General System Architecture and COTS Prototyping of an AIoT-Enabled Sailboat for Autonomous Aquatic Ecosystem Monitoring

Article *in* IEEE Internet of Things Journal · January 2023

DOI: 10.1109/JIOT.2023.3324525

André Araújo
Fluminense Federal University

Dickson Daniel
Fluminense Federal University

Raphael Guerra
Fluminense Federal University

Diego Brandão
Federal Center for Technological Education Celso Suckow da Fonseca

General system architecture and COTS prototyping of an AIoT-enabled sailboat for autonomous aquatic ecosystem monitoring

André P. D. Araújo, Dickson H. J. Daniel, Raphael Guerra, Diego N. Brandão, Eduardo Charles Vasconcellos, Alvaro P. F. Negreiros, Esteban W. G. Clua, Luiz M. G. Goncalves and Philippe Preux

***Abstract*—Unmanned vehicles keep growing attention as they facilitate innovative commercial and civil applications within the Internet of Things (IoT) realm. In this context, autonomous sailing boats are becoming important marine platforms for performing different tasks, such as surveillance, water, and environmental monitoring. Most of these tasks heavily depend on artificial intelligence (AI) technologies, such as visual navigation and path planning, and comprise the so-called Artificial Intelligence of Things (AIoT). In this paper, we propose (i) the OpenBoat, an automating system architecture for AIoT-enabled sailboats with application-agnostic autonomous environment monitoring capability, and (ii) the F-Boat, a fully functional prototype of OpenBoat built with Commercial Off-The-Shelf (COTS) components on a real sailboat. F-Boat includes low-level control strategies for autonomous path following, communication infrastructure for remote operation and cooperation with other systems, edge computing with AI accelerator, modular support for application-specific monitoring systems, and navigation aspects. F-Boat is also designed and built for robustness situations to guarantee its operation under extreme events, such as high temperatures and bad weather, through extended periods of time. We show the results of field experiments running in Guanabara Bay, an important aquatic ecosystem in Brazil, that demonstrate the functionalities of the prototype and demonstrate the AIoT capability of the proposed architecture.**

***Index Terms*—IoT, AIoT, Autonomous Robots, SailBoats, USV.**

We thank CAPES-Brazil, grants numbers 001 and 88881.506890/2020-01, and CNPq-Brazil, grant number 311640/2018-4, for funding this research. Ph. Preux acknowledges the support of the Métropole Européenne de Lille (MEL), ANR, Inria, Université de Lille, through the AI chair Apprenf number R-PILOTE-19-004-APPRENF, as well as the Scool research group.

André Araújo (corresponding author; e-mail: andrepda@id.uff.br), Dickson Daniel (e-mail: dicksond@id.uff.br), Raphael Guerra (e-mail: rguerra@ic.uff.br), Eduardo Vanconcellos (e-mail: evasconcellos@id.uff.br) and Esteban Clua (e-mail: esteban@ic.uff.br) are with the Institute of Computing at Fluminense Federal University, Niterói/RJ, Brazil.

Diego Brandão (e-mail: diego.brandao@cefet-rj.br) is with the Computer Department at Federal Center for Technological Education Celso Suckow da Fonseca (CEFET/RJ), Rio de Janeiro/RJ, Brazil.

Alvaro P. F. Negreiros (e-mail: alvaro.negreiros@gmail.com) and Luiz Gonçalves (e-mail: lmarcos@dca.ufrn.br) are with the Graduate Program in Computer and Electrical Engineering at the Rio Grande do Norte Federal University (UFRN), Natal/RN, Brazil.

Philippe Preux (e-mail: philippe.preux@univ-lille.fr) is with the Université de Lille and CNRS CRIStAL laboratory and with Inria, Villeneuve d'Ascq, France.

I. INTRODUCTION

Aquatic ecosystems are critical components to the maintenance of Earth's life. Nonetheless, various factors can contaminate them, including pollution and extreme events like storms and floods due to climate change. In this context, the use of Autonomous Surface Vehicles (USV) of type sailboats powered by the Internet of Things (IoT) and Artificial Intelligence (AI) can be essential for monitoring and protecting these ecosystems, leveraging the Artificial Intelligence of Things (AIoT) systems. AIoT enables real-time data collection, analysis, and decision-making, improving monitoring accuracy and efficiency [1].

Robotic sailboats powered by these technologies can autonomously navigate the seas and oceans, collecting data and performing various tasks without human intervention. It can operate for extended periods, cover vast distances, and collect data that would be difficult or impossible to obtain using traditional research vessels [2]. Operating over extended periods in marine environments demands robust and resilient vessels, components, and software to deal with extreme weather events.

The IoT capabilities of an autonomous sailboat enable it to collect and transmit data on various environmental parameters such as water temperature, pH, turbidity, and salinity, among others. This USV can also have sensors to detect marine life, pollutants, and other relevant information. Such data can help scientific research, aquatic conservation efforts, and understanding extreme events, such as severe storms, heavy rainfalls, and tidal surges that can cause flooding [3], [4].

In addition, AI algorithms embedded in the robotic sailboat can analyze data while being collected, supporting decision-making based on that data. These algorithms can adjust the sailboat course based on changing weather conditions or avoid obstacles and hazards.

Despite advances in the AIoT, developing autonomous sailboats remains a challenge, mainly due to the inherent difficulties of navigation control, infrastructure, and energy management, leading much research to be restricted to simulation environments [5], [6], [7], [8], [9], [10], [11].

Towards solving these issues, we propose (i) the OpenBoat, an autonomous system architecture for AIoT-enabled sailboats with application-agnostic autonomous environment monitoring capability and robustness to guarantee its operation under extreme events, such as high temperatures and bad weather; and (ii) the F-Boat, a fully functional prototype of OpenBoat built with Commercial Off-The-Shelf (COTS) components on a real sailboat (Figure 1). F-Boat includes low-level control strategies for autonomous path following, communication infrastructure for remote operation and cooperation with other systems, edge computing with AI accelerator, modular support for application-specific monitoring systems, and navigation aspects. F-Boat allows the evaluation of different testbed experiments, such as navigation and computer vision algorithms, communication technologies, and others. In addition, F-Boat is designed and built for applications that include continuous autonomously monitoring of aquatic ecosystems for long periods, under adverse temperature conditions and heavy rainfall. Such applications include surveillance (criminal behavior, such as arms and drug smuggling) and environmental protection (water quality and wildlife preservation), among others. Project documentation, source code, images and videos of the building process and field deployment are available at https://www.natalnet.br/nboat/ and https://github.com/medialab-fboat.

We show the results of field experiments that demonstrate the functionalities of the prototype and validate the AIoT capability of the proposed architecture. We chose Guanabara Bay, an important marine and coastal ecosystem in southeastern Brazil, for the experiments. That area is also prone to extreme events that can significantly impact its ecosystems and population, including severe storms, heavy rainfalls, and tidal surges with eventual flooding [3], [4].

