The Atlantic Testing Platform for Maritime Robotics

Topic	ICT-09-2019-2020 (H2020)
Acronym	ATLANTIS
Title	The Atlantic Testing Platform for Maritime Robotics: New Frontiers for Inspection and Maintenance of Offshore Energy Infrastructures.
Project number	871571
Delivery date	30.11.2022
Deliverable number	D3.2 (D11)
Dissemination level	Public
Lead Beneficiary	SPACEAPPS

IT Systems and Supervisory Control Centre

SPACEAPPS, VTT, INESC, IQUA, UdG, ECA

Actions

	Action	Organisation	Date
Technical manager	Requested deliverable from the WP Leader and the deliverable Responsible.	VTT	16.11.2022
Deliverable Responsible	Prepared draft of the deliverable.	SPACEAPPS	24.11.2022
Technical Manager	Approved the draft as the first version.	VTT	30.11.2022
Quality Manager	Approved the final version of the document and saved to the “Final” folder.	UdG	30.11.2022
Project Coordinator	Approved the final version and sent to the EC.	INESC	30.11.2022

Disclaimer

This document does not represent the opinion of the European Union nor the European Commission is responsible for any use that might be made of its content. The ATLANTIS consortium cannot warrant that information contained in this document is free from risk and, neither the European Commission nor the ATLANTIS consortium parties are responsible for any use that may be made of the information contained therein.

This document may contain material, which is the copyright of certain ATLANTIS consortium parties, and may not be reproduced or copied without permission. The commercial use of any information contained in this document may require a license from the proprietor.

The sole responsibility for the content of this publication lies with the authors and all ATLANTIS consortium parties have agreed to full publication of this document.

This project has received funding from the European Union’s Horizon 2020 research and innovation programme, under the Grant Agreement no. 871571.

Acronyms

SCC Supervisory Control Centre

MCT Mission Control Team

AUV Autonomous Unmanned Vehicle

USV Unmanned Surface Vehicle

UxV Unmanned Vehicle

ROS Robot Operating System

GIS Geographic Information System

REST Representational State Transfer

UI User Interface

PDDL Planning Domain Definition Language

API Application Programming Interface

IMR Inspection, maintenance, and repair

ASV Autonomous Surface Vehicle

This project has received funding from the European Union's Horizon 2020 research and innovation programme, under the Grant Agreement no. 871571.

Table of Contents

This project has received funding from the European Union's Horizon 2020 research and innovation programme, under the Grant Agreement no. 871571.

List of Images

List of Tables

This project has received funding from the European Union's Horizon 2020 research and innovation programme, under the Grant Agreement no. 871571.

1. Introduction

The Supervisory Control Centre (SCC) is a technological ecosystem with the purpose to connect the Pilot operational team to the testbed and one or more Unmanned Vehicles (UxVs) performing IMR operations. The activities performed by the UxVs are associated with well-defined objectives. One activity that may include multiple actions will be called a Mission.

1.1. Purpose of the system

Design of the interoperability is part of the overall architecture of the ATLANTIS testbed, and it makes the link between vehicles, control centre and the rest of the pilot. Looking at the proposed architecture (Figure 1), we can identify that the communication, O&M data processing and the Supervisory Control Centre (SCC) will be stable systems during various phases of the testbed exploitation, while the sensors, robots and robot HMIs are elements specific for different use cases being tested on the platform.

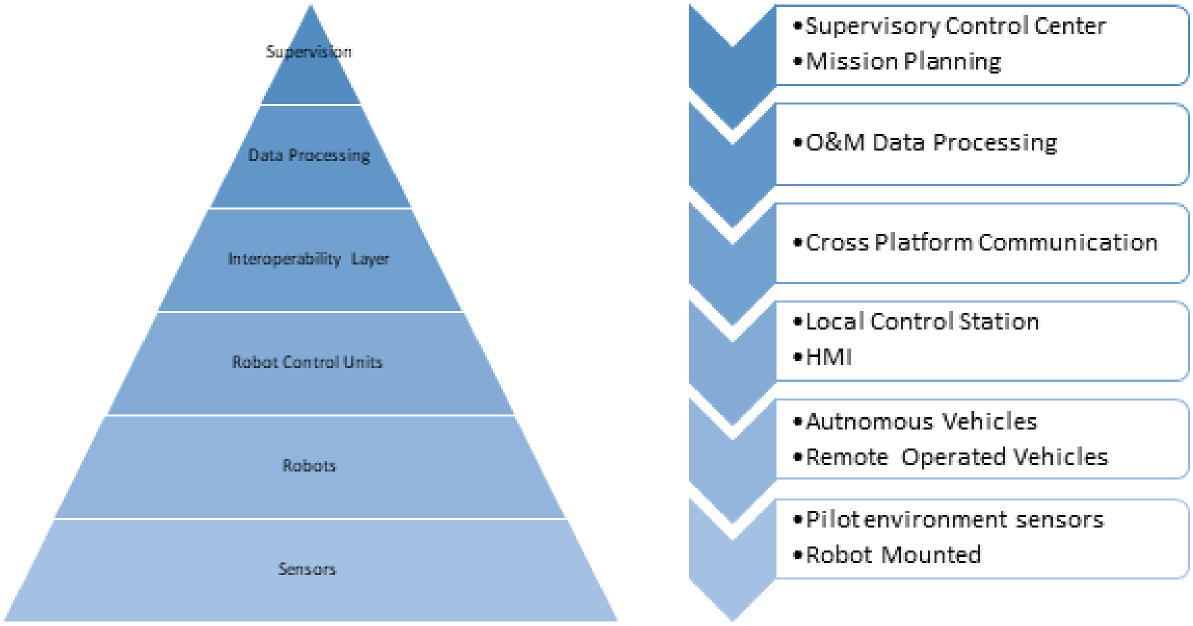


Figure 1. Technological layers in ATLANTIS System

By using this toolchain connected to the UxVs and the Pilot infrastructure, the operators will be able to:

- Monitor the state of the connected vehicles based on the sensors mounted on them. The state of the vehicle can include the GPS coordinates, altitude, orientation, velocities, battery levels or power related data or a non-directly observable state (Ex. Task, failure etc.)
- Visualise and analyse all the data received from the vehicles in the past or replay a mission that has already been performed.
- Issue high level commands to the vehicles and monitor the execution.
- Define missions together with their goals.

This project has received funding from the European Union's Horizon 2020 research and innovation programme, under the Grant Agreement no. 871571.

- Request the software planning tool to analyse mission input and output in order to produce a plan that include all UxVs for reaching the goal
- Deploy mission plans to the remote vehicles
- Download survey data for specific analysis.

From an abstract architectural perspective, the layer that links the robots with each other and with the onshore control centre is the *Interoperability Layer*. This layer has the main role of unifying the data types coming and going to the deployed robots. The higher layers will be able to have a unified access point to the data exchanged between robots named “data gateway”. The connection with the supervisory control centre will act just like another node in the network of assets.

Looking downward from the interoperability communication layer, there are four main data categories that are supported between assets:

- Info: each asset publishes the information about itself to all peers, thus being discoverable. The other nodes can exchange data with this asset based on this information.
- Commanding: a robot can receive commands from a control node.
- Monitoring: timestamped data used for generating situational awareness and the full history of the operation.
- Data retrieval: large amounts of data after performing a survey (images, pointclouds, raw or processed data).

Each one of the robots has a specific set of commands that can be engaged by the control centre. The interoperability layer has the role to advertise towards the control centre which are the capabilities of each robot and to accept from the supervisory control centre a command format that is generic for all the robots. Inside the interoperability, the commands received from the control centre are translated into each robot's command specifications.

In order to translate the generic command structure to the specific format, the interoperability layer has to understand each of the robot's capabilities. This will be done in an automatic way the first time a robot is connected to the ATLANTIS Testbed system. The configuration of the robot will be passed to the interoperability layer through a configuration file. The configuration file will be stored on the robot on-board computer and it will contain the list of supported commands, telemetry and data payloads, along with the robot namespace.

The interoperability software can communicate with an instance of the SCC launched in the cloud. This is possible by using access tokens and secure web protocols.

From a deployment view, the interoperability layer software will run on both the onshore facility and the offshore unmanned vehicles. The module running on shore will act as a data gateway between the SCC/Data Processing units and the rest of the system. The physical link between the onshore and offshore parts of the interoperability layer can be a satellite link, 4G, or the existing WindFloat communication infrastructure.

On the robot side, the module is deployed as a library. The library can be a dependency for the robot or the local robot control centre. From the Interoperability layer each robot control unit or robot will be linked through an API.

This project has received funding from the European Union’s Horizon 2020 research and innovation programme, under the Grant Agreement no. 871571.

1.2. ATLANTIS defined scenarios

The ATLANTIS vehicles should be used within different application contexts where complex missions should be performed autonomously without the support of a vessel. During task T2.1 the consortium has defined the IMR scenarios that are the main use cases for the ATLANTIS pilot.

In the case of SCC development we have been focused on four representative scenarios from the 8 proposed. By fulfilling the requirements for these four scenarios, the control centre will support all the others since the same communication visualisation and data storage apply. The four scenarios and the role of the control centre for each one of them is presented below.

Scenario 1: Inspection of blades and tower at WFA

For this scenario, there are two distinct operations that have to be performed: reach the inspection site and inspect the blades. In an offshore environment, reaching out the test site is best achieved using a surface vehicle, while the inspection of the blades is done by an aerial vehicle.

The proposed procedure for the inspection of the blade and towers involves an UAV equipped with inspection sensors mounted stable on a surface vessel equipped with a platform. The Autonomous Surface Vehicle (ASV) will navigate to the proximity of the tower in a specific predefined location. Once the ASV reaches the destination point the UAV will take off and perform the inspection of the tower and blades by following a predefined trajectory. During the inspection, the UAV is using the onboard camera to capture images of the installation. The resulting set of images or video files are the main output of the inspection activity. After the inspection flight, the UAV will land back on the USV which awaits in a rendez-vous spot.

From the SCC perspective the operators will have:

- ASV status monitoring
- UAV status monitoring
- GIS view monitoring the position of the vehicles
- Operation planning and execution
- Survey data view

Scenario 2: IMR of the transition piece or the floating structure

Inspection and Maintenance of the transition piece using unmanned vehicles involves beside reaching the target, accurate control over a surface vehicle for inspecting or performing maintenance tasks. In this scenario, the involved robot is an ASV that has onboard devices specific for inspection and maintenance.

From the SCC perspective the operators will have:

- ASV status monitoring
- GIS view monitoring the position of the vehicle
- Operation planning and execution
- Survey data view

Scenario 3: Repair of underwater floating wind turbine cables protection systems

This project has received funding from the European Union's Horizon 2020 research and innovation programme, under the Grant Agreement no. 871571.

As opposed with the first two scenarios, in this case the vehicle is submersible and can be either autonomous (AUV) or controlled by a human driver (ROV). The movement of the vehicle will include submerging and emerging. For the communication a much lower bandwidth is available through the acoustic channel since no RF links can be established in the case of non tethered AUVs.

From the SCC perspective the operators will have:

- USV / ROV status monitoring
- GIS view monitoring the position of the vehicles
- Operation planning and execution
- Survey data view

Scenario 4: Underwater monitoring over extended time periods

Another component is introduced in this scenario: the docking station. This station is well anchored to the seabed and has the role to hold, recharge and act as a communication proxy for an AUV that performs regular surveys over a specific area or component of the floating structure. The SCC must consider docking and undocking procedures for this scenario.

From the SCC perspective the operators will have:

- AUV status monitoring
- Docking Station status monitoring
- GIS view monitoring the position of the vehicles
- Operation planning and execution
- Survey data view
- Docking / Undocking status

1.3. Utilisation of the Control Centre

The SCC communications and computer systems are linked with the vehicles and the sensors mounted in the pilot. All the data produced is permanently recorded in the data store of the control centre and can be accessed at all time.

For specific IMR operations the SCC provides the tools to plan, supervise and analyse. One complete operation with a well-defined goal is called a *mission* and the control centre facilitates the three stages that form its life cycle: *Mission Preparation*, *Mission Execution* and *Mission Analysis*. The elements of each one of these stages, from the supervision perspective are described below:

Mission Preparation:

- The two vehicles are connected with the interoperability layer and there is a stable communication between the robots and the SCC
- The robots are publishing the telemetries associated with their operations. These telemetries should include:
    - Positional data as geographical position or a frame in a pre-defined coordinate system
    - Battery status
    - Alarms
    - State of the subsystems
- The control centre UI is configured to visualise the robot state in a single ecological view.

This project has received funding from the European Union's Horizon 2020 research and innovation programme, under the Grant Agreement no. 871571.

- For each type of mission the operator will define the parameters of the mission, parameters of the robot, the initial state and the goal of the mission using UI tools.
- The SCC computes the sequence of operations to be executed by either the autonomous vehicle or by the operator controlling the ROV.
- The SCC planner computes the estimated duration of each of the activities to be performed in the IMR mission.
- An operator has inspected and validated the sequence of operation.

Mission Execution

- The vehicles assigned to perform the mission are running, available and in a state compatible with the sequence operations to be performed.
- An operator launches the sequence of operations to be executed by the autonomous vehicles or by the ROV operators.
- The vehicles and vehicle operators acknowledge receiving the plan and the plan execution is starting.
- The vehicles publish the beginning and end of each step and the supervisory control center displays it on the mission control views.
- The vehicles are publishing their state and the state of the environment. The mission control views are displaying the remote status creating situational awareness from the operators perspective.
- In case of step abortion or mission failure the operator is informed about the event and about the procedure that was automatically engaged by the vehicle.
- The vehicles upload the mission results (e.g. survey images) to the SCC and the operator is able to either visualise or download these data.

Mission Analysis

- The operator can select a particular executed mission and access the mission configuration and established plan.
- The operator can access the survey data of a mission.
- The operator is consulting the telemetry produced by the vehicles during the mission by selecting the time interval in which the mission was executed.
- The data is persistent and can be used for the forensic analysis of each individual mission.
- The telemetry of robots and sensors mounted in the pilot is accessible during and outside of the mission time windows.

| Page

This project has received funding from the European Union's Horizon 2020 research and innovation programme, under the Grant Agreement no. 871571.

2. IT systems

The infrastructure of the SCC onshore is composed of desktop hosting the SCC server and 3 other desktops as clients of the SCC. Onshore, robots can connect to the SCC using Wifi. The SCC is linked to a router providing Wifi. Thus, when a robot connects to this network, it is on the local network of the SCC and then can be discovered automatically by the SCC.

Robots that are operating offshore can also be connected to SCC using a 4G Router with Wifi installed on a vessel near the site of operation. Using the VPN, robots can be on the same network as the onshore SCC which allows the communication between them.

