Control and Robotics Express Communications
https://www.sciltp.com/journals/crec

Article

An Open-Source, Low-Cost Framework for Real-Time IoT–WebGIS Integration in Distributed Monitoring and Automation System

Filippo D'Ippolito [1,*], Giovanni Garraffa [1], Marcello La Guardia [2] and Antonino Sferlazza [1]

[1] Department of Engineering, University of Palermo, Viale delle Scienze, Edificio 10, 90128 Palermo, Italy
[2] Department of Engineering, University of Messina, 98158 Messina, Italy
* Correspondence: giovanni.garraffa@unikore.it

How To Cite: D'Ippolito, F.; Garraffa, G.; La Guardia, M.; et al. An Open-Source, Low-Cost Framework for Real-Time IoT–WebGIS Integration in Distributed Monitoring and Automation System. *Control and Robotics Express Communications* **2026**, *1*(1), 3.

Received: 14 January 2026
Revised: 6 May 2026
Accepted: 19 May 2026
Published: 29 May 2026

Abstract: Recent advances in Internet of Things (IoT) and Geographic Information Systems (GIS) enable re-al-time monitoring and visualization of geospatial data through low-cost solutions. However, in-tegrating IoT devices with WebGIS platforms still requires complex configuration of multiple components, including hardware, databases, and geospatial services. This work presents an open-source, low-cost framework that simplifies IoT–WebGIS integration through a containerized and automated architecture. The proposed platform combines real-time sensor acquisition (tem-perature, humidity, and GNSS positioning) with cloud-based data storage and interactive WebGIS visualization. The system is fully open-source, reducing costs to hardware components only, and is designed to be easily deployable without advanced configuration skills. The framework is validated through experimental implementation, demonstrating its capability for real-time monitoring and data management. The source code is provided as supplementary material to ensure reproducibility and facilitate further developments.

Keywords: IoT; cloud computing; GIS; data management; real time monitoring; WebGIS

1. Introduction

The Cloud Computing is one of the concepts more developed in recent years in the world of computer science. The United States National Institute of Standards and Technology defined the Cloud Computing as "a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction" [1]. Every kind of dataset can be hosted on the web (services, resources, data). The key features of Cloud Computing are elasticity, availability of on-demand services, remote access from everywhere, sharing of resources [2].

At the same time the opportunities offered by Internet of Things (IoT) recently emerged due to the possibility to connect smartphones, computers, actuators, cars, home appliances, and, generally, electronic devices to share real time information on the web [3,4].

As analyzed in a scientific review study in this field [5], the International Telecommunication Union (ITU) defined the IoT as "a global infrastructure for the Information Society, enabling advanced services by interconnecting (physical and virtual) things based on, existing and evolving, interoperable information and communication technologies".

The integration of smart services connected to IoT lead the society to enhance sustainability, inclusion, economical aspects [2]. IoT technology is considered as one of the most relevant technologies of the future, offering the possibility of creating a global network of interacting machines and devices [6]. IoT systems rapidly diffused their integration in the “smart home” world, where users can remotely control the lighting of house environments [7], receive alarm signals on their smartphones when smoke is indoor detected [8], real-time observe the activities of the kids [9]. The spread of this technology underlined new risks related to the possibility of attacks that can compromise personal safety, and it has been necessary to provide network security solutions [10–12].

Recent experimentation employed IoT low-cost technology to analyze environmental conditions datasets (carbon dioxide, methane, carbon monoxide, rain, dust particulates, humidity, and temperature) useful to monitor the pollution of urban centers with a reduced cost [13]. IoT technology has been also recently experimented in the field of agriculture to avoid wastes of water, developing a low-cost smart irrigation system including irrigation schedule estimation, neural based decision making and remote data viewing [14]. The integration of IoT systems allowed on recent research to employ low-cost technology for BIM (Building Information Modelling) and real time monitoring integration in the world of constructions on distinct phases of the life cycle of the structures [15–17]. The definition of smart building came properly from the integration of IoT technology [18–20].

Considering the word of territorial analysis, last decades are characterized by the spread of territorial analysis models based on GIS. GIS solutions connect geographic and semantic information of geospatial datasets. These geospatial tools allowed to make territorial operations in several fields of research, providing the integration of different kinds of dataset (transportation, security, business, terrain analysis, etc. [21–25]. GIS applications allow to work with large geographic dataset associating geospatial representation with semantic metadata description, merging spatial data with computer and network technologies.

The diffusion of mobile devices as laptops, tablet and smartphones and the continuous improvement of internet connection made even more useful to share geospatial information on web, employing the WebGIS solution [26]. WebGIS allowed to share static web maps, to make real-time data integration, to access from different independent GIS analysis tools [27,28].