Fig. 1. F-Boat - The Sailboat Autonomous Robot.

The paper follows with Section II that presents related works and Section III that presents the main components of a sailboat. Then, several aspects of the Internet of Things architecture are introduced and analyzed in Section IV. Section V presents the OpenBoat architecture, while the F-Boat prototype is presented in Section VI. Section VIII presents the experiments and results obtained by monitoring Guanabara Bay. Finally, section IX presents the final considerations and proposals for future works.

II. Related Work

This work deals with the architectural design and prototyping of AI-based autonomous sailboats powered with IoT systems in order to establish seamless connectivity with both onshore control centers and other vessels. There are several approaches related to the architectural design of autonomous sailboats that inspire this work [12], [13], [14]. However, these approaches include neither IoT support nor AI computing capability. Therefore, we also broadened the search scope to include the architectural design of any AI-enabled and IoT-enabled unmanned vehicles.

In a previous work [14], we introduced a sailboat prototype built for experimental validation of the presented architecture. The solution allows for performing long-endurance autonomous missions such as ocean monitoring by introducing sailboat techniques and discussing the needed sensors, actuators, and control laws. The paper also presents the mathematical modeling of the sailboat and control strategies developed using PID and fuzzy controllers to control the sail and the rudder. Furthermore, the paper presents a study of the hardware architecture that enables the system overall performance to be increased. The sailboat movement can be planned through predetermined geographical way-points provided by a base station. Simulated and experimental results are presented to validate the control architecture, including tests performed on a lake. Underwater robotics can rely on such platform as a surface base vessel. However, the paper does not discuss the cost or feasibility of implementing the proposed control architecture on a larger scale. It also does not compare the performance of the proposed control architecture to other existing control architectures for autonomous sailboats.

Burke [15] presents an open-source cloud (4G) connected and controlled self-flying airplane based on a secure, encrypted flight stack with a weight of 300g as an example of a new class of Unmanned Aerial Vehicles (UAVs). The avionics portion weighs only 40g, about 10 times lighter than the previous avionics package achieving the same mission. This new class of aerial vehicles is enabled due to multiple recent advances, including Linux single-board computers, cloud computing, software abstraction, secure encryption, and ubiquitous wireless communications. Among the contributions of the paper, it is presented an Internet connectivity for control from any point on earth.

Freimuth and König [16] propose a framework that uses Four Dimensional Building Information Modelling (4D-BIM) to extract locations of objects that require inspection and compute viable flight missions around the known structures of the construction site. During flight, the UAV uses stereo cameras to detect and avoid obstacles unknown to the model. An additional software component compares the captured point cloud data with the model data, enabling automatic per-object completion checking or reconstruction. The prototype is developed in the Robot Operating System (ROS) and evaluated in Software-In-The-Loop (SITL) simulations to be executable on real UAVs.

Kitts et al. [17] detail the development of an innovative robotic boat that can perform bathymetric mapping of very shallow coastal, estuarine, and inland waters. The boat uses a small waterplane area to provide natural platform stability for a multibeam sonar payload, and a navigation system automatically guides the boat in a *lawn-mowing* pattern to map a region of interest. The boat is operational and used to generate science-quality maps for scientific and civil use; it is also used as a testbed for evaluating the platform for other types of scientific missions and for demonstrating advanced control techniques. It is important to note that the boat was developed as part of a low-cost student design program, which may affect its scalability and robustness.

Mendonza-Chok et al. [18] introduce a new methodology that establishes a hybrid control architecture based on systems engineering concepts, initially defining requirements, risk management, design flexibility, logical decomposition, functional classification, verification-validation-integration, and the technological plan of the resulting prototype.

Garcia and Valvanis [19] detail the technical aspects of a small UAV helicopter specifically designed for testbed. The design aims to provide a general framework allowing researchers to supplement the system with new technologies and add innovation to the vehicle.

Koubaa et al. [20] contribute to the Internet-of-Drones (IoD) concept, its requirements, security considerations, challenges, and importance. Among different applications, they propose a dronemap Planner, a cloud-based management of drones connected through the Internet with cloud integration. The network's reliability is crucial for the system's successful operation. Hard real-time control, which requires zero faults and zero deadline misses, imposes the usage of a very reliable and high-quality-of-service network.

OpenUAV [21] consists of an open-source, web, and cloud-based simulation testbed specifically designed for UAVs. Their solution leverages Containers as a Service (CaaS) technology to enable simulations on the cloud. The paper also includes two use cases demonstrating the use of OpenUAV for machine learning and multi-UAV swarms.

Kuan-Ting Lai et al. [22] present techniques for integrating ArduPilot with the Android mobile platform, which provides AI computing power and 4G/5G connectivity to DIY drones. The paper also presents embedded control software that cooperates with the AI Wings cloud, a cloud server for commanding drone fleets securely, software/hardware design for AIoT drones, and VR simulation for training and testing AI models. The authors propose an authentication protocol based on elliptic-curve cryptography with pseudo-identities and time freshness for checking secure communication. Their work demonstrates the system's effectiveness by building an experimental medical drone service that delivers an automated external defibrillator to people with a sudden cardiac attack in the shortest time possible.

III. Structural Elements of a Sailboat

Sailboat sails are designed to work as an airfoil, similar to an airplane wing. In this way, a sailboat can move forward with the wind blowing from almost any direction. A synergy between the boom (sail) and the rudder is essential for maneuvering a sailboat [23]. Depending on the wind force, even maintaining a straight course sometimes demands a combined operation of the boom and the rudder. In such a way, to develop a controller for a sailboat, both the sail and the rudder should be considered for developing its propulsion and guidance systems.

There are several types of sailboats with different shapes and functionalities. The baseline shape of F-Boat is illustrated in Figure 2 with its main parts being:

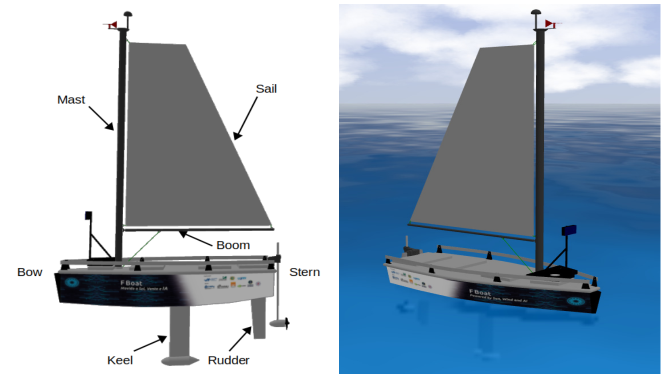

Fig. 2. Overview of main sailboat parts.