- Communication between SCC and robots onshore (wifi local network)
- Communication between SCC and offshore (4G modem - VPN)
- SCC as a desktop
- 3 SCC clients desktop

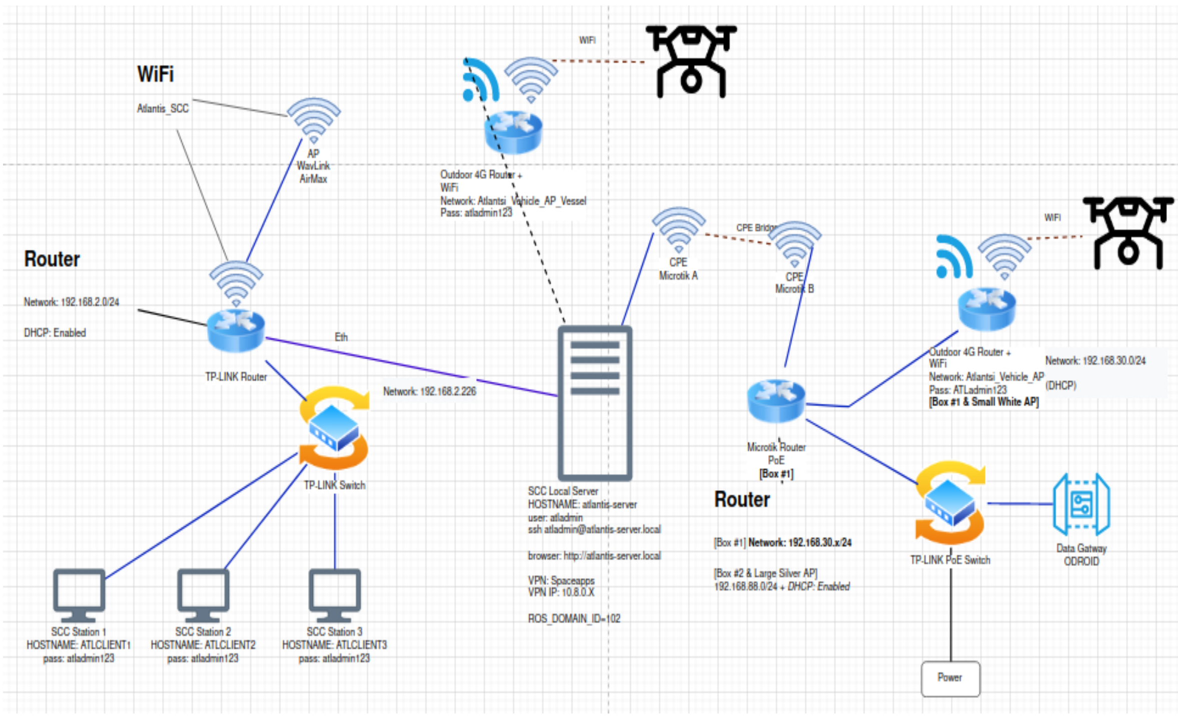


Figure 2 : Overall SCC network architecture

2.1. SCC Hardware Requirements Compliance Matrix

Mission planner is a high-level software operated by the SCC, aimed at providing a formal definition of the mission. The mission planner should control both the USV and the AUV vehicles and should allow the user to define each mission parameters.

The compliance for the proposed specifications given in D2.2 - System Requirements and Platform Interfaces. Below there are listed the proposed SCC requirements and the compliance of the resulting work. All the requirements have been defined in D2.2, chapter 5.8 SCC.

This project has received funding from the European Union's Horizon 2020 research and innovation programme, under the Grant Agreement no. 871571.

Requirement	Description	Compliance	Observation	Setup
HW_Ga_1 SCC Satellite communication	The SCC will communicate with the satellite linked assets through the public network and the satellite communication service provider infrastructure	Yes	Satellite communication tested but not mounted on the boat	No
HW_Gb_1 SCC Wired Ethernet	The computers and servers installed onshore will be connected using an Ethernet local network	Yes		Yes
HW_Gc_1 SCC Wired Optic	Given access to the windfloat line communication infrastructure, the SCC will be connected to it using an Ethernet link. The connection to the WFA will be done through a gateway computer which will separate the remote network.	No	The wired connection with the WFA infrastructure is not available to be used outside of WFA. Communication will be done using the alternative communication setup.	No
HW_Gl_1 SCC Local Servers	The SCC components and the O&M will be deployed on the onshore facility on computing systems with enough storage and computing power to run the desired applications.	Yes		Yes
HW_Gl_2 SCC Operator Computers	The operators will use pc’s to access the SCC user interface. Additional monitors will be available for a better view of the operation.	Yes		Yes

Table 1. Compliance Matrix of the SCC

This project has received funding from the European Union’s Horizon 2020 research and innovation programme, under the Grant Agreement no. 871571.

3. Remote integration results

A process of remote integration with the robot from partners has been started to integrate the SCC using an online instance. This process has allowed us to do a basic integration with the robots. These remote integrations helped us fine tune the software from the SCC and also helped the partners on the software side of their robots. One of the main goals was to prepare, at best, the field test since they are limited in time. This process gave us more insight about the needs of the different robots.

The following data has been integrated remotely:

- Imagery
- Video
- Numerical Telemetry
- Mission planning
- Joint States
- GIS + Odometry (Position & orientation)

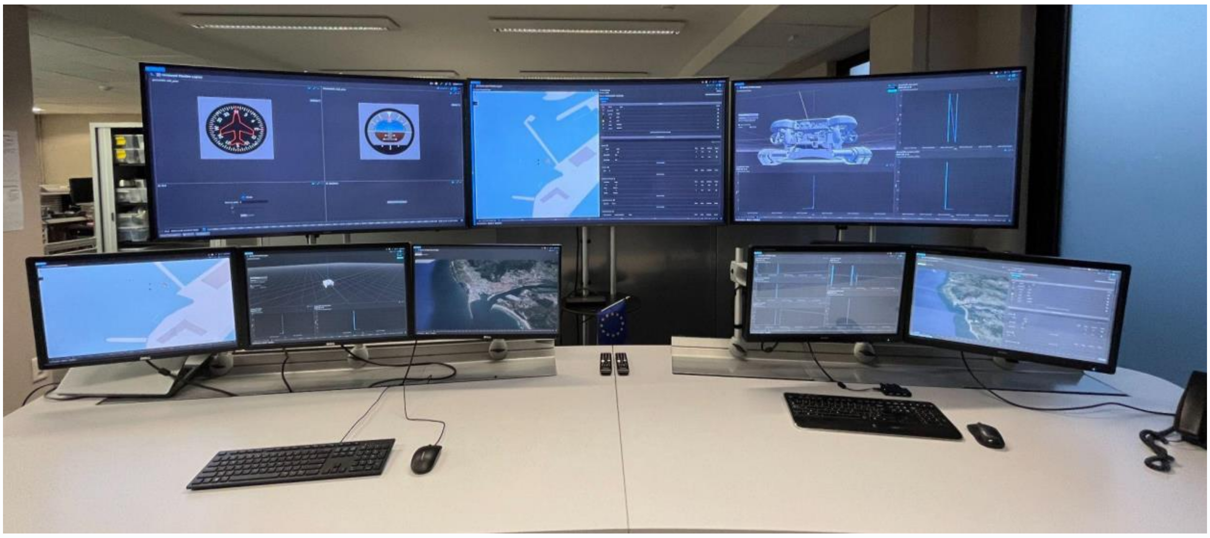


Figure 3: Remote instance of the SCC setup in the Space Applications laboratory.

3.1. Architecture Overview

The SCC has been deployed on a server possessed by Space Applications Services. The goal here was to provide an instance for partners in the project. With this instance in place, the partners had access to the SCC at any time. To connect the robots to the SCC, we use a ROS bridge with a specific domain ID to discover the robots on the network. To do so, we need the robots and the bridge on the same network. One of the easiest solutions was to set up a VPN connecting the robots and the server. Once the robots are connected to the VPN, the ROS bridge can easily discover them. Afterwards, they will be connected to the SCC and will be added to the user interface of the SCC.

In the original scenario, the robots and the SCC are on the same local network. The use of the VPN mimics that behaviour over the internet. With this setup, we organised sessions of remote integrations

This project has received funding from the European Union's Horizon 2020 research and innovation programme, under the Grant Agreement no. 871571.

with partners. They were able to test on their side if all the telemetries were discovered and visible on the UI. If problems occurred on the SCC side, they were relayed to Space Apps for further investigation and solving.

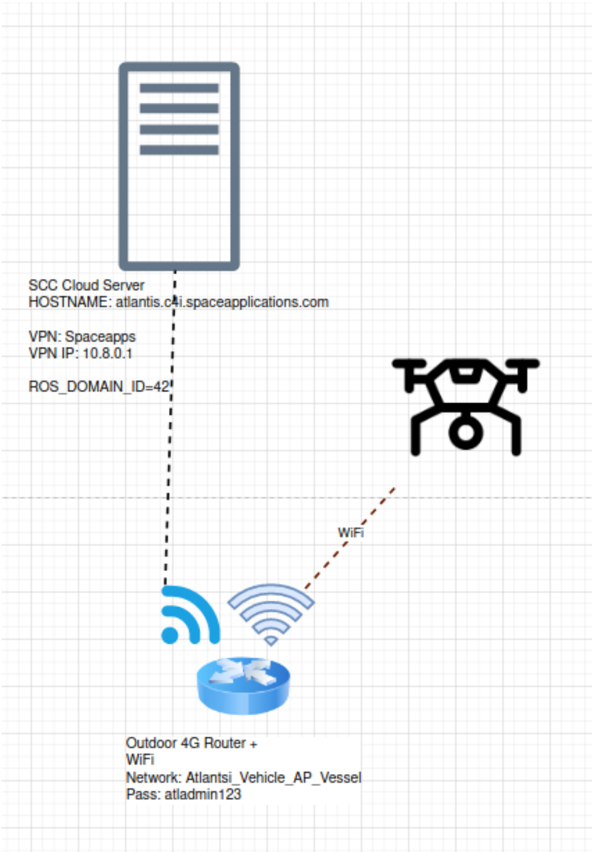


Figure 4 : Network architecture of the offshore set up and the cloud

3.2. ROV - RovingBat

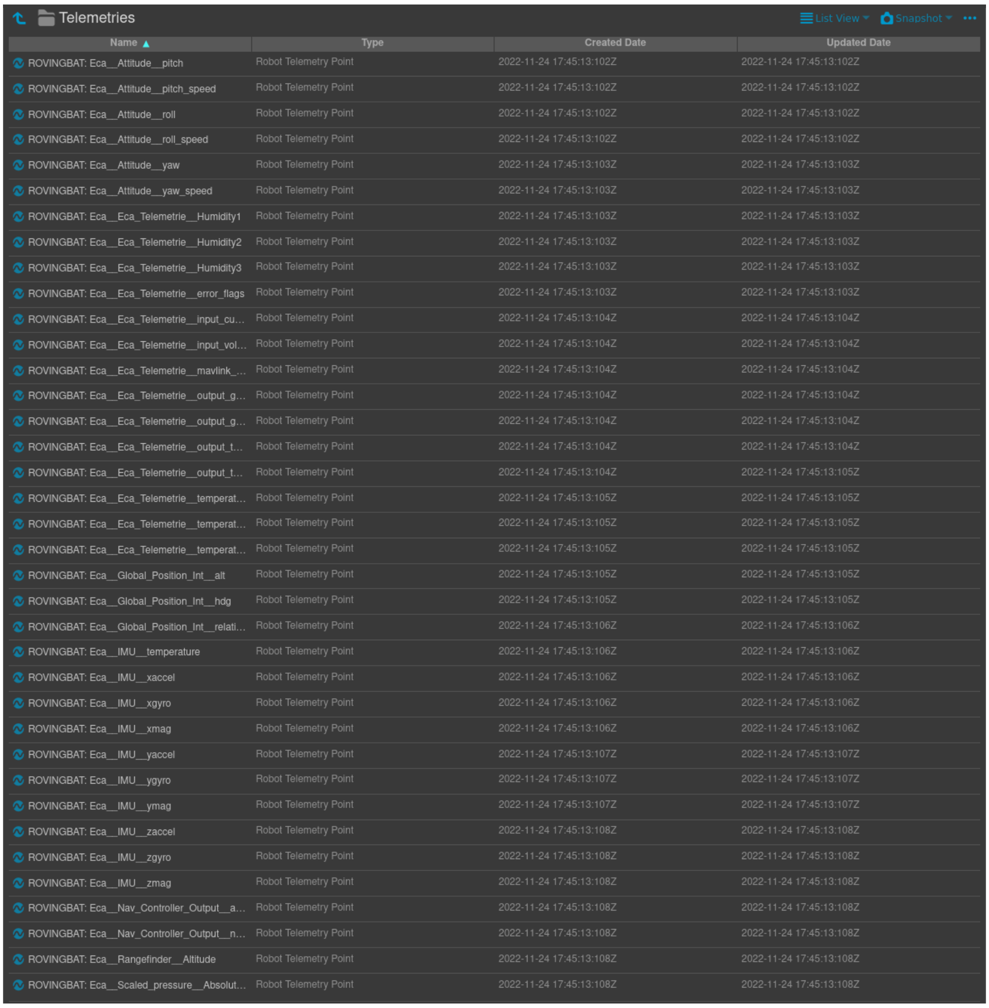
With ECA, we first integrated their ROV remotely by receiving the configuration file of the robot.

Once the robot was discovered, we check if all the telemetries from the config file appear on the UI of the SCC. At first, an issue occurred as the numerical telemetries were visible but the data from the telemetries were not. This issue was caused by the backslash symbol in the names of the telemetries written in the configuration file of the robot. Once the issue was solved, we were able to monitor the live numerical data flowing from the RovingBat.

The RovingBat has a robotic arm from REACH Robotics. This arm also needed to be integrated with the SCC. The relevant data useful for the operator are the angles of the different joints of the arm. We call them “joint states”. This data is not relevant as numbers but is relevant when the arm is represented as a 3D model with each joint at the same value as the real robotic arm. The SCC has a local 3D map that allows the display of 3D models. Once the 3D model of the RovingBat and the arm has been uploaded to the SCC, our goal is to update the joint states received by the arm. We faced some issues related to the format of the data but these issues were solved pretty quickly and thus the joint states telemetry was successfully integrated.

This project has received funding from the European Union’s Horizon 2020 research and innovation programme, under the Grant Agreement no. 871571.

The RovingBat has multiple cameras. The SCC accepts only RTSP streams. When doing the remote integrations with such devices, they need to be also on the same network as the SCC. Thus the RTSP streams should have the same subnet as the VPN. This was one of the first issues we faced but due to limited time we could not integrate remotely.