The OGC (Open Geospatial Consortium), the core of the open interoperability standards for geospatial information, provided the main services of map visualization, WMS (Web Map Services), and the main services for loading geospatial features, WFS (Web Feature Services) [29]. The diffusion of further technological solutions like GPS (Global Positioning System) on smartphones offered new possibilities on geospatial dataset integration on WebGIS. In this field also UWB (Ultra Wide Band) solutions allowed to employ localization system where GPS signal is not present [30,31].

With the evolution of geospatial technologies, the use of WebGIS opensource application has been implemented in many fields of research: historical buildings management and accessibility [32,33], power plants localizations [34], pandemic evolution [35], study of ecosystems [36], etc. Last years have been characterized by the diffusion of collaborative web platforms, useful to coordinate and manage environmental problems [37].

The implementation of low-cost facilities based on open standard solutions is more scientifically relevant than the study of solutions offered by commercial ones. In fact, low-cost solutions offer the opportunity of working with acquisition/elaboration/fruition of datasets obtained by networks of sensors to students, researchers, and teams with few economic resources. In this way it is not necessary to buy or rent property licences of hardware/software commercial solutions which can be prohibitively expensive.

The integration of IoT and WebGIS open-source solutions results strategic due to the wide range of possibilities of analysis offered by these technologies, and as will be shown, it represents a field of growing scientific interest. The solution shown in this paper is a smart ready to use equipment that integrate IoT and WebGIS technologies.

The platform employs an automated process generating a chain of single modules based on containerization and virtualization. The components of the framework are totally opensource, and the costs for users are limited only to the IoT boards and sensors. The system allows users to run a real time dataset acquisition with a WebGIS browser visualization, with the possibility to download and share live and historical data. Users don’t need any geospatial server, database, or IoT server configuration, because the automated process has packaged every component of the chain.

The focus of this research regards the construction of the framework that define the platform and its functions. All the functionalities that regards the acquisition, the storage, and the real-time fruition were tested, with an example of data analysis acquisition based on IoT technology.

The main contributions of this work can be summarized as follows:

- Development of a fully open-source and low-cost IoT–WebGIS integration framework;
- Implementation of containerized architecture enabling one-click deployment of all system components;
- Integration of real-time IoT data acquisition with interactive WebGIS visualization;
- Provision of a ready-to-use solution aimed at reducing technical barriers for non-expert users;

- Release of the full source code to ensure reproducibility and extensibility.

The next paragraph will show the last advances and examples on IoT and WebGIS integration. Then, the deployment of the platform will be described, analyzing every part of the framework. Then, the results will be shared, and, in the final paragraph the results will be discussed, showing also possible future implementations based on this solution.

2. Iot and WebGIS Integration: State of Art

As seen before, recent advances in IoT low-cost technology contributed to the diffusion of smart applications on several fields of research. IoT technology allows to collect data acquisition from remote sensors placed in any location. The addiction of the georeferenced information enhances the usability of IoT technology. In fact, sensors localization is necessary to integrate the monitoring with the contextual environment. In this field GIS integration offers new possibilities to IoT applications. Furthermore, the need to obtain a real time acquisition and to visualize it through remote connection lead necessary to integrate IoT systems with WebGIS. In fact, a WebGIS platform could integrate static geospatial information (provided by raster maps) and vectorial datasets (provided by remote databases) with real time dynamic information provided by IoT systems.

Last years are characterized by this kind of integration in many sectors, because IoT and WebGIS integration represents the basic smart solution for smart city evolution and infrastructure development [38]. For instance, some researchers developed a spatial information infrastructure platform (PAMS) in precision agriculture integrating WebGIS and IoT, to monitor and manage the production of the soils [39]. Even in the field of agriculture, this kind of integration has been experimented to evaluate the agricultural environment from on-farm sites to regional scales [40], or to create a decision support system based on a close-range sensor-based data management platform [41]. In fact, recent smart agriculture systems adopt IoT cloud-based strategies for real-time fire monitoring systems to mitigate agricultural risks [42].

IoT based technology, as affirmed before, has been recently used for online monitoring in smart building applications. For instance, IoT solution integrated in a WebGIS platform has been used for Indoor Air Quality (IAQ) assessment analyzing radon gas values in indoor building environments [43]. In the field of mobility and transportation, IoT system has been used for monitoring traffics and for highlighting critical transport infrastructures integrated with WebGIS real time emergency management on road network [44]. Some years ago, considering the emergency caused by natural disasters, it has been developed and IT-based volcanic disasters response system simulating volcanic activity adopting proper scenarios with IoT GIS integration [45]. Other applications considered the development of a WebGIS platform using Google Maps API (Application Programming Interface) services for data plotting of real-time flood monitoring, connecting in-situ instrumentation with the geospatial system [46].