- **Hull:** The body of the boat.
- **Boom**: a horizontal pole attached to the mast, which extends the mainsail's foot and controls the attack angle (angle between the sail chord line and the apparent wind).
- **Mast**: a vertical pole that supports the sails.
- **Mainsail**: responsible for taking the major part of the wind, being the biggest sail of the boat; the aperture of the sail (attack angle) can be controlled by releasing or pulling a cable attached to the boom.
- **Bow**: the front part of the boat.
- **Stern**: the back part of the boat.
- **Rudder**: a wing-shaped device that changes the boat's orientation in the water.
- **Rudder stem**: a horizontal pole or rod to control the rudder.
- **Keel**: an element fixed on the bottom of the hull to prevent sideways drift and provide stability.

IV. IoT Reference Architecture

The term *Internet of Things* (IoT) denotes a collection of systems and devices that link real-world sensors and actuators to the Internet. Such devices include internet-connected cars, wearable devices like health and fitness monitoring, watches, human-implanted devices, smart meters, home automation systems, lighting controls, smartphones, and wireless sensor networks.

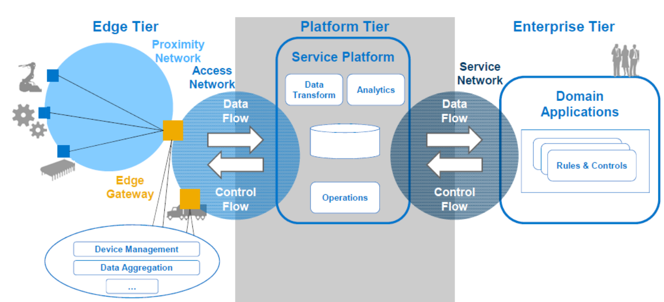

Fig. 3. Internet of Things Reference Architecture [24].

There are many architecture proposals to implement the Internet of Things [24], [25], [26], [27]. Despite their differences in abstraction level, these proposals share many similarities, mainly covering the same elements. In the remainder of this section, we will briefly overview the architecture proposed by the Industrial Internet Consortium [24]. This architecture consists of edge, platform, and enterprise tiers connected by proximity, access, and service networks (see Figure 3).

According to the International Electrotechnical Commission [26], the devices are typically microcontrollers with low processing power, memory capabilities, and intermittent network connections. Combining wireless and wired technologies, such as RFID, Bluetooth, Cellular, ZigBee, Z-Wave, Thread, and Ethernet, can create these networks. As shown in Figure 4, the edge tier employs the proximity network to obtain data from sensors and actuators. Data are transmitted through the access network to the platform tier (i.e., cloud servers), which processes, stores, and relays these data to the enterprise tiers. Communication between the platform and enterprise tiers is accomplished through the service network. The enterprise tier offers end-user interfaces, control commands, and domain-specific applications.

The architecture pattern known as gateway-mediated edge connectivity and management involves a gateway acting as a mediator between a LAN of edge nodes on one side and a WAN on the other. This gateway functions as an endpoint for the WAN network and may also act as a management entity for the edge devices on the LAN, keeping them separate from the WAN [24]. Moreover, edge gateways can move data storage and processing capabilities closer to the endpoints, which can help reduce latency and increase network failure resilience.

V. OpenBoat Architecture

OpenBoat is a modular, abstracted, and flexible architecture for an IoT-enabled autonomous sailboat. An OpenBoat-based sailboat must sail autonomously, collect and preprocess data onboard, and communicate.

Due to many uncertainties and non-linearities, marine environments are challenging for several computational systems, mainly regarding communication. In this work, we aim to use a robotic sailboat for monitoring applications in a marine region considered extremely important for the population, named Guanabara Bay, in Rio de Janeiro. In addition to the marine environment by itself, this region has strong marine currents due to the rise and fall of the tide and extreme wind situations that occur at certain times of the year. Also, in the deepest channels, waves are up to 2 meters in height. Another possible application region, at the east of Natal, Northeastern Brazil, also involves substantial differences between low and high tides (up to 3 meters) and other weather conditions characteristic of the sea environment, such as extreme winds up to 35 knots. Thus, not only to ensure the sailboat's navigability, depending on the weather in extreme conditions, but also to ensure that the communication part works properly, the IoT architecture must be thought of from the beginning of the sailboat's construction, and some functionalities require adequate equipment and material. Hence, since its initial conception, the proposed IoT architecture had to be designed to work with the adverse situations of the sea, which is considered an extremely dynamic environment. We had to design an architecture that meets all the desired robustness requirements. The robustness part of the architecture will be discussed below, in Section VII.

Therefore, the architecture of OpenBoat considers a specialized monitoring system consisting of specific sensors and a microcontroller responsible for collecting, processing, and relaying data to upper layers. For example, this monitoring system can be a data collection and processing system for water quality (e.g. pH, turbidity, gas dilution, etc.).

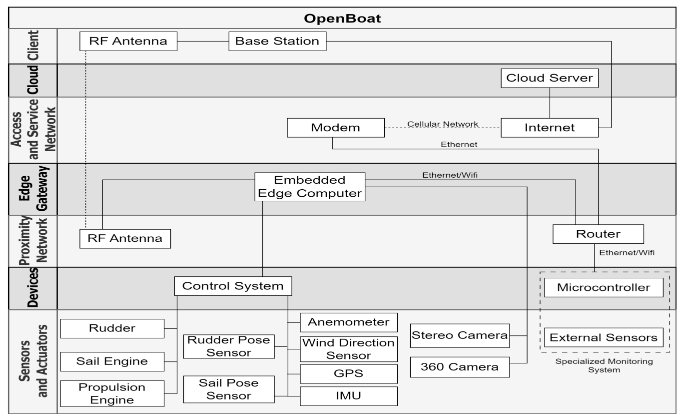

Fig. 4. OpenBoat Reference Automating Architecture.

The OpenBoat architecture is divided into six layers based on the IoT reference architecture in Section IV. These layers are the Client, Cloud, Access and Service Network, Edge Gateway, Devices, and Sensors and Actuators (see Figure 4). The following subsections describe each of these layers in detail.

A. Sensors and Actuators Layer

The Sensors and Actuators Layer is the lowest level in our architecture. It contains the components necessary for the boat to capture data from the physical environment and perform navigation actions (e.g. rudder and sail pose). Thus, upper layers can be provided with data that enables autonomous vessel navigation and specialized monitoring. OpenBoat requires the use of an anemometer to obtain wind speed, a windsock for wind direction (essential to assist in the sail angle), a GPS for boat geolocation (position), and an inertial measurement unit that provides the sailboat angles in relation to a global origin (for direction determination).

Among the actuators, we include a propulsion motor for controlling the rudder and another actuator for controlling the sail angle. The external sensors of the specialized monitoring system also comprise this layer. Finally, we require stereo and panoramic cameras, necessary for computer vision and real-time navigation view by a remote operator.