Telemetries

Name	Type	Created Date	Updated Date
ROVINGBAT: Eca__Attitude__pitch	Robot Telemetry Point	2022-11-24 17:45:13:102Z	2022-11-24 17:45:13:102Z
ROVINGBAT: Eca__Attitude__pitch_speed	Robot Telemetry Point	2022-11-24 17:45:13:102Z	2022-11-24 17:45:13:102Z
ROVINGBAT: Eca__Attitude__roll	Robot Telemetry Point	2022-11-24 17:45:13:102Z	2022-11-24 17:45:13:102Z
ROVINGBAT: Eca__Attitude__roll_speed	Robot Telemetry Point	2022-11-24 17:45:13:102Z	2022-11-24 17:45:13:102Z
ROVINGBAT: Eca__Attitude__yaw	Robot Telemetry Point	2022-11-24 17:45:13:103Z	2022-11-24 17:45:13:103Z
ROVINGBAT: Eca__Attitude__yaw_speed	Robot Telemetry Point	2022-11-24 17:45:13:103Z	2022-11-24 17:45:13:103Z
ROVINGBAT: Eca__Eca_Telemetrie__Humidity1	Robot Telemetry Point	2022-11-24 17:45:13:103Z	2022-11-24 17:45:13:103Z
ROVINGBAT: Eca__Eca_Telemetrie__Humidity2	Robot Telemetry Point	2022-11-24 17:45:13:103Z	2022-11-24 17:45:13:103Z
ROVINGBAT: Eca__Eca_Telemetrie__Humidity3	Robot Telemetry Point	2022-11-24 17:45:13:103Z	2022-11-24 17:45:13:103Z
ROVINGBAT: Eca__Eca_Telemetrie__error_flags	Robot Telemetry Point	2022-11-24 17:45:13:103Z	2022-11-24 17:45:13:103Z
ROVINGBAT: Eca__Eca_Telemetrie__input_cu...	Robot Telemetry Point	2022-11-24 17:45:13:104Z	2022-11-24 17:45:13:104Z
ROVINGBAT: Eca__Eca_Telemetrie__input_vol...	Robot Telemetry Point	2022-11-24 17:45:13:104Z	2022-11-24 17:45:13:104Z
ROVINGBAT: Eca__Eca_Telemetrie__mavlink_...	Robot Telemetry Point	2022-11-24 17:45:13:104Z	2022-11-24 17:45:13:104Z
ROVINGBAT: Eca__Eca_Telemetrie__output_g...	Robot Telemetry Point	2022-11-24 17:45:13:104Z	2022-11-24 17:45:13:104Z
ROVINGBAT: Eca__Eca_Telemetrie__output_g...	Robot Telemetry Point	2022-11-24 17:45:13:104Z	2022-11-24 17:45:13:104Z
ROVINGBAT: Eca__Eca_Telemetrie__output_t...	Robot Telemetry Point	2022-11-24 17:45:13:104Z	2022-11-24 17:45:13:104Z
ROVINGBAT: Eca__Eca_Telemetrie__output_t...	Robot Telemetry Point	2022-11-24 17:45:13:104Z	2022-11-24 17:45:13:105Z
ROVINGBAT: Eca__Eca_Telemetrie__temperat...	Robot Telemetry Point	2022-11-24 17:45:13:105Z	2022-11-24 17:45:13:105Z
ROVINGBAT: Eca__Eca_Telemetrie__temperat...	Robot Telemetry Point	2022-11-24 17:45:13:105Z	2022-11-24 17:45:13:105Z
ROVINGBAT: Eca__Eca_Telemetrie__temperat...	Robot Telemetry Point	2022-11-24 17:45:13:105Z	2022-11-24 17:45:13:105Z
ROVINGBAT: Eca__Global_Position_Int__alt	Robot Telemetry Point	2022-11-24 17:45:13:105Z	2022-11-24 17:45:13:105Z
ROVINGBAT: Eca__Global_Position_Int__hdg	Robot Telemetry Point	2022-11-24 17:45:13:105Z	2022-11-24 17:45:13:105Z
ROVINGBAT: Eca__Global_Position_Int__relati...	Robot Telemetry Point	2022-11-24 17:45:13:106Z	2022-11-24 17:45:13:106Z
ROVINGBAT: Eca__IMU__temperature	Robot Telemetry Point	2022-11-24 17:45:13:106Z	2022-11-24 17:45:13:106Z
ROVINGBAT: Eca__IMU__xaccel	Robot Telemetry Point	2022-11-24 17:45:13:106Z	2022-11-24 17:45:13:106Z
ROVINGBAT: Eca__IMU__xgyro	Robot Telemetry Point	2022-11-24 17:45:13:106Z	2022-11-24 17:45:13:106Z
ROVINGBAT: Eca__IMU__xmag	Robot Telemetry Point	2022-11-24 17:45:13:106Z	2022-11-24 17:45:13:106Z
ROVINGBAT: Eca__IMU__yaccel	Robot Telemetry Point	2022-11-24 17:45:13:107Z	2022-11-24 17:45:13:107Z
ROVINGBAT: Eca__IMU__ygyro	Robot Telemetry Point	2022-11-24 17:45:13:107Z	2022-11-24 17:45:13:107Z
ROVINGBAT: Eca__IMU__ymag	Robot Telemetry Point	2022-11-24 17:45:13:107Z	2022-11-24 17:45:13:107Z
ROVINGBAT: Eca__IMU__zaccel	Robot Telemetry Point	2022-11-24 17:45:13:108Z	2022-11-24 17:45:13:108Z
ROVINGBAT: Eca__IMU__zgyro	Robot Telemetry Point	2022-11-24 17:45:13:108Z	2022-11-24 17:45:13:108Z
ROVINGBAT: Eca__IMU__zmag	Robot Telemetry Point	2022-11-24 17:45:13:108Z	2022-11-24 17:45:13:108Z
ROVINGBAT: Eca__Nav_Controller_Output__a...	Robot Telemetry Point	2022-11-24 17:45:13:108Z	2022-11-24 17:45:13:108Z
ROVINGBAT: Eca__Nav_Controller_Output__n...	Robot Telemetry Point	2022-11-24 17:45:13:108Z	2022-11-24 17:45:13:108Z
ROVINGBAT: Eca__Rangefinder__Altitude	Robot Telemetry Point	2022-11-24 17:45:13:108Z	2022-11-24 17:45:13:108Z
ROVINGBAT: Eca__Scaled_pressure__Absolut...	Robot Telemetry Point	2022-11-24 17:45:13:108Z	2022-11-24 17:45:13:108Z

Figure 5 : Telemetry sample from RovingBat

3.2. AUV - Sparus & Girona

The AUV was integrated with the SCC using the interop. The following has been tested:

- **VPN connection:** SCC and robot connected to the same network using OpenVPN connection.
- **Discovery and identification of the Robot:** Robot automatically discovered by the SCC by receiving the robot config.
- **Boolean and Numerical Telemetry:** Reception and visualisation of the robot telemetries.
- **Poses:** Reception and visualisation of the pose (position and orientation) of the robot in real time on the SCC.
- **Mission Planning:** Mission configuration, planning and deployment of the mission on the robot have been successfully tested.

This project has received funding from the European Union's Horizon 2020 research and innovation programme, under the Grant Agreement no. 871571.

3.2. ASV - Zarco

The ASV to be used during Scenario 1 in the ATLANTIS demonstration is Zarco (Figure 23), provided by INESC TEC. It will serve as a surface vehicle bringing the UAV close to the target of inspection, wait for the inspection to finish and upload data.

The ASV was integrated with the SCC using the interop. The following has been tested:

- **VPN connection:** SCC and robot connected to the same network using OpenVPN connection.
- **ROS1 to ROS2 bridge:** ROS1 messages translated and republished in a ROS2 format.
- **ROSbag record and replay:** ROS2 record of the telemetries topic and the config topic on the robot side and replay of the data on the SCC side.
- **Discovery and identification of the Robot:** Robot automatically discovered by the SCC by receiving the robot config.
- **Boolean and Numerical Telemetry:** Reception and visualisation of the robot telemetries.

The Zarco config was published and successfully added to the SCC list of robots (Figure 6)

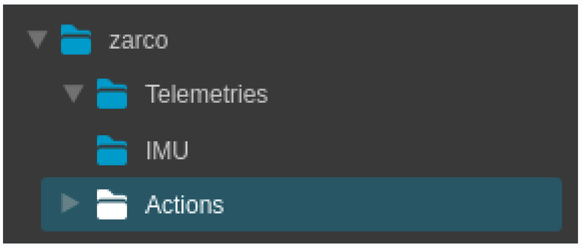


Figure 6. Zarco robot showing in the SCC UI after discovery

3.2. UAV - Crow

In the scope of Scenario 1, a collaboration between an ASV and a UAV is foreseen. Both of these vehicles are provided by INESC and will be connected to the rest of the system. The name of the UAV is CROW.

The UAV was integrated with the SCC using the interop. The following has been tested:

- **VPN connection:** SCC and robot connected to the same network using OpenVPN connection.
- **ROS1 to ROS2 bridge:** ROS1 messages translated and republished in a ROS2 format.
- **ROSbag record and replay:** ROS2 record of the telemetries topic and the config topic on the robot side and replay of the data on the SCC side.
- **Discovery and identification of the Robot:** Robot automatically discovered by the SCC by receiving the robot config.
- **Boolean and Numerical Telemetry:** Reception and visualisation of the robot telemetries.

This project has received funding from the European Union's Horizon 2020 research and innovation programme, under the Grant Agreement no. 871571.

4. Onshore integration results

After the integration was performed remotely using the cloud deployment of SCC, the robots were integrated to the local instance of the SCC at Viana do Castello.

4.1. Network architecture

On the onshore site, the SCC has been set up to allow any robot being part of the same network to be registered, operated, and monitored from the SCC. For this purpose, the following configuration has been used (Figure 7):

- 1 SCC server: Desktop with port 22 (SSH) and 80 (HTTP) opened.
- 3 SCC clients: small computers acting as user interface clients.

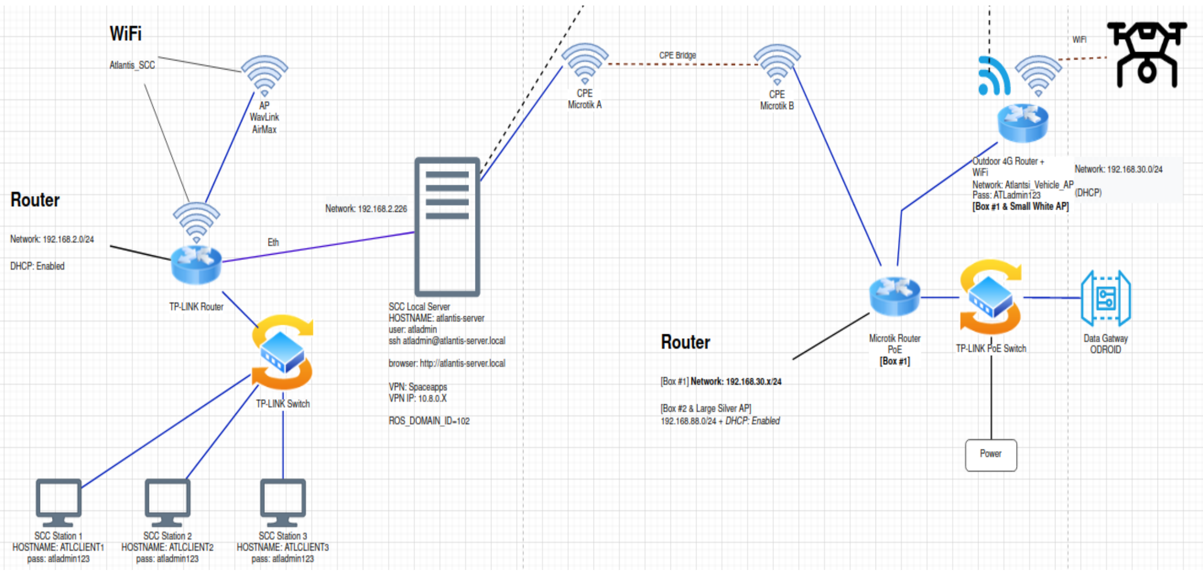


Figure 7 : Network architecture of the SCC and the onshore set up

4.2. SCC setup

A local instance of the SCC has been installed onsite at Viana Do Castello. The SCC is running on a Ubuntu 20.04 computer. The different software needed by the SCC were deployed on the computer using docker and services. A VPN connection has also been set up on the computer to be able to SSH on the desktop and to visualise and interact with the graphical user interface from anywhere.

Any computer connected to the VPN can then access to the graphical user interface, or the computer itself, by connecting to the SCC IP address in the VPN network or in the local network, on port 80 for the UI and port 22 for the SSH connection.

This also makes the UI available on any device equipped with a web browser such as laptops, tablets, or even smartphones.

This project has received funding from the European Union's Horizon 2020 research and innovation programme, under the Grant Agreement no. 871571.