The development of an integrated GIS and IoT system able to improve current systems and to create more reliable solutions represents today one of the main areas of research [47]. Last advances on IoT and WebGIS integration considered possible open-source solutions for developing monitoring geospatial platforms aimed at real-time environmental survey [48,49]. This kind of solution is very smart but at the same time needs the management of configuration files and require programming skills by users. To create an open-source WebGIS framework, it is necessary to integrate several open-source tools with PHP MapScript programming languages, RDMS (Relational Database Management System) database and geospatial server [50].

2.1. The Proposed Solution

With reference to the examples shown below, our work proposed an open-source solution that strongly simplify the users' operations on IoT real-time data analysis and visualization process. We developed a solution that encapsulates a platform able to real-time visualize the IoT acquisition in an integrated WebGIS/chart visualization on the web.

Considering our choice of employing only opensource solutions, we wanted to make the platform openly available as supplementary material to share the results of the work and, at the same time, give a ready-to-use solution for user experiences. Our aim was to offer users the possibility to freely give the proper expertise to implement new functions, optimize the existing ones or simply customize the existing configurations for personal purposes, without the necessity of rewriting the entire code.

3. Material and Methods: The Infrastructure of the Smart Monitoring Platform

3.1. The Overall Framework

The overall framework is made up by two main parts, the hardware, and the software part (Figure 1). The hardware part, or mobile part, includes the electronic board with a WiFi microcontroller unit, the sensors (in this version humidity, temperature, and GPS) and the Lithium battery. The software part is made up by a cloud server infrastructure responsible to take account for data storage, computation, and results presentation. All the hardware and software components are supplied in an open-source way. Specifically, the integration workflow begins with the hardware nodes collecting environmental parameters, which are then securely transmitted to the cloud infrastructure. Here, the internal software modules automatically interact to parse, store, and seamlessly push the data to the web interface.

The mobile part (hardware), based on IoT technology, is used as example for real time acquisition process. It is connected through WiFi connection to the Cloud Sever Infrastructure (software) part that allow the acquisition, the storage, and the real-time fruition of the dataset on a WebGIS/chart visualization (with the possibility to combine historical and real-time data acquisitions) in an encapsulated solution, avoiding any installation or configuration on client-side.

Every user with few programming skills can customize the code to let the system fit its own goals. This contribution is completed with a package of the system project attached as supplementary material.

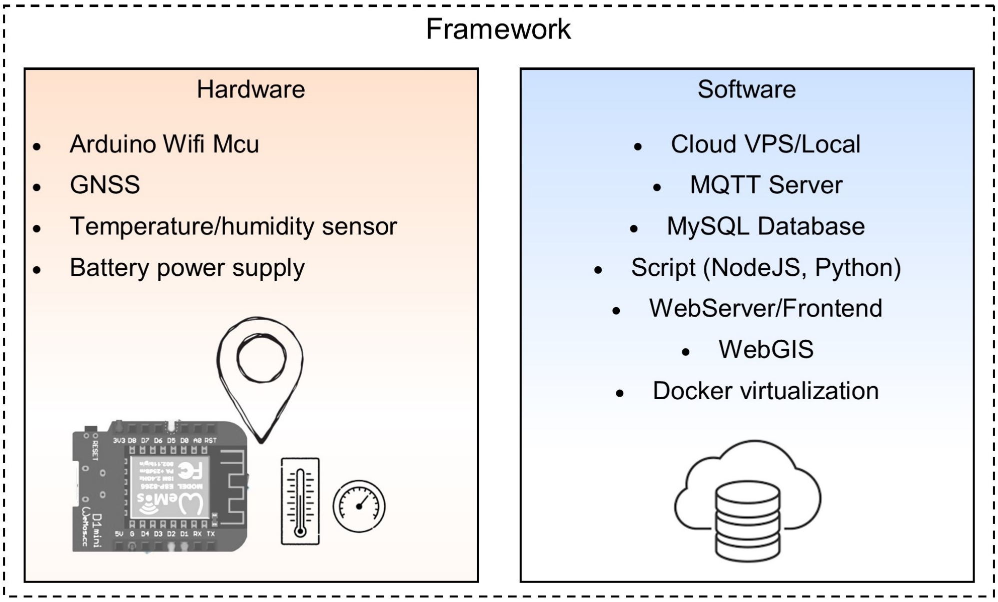


Figure 1. Architecture of the overall proposed framework, highlighting the interaction between the mobile hardware acquisition unit and the cloud-based software infrastructure.

