business continuity and disaster recovery; with a Recovery Point Objective (RPO) and Recovery Time Objective (RTO) in accordance with the Service Level Agreement (SLA) for the services provided. The ability to find out what is happening on any operational system at any given time is crucial to provide a 24/7 Service. In this sense, Nagios is an open source system that offers monitoring and alerting services for servers, models, applications and services, see Figure 10.

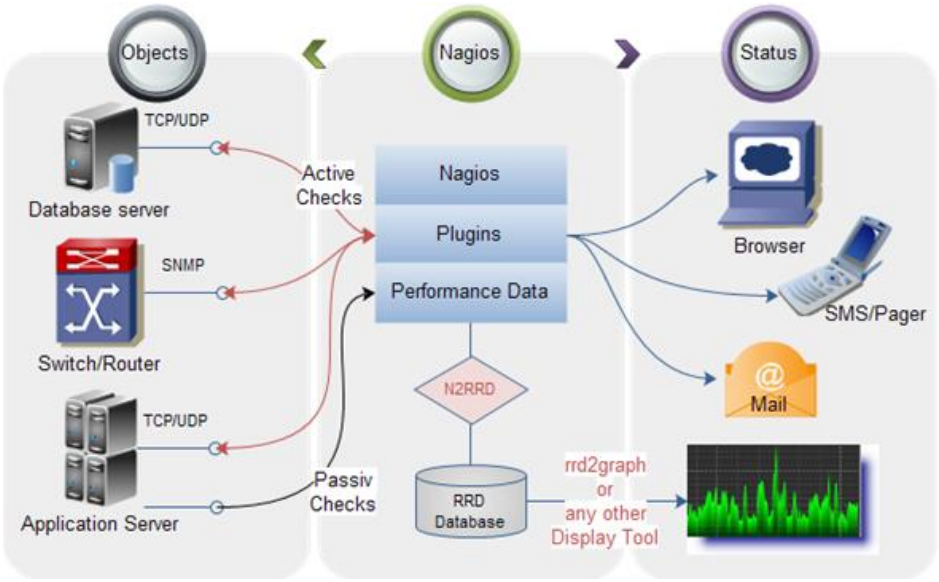

Figure 10. Monitoring System based on Nagios

Therefore, the status of the SOSeas service is constantly monitored. In order to provide access to the status of the service and all their components, a specific dashboard has been designed and developed. The dashboard of the Monitoring System, see Figure 11 showing the operational status of data providers. The System provides information about the lots of metocean data (wind, waves and currents), as well as information on the RAM System, CPU, storage, etc.

Figure 11. Dashboard of the Monitoring Service

In addition, the Monitoring System can be configured to receive alerts via mail or through team communication channels, such as Teams or Slack, see Figure 12.

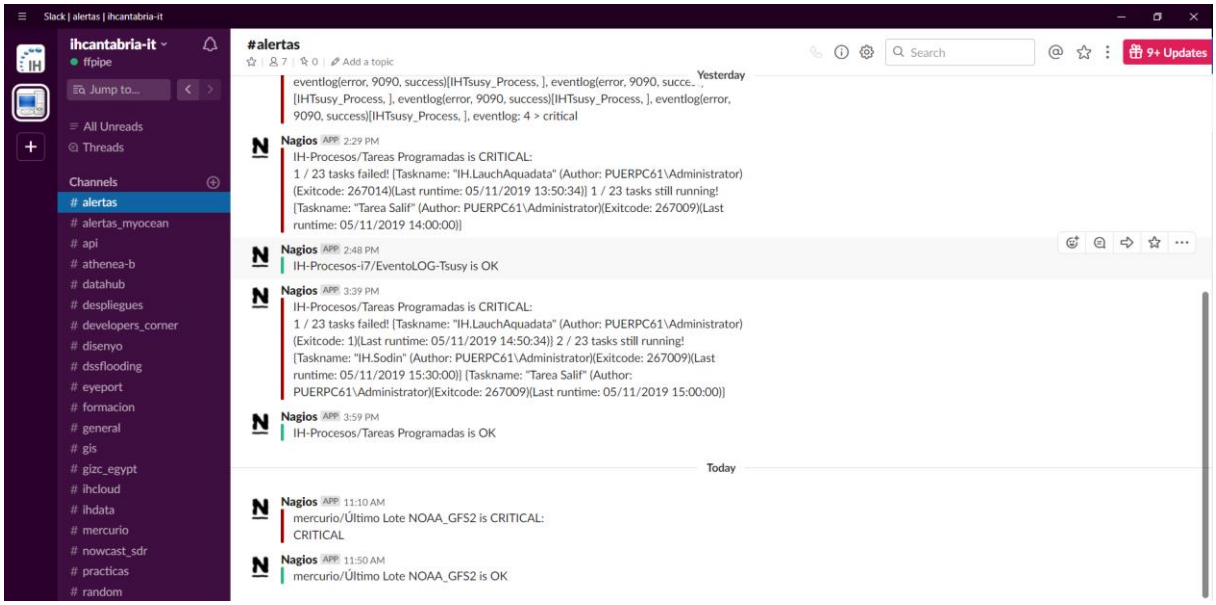

Figure 12. Nagios alerts integrated with Slack

5 WEkEO

Copernicus launched an initiative to facilitate access to Copernicus data and information services called "Data Information and Access Service" (DIAS) on the 20th of June 2018. DIAS is designed to improve users' ability to access as well as process Copernicus data and information by standardizing access to data through five cloud-based platforms: CREODIAS, MUNDI, ONDA, SOBLOO and WEKEO.

IHCantabria was selected as a user for the WEkEO Beta testing programme. Two virtual machines were requested to test the WEkEO Platform, see Figure 13.