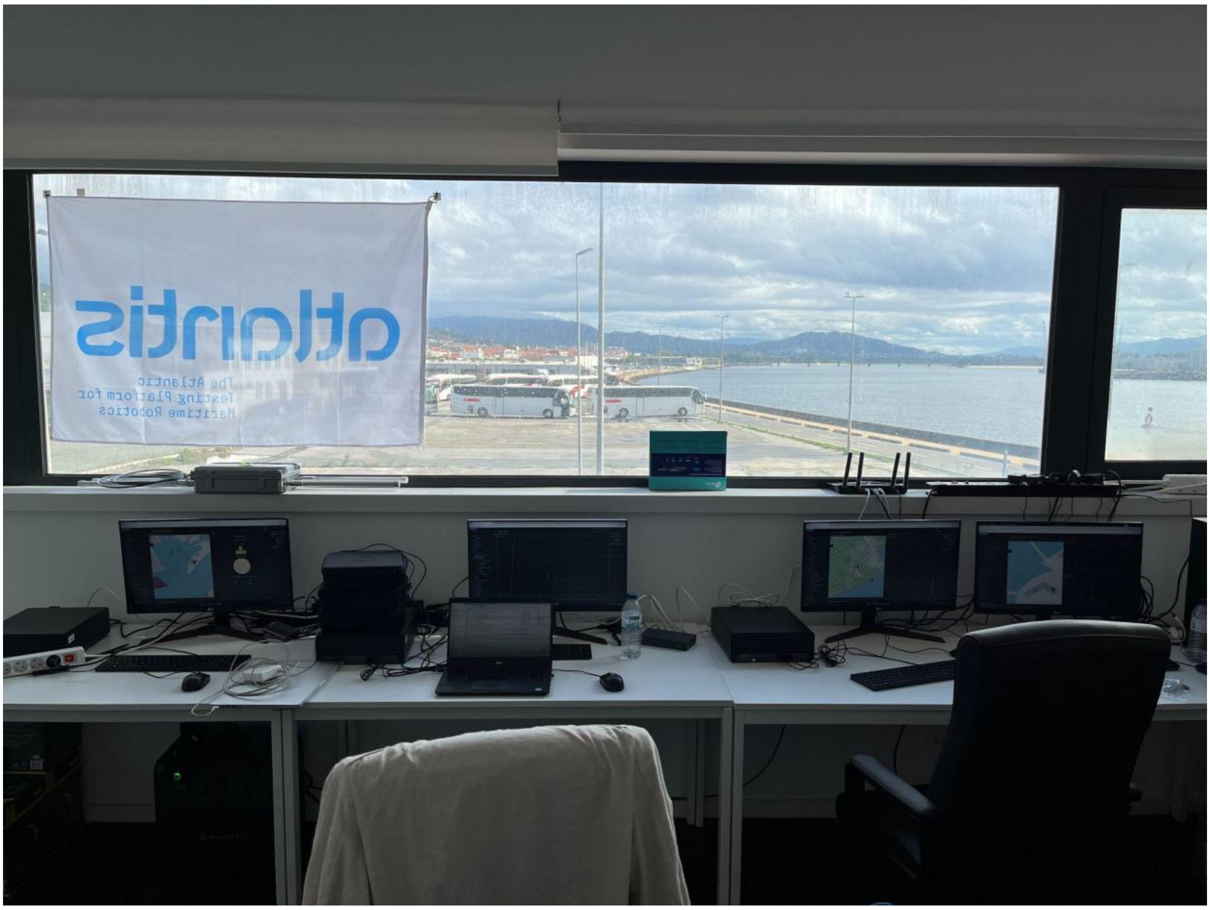


Figure 8. Onsite SCC desk in the ATLANTIS Pilot equipped

4.3. Robot monitoring results

During the onshore integration periods, the different robots have been tested and integrated to the SCC. The following data have been integrated:

- Robot configuration.
- Numerical and boolean telemetries.
- Images.
- Odometry.
- RTSP video streams.
- Mission deployment.
- 3D model articulations.

Robots	Config	Numerical	Odometry	Images	Videos	3D	Mission
UAV - Raven	X	X	X	X			
USV - Nautilus	X	X	X	X			
USV - Zarco	X	X	X				

This project has received funding from the European Union's Horizon 2020 research and innovation programme, under the Grant Agreement no. 871571.

ROV - RovingBat	X	X	X		X	X	
AUV - Girona1000	X	X	X				X

Table 2: Interoperability tests results by robots.

Grey cells = not relevant for the robot.

4.3.1. Interoperability integration

For all the tests performed during the remote and onsite integration sessions, each robot provider was using the CPP interoperability library provided by Space Applications to integrate their robots with the SCC. For every robot, a configuration file has been defined to receive data and send commands from and to the robot.

This file was sent every second by the robots to indicate that they are online. The configuration file contains all the ROS2 topics used by the robot to send data classified by data type. In addition, it also contains a security access token to connect to the SCC. The token was required for the cloud instance of the SCC but not for the local instance onsite as the robots were in the same network as the SCC.

All the robots successfully managed to connect to the SCC onsite for all the onsite and remote integration sessions.

4.3.2. GIS monitoring

During the July 2022 and the September 2022 integration sessions the Zarco GIS monitoring has been tested. The USV indicated the ROS topic used for the GPS in its configuration file and the SCC was able to display it as a GIS map. On Figure 9 we can see Zarco moving on the ground in front of the SCC building:

- The live position of the robot is represented by an icon, an orange boat icon for Zarco
- By default, an icon is facing the north.
- The history of the GIS position is displayed as a path from the same colour as the one of the robots to identify each one of them.

This project has received funding from the European Union's Horizon 2020 research and innovation programme, under the Grant Agreement no. 871571.

Figure 9: USV Zarco GIS view on the SCC.

During the same integration sessions the Raven UAV GIS monitoring has also been tested in addition to the odometry. The configuration of the UAV contained the GPS ROS2 topic and the odometry ROS2 topic.

At the opposite to the Zarco USV, Raven initialised only once its GPS coordinates and then sent its coordinates in its local coordinates system. As shown on Figure 9:

- The SCC computes the GIS coordinates of the robot by combining its GPS position to its local position.
- The orientation is also displayed by rotating the icon of the robot.
- The altitude coming also from the GPS and the odometry data of the robot is displayed as a graph.
- The SCC will compute the altitude by also combining the two data.

Raven also sent its front camera images as ROS2 images to the SCC using the CPP API. Images, GPS, odometry and numerical telemetries were received simultaneously on the SCC. On Figure 11 we can see a synchronised view of the GPS position, altitude and images. The data visible on the SCC on Figure 11 represents the real situation Figure 10.

The GIS position and the altitude are both stored in the database of the SCC to be accessible after the tests.

This project has received funding from the European Union’s Horizon 2020 research and innovation programme, under the Grant Agreement no. 871571.

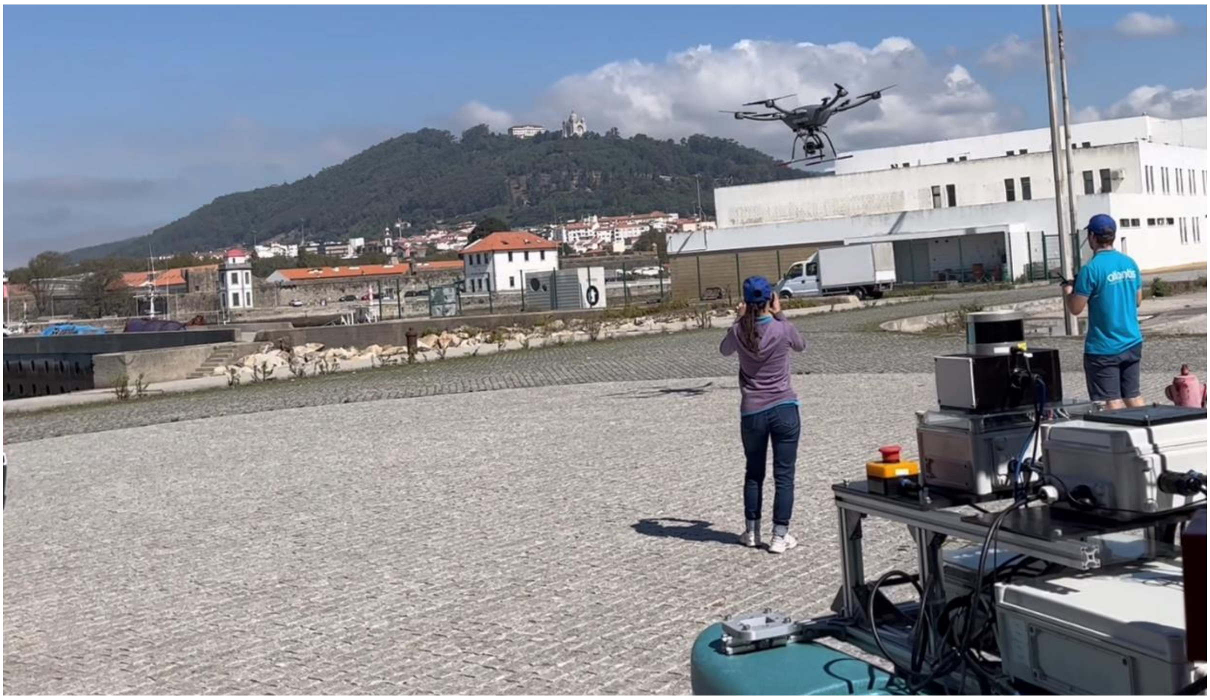


Figure 10: Raven UAV flying at Viana do Castello.

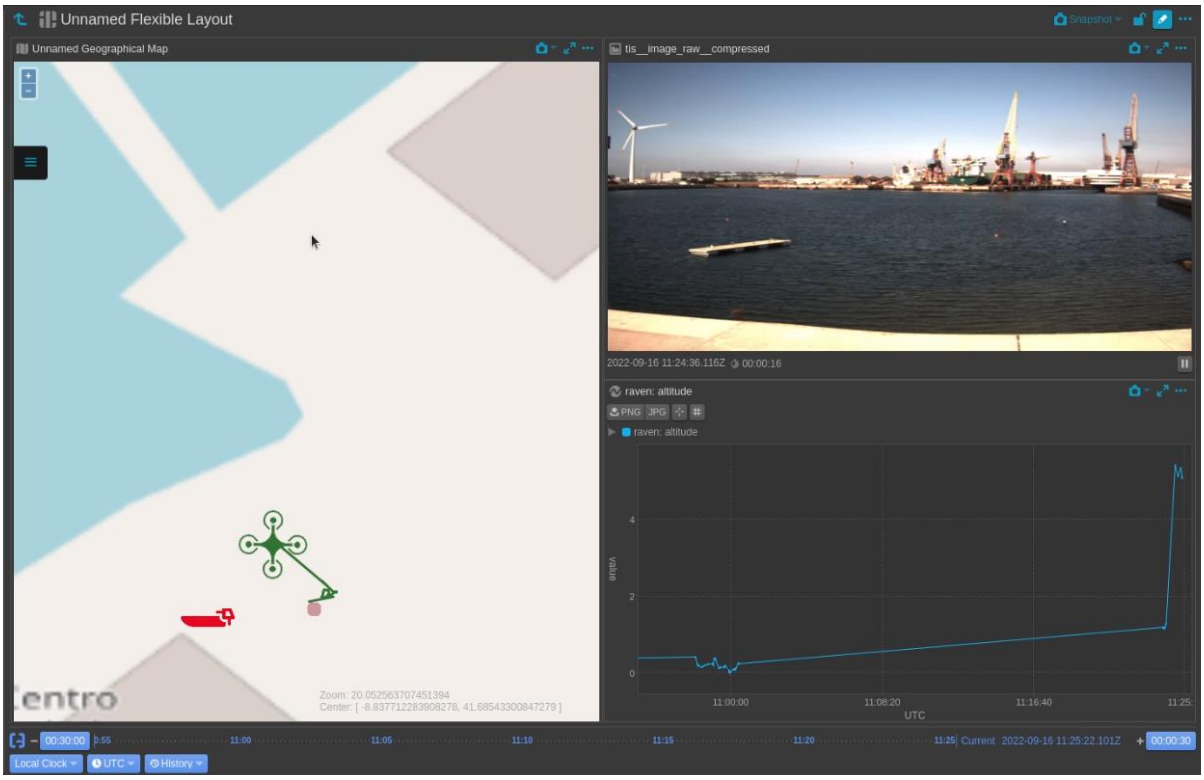


Figure 11: GIS, odometry, live images and altitude telemetries from Raven UAV monitored on the SCC.

This project has received funding from the European Union’s Horizon 2020 research and innovation programme, under the Grant Agreement no. 871571.

4.3.3. Video streams

The ROV from ECA has been integrated during the November 2022 tests. Prior to those tests, remote integration has been done to integrate the numerical telemetries and odometry. During the integration of the RovingBat one of the main focus was the RTSP video streams integration.

The ROV possesses multiple cameras. The following two RSTP streams were integrated:

- Navigation camera: Camera in front of the ROV.
- TZ camera: Camera oriented toward the robotic arm of the ROV.

The two RTSP streams links were added to the configuration file of the ROV using the IP address of each camera in the SCC network. Each link is associated with a name that is then automatically displayed on the SCC once the ROV publishes the ROS2 configuration of the robot.

On Figure 12 we can see the RovingBat facing a wood panel and on Figure 13 the RTSP stream of the front camera (Navigation camera) displayed in real time on the SCC. Figure 14 shows the TZ camera stream with the lens focusing on cable displayed in real time on the SCC.

Each RTSP stream was recorded into the SCC database. The streams are truncated into small videos of 5min that can be played directly on the SCC even during the recording of the stream.

Figure 12: RovingBat ROV on the coast at the SCC.

This project has received funding from the European Union's Horizon 2020 research and innovation programme, under the Grant Agreement no. 871571.

Figure 13: Navigation camera RTSP stream from RovingBat on the SCC

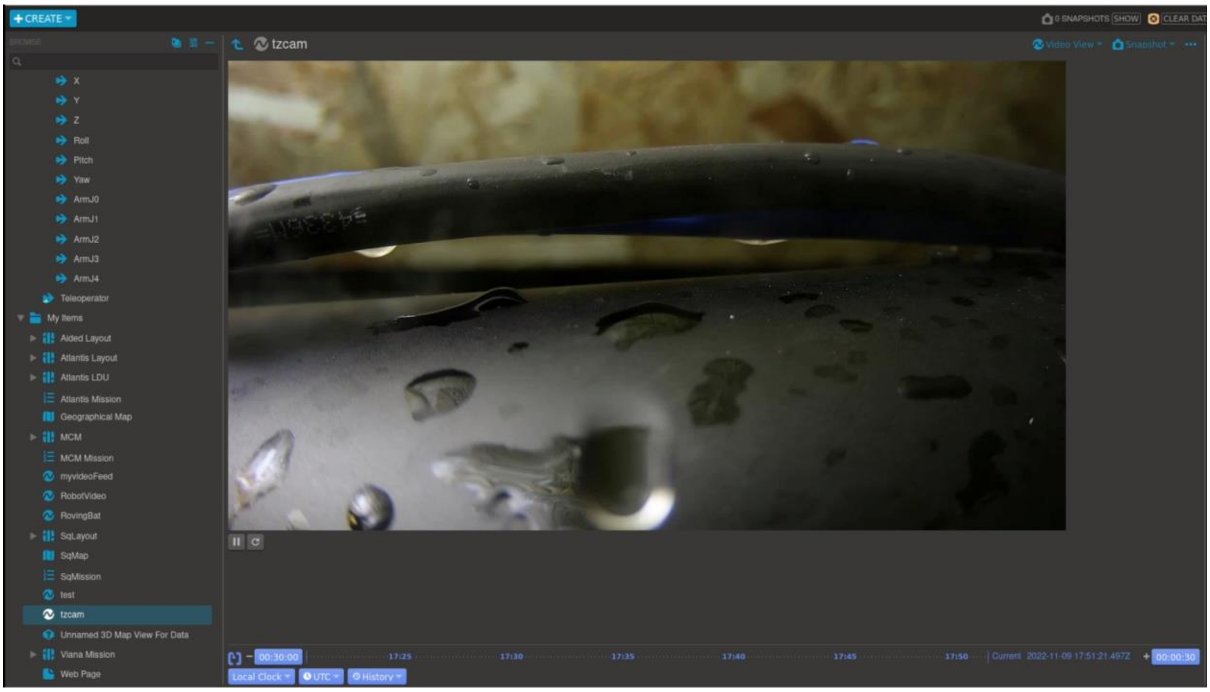


Figure 14: TZ camera RTSP stream from RovingBat on the SCC

4.3.4. 3D model articulations

The second focus of the integration with the ROV was joint states integration. The integration performed remotely did not display well the different articulations of the RovingBat in the SCC. To

This project has received funding from the European Union's Horizon 2020 research and innovation programme, under the Grant Agreement no. 871571.

display the articulations of the 3D model of the RovingBat, ECA provided us the URDF model of the ROV with the robotic arm integrated. The URDF model was uploaded into the SCC local map plugin and the robot was displayed on the graphical user interface as shown on Figure 15 and Figure 16.