B. Devices Layer

The Devices Layer contains microcontrollers directly connected to the sensors and actuators. The Control System is responsible for lower-level control, directly controlling the actuators and collecting navigation sensor data. There is currently a range of devices specialized in this function, such as the many versions of Pixhawk and the Navio2. These devices run navigation control firmware, such as PX4 or Ardupilot. The Control System connects to an Edge Computing component in order to enable remote operation and AI-based higher-level control such as path planning, complex maneuvers, and computer vision assistance.

The second component of the Devices layer is the microcontroller of the specialized monitoring system. It must be able to connect to the specialized monitoring sensor and extract the data captured by this component. Additionally, this microcontroller needs to be connected to the Proximity Network Layer to relay specialized monitoring data to local and remote repositories.

C. Proximity Network Layer

In this layer, we have two important components: the radio antenna and the router. The first enables data exchange with the ground station via radio frequency when the ground station is near the boat. This way, a boat that follows the OpenBoat architecture has the ability to communicate with the ground station without the Internet. The router interconnects all the components of the device and edge layers, as well as the modem, for external network access. This way, we have all the boat components connected to a LAN with a gateway for Internet access.

D. Edge Gateway Layer

The Edge Gateway layer has an Embedded Edge Computer responsible for data processing, mainly machine learning and filtering algorithms, data storage, and data transmission to remote servers (e.g. the cloud) [25]. Bringing processing and storage capability to the border reduces latency and network traffic and offers robustness against access network failure.

Given the heterogeneity of the devices that make up the OpenBoat, the use of a middleware is necessary so that there is a centralized awareness of context, the ability to interact with all devices, as well as to trigger events arising from the recognition of a specific condition, such as high latency in the internet connection, which can influence the operation of the boat. For example, the middleware should be able to interrupt the sending of data to the cloud and perform local storage, make the local database available, perform switching related to unloading processing to the cloud, and host APIs, among other important functions.

The software responsible for this resource management is the IoT Middleware. There are many commercial and open-source solutions tailored to this end, such as AWS IoT Greengrass [28], Azure IoT Edge [29], ThingsBoard [30], ThingWorx [31], ThingSpeak [32], OpenRemote [33], among others. They all offer edge computing capability with cloud synchronization, but some are not interoperable with different vendors. IoT Middleware is also a very active research field [34], [35], [36].

The Embedded Edge Computer can also run higher-level control algorithms that interact with the actuators of the boat for navigation purposes. These algorithms include path planning and obstacle avoidance. The first is highly complex due to the roles of wind speed and direction in navigation actions. Obstacle avoidance benefits from computer vision. Running these algorithms in a powerful embedded computer allows for deploying advanced AI techniques.

E. Access and Service Network Layer

The Access and Service Network layer offers Internet access to the boat through a regular modem connected to the WAN interface of the router in the Proximity Network Layer.

F. Cloud Layer

The Cloud Layer refers to cloud resources such as processing and storage services essential for autonomous vehicles. Currently, there are some very robust providers with services to meet the needs of the Internet of Things (e.g. AWS, Azure, and Google). OpenBoat uses cloud services for heavy machine learning processing offloading, real-time telemetry logging, and remote driving, among others.

G. Client Layer

We propose the inclusion of a second RF antenna and a Base Station in the Client Layer as an option for connecting the base station and the sailboat. It can be used for sending telemetry directly via radio frequency without going through the internet.

Finally, in the client layer, there is a base station. Also known as a Ground Control Station (GCS), it is a simple workstation with a computer, laptop, or desktop, capable of presenting graphically, numerically, or textually the actuators behavior and the values of the sensors, in real time. Examples of software used as GCS include Mission Planner, APM Planner, QGroundControl, LibrePilot, and MAVProxy.

VI. F-Boat Architecture

F-Boat is a robotic sailboat built with Commercial Off-The-Shelf (COTS) components for cost-effectiveness, time efficiency, reliability, flexibility, and accessibility. Project documentation, source code, and images and videos of the building process and field deployment are available at https://www.natalnet.br/nboat/ and https://github.com/medialab-fboat.

The baseline design of F-Boat is shown in Figure 5 and follows the seven-layer model of OpenBoat as shown in Figure 4. These layers are Client, Cloud, Access and Service Network, Edge Gateway, Proximity Network, Devices, Sensor, and Actuator, and their implementations are described in the following.

A. Sensor and Actuator Layer

F-Boat is equipped with a Nagano electrical winch with a pulling force of 3000 pounds to control the sail. This winch is responsible for releasing and collecting the cable that limits

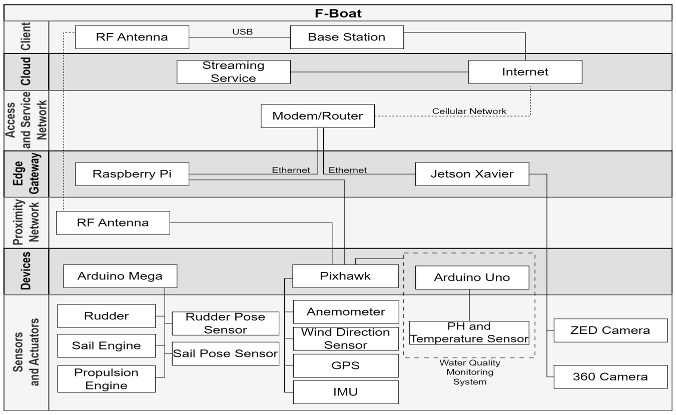

Fig. 5. F-Boat Automation Architecture.

the range of motion of the boom, which moves freely to both sides within this range of motion. A Bochen potentiometer model WXD3-13 with a multi-turn design and resistance of 4.7 kilo-Ohms feedbacks on the rotational state of the winch, enabling the calculation of the amount of released cable.

A linear actuator Vinitrônica model V1 steers the rudder. A linear potentiometer model WH148-1 with a resistance of 10 kilo-Ohms enables the calculation of the rudder angle as a function of the position of the linear actuator. There is also an electrical propeller for conventional boats (PHA-DIGSW54 54Lbs). This propeller is for emergencies and complex maneuvers.

The sensors for allowing autonomous navigation consist of a Davis 6410 anemometer (capable of measuring both wind speed and direction), a Holybro GPS module model NEO-M8N with an embedded compass, and an inertial measurement unit (IMU) with 9DoF (3-axis accelerometer, 3-axis magnetometer, and 3-axis gyroscope). Besides, a 360 degrees panoramic camera and a Zed 3D stereo camera provide real-time video streaming for human remote operation and computer vision-based autonomous navigation.

Finally, specific sensors and actuators of the payload monitoring system complete the list of equipment in this layer. In this work, we use water temperature and water pH sensors.