3.2. The Mobile Part (Hardware)

With regards to the mobile part, or hardware part of the project, it is built around a WiFi Arduino commercial board, the Wemos D1 that, in turn, is built on the ESP8266 Mcu (Figure 2). The final board is then equipped with sensors to acquire in real-time environmental data like temperature, humidity, and position coordinates and with a battery for mobile operations (Figure 3). The acquisition hardware equipment is internet connected. Once defined the sample rate of the project, the data acquired is instantly available, unless network latency affects the application when operating in "Live data" mode (as will be described in the next paragraph). In detail, the hardware IoT sensors are:

- DHT22 sensor, on I2C digital bus, used to acquire temperature and humidity data.
- GY-NEO6MV2 GPS module used to acquire the georeferenced position.

Moreover, a custom firmware, built with the open-source Arduino IDE and Wiring C language, was developed to perform a cyclic sensor read operation and to send data on the cloud through the WiFi interface.

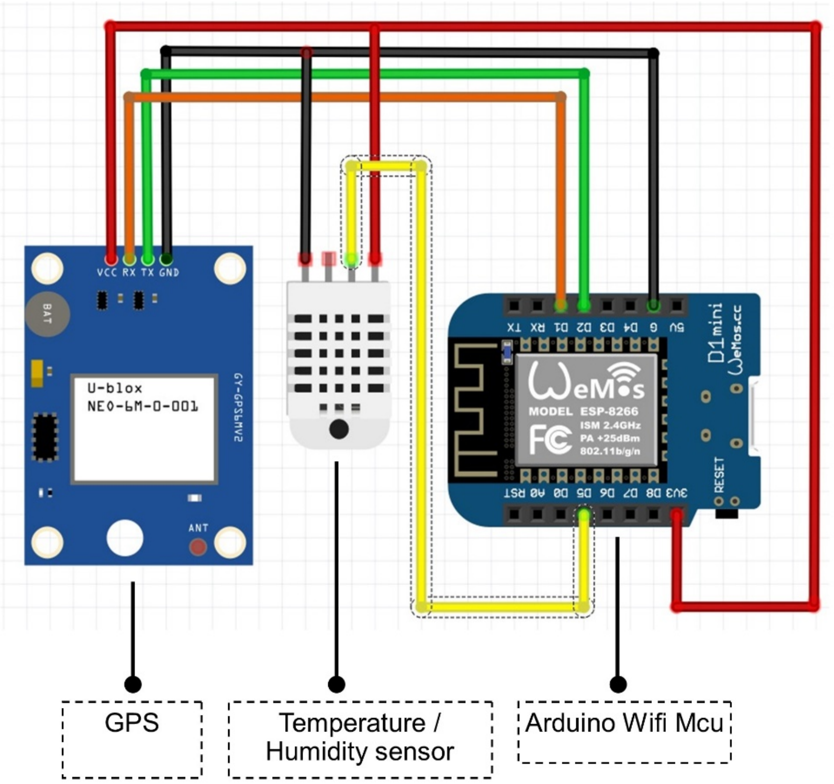


Figure 2. Detailed view of the mobile IoT hardware unit, showcasing the integration of the microcontroller board and the environmental sensors.

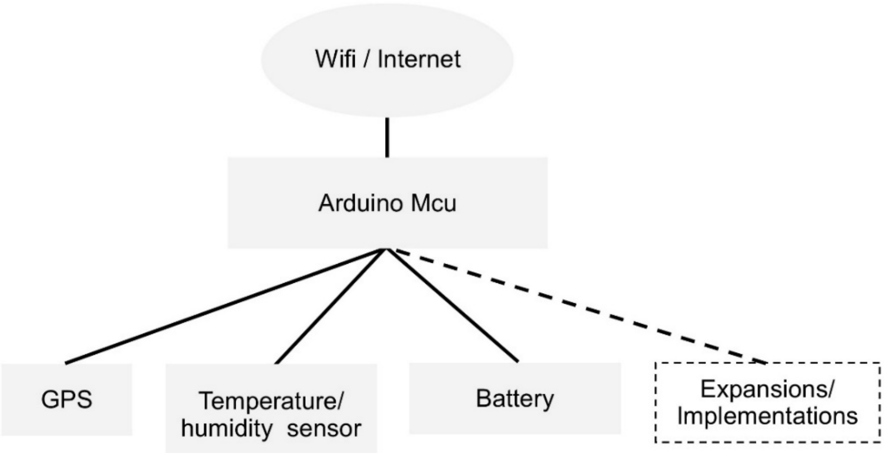


Figure 3. Schematic diagram illustrating the physical hardware connections and wiring components within the mobile data acquisition unit.