The joint states ROS2 topics of the REACH BRAVO arm embedded on the RovingBat were added to the configuration file of the ROV. These topics should be correctly mapping the URDF joint states to be able to visualise the articulations of the ROV 3D model moving in real time on the SCC.

The odometry of the robot is also displayed on the local map plugin shown on Figure 15 and 16:

- The 3D model of the ROV follows the orientation of the robot sent using an odometry ROS2 topic.
- It also displays the local position of the robot received in the same odometry topic.
- It will not take in consideration the GPS data and display the robot only in its local coordinates.

The joint states values were only displayed in real time and not recorded into the database. ROSBags were recorded to be able to replay the data.

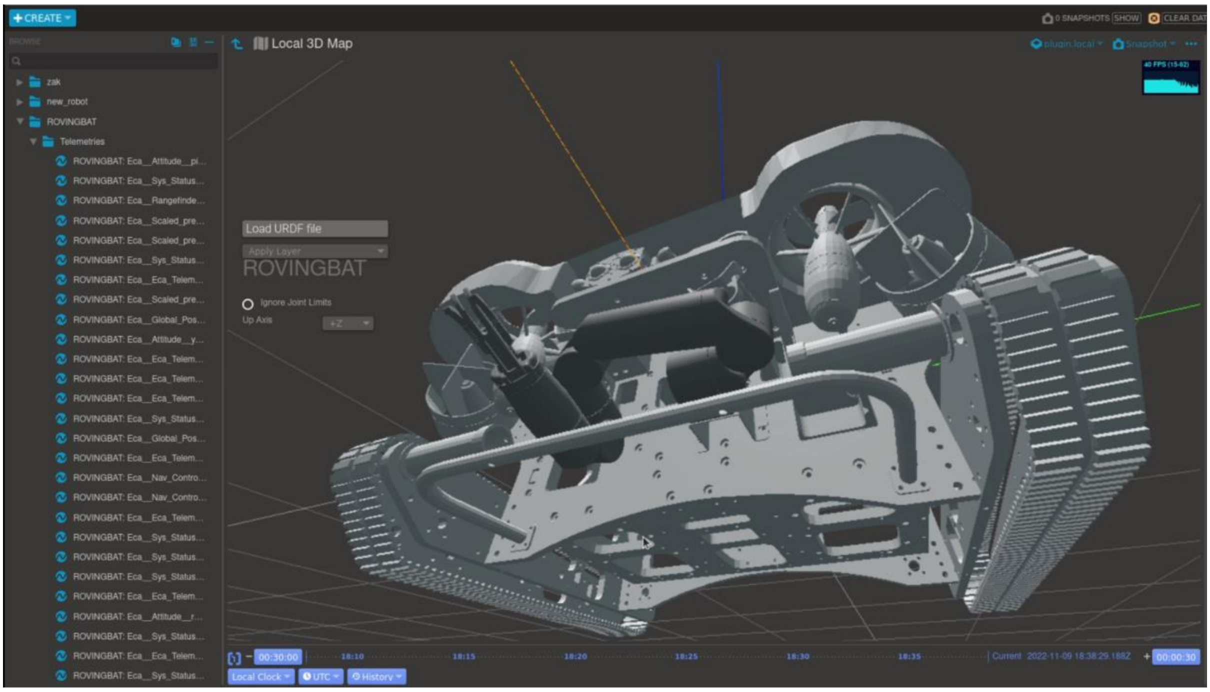


Figure 15: RovingBat Robotic Arm default position displayed on the SCC.

This project has received funding from the European Union's Horizon 2020 research and innovation programme, under the Grant Agreement no. 871571.

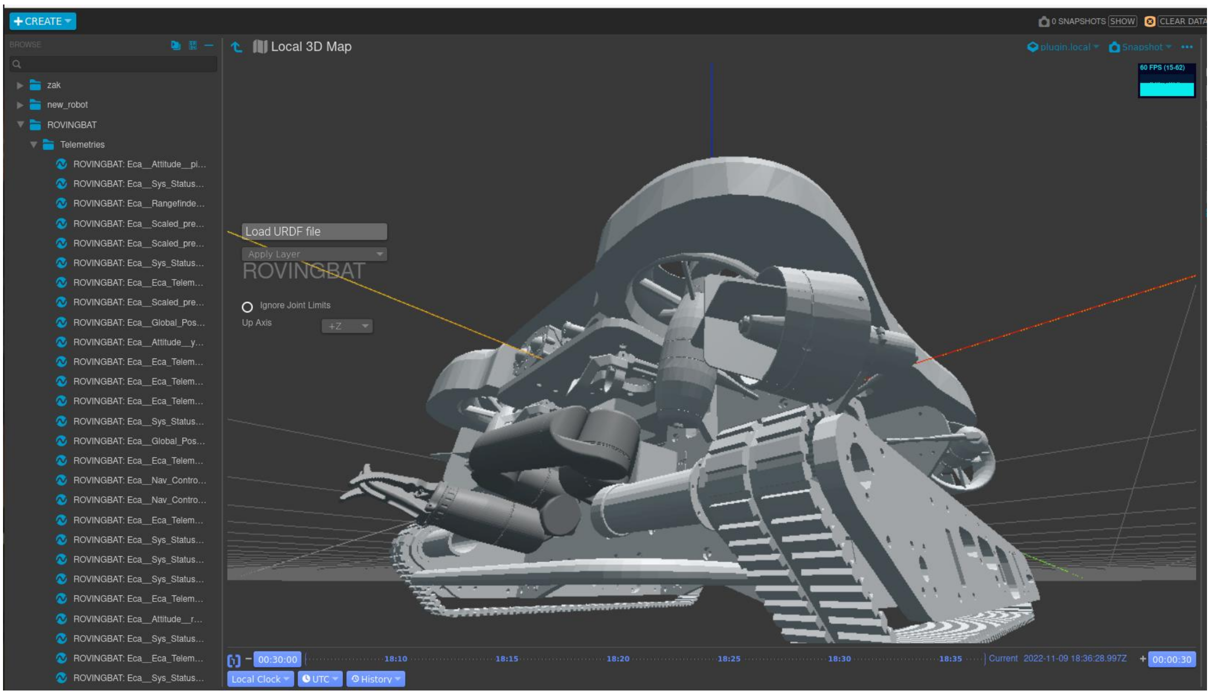


Figure 16: RovingBat Robotic Arm movements displayed on the SCC.

4.3.5. Mission configuration and deployment

The GIRONA 1000 from IQUA has been integrated first remotely using the cloud instance of the SCC before the onsite tests. During the November 2022 integration period we had the opportunity to integrate the robot onsite and also to test the deployment of a mission around the floating structure at the coastal testbed.

During the remote integration, mission planning had already been tested with the AUV. For the onsite tests the same mission has been recreated on the local instance of the SCC as shown on Figure 17, 18 and 19. For this, IQUA created the mission on the graphical user interface using the mission plugin of the SCC. For the tested mission, the floating structure was located in the coastal testbed. The objective of the mission was to send to AUV first assess and then inspect the floating structure. The mission is configured by defining the following parameters:

- Floating structure: represented as a point but assigned to the mission as a target.
- Initial point and end point of the mission: assigned as points.
- Robot speed: Estimated speed of the robot. Taking in consideration to optimise the battery consumption.
- Operations: assess and inspection operations. (Figure 18)
- Operation priorities: assess should be performed before inspection (Figure 19)

This project has received funding from the European Union's Horizon 2020 research and innovation programme, under the Grant Agreement no. 871571.

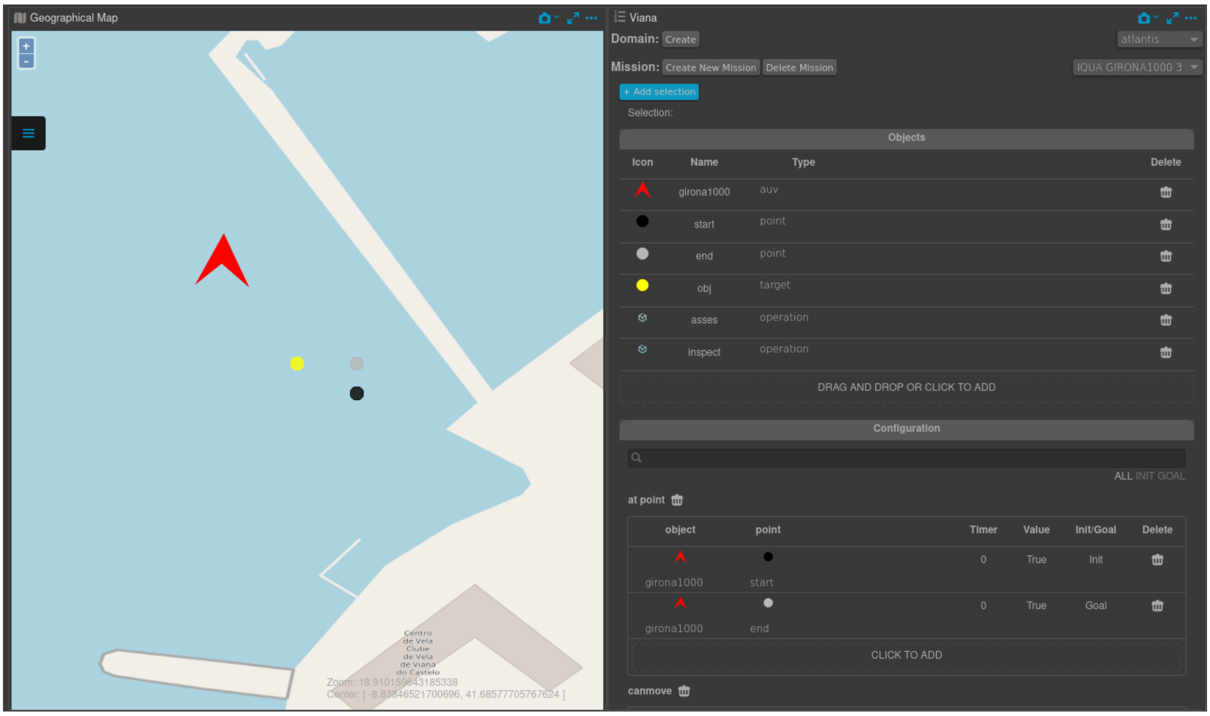


Figure 17: Mission configuration including Girona 1000 on the SCC

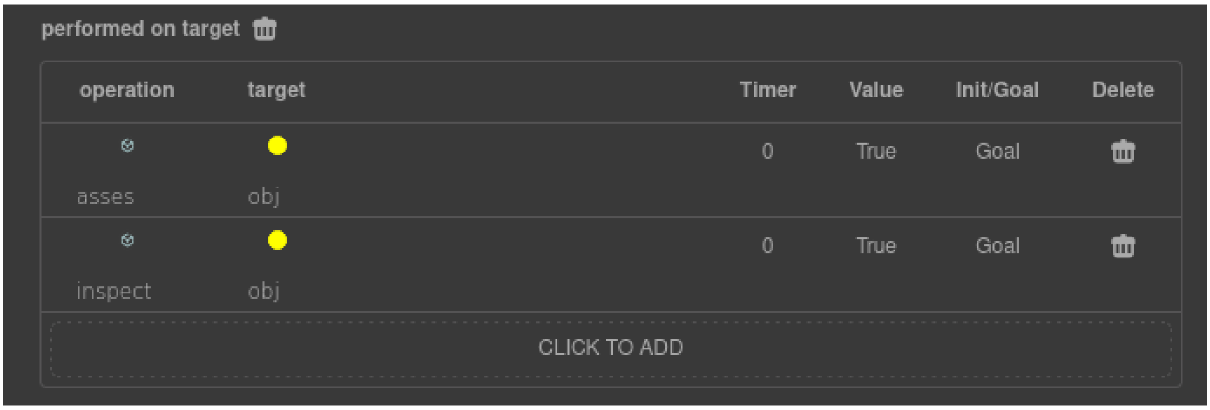


Figure 18: Definition of actions to perform on the selected target

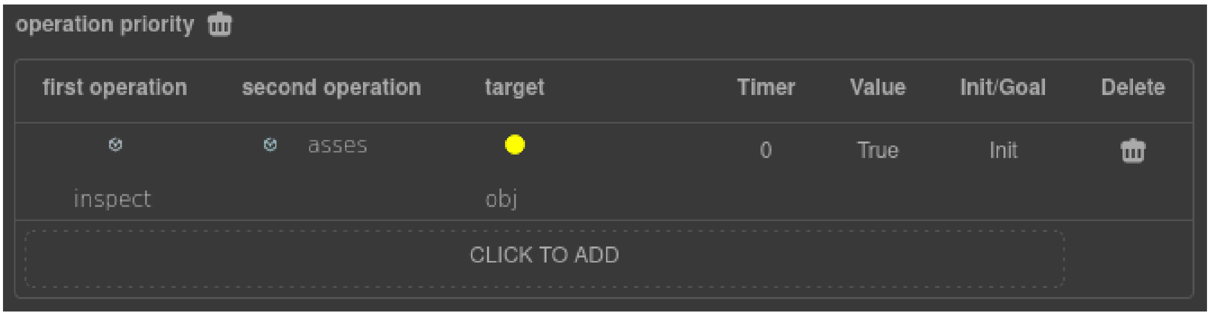


Figure 19: Definition of operation priority to perform on the selected target

The mission planner was then started using the “plan” button on the UI. Then the mission planning process created a mission plan for the AUV based on the user inputs:

1. Move: Move to the initial point.

This project has received funding from the European Union’s Horizon 2020 research and innovation programme, under the Grant Agreement no. 871571.

2. Asses: Asses the structure.
3. Inspect: Inspect the structure.
4. Move: Move to the final point.

The list of actions was finally deployed on the AUV using the “deploy” button of the SCC. As shown on figure 20, during the deployment of the mission the SCC indicated the status of each one of the actions of the mission plan:

- Not completed: the action is not yet started.
- In progress: the action is in progress now.
- Completed: the action is completed.

The mission deployment was monitored on the SCC. In addition to the status of each one of the actions of the plan, the SCC displays the position of the robot on the map along the geographical user inputs for the mission. Figure 22 shows the live position of the AUV and its path since the beginning of the mission on the left side of the picture. On the right side the mission plan and configuration is displayed.