B. Devices Layer

The main component of this layer is the Pixhawk, a hardware platform designed for autopilot systems. The autopilot system is Ardupilot, a full-featured and reliable open-source firmware used by industry, research organizations, and amateurs.

This layer also features an Arduino Mega that acts as a motor driver for the rudder actuator, the sail actuator, and the propeller. This Arduino makes both the linear actuator connected to the tiller extension and the electric winch work as PWM servo motors that adjust the angle of the rudder and the sail, respectively. This Arduino also runs an algorithm that calculates the target angle of the sail as a function of the apparent wind direction. We run this algorithm on Arduino for simplicity as changing Ardupilot's default sail control is more complex.

Finally, the microcontroller of the payload monitoring system completes this layer. In this work, we use an Arduino Uno to collect, process and retransmit data from the temperature and the pH sensors.

C. Proximity Network Layer

The proximity network layer consists of an RC receiver, a 433MHz RF antenna, and an Ethernet LAN (included in the modem device). The RC receiver allows remote operation with an RC controller; the 433MHz RF antenna is an alternative connection between the base station and the sailboat without going through the Internet; finally, the Ethernet LAN interconnects local devices and a cellular gateway.

D. Edge Layer

F-Boat has an embedded single-board computer (SBC) NVidia-Xavier that provides enough processing power to allow computer vision tasks in real time. This boosts the stages of perception (object detection, classification, tracking, among others), localization (providing a reference path to the robot), planning (to define a new route), and controlling (sending commands to actuators). Basic sensory data necessary for short-term navigation include wind direction, GPS position, compass, rudder angle, and sail angle. The literature is particularly vast on object detection and collision avoidance [37], [38].

A Raspberry Pi 3B model is in the Edge Gateway Layer. This device is an SBC that manages the communication between the autopilot system (Pixhawk/Ardupilot) and other devices. The Raspberry Pi also runs the IoT middleware responsible for resource management and collaboration with the cloud system.

E. Access and Service Network Layer

Serving the Access and Service Network layer, F-Boat has a modem/router model Aquário MD-4000, which is capable of interconnecting the Raspberry Pi and Jetson Xavier in an Ethernet network and serves as a modem for Internet access using a 4G cellular network. This way, the boat's components are interconnected in the same Ethernet LAN with access to the Internet.

F. Cloud Layer

To date, the cloud layers offer only scalable live video streaming via Facebook Watch and WebRTC. In the future, we plan to add other services to this layer such as processing and storage.

G. Client Layer

The base station is a laptop running Mission Planner software that shows graphically, numerically, or textually the real-time status of the sensors and actuators. Mission Planner can also be used to send navigation commands to F-Boat, allowing for either manual control or autonomous navigation by sending directions, desired speeds, and waypoints for the boat to reach. These capabilities can lead to more accurate and safer navigation.

VII. Architecture's Resilience and Fault Tolerance

Robustness requirements were considered in the IoT architecture itself, being thought of, and in a way imposed, from the hardware design part (physical construction). Thus, we distinguish two aspects: the hardware construction characteristics that allow communication robustness and the IoT robustness. To meet these requirements effectively, we tried to answer some questions such as: 1) how the physical construction of the sailboat should be, in terms of communication, both between internal components and between the sailboat and the land station and other instances, so that the robustness of the IoT system is guaranteed? 2) What happens to communication in case of bad weather conditions? 3) What is the attitude to be taken by the sailboat in case of a loss of communication?

Regarding the design and physical project of the sailboat, which is open source (https://www.natalnet.br/nboat/ and https://github.com/medialab-fboat), some relevant points were considered, which also bring robustness to the IoT part as listed below. The hull offers separate and watertight compartments with different functionalities, in which the (separate) behaviors of the proposed control architecture must be implemented [39]. The Sensors and Actuators layer, and the Devices layer are connected to each compartment, which guarantees that the control part of the architecture is distributed and one behavior does not depend on the other to work. The communication between these behaviors (over devices and sensors) can be done redundantly, both via the onboard WiFi and via a wired connection (Ethernet). The behaviors are effectively controlled on a SBC (single board computer), which also implements behaviors from the vision part (N-Vidia Xavier), over which we use reinforcement learning to maintain the sailboat's seaworthiness [39].

Communication between the sailboat and the shore (base) station (with higher parts of the architecture) is handled in several ways, depending on the application. On larger distances, it is currently done using a 4G cellular network. Still, it could even use satellite communication, which is the primary form of communication on the high seas. Also, communication can be done by RC radio and wifi (at very short distances). This guarantees operation in any location as long as the sailboat is operational.

The physical characteristics of the sailboat construction [13], [40] is also considered to guarantee the navigability even in case of serious problems or bad weather conditions, which in turn also helps in the robustness of the IoT architecture. The bottom of the sailboat has 3-4cm of foam (which can help maintain buoyancy, even if a lot of water comes into the boat). The electronic components are all placed inside watertight boxes in each compartment. The wired part (Ethernet) is always placed inside ducts, with total sealing between each compartment, to prevent water from entering. The sail was designed to be undersized, giving less speed but providing more stability. The sailboat is a monohull with a low center of gravity, unlike catamarans. This ensures buoyancy even in case of bad weather conditions. Furthermore, we have the possibility of using a motherboard developed by the team (see Figure 6) with slave distribution boards to avoid jumpers, providing more robustness in the internal communication.

Fig. 6. Mother and slave boards specific developed and available for increasing components connectivity.

With regards to energy robustness, the sailboat is currently self-sustainable (solar energy), having a battery bank (105A+105A in parallel), which provides much more power than it would consume in 1 day if fully charged, and if it is propelled only by the sail. The amount of energy provided by the system lasts more than 48 hours. In the load tests we carried out, even at night or in case of heavy clouds, the system maintains itself, guaranteeing navigability (and consequently communication). Of course, disasters can occur, and no marine system is robust enough to prevent accidents [41]. For example, in the case of a collision with an other vessel or a wreckage, or in extreme storms or hurricanes, the communication part may be affected. But, it is worth noting that, regarding the situation of loss of communication, the navigation system based on Pixhawk offers a failsafe functionality, which can be activated. In this case, the system can trigger an alternative route back home (or to another location), which can be followed if the sailboat's seaworthiness is functional.

With regards to maintaining the robustness, communication between the upper levels can be achieved through simple measures already widely adopted in the various architectures operating on land that exist. We believe that the critical parts of our application is the internal communication between the robotic sailboat components and the communication between the autonomous vehicle and the ground station, which is why these aspects were the most discussed in this section.

VIII. Experiments and Results

Guanabara Bay provides social, economical, and environmental asset services for more than 10 million people. However, this area is also prone to extreme events that can increase the sediment load and nutrient input into the bay, affecting water quality and ecosystem health [4]. Thus, monitoring the bay should be essential.