Several open-source libraries (for sensor and connectivity interface) located on the Arduino open-source community were applied to implement the ESP8266 firmware. The firmware follows a quite simple flow. At startup the board input/output, digital bus and Wifi peripherals are configured in the so called "setup" procedures. Then, in a cyclic loop, whose interval has been set to 5 s (it is fully customizable), new sensor data are read and sent to the cloud through the WiFi interface by using the MQTTs protocol. The security of the communication between the mobile devices and the cloud infrastructure is granted using the MQTTs protocol instead of the standard MQTT. At the same time the cryptography TLS standard ensures data consistency and avoids data sniffing.

The mobile board was developed to be portable and replicable, with the possibility to build several different cards to log data from many scenarios. Each card is identified by its own unique MAC address stored, by factory, in the ESP8266 ROM and it is recognized by the system at the first connection to the cloud. The MAC address is also the unique identifier that allows to distinguish a board from the other ones. The open source and open hardware paradigm represent another strategic peculiarity of the boards. These features allow anyone that possesses a small confidence with the Arduino environment to expand the board capabilities by adding sensors and functionalities by simply modify the supplied code. To work, the board needs only an internet WiFi connection that can be supplied with an Access Point or by a smartphone with tethering capabilities. The WiFi configuration (SSID and password) are stored on separated header file. In this way, each user can edit them prior to compile the Arduino code to match its own net configuration. A detailed tutorial on how to perform the initial configuration and first startup is supplied with the software package delivered as supplementary material.

Authors of this contribution are still available to consider other board/sensor implementation to allow the board to work on other application scenarios.

3.3. The Cloud Server Infrastructure (Software)

With regards to the cloud server infrastructure, it is made by several software components built on top of a docker virtualized environment to facilitate the distribution and deploy of the code (Figure 4). The aim of this contribution is to allow the user to build its own infrastructure to work with data logging on a local environment. To get the same infrastructure work on a production environment the user should cover some other implementation aspects that are out of the scope of this work. Authors are still available to cooperate with other researcher to implement the same framework in a production environment.

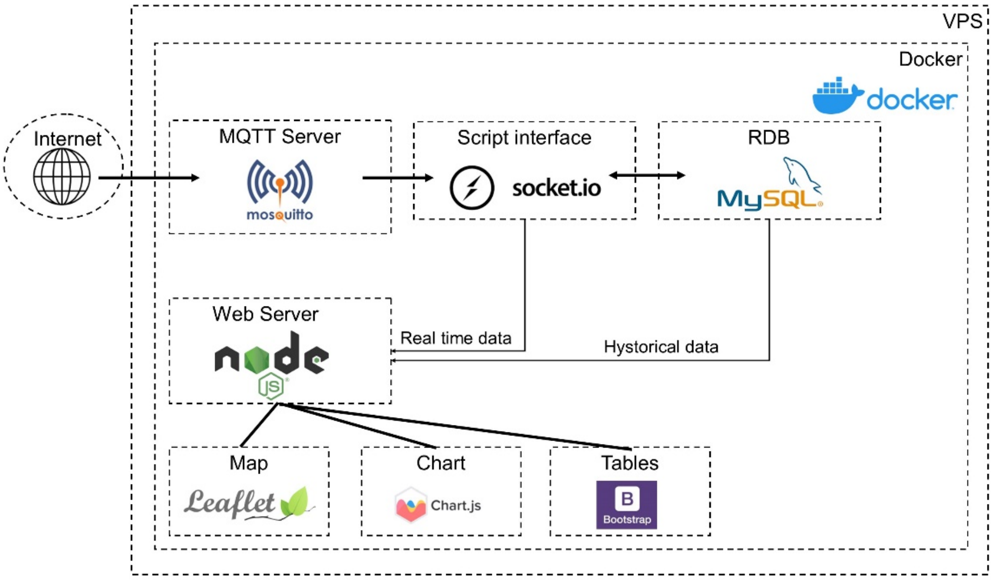


Figure 4. Operational flowchart of the cloud server infrastructure, detailing the automated interactions among the MQTT broker, database, and WebGIS services.

As early mentioned, the cloud infrastructure is built on top of a docker environment which is an open platform to develop, deliver and run applications in an isolated environment. In conjunction with the use of Docker, for this application, also the Docker-Compose utility has been used. Docker-Compose is a tool to define and share multi-container applications by defining them on a YAML file. The main advantage for the final user is the possibility of running the overall framework (composed by several components) with a single click operation. This containerization process ensures that all dependencies are pre-configured within isolated environments. The Docker-Compose utility acts as an orchestrator, establishing dedicated virtual networks that allow the MQTT broker, database, and web server to communicate flawlessly without manual network setup. This solution avoid singular installations and the configuration of each service (web server, database, api, mqtt server). A Docker-compose file is supplied with this contribution with the source files to let the user the building of the overall framework with the smallest possible number of operations. Furthermore, this container-based architecture promotes a high degree of modularity. Because each component operates within its own dedicated container, users can independently update, modify, or scale individual modules without disrupting the overall system workflow.