All the telemetries sent by the robot during the deployment of a mission are recorded in the database and can be accessible later. The status of the actions during the deployment of a mission is not recorded.

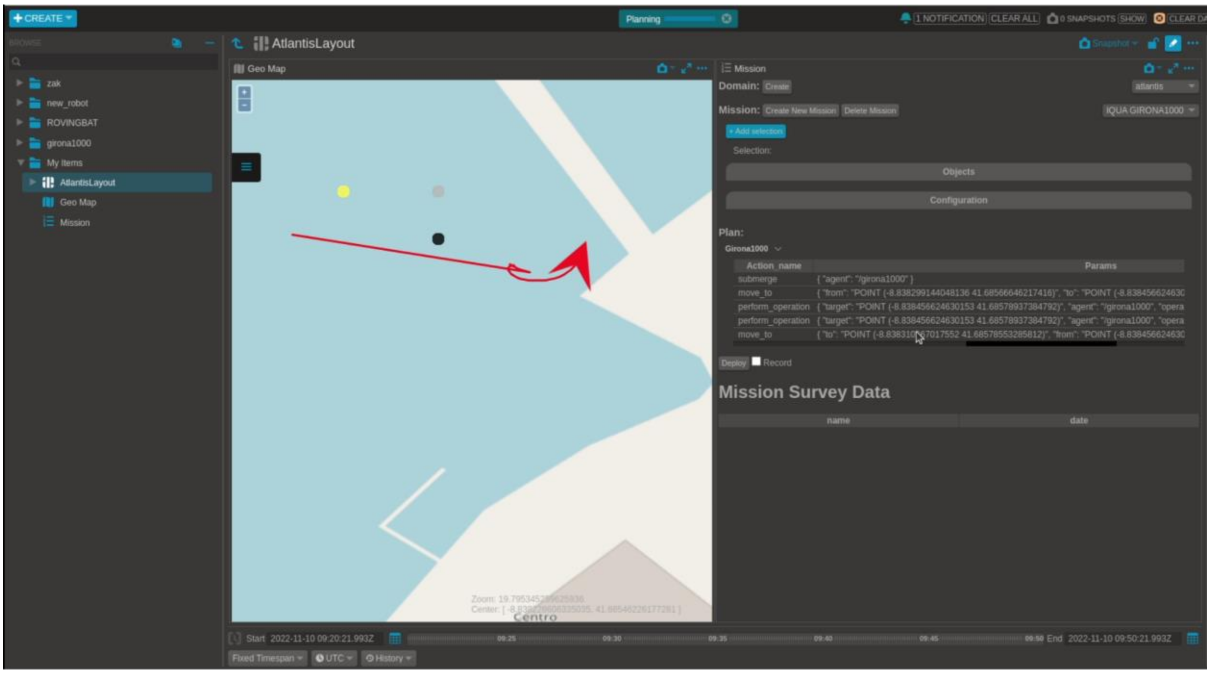


Figure 20: Girona 1000 AUV Mission execution monitoring on SCC

In the next figure, the mission has been deployed by the operator. We can see the mission's steps status in the top right corner. On the down right corner, we can see the live heading value of the robot. On the left side, we can see the robot heading to the first point according to the plan.

This project has received funding from the European Union’s Horizon 2020 research and innovation programme, under the Grant Agreement no. 871571.

Figure 21: Girona 1000 AUV around the floating structure.

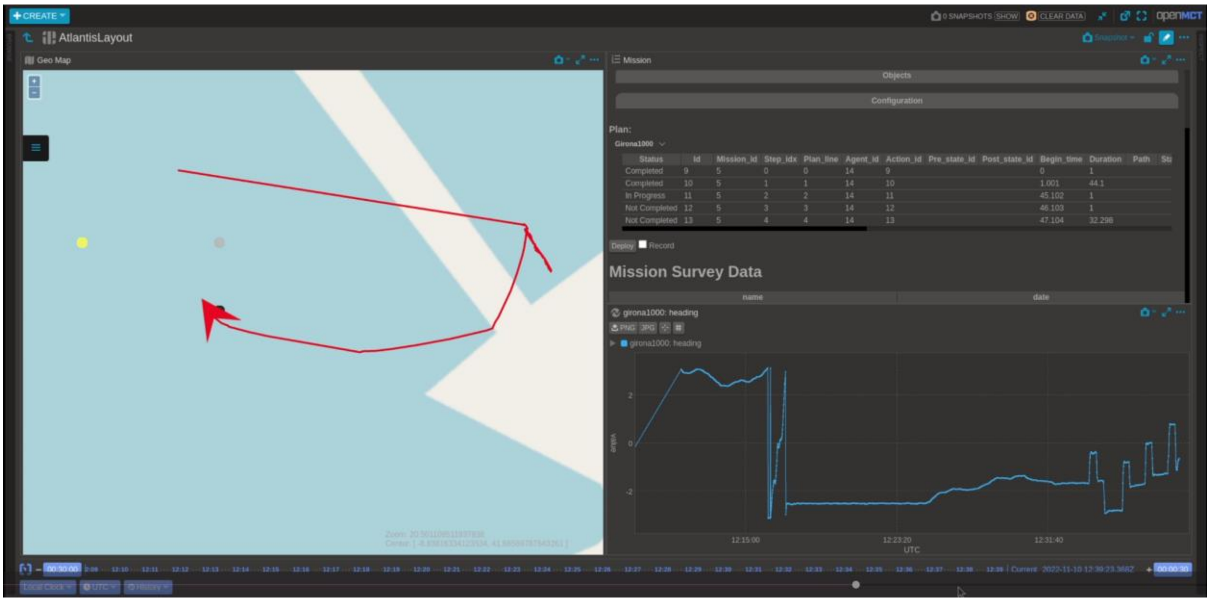


Figure 22: Girona 1000 AUV Mission execution monitoring with heading on SCC

4.3.6. Multi robot integration

During the September 2022 integration session, after having successfully integrated the Raven UAV and the Zarco USV separately, we integrated both robots at the same time. The CPP API was used in order to integrate the robots simultaneously to the SCC.

The following scenario was tested:

This project has received funding from the European Union's Horizon 2020 research and innovation programme, under the Grant Agreement no. 871571.

- Zarco and Raven connected to the SCC local network.
- SCC graphical user interface available over the SCC network.
- Numerical telemetries sent by Zarco and Raven.
- GPS data sent by Zarco and Raven.
- Images sent by Raven.
- Odometry sent by Raven.
- Zarco and Raven manually moved on the ground in front of the SCC (Figure 23).

For this scenario, remote monitoring of the SCC has also been tested. The laptops used on Figure 24 were connected to the SCC local network over wifi. The SCC user interface is displayed on the web browser of the laptop by accessing the IP address of the SCC on the local network on port 80.

Figure 23: Zarco USV and Raven UAV moved on the ground in front of the SCC.

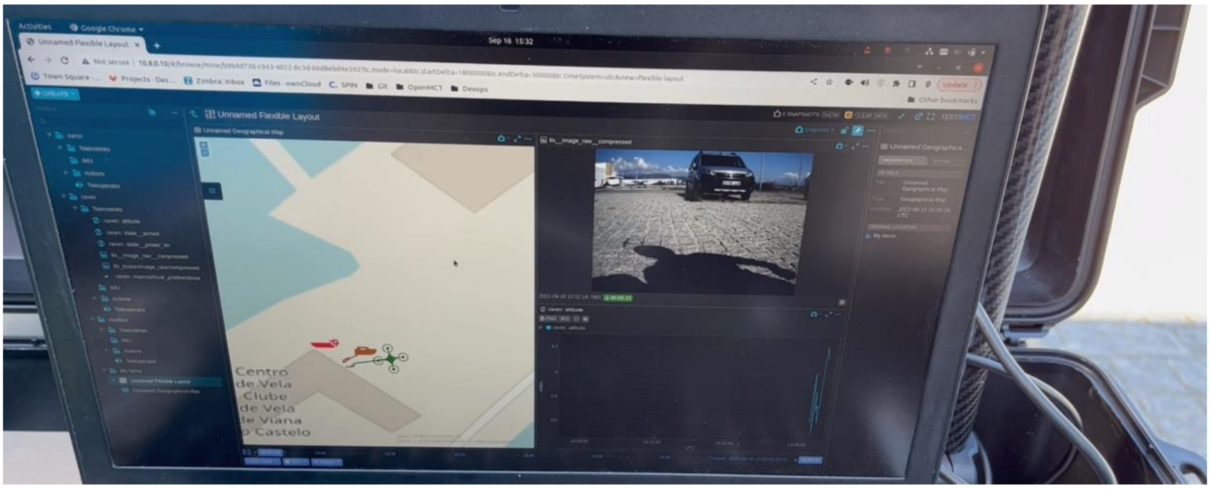


Figure 24: Zarco USV and Raven UAV GIS, odometry and images displayed on a remote SCC client.

This project has received funding from the European Union's Horizon 2020 research and innovation programme, under the Grant Agreement no. 871571.

5. Offshore integration

5.1. Network architecture

During the September 2022 integration session, the Nautilus USV was integrated with the SCC. The following network configuration were used (Figure 25):

- SCC is connected to its local network.
- SCC is connected to an openVPN server.
- 4G router and wifi access point on the offshore boat.
- Robot connected to the 4G router and to the openVPN server.

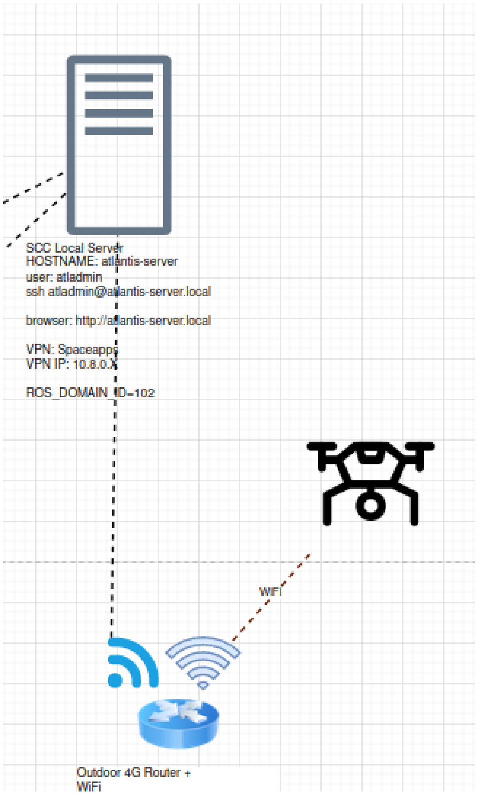


Figure 25 : Network architecture of the ATLANTIS server and the offshore set up

5.2. Interoperability integration

The Nautilus USV were integrated to the SCC using the CPP API while being offshore. It was connected to the SCC through the openVPN server. On one hand, ROS2 data was sent from the USV over the VPN to the SCC and the SCC user interface was accessible on laptop offshores using the same VPN connection.

The configuration of Nautilus was sent from offshore and was successfully received on the SCC. The robot was displayed online on the SCC and was subscribing to robot data.

This project has received funding from the European Union's Horizon 2020 research and innovation programme, under the Grant Agreement no. 871571.

Due to the difficulties to control the robot and the communication with the SCC at the same time, only the configuration of the robot was sent to the SCC. The configuration file was sent as ROS2 data to the SCC so ROS2 data transfer from offshore has been tested and validated.

5.3. ROSBags replay

The Nautilus data sent during the offshore tests has been recorded locally on laptops. The data was then replayed on the local instance of the SCC. The following data has been replayed:

- Configuration file
- Numerical and boolean telemetries
- GPS and odometry
- Images

All the data were successfully replayed on the SCC. The configuration file needs to be published to indicate to the SCC which topics to subscribe as for real time integration. The most important difference with real time integration is the timestamp of the data that are published.

During real time integration all the data are received right after they are originally published. In that way the SCC directly displays the data as being published in real time. When receiving data coming from ROSbags, the timestamp is the same as the one used during the record of the ROSbags. The SCC will then display data from the past instead of displaying real time data.

Figure 26: Offshore wind turbines photo taken from the boat.

However, on the graphical user interface the GIS position of the robot is still displayed as if it was received in real time. This means that the UI is not displaying the real current position of the robot while replaying a mission using ROSbags but the position sent by the replay of the ROSbags (Figure 27).

Except the GIS position all the other data (images, telemetries, …) are displayed based on their timestamps. So the latest data received by the robot for a specific topic will not be replaced by the one received while replaying the data.

This project has received funding from the European Union's Horizon 2020 research and innovation programme, under the Grant Agreement no. 871571.

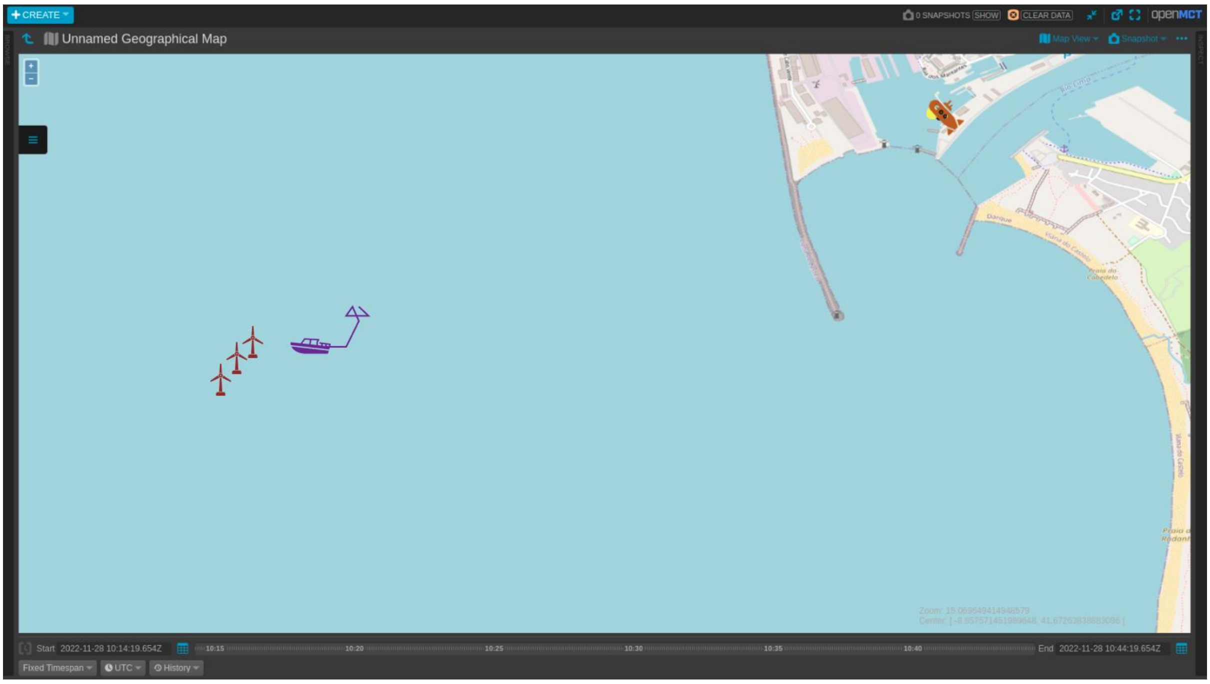


Figure 27: Nautilus ROSbag data replayed on the SCC.