In this experiment, manual remote control was initially used to guide the vessel with a radio controller. Next, tests of the automatic control were performed. For this stage, Ardupilot's guided navigation system controlled the sail and the rudder.

In the last set of tests, we generated different navigation scenarios.

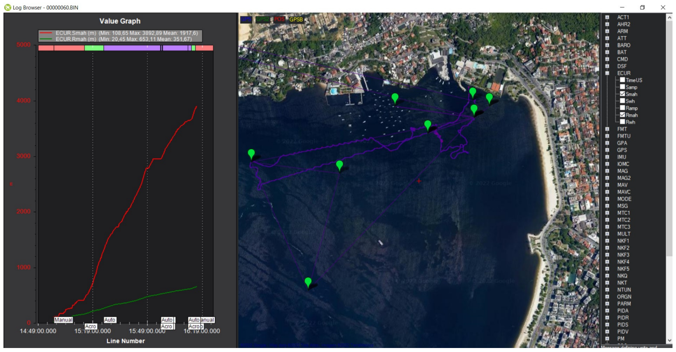

Fig. 7. Graphical representation of the Energy consumption of the Rudder (green) and the Sail (red).

The energy consumption of the rudder and sail was relatively low, as shown in Figure 7, with the rudder consuming less than the sail: 1000 mAh and 4000 mAh respectively. For the sail, it would give $4000/1.5 = 2.67$A. The F-Boat has two 105A batteries attached to a solar panel that manages to produce all the energy needed for the complete autonomy of the vehicle. Figure 8 shows the total amount of the vessel's consumption. Battery voltage remains between 11v and 12v, and current remains constant between 231.68mA and 5562.85mA.

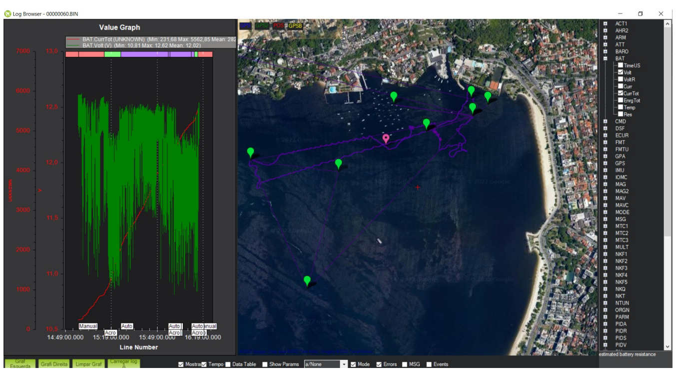

Fig. 8. Graphical representation of energy consumption: voltage (red) and current (green).

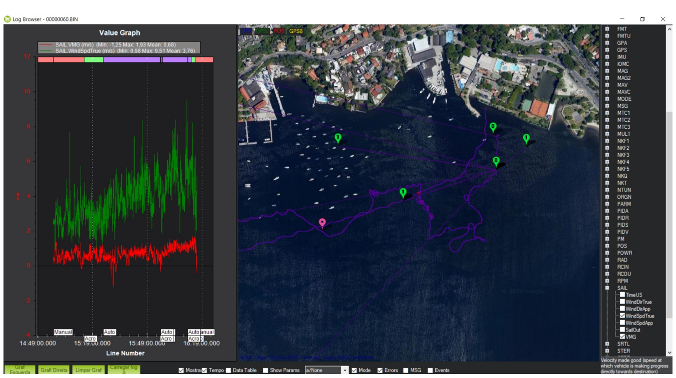

Fig. 9. Graphical representation of real wind speed (red) and apparent wind (green).

Figure 9 illustrates the speed on the boat, where the VMG (correct speed or speed at which the vehicle is advancing directly towards the destination) and the actual wind (two meters per second) are found. The VMG wind of the vessel is close to 1.5m/s, while the real wind has values between 2m/s and 8m/s.

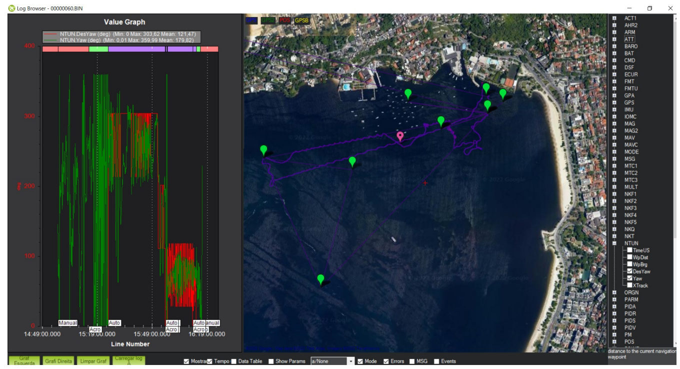

Fig. 10. Graphical representation of current direction (green) and desired direction (red).

It is possible to see the path taken by the F-Boat, given the defined *waypoints*. The path navigated by the vessel, as shown in Figure 10, is started manually at some *waypoints* and then passed via ACRO mode to the autopilot.

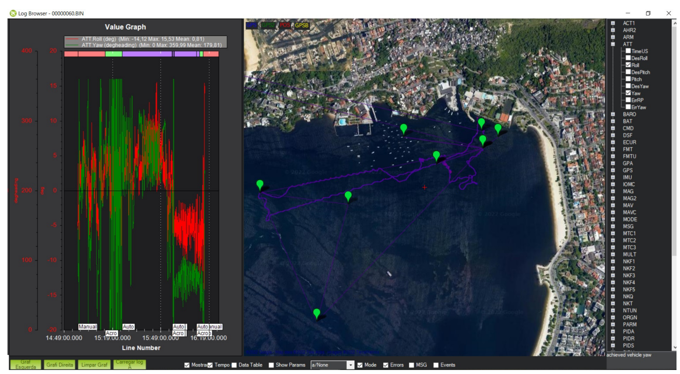

Fig. 11. Graphical representation of the Roll direction in red and the Yaw direction in green.

Figure 11 illustrates the vessel's heading, measured in degrees of the roll (*roll*) and pitch (*yaw*) angles, which are varying between 15 and -15 for *roll* and 0 to 400 for *yaw*.

As previously mentioned, the vessel used the PI controller in the actual tests to fix errors in the input responses of the navigation system to the actuators. Figure 12 shows in green and blue the control of the winch actuator for sail control using PI, and the control of the rudder actuator using PI in red.