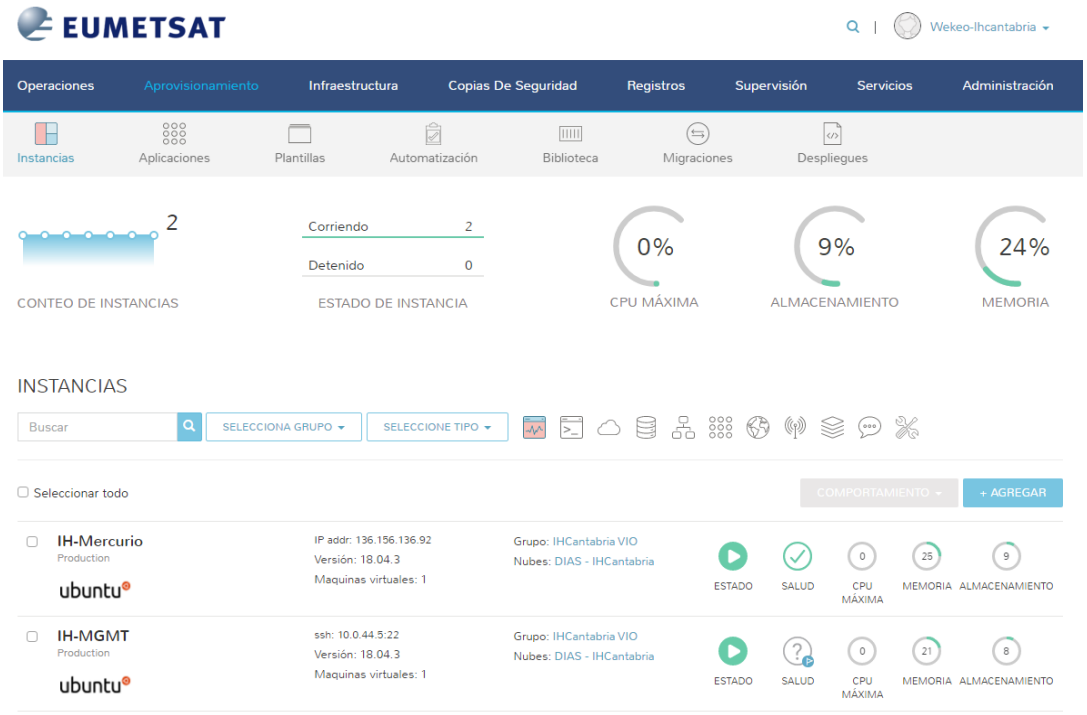

Figure 13. Virtual Machines at the WEkEO Platform

One of the main sections of the BackEnd architecture designed for the operational systems is the collection, transformation and publication of metocean products through standard interoperability protocols. The middleware in charge of the ETL (Extract, Transform & Load) and interoperability is called “Mercurio”, which collects meteocean data from different data providers, standardize and share them making use of THREEDs technology. Then, different Decision Support Systems are able to make use of the standardized products through interoperability access (WCS, WMS, etc.). Mercurio was also deployed at the WEkEO platform, see Figure 14, providing access to metocean products from different sources (CMEMS, NOAA, meteoGalicia, *Puertos del Estado*, etc.).

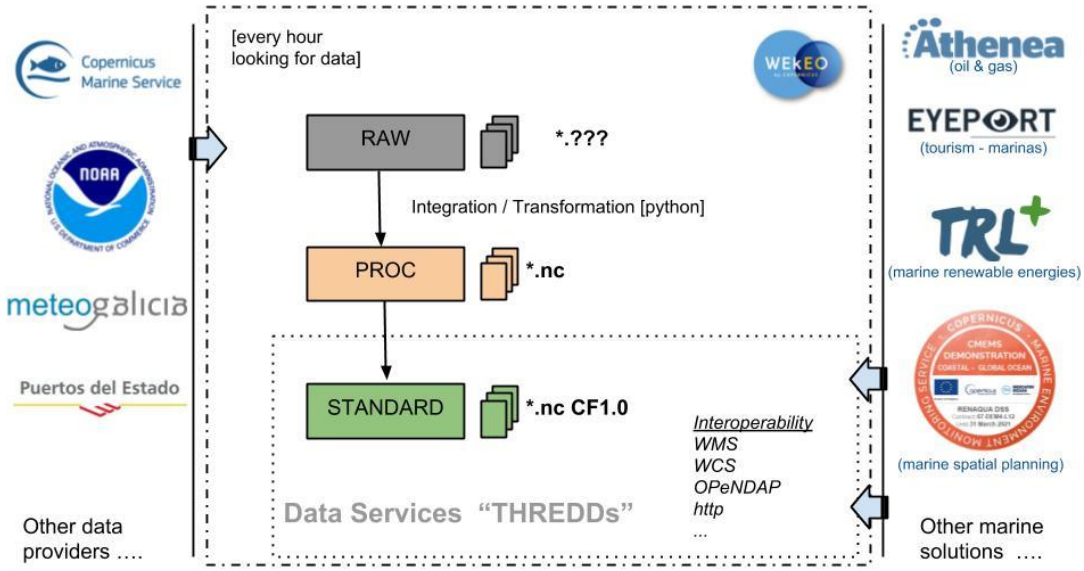

Figure 14. Architecture at WEkEO platform

In relation with the SOSeas Service, the main feature that WEkEO provides to the Service is the fast access to Global products. All metocean products must be transformed and standardized in order to be shared through interoperability protocols (THREDDs Data Server). However, Global Products can be extremely large to perform, in a timely and operational manner, such transformations. In this sense, the SOSeas Service points to the CMEMS Global and NOAA products that are currently being collected, standardized and shared at the Mercurio hosted at the WEkEO Platform. The following url provides access to the products hosted at WEkEO cloud platform.