This project has received funding from the European Union's Horizon 2020 research and innovation programme, under the Grant Agreement no. 871571.

6. Conclusion

The scenarios defined in previous deliverables were tested and integrated to the SCC. Remote integration sessions performed prior to the onsite tests allowed to identify the potential issues of the integration of each of one the robots, on the robot side and on the SCC side.

The three onsite integration sessions were used to validate the outcome of the remote integrations in real conditions. All the tests that were planned were performed and successfully integrated. Each robot was able to be monitored from the SCC using numerical telemetries, GPS position, odometry, images and videos.

The mission plan integration was tested with IQUA on the Girona 1000. This test took in consideration, monitoring and commanding of the robot as well as user interactions with the SCC to define a mission. Commanding and monitoring of Girona 1000 were successfully integrated to the SCC using the interop library.

Offshore communication was tested as well as the interoperability CPP library while integrating Nautilus. Nautilus were successfully integrated offshore. For the data that could not be sent while offshore, the replay of the data after the execution of the mission also showed good results on the graphical user interface of the SCC.

As a general overview, all the features needed from the SCC were successfully tested and integrated with robots. In the next phases, the mission configuration and deployment will be tested and integrated with all the robots being part of the project.

This project has received funding from the European Union's Horizon 2020 research and innovation programme, under the Grant Agreement no. 871571.

7. Annexe I

7.1. Robot connection

Thanks to the UV-Interop library the specific configuration file is sent to the UI. The recommended way to define the robot capabilities is by writing a YAML file that includes:

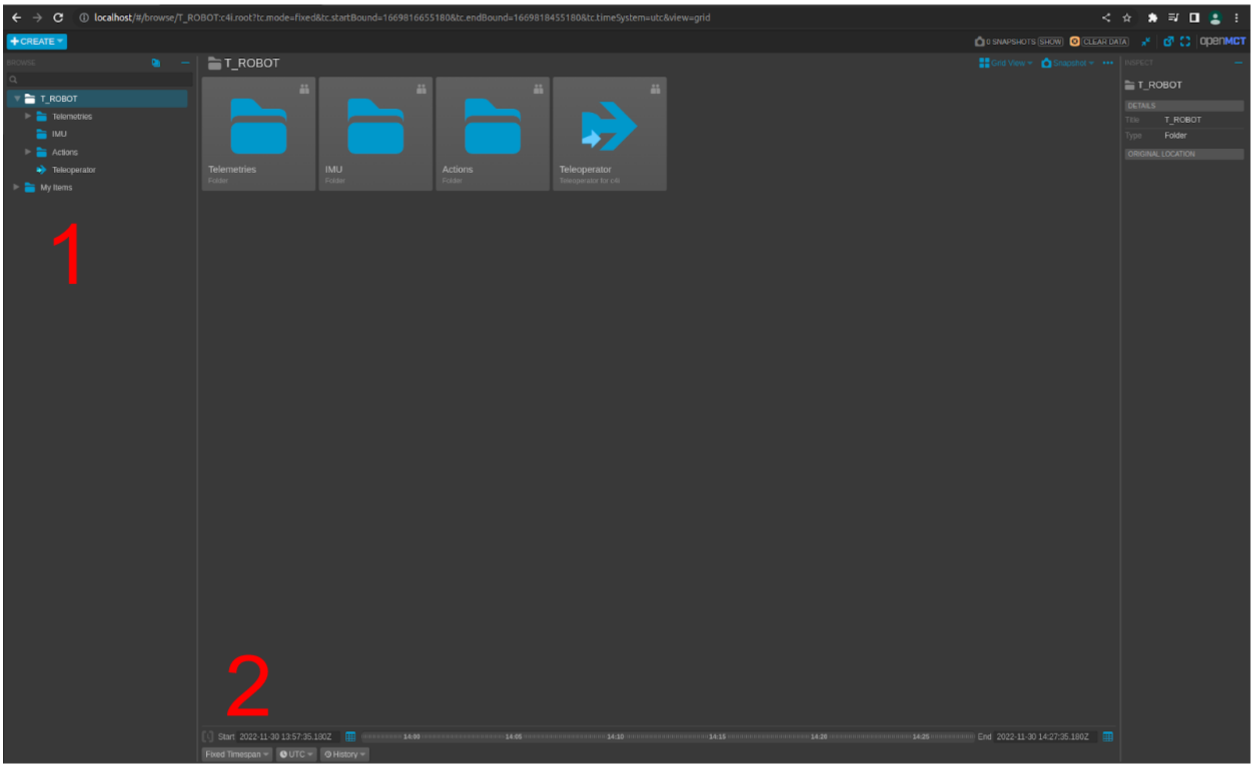
- A unique string identifier of the robot called namespace
- Telemetry topics for data:
  - Boolean
  - Numerical
  - Imagery
  - Position and orientation
  - GPS position
- An identifier for the algorithm responsible for executing the mission plan
- List of commands based on their payload types
  - Plain (no payload)
  - Waypoint bath
  - Switch
  - Numeric

```yaml
### Namespace

namespace: /TEST_ROBOT

### Telemetry (for ROS: name of topics)

boolean_telemetry_topics:

 - submerged

 - floating

 - wifi_on

 - docked

 - lidar_on

 - power_on_MBES

numeric_telemetry_topics:

 - battery

 - altitude

 - temperature

 - motor1/current
```


This project has received funding from the European Union's Horizon 2020 research and innovation programme, under the Grant Agreement no. 871571.

```yaml
imagery_telemetry_topics:
 - webcam
pose_telemetry_topics:
 - robot_pose
gps_telemetry_topics:
 - gps_topic
### Commands (for ROS: name of action servers)
mission_execution: plan_execution
command_plain: #no parameters
 - recharge:
   conditions:
     - docked
command_waypoint: #point payload
 - move_to_point:
command_path: #path payload
 - follow_path:
command_switch:
 - power_on
command_numeric:
 - set_heater_temperature:
   unit: K
   min_val: 270
   max_val: 350
   goal:
     - target_value
   feedback:
     - status_feedback
     - feedback_value
   result:
     - status_result
     - result_value
```


This project has received funding from the European Union's Horizon 2020 research and innovation programme, under the Grant Agreement no. 871571.

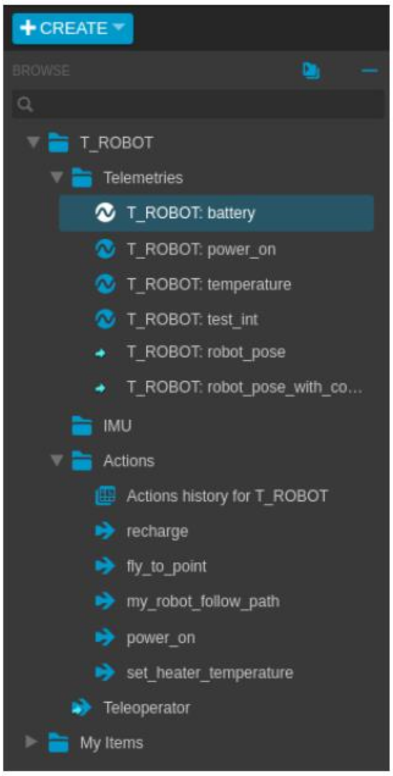


7.2. Telemetries

Once the robot is connected it is possible to access any topics and actions from the configuration file from the items menu on the left.

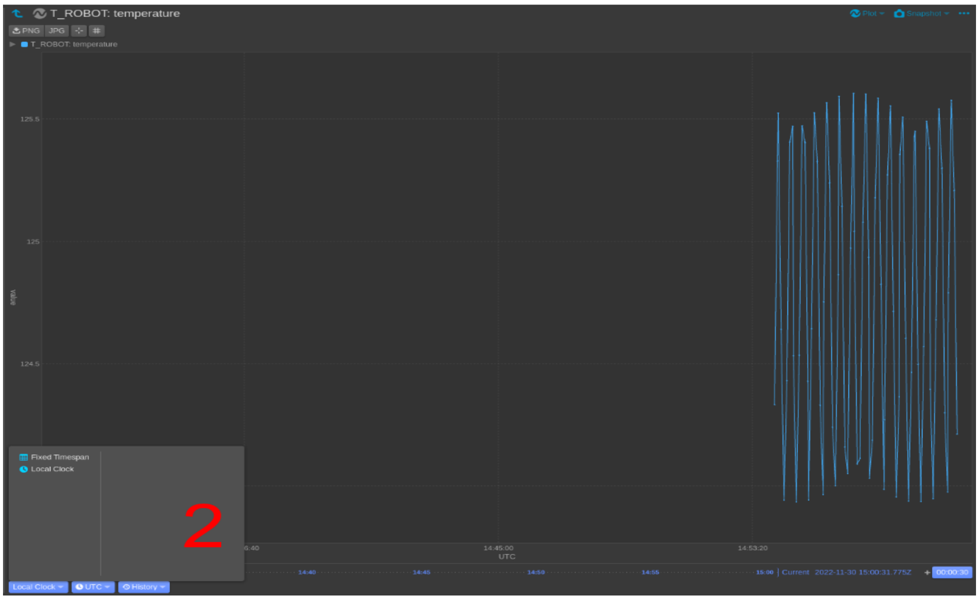
This project has received funding from the European Union's Horizon 2020 research and innovation programme, under the Grant Agreement no. 871571.

1



Then to display the chosen topic, it is possible to choose a fixed **Timespan** or the live data from the **Local Clock** (see picture below).

Additionally, it is possible on the timeline at the bottom of the screen or directly on the graph to select a specific part of the data.

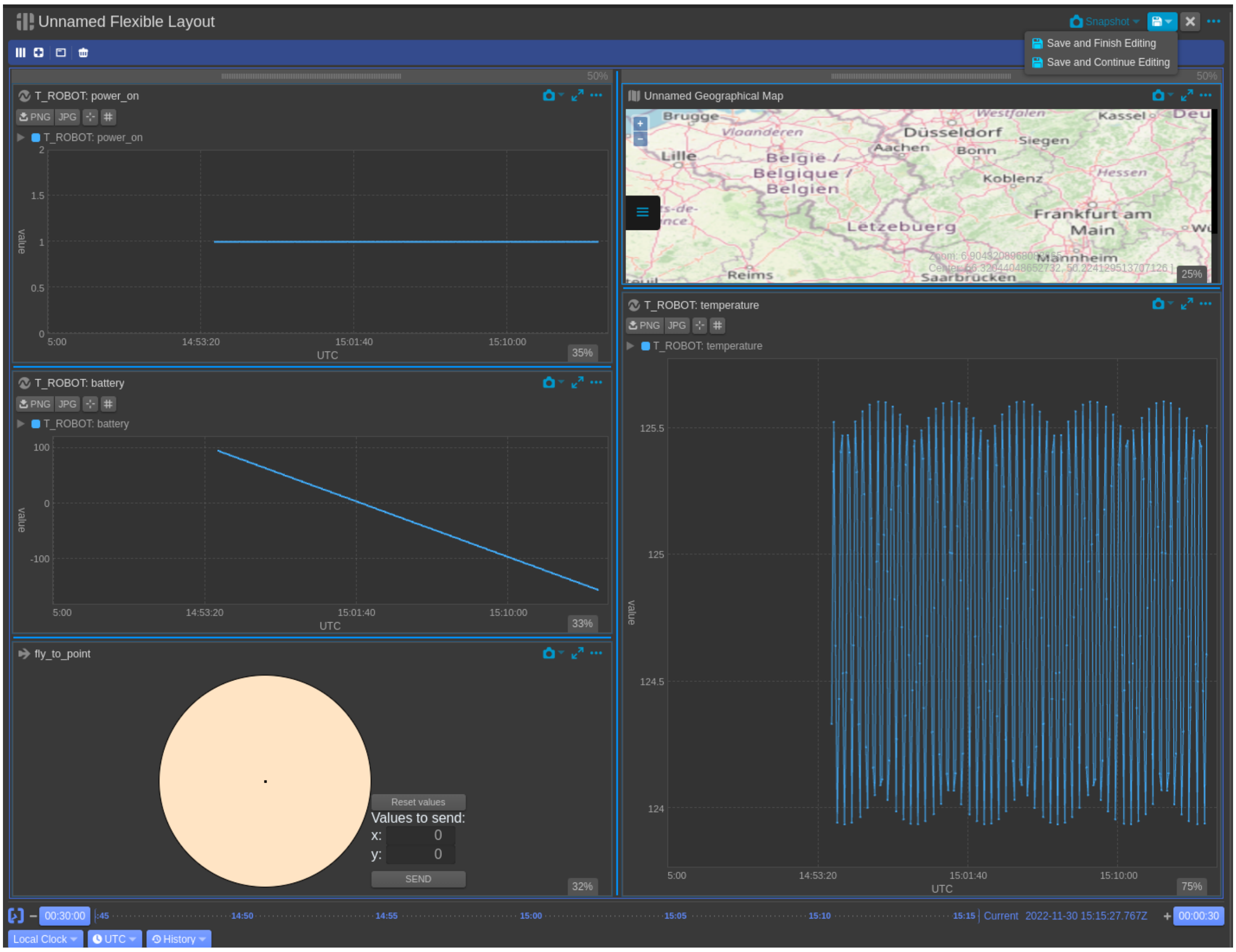


This project has received funding from the European Union's Horizon 2020 research and innovation programme, under the Grant Agreement no. 871571.

7.3. Flexible Layout

In order to customize the displayed data by combining the wanted topics, actions or maps, we need to create a **Flexible Layout** from the **Create** menu at the top left corner (1).

Once the edit of the **Flexible Layout** is done, you can save it by using the save button on the top right corner (see picture below), the layout can be modified as much as required by using the Edit button.