The framework is composed by these key services:

- MQTT broker/server
- NodeJS MQTT engine
- MySql Database
- Redis Server for queuing
- Express Webserver

As for the MQTT broker the Mosquitto server was chosen. Mosquitto is a ligthweight open-source message broker that implements the MQTT protocol, and it is suitable for IoT applications. The MQTT server acts as an endpoint to receive the data coming from the Arduino board and offers websocket connectivity to interface with other software modules like the MQTT engine.

The MQTT engine is a custom software module, written in NodeJs, that listen for the incoming data packets from the MQTT server and adds two main services. The first one is the log of the data on the MySql database. Since every received packet is identified by the board MAC address, the MQTT engine firstly check if the device is present on the database devices table (adding it if not present) and then stores the new data on the values table adding the MAC address and timestamp references. The second service added by the MQTT engine is to instantiate a websocket connectivity to send data back to the webserver for real-time visualization. This specific module interaction creates an automated pipeline bridging the data ingestion layer and the presentation layer, where raw sensor packets are immediately translated into actionable geographic information on the WebGIS interface.

MySql is an open-source relational database management system (RDBMS) that gives to the framework the capability to store data and device information to perform historical query on past data. In the following paragraph it will be shown that the front-end (the web server application) gives user the possibility to show real-time data and/or to query for historical data.

Redis is a simple open-source in-memory data store, used, in this application, to take account of asynchronous jobs computing. By managing these asynchronous tasks, the Redis module ensures a decoupled and highly responsive architecture, preventing the database from experiencing bottlenecks during high-frequency data ingestion from multiple IoT nodes. As for the other module no separate installation/configuration is needed. As will be shown in the implementation tutorial the user will be able to deploy all the entire framework by simply launch the docker-compose command.

The Web server, namely the front-end, is the main software component that allow the final user to show real-time and historical data coming from the sensor boards. It has been developed on the top of the NodeJs Express. It is an application framework that provides a robust set of features for web and mobile applications. The frontend was developed to improve the user-experience using Bootstrap components for the responsive graphics part in conjunction with some other open-source components. Data are represented on a map and on a chart and the user can download all the sensor data on an excel spreadsheet. About the map, data are represented through a clickable marker. When the user clicks on a marker on the map a popup containing sensor data, MAC address of the board and the timestamp are shown. The map was implemented using Leafleat, an open-source Javascript library to develop interactive geographic map for WebGIS application. As early mentioned, data are also represented on a chart developed using ChartJS that is a free, open-source Javascript library for data visualization.

The web application allows users to choose a device by the MAC address and then to select three way of data retrieve. In the next paragraph the functionalities of the developed platform and the possibilities offered by the package to users will be described in detail.