IX. Conclusion

To date, our proposed architecture for automating sailboats in the context of the Internet of Things (IoT) and Artificial Intelligence (AI) technologies, the so-called Artificial Intelligence of Things (AIoT), has been implemented, tested, and validated in practice. We call our proposed general sailboat automating architecture OpenBoat. This architecture consists

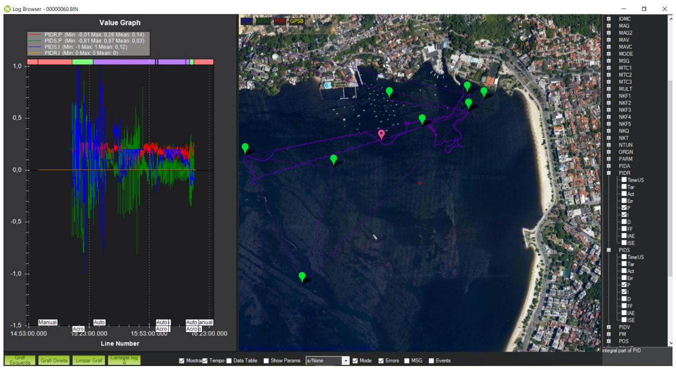

Fig. 12. Graphical representation of the vessel's PI Control for the rudder (red) and for the sail(blue and green).

of six layers that address traditional automation, AI-processing units, interconnectivity, and interoperability with other subsystems. These six layers are Client, Cloud, Access and Service Network, Edge Gateway, Devices, and Sensors and Actuators.

We also introduced F-Boat, an AIoT-enabled autonomous sailboat prototype built with Commercial Off-The-Shelf (COTS) components and following the OpenBoat reference architecture. Building prototypes with COTS components offers numerous advantages in cost-effectiveness, time efficiency, reliability, flexibility, and accessibility. By leveraging these readily available components, we can streamline the development process, reduce costs, and benefit from established quality standards. F-Boat is designed and built for robustness to guarantee its operation under extreme events, such as high temperatures and bad weather, for extended periods of time. Project documentation, source code, and images and videos of the building process and field deployment are available at https://www.natalnet.br/nboat/ and https://github.com/medialab-fboat.

The results of field experiments demonstrate the functionalities of the prototype and the AIoT capability of the general architecture. We conducted these experiments at Guanabara Bay, an important marine and coastal ecosystem in southeastern Brazil. This area is prone to extreme events that can significantly impact its ecosystems and population. Extreme events in this area include severe storms, heavy rainfall, and tidal surges that can cause flooding.

Our architecture favors tasks that require a higher level of persistence where energy self-sufficiency is desired, such as monitoring and patrolling large water spaces like rivers, lakes, and the ocean, mainly in coastal regions. These ecosystems provide a wide range of goods and services essential for human well-being, including food, climate regulation, coastal protection, nutrient cycling for plants and animals, among others. However, these ecosystems are vulnerable to extreme events, which can significantly impact their biodiversity, productivity, and the services they provide. Such events in the marine and coastal ecosystems include storms, floods, heat waves, cold spells, droughts, and sea-level rise, causing physical damage, such as the destruction of habitats, loss of biodiversity, and changes in the structure and functioning of these ecosystems.

Future work includes using F-Boat as a testbed to compare and evaluate techniques for autonomous sailing. Often, sailing is not just pointing the boat in the direction to be followed, and maneuvers and course changes are commonly necessary. Deploying AI technologies to make maneuver and course decisions is a promising research field with great potential to improve traditional control theory-based techniques.

REFERENCES

[1] S. Samadi, "The convergence of ai, iot, and big data for advancing flood analytics research," *Frontiers in Water: Rising Stars 2021*, 2022.

[2] J. C. Molina-Molina, M. Salhaoui, A. Guerrero-González, and M. Arioua, "Autonomous marine robot based on ai recognition for permanent surveillance in marine protected areas," *Sensors*, vol. 21, no. 8, 2021. [Online]. Available: https://www.mdpi.com/1424-8220/21/8/2664

[3] W. Luiz-Silva and A. C. Oscar-Júnior, "Climate extremes related with rainfall in the state of rio de janeiro, brazil: a review of climatological characteristics and recorded trends," *Natural Hazards*, vol. 114, no. 1, pp. 713–732, 2022.

[4] A. S. Fries, J. P. Coimbra, D. A. Nemazie, R. M. Summers, J. P. S. Azevedo, S. Filoso, M. Newton, G. Gelli, R. C. N. de Oliveira, M. A. R. Pessoa *et al.*, "Guanabara bay ecosystem health report card: Science, management, and governance implications," *Regional Studies in Marine Science*, vol. 25, p. 100474, 2019.

[5] J. a. Esteves, L. i. Gomes, and A. Costa, "Collision avoidance system for an autonomous sailboat," in *IECON 2017 - 43rd Annual Conference of the IEEE Industrial Electronics Society*, 2017, pp. 3539–3544.

[6] H. Abrougui, S. Nejim, and H. Dallagi, "Modeling and autopilot design for an autonomous catamaran sailboat based on feedback linearization," in *2019 International Conference on Advanced Systems and Emergent Technologies (IC_ASET)*, 2019, pp. 130–135.

[7] D. H. dos Santos and L. M. G. Gonçalves, "Performance evaluation of propulsion control techniques for autonomous sailboat," in *2020 Latin American Robotics Symposium (LARS), 2020 Brazilian Symposium on Robotics (SBR) and 2020 Workshop on Robotics in Education (WRE)*, 2020, pp. 1–6.

[8] I. Astrov and A. Udal, "Neural predictive tracking control of catamaran model sailboat for situation awareness applications," in *2020 IEEE 24th International Conference on Intelligent Engineering Systems (INES)*, 2020, pp. 153–158.

[9] C. Viel, U. Vautier, J. Wan, and L. Jaulin, "Platooning control for sailboats using a tack strategy," *International Journal of Control, Automation and Systems*, vol. 17, no. 9, pp. 2310–2320, 2019.

[10] A. G. d. Silva Junior, D. H. d. Santos, A. P. F. d. Negreiros, J. M. V. B. d. S. Silva, and L. M. G. Gonçalves, "High-level path planning for an autonomous sailboat robot using q-learning," *Sensors*, vol. 20, no. 6, 2020. [Online]. Available: https://www.mdpi.com/1424-8220/20/6/1550

[11] C. Liu, S. Yang, T. Duan, J. Huang, and Z. Wang, "Motion control of an one-meter class autonomous sailboat," in *2018 IEEE 8th International Conference on Underwater System Technology: Theory and Applications (USYS)*. IEEE, 2018, pp. 1–6.

[12] F. Plumet, C. Pêtrès, M.-A. Romero-Ramirez, B. Gas, and S.-H. Ieng, "Toward an autonomous sailing boat," *IEEE Journal of Oceanic Engineering*, vol. 40, no. 2, pp. 397–407, 2015.

[13] A. G. Júnior *et al.*, "N-boat: An autonomous robotic sailboat," in *2013 Latin American Robotics Symposium and Competition*, 2013, pp. 24–29.