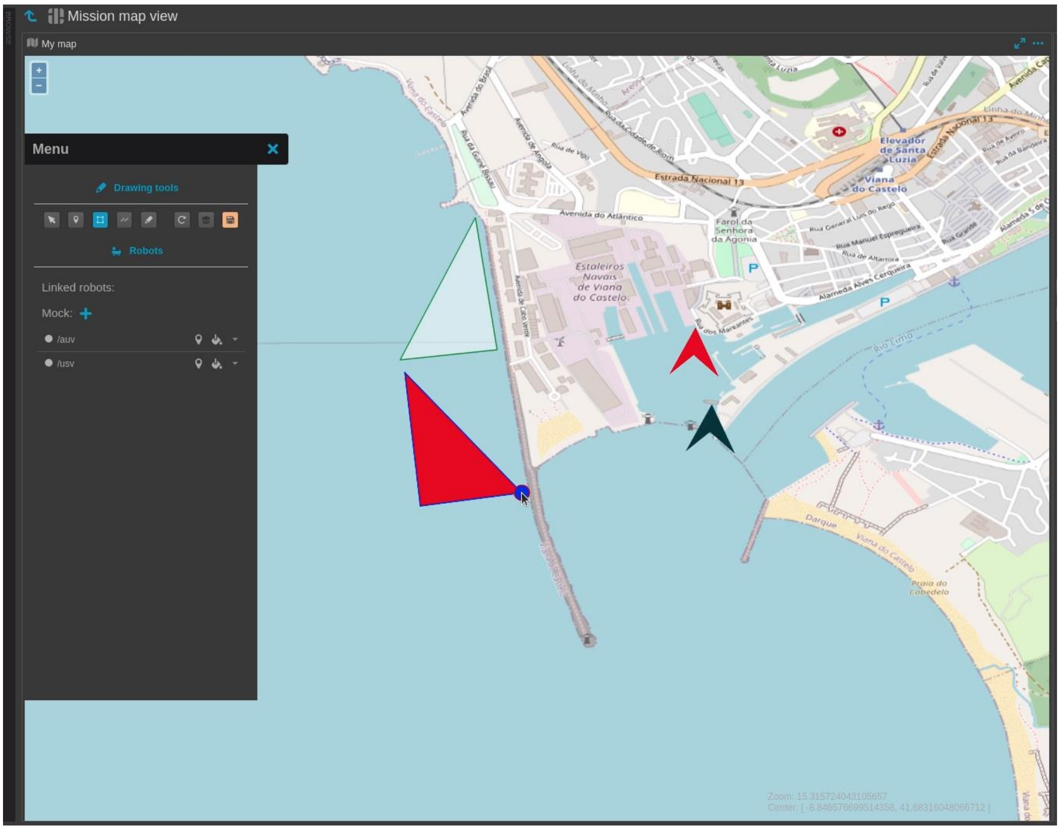


7.4. Geographical Map Display

The map view allows the user to check the real time position and orientation of the agents in the world. A toolbar can be displayed to do the following operations:

- interact with the view by drawing geometries on the map directly .
- interact with the robots registered in the SCC.
- center the view on any desired agent.
- create a mock agent and position it on the map.
- customize the agent display to efficiently identify the different robots.

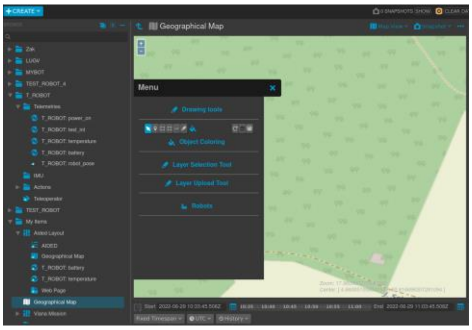
This project has received funding from the European Union's Horizon 2020 research and innovation programme, under the Grant Agreement no. 871571.



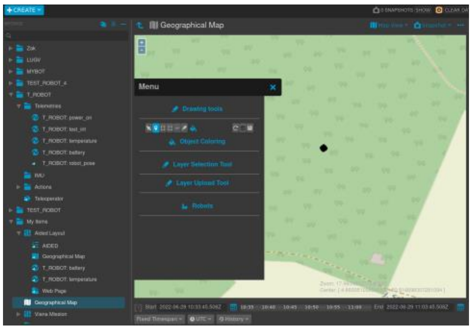
The map view has a menu on the left that allows us to draw geometries on the map. The following sequence of figures shows the steps on how to use this feature.

In the case of missions, the geometries defined using the drawing tools can be associated with object types and named in such a way that the user can introduce the required configuration and the plan executor of the robot will navigate to and from these real objects

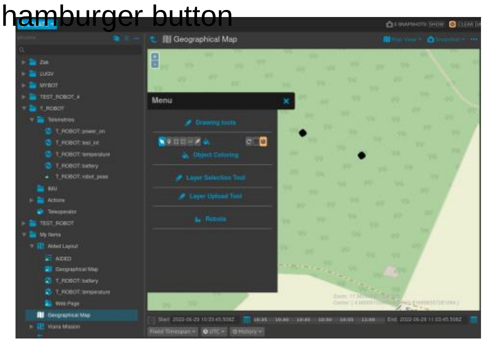
This project has received funding from the European Union's Horizon 2020 research and innovation programme, under the Grant Agreement no. 871571.



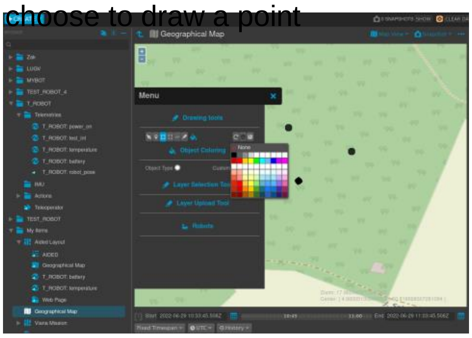
1. Show the menu by clicking on the hamburger button



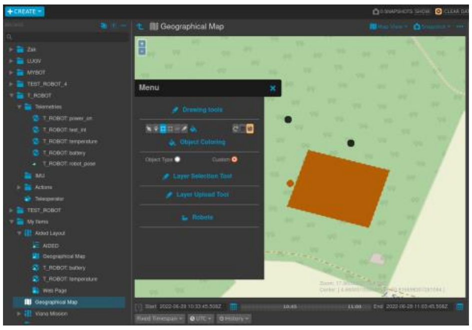
2. Select the tool to draw. Here we choose to draw a point



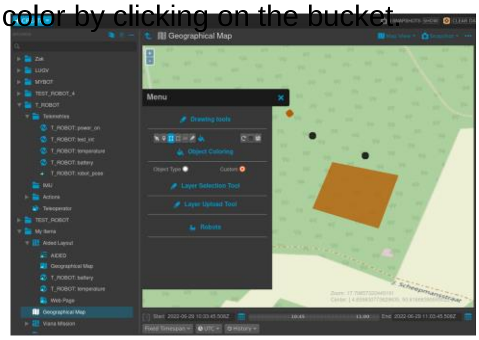
3. After drawing, we click the save button highlighted in orange



4. To draw a polygon, we select the polygon tool. We can also change the color by clicking on the bucket



5. The polygon has dotted lines which means that it has not been saved



6. After clicking save, the polygon appears with plain lines

7.5. Mission Planning

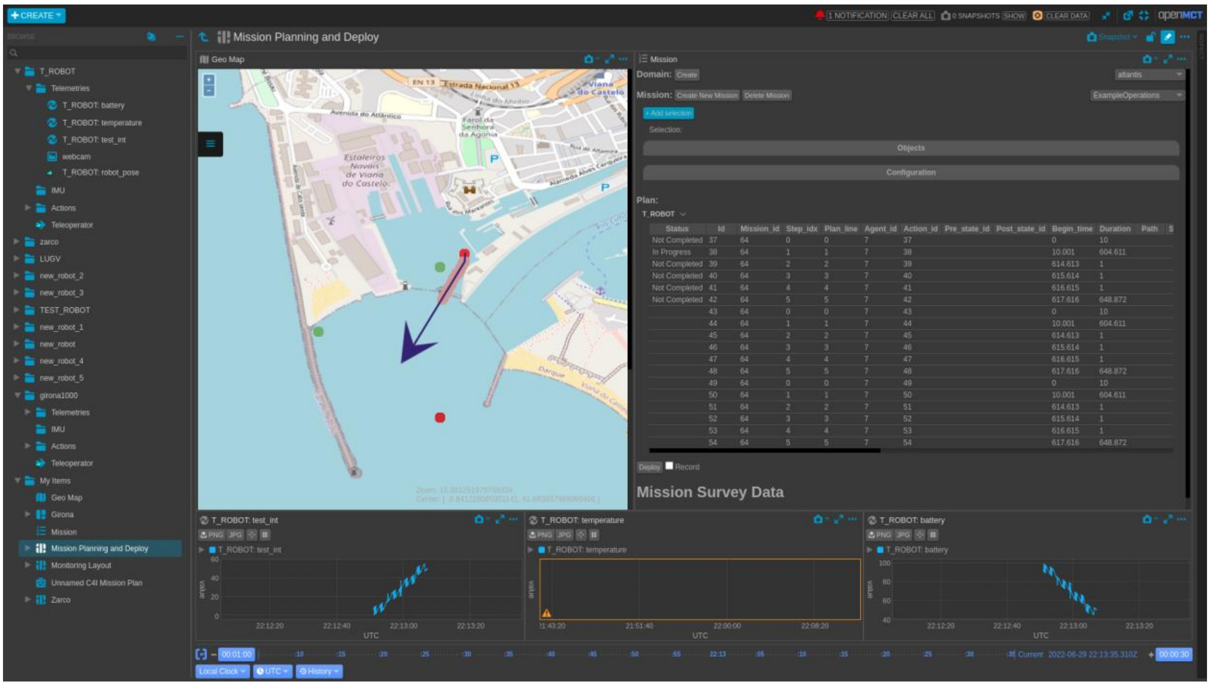
This section details the technologies, features and use cases of the mission planning tool embedded in the SCC (**Mission** feature in the **Create** menu). The characteristics of the task planning and of the underlying paradigm is briefly described along with its integration within the graphical user interface. The mission planner can be decomposed in three sections : the task planner, the deployment and the mission monitoring.

Domain definition

This project has received funding from the European Union's Horizon 2020 research and innovation programme, under the Grant Agreement no. 871571.

The PDDL domain file is already uploaded in the UI and can be selected using the **Create** button (from the **Mission** feature). This feature allows users to change and select a domain corresponding to the mission characteristics.

Using the domain content present in the database, it is possible to define multiple missions based on the same domain.



5.2 Mission Configuration

To simplify the configuration of a mission by an operator, a mission plugin is available on the graphical interface. This plugin interfaces with the database to create and store mission related data, and link them to other data like the robots registered in the system or any geometries defined using the **Flexible Layout** described above. The mission plugin is designed to handle PDDL syntax to be used in association with the map view.

A mission is firstly defined by:

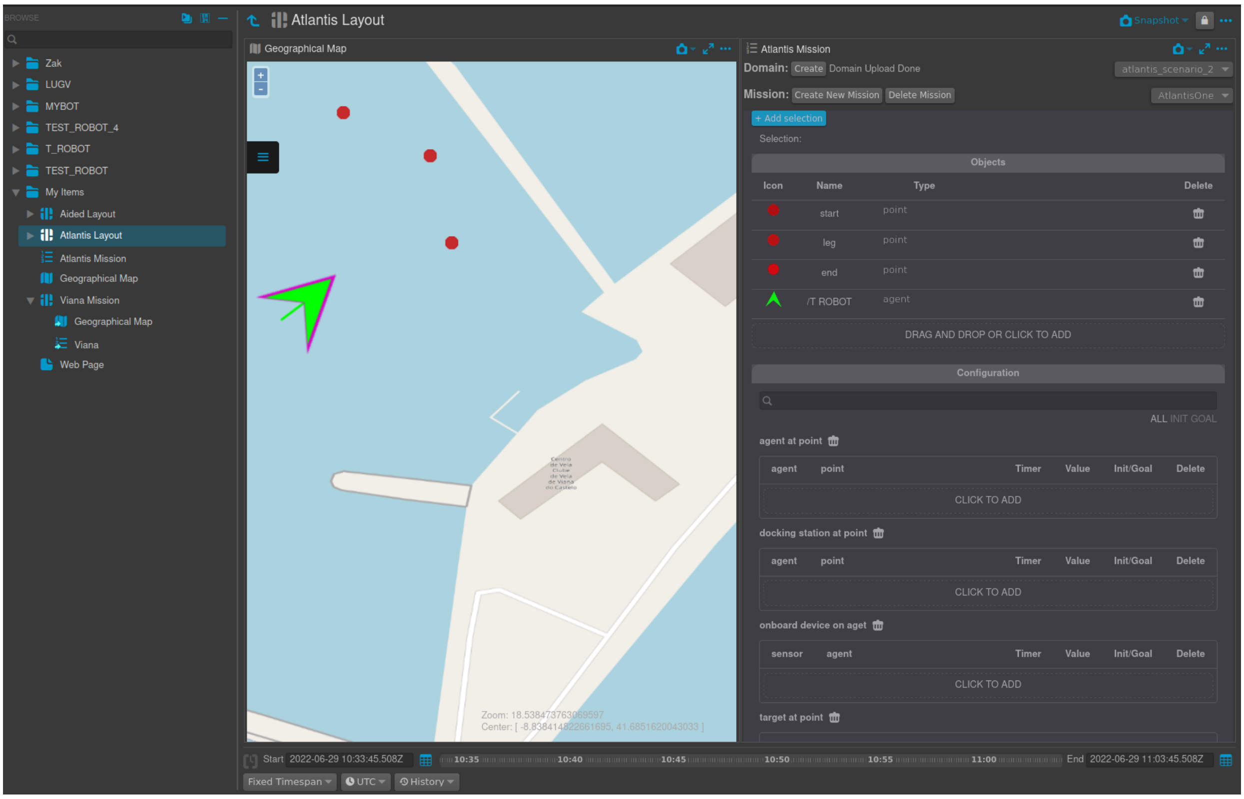
- a name.
- a short description.
- a PDDL domain.

Using the plugin an operator can create a mission, by selecting or uploading a PDDL domain in the database. Then the mission can be configured using the domain specifications. The operator is able to add robots, assets or geometries to a mission by linking it to an object type taken from the domain. For example using a domain containing the types AUV and USV, the operator will be able to add a robot as an AUV, or as an USV to the mission.

Robots and objects (such as geometries or assets) to add to the mission can be selected using the map plugin and then the corresponding object type can be chosen in the mission plugin. The state of each one of these objects can be configured in the mission view using the domain predicates and numerical

This project has received funding from the European Union's Horizon 2020 research and innovation programme, under the Grant Agreement no. 871571.

values. Two states need to be configured by the operator , the initial state of the mission, before the deployment and the goal state, at the end of the mission execution. The state of an object can affect its position, battery level, etc (depending on the domain specifications and the type associated with each object).



This project has received funding from the European Union's Horizon 2020 research and innovation programme, under the Grant Agreement no. 871571.