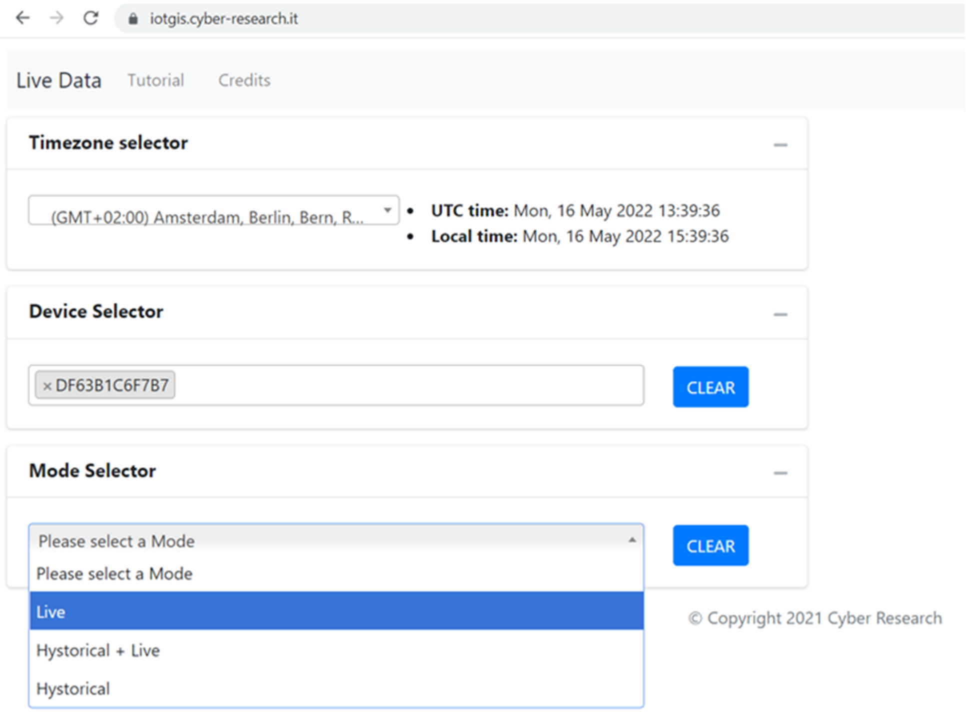
4. Results

The web application is freely available for user as supplementary material. It integrates IoT and WebGIS technologies incapsulated in a Docker structure, as seen before. The first activity that user can do is to select an IoT device, that represents the source of the dataset to analyze.

After the device selection (it can be also a multiple choice) user is prompted to select one of the three different available query options. In detail, three mode options are (Figure 5):

- Live data.
- Historical + live data.
- Historical data.

In "Live data" mode the system starts with an empty map connected with a chart (Figure 6). When new data are available the map and the chart are filled in real-time. With the real-time measures available on "Live data" mode, it will be possible to obtain, with further integration in the future, virtual real-time replicas of physical resources (objects, sites, infrastructures) in the same web platform. This is the base to create, in fact, the so-called "digital twin", that now represents a research topic of growing interest both in academic and industrial fields [51–54]. In "Historical + live data" mode the user can select a start date/time for database data retrieving. In this case, if the database contains data from the selected start date, the map and the chart will be pre-filled with the queried data, then, real-time data will be added as they arrive. The last mode, "Historical data" mode, has no real-time capabilities and allows the user to select a start and stop date/time, namely a range, to retrieve data from and to a specific date/time. No real-time data are added in this mode.



(GMT+02:00) Amsterdam, Berlin, Bern, R...

- **UTC time:** Mon, 16 May 2022 13:39:36
- **Local time:** Mon, 16 May 2022 15:39:36

× DF63B1C6F7B7

Please select a Mode

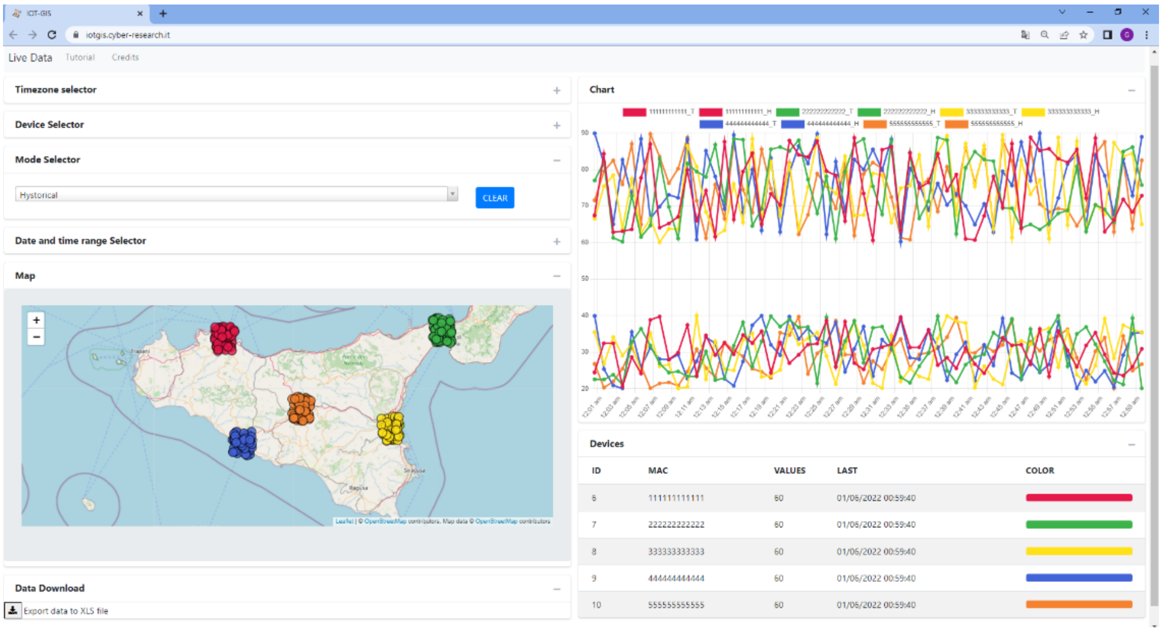
Please select a Mode

Live

Hystorical + Live

Hystorical

Figure 5. User interface of the web application's main window, displaying the configuration forms for selecting specific IoT devices and data query modes.