[14] D. H. Santos, A. G. S. Junior, A. Negreiros, J. ao V. B. Souza e Silva, J. E. Alvarez-Jácobo, A. Araújo, R. V. Aroca, and L. M. G. Gonçalves, "Design and implementation of a control system for a sailboat robot," *Robotics - MDPI*, no. 24, 2016. [Online]. Available: https://www.mdpi.com/1424-8220/20/6/1550

[15] P. J. Burke, "A safe, open source, 4g connected self-flying plane with 1 hour flight time and all up weight (auw) ¡300 g: Towards a new class of internet enabled uavs," *IEEE Access*, vol. 7, pp. 67 833–67 855, 2019.

[16] H. Freimuth and M. König, "A framework for automated acquisition and processing of as-built data with autonomous unmanned aerial vehicles," *Sensors*, vol. 19, no. 20, 2019. [Online]. Available: https://www.mdpi.com/1424-8220/19/20/4513

[17] C. Kitts, P. Mahacek, T. Adamek, K. Rasal, V. Howard, S. Li, A. Badaoui, W. Kirkwood, G. Wheat, and S. Hulme, "Field operation of a robotic small waterplane area twin hull boat for shallow-water bathymetric characterization," *Journal of Field Robotics*, vol. 29, 11 2012.

[18] J. Mendoza-Chok, J. C. C. Luque, N. F. Salas-Cueva, D. Yanyachi, and P. R. Yanyachi, "Hybrid control architecture of an unmanned surface vehicle used for water quality monitoring," *IEEE Access*, vol. 10, pp. 112 789–112 798, 2022.

[19] R. Garcia and K. Valavanis, *The Implementation of an Autonomous Helicopter Testbed*, 12 2008, vol. 54, pp. 423–454.

[20] A. Koubâa, B. Qureshi, M.-F. Sriti, A. Allouch, Y. Javed, M. Alajlan, O. Cheikhrouhou, M. Khalgui, and E. Tovar, "Dronemap planner: A service-oriented cloud-based management system for the internet-of-drones," *Ad Hoc Networks*, vol. 86, pp. 46–62, 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1570870518306814

[21] M. Schmittle, A. Lukina, L. Vacek, J. Das, C. P. Buskirk, S. Rees, J. Sztipanovits, R. Grosu, and V. Kumar, "Openuav: A uav testbed for the cps and robotics community," in *2018 ACM/IEEE 9th International Conference on Cyber-Physical Systems (ICCPS)*, 2018, pp. 130–139.

[22] K.-T. Lai, Y.-T. Chung, J.-J. Su, C.-H. Lai, and Y.-H. Huang, "Ai wings: An aiot drone system for commanding ardupilot uavs," *IEEE Systems Journal*, pp. 1–12, 2022.

[23] T. Whidden and M. Levitt, *The Art and Science of Sails*. Brooklin: Seapoint Books, 2016.

[24] I. Consortium *et al.*, "The industrial internet of things volume g1: Reference architecture, version 1.9, 2019-june-19," *Ind. Internet Consort*, 2019.

[25] W. Yu *et al.*, "A survey on the edge computing for the internet of things," *IEEE access*, vol. 6, pp. 6900–6919, 2017.

[26] I. E. Kommission and I. . P. Team, *IoT 2020: Smart and Secure IoT Platform : White Paper*, ser. White paper. International Electrotechnical Commission, 2016. [Online]. Available: https://books.google.com.br/books?id=aItwAQAACAAJ

[27] B. Mazon-Olivo and A. Pan, "Internet of things: State-of-the-art, computing paradigms and reference architectures," *IEEE Latin America Transactions*, vol. 20, no. 1, p. 49–63, May 2021. [Online]. Available: https://latamt.ieeer9.org/index.php/transactions/article/view/5037

[28] AWS, "Aws iot greengrass," 2023. [Online]. Available: Availableathttps://aws.amazon.com/greengrass/(05/15/2023)

[29] Azure, "Azure iot edge," 2023. [Online]. Available: Availableathttps://azure.microsoft.com/en-us/products/iot-edge(05/15/2023)

[30] ThingsBoard, "Thingsboard open-source iot platform," 2023. [Online]. Available: Availableathttps://thingsboard.io/(05/15/2023)

[31] ThingWorx, "Thingworx iiot solutions platform," 2023. [Online]. Available: Availableathttps://www.ptc.com/en/products/thingworx(05/15/2023)

[32] ThingSpeak, "Thingspeak for iot projects data collection in the cloud with advanced data analysis using matlab," 2023. [Online]. Available: Availableathttps://thingspeak.com/(05/15/2023)

[33] OpenRemote, "The open source iot platform for manufacturers and integratorb," 2023. [Online]. Available: Availableathttps://www.openremote.io/(05/15/2023)

[34] A. H. Ngu, M. Gutierrez, V. Metsis, S. Nepal, and Q. Z. Sheng, "Iot middleware: A survey on issues and enabling technologies," *IEEE Internet of Things Journal*, vol. 4, no. 1, pp. 1–20, 2017.

[35] J. Zhang, M. Ma, P. Wang, and X. dong Sun, "Middleware for the internet of things: A survey on requirements, enabling technologies, and solutions," *Journal of Systems Architecture*, vol. 117, p. 102098, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1383762121000795

[36] M. A. A. da Cruz, J. J. P. C. Rodrigues, J. Al-Muhtadi, V. V. Korotaev, and V. H. C. de Albuquerque, "A reference model for internet of things middleware," *IEEE Internet of Things Journal*, vol. 5, no. 2, pp. 871–883, 2018.

[37] Y. Huang, L. Chen, P. Chen, R. R. Negenborn, and P. Van Gelder, "Ship collision avoidance methods: State-of-the-art," *Safety science*, vol. 121, pp. 451–473, 2020.

[38] J. N. Yasin, S. A. Mohamed, M.-H. Haghbayan, J. Heikkonen, H. Tenhunen, and J. Plosila, "Low-cost ultrasonic based object detection and collision avoidance method for autonomous robots," *International Journal of Information Technology*, vol. 13, pp. 97–107, 2021.

[39] A. P. F. Negreiros, E. C. Vasconcellos, D. H. Santos, E. W. G. Clua, and L. M. G. Gonçalves, "Autonomous sailing behavioral learning high level control architecture," in *Proceedings of Latin American Robotics Symposium*. IEEE Xplore, October 2023, pp. 1–6.

[40] Á. P. F. d. Negreiros, "N-boat: Project and development of an autonomous robotic sailboat (in portuguese)," Master's thesis, Graduate Program in Computer and Electrical Engineering, Federal Universty of Rio Grande do Norte, Brasil, 2019.

[41] M. Paravisi, D. H Santos, V. Jorge, G. Heck, L. M. Gonçalves, and A. Amory, "Unmanned surface vehicle simulator with realistic environmental disturbances," *Sensors*, vol. 19, no. 5, p. 1068, 2019.