Mode Selector: Hystorical

ID	MAC	VALUES	LAST	COLOR
6	111111111111	60	01/06/2022 00:59:40	🟥
7	222222222222	60	01/06/2022 00:59:40	🟩
8	333333333333	60	01/06/2022 00:59:40	🟨
9	444444444444	60	01/06/2022 00:59:40	🟦
10	555555555555	60	01/06/2022 00:59:40	🟧

Figure 6. Real-time visualization of the platform showing (on the **left**) spatial distribution of IoT data on the WebGIS map and (on the **right**) temporal trends in the chart interface. The system dynamically updates both components as newly received data.

The web application presents also other pages that integrates the platform: the credits page and the tutorial page. In the credits page all the used software, modules, and libraries are listed and linked. In the tutorial page all the operations needed to build the sensor board and deploy the IoT framework are shown.

The platform is created to be employed by users for several integrations allowing two different possibilities of use: participatory and personal. The participatory use integrates the sensor dataset of users in the platform shared by the authors, where each user can visualize the datasets of everyone (like a participatory WebGIS). In the personal use, users can download the package provided by authors from the website and then properly configure the personal hardware in a local environment. Then user can activate the docker opensource platform following the tutorial available on the website. In this case, it is possible to build-up the system creating a personal local version of the platform.

5. Discussion

Considering the analysis of the acquired dataset, it is necessary to underline that the quality of IoT acquisition is not the focus of this research. As affirmed before, the IoT acquisition is not analyzed but is considered as example for testing the functionalities of the framework. The platform represents a ready-to-use application that every user can implement with his sensor network.

While specific quantitative evaluation metrics such as network latency, data throughput, and maximum scalability, depend highly on the end-user's local network conditions and hardware capabilities, the proposed containerized architecture is intrinsically designed to support scalable deployments. During the tests, the system smoothly handled the predefined 5-s cyclic data payload without noticeable bottlenecks. However, since the primary focus of this study is to provide an accessible, ready-to-use integration framework rather than a production-grade stress test, comprehensive quantitative evaluations under massive concurrent connections are deferred to future investigations.

The results obtained from the research confirm the effectiveness of the proposed framework in enabling real-time monitoring and visualization with minimal configuration effort. Despite its advantages, the proposed framework presents some limitations and potential real-world deployment challenges. The system performance depends on network stability, which may affect real-time data transmission in remote areas. Additionally, deploying the hardware in practical scenarios introduces physical and infrastructural challenges, such as ensuring a continuous power supply for the mobile IoT boards in off-grid locations, protecting sensors from harsh environmental conditions, and securing the network against potential cyber threats. Furthermore, the current implementation focuses on basic sensor integration and does not include advanced data analytics modules. Future work will address scalability, security aspects, and integration with machine learning-based analysis tools.

6. Conclusions and Open Scenarios

The developed framework allows to create a smart monitoring system integrated with a WebGIS real-time visualization. The encapsulation of all the modules offers users the possibility to directly run the application, starting automatically, without the necessity of installing or configuring any software. Only the IoT hardware configuration is required. All the modules are opensource. The developed solution avoids the need of programming skills by users, offering a simple starting procedure, avoiding, at the same time, any cost of property software integration.

In fact, electronic, telecommunication, informatics, and geomatics capabilities are normally requested to the user for implementing and configuring the entire toolchain (sensors dataset acquisition from different sources, storage of information into a relational database, online real time fruition through maps and charts) necessary to realize the entire IoT-GIS infrastructure [55,56]. Our solution solves this challenge giving user the opportunity to employ a ready to use framework that answers to these requirements (acquisition, storage, and fruition of the dataset), breaking down the barriers on the basis to reduce the effort in terms of man-hours (Table 1). This is the scientific contribution and the proposed novelty of our work. The provided software package offers the possibility to locally test the application. This solution makes the most of IoT and WebGIS integration offering a smart and opensource solution that could be employed by researchers, municipal technicians, planners, and software developers.

Table 1. Main features and innovative aspects of the work.

Main Features	
Acquisition	Possibility to integrate Arduino board with IoT sensor acquisitions and to send data through Wifi module
Storage	Possibility to store data and device information for making historical query on past data
Fruition	With "Live data", "Historical data" and "Historical + live data" options user can analyze the real time and historical data with an integrated map and chart visualization.
Innovative Aspects	
The encapsulation of the framework offers users a ready-to-use platform, without the need to install and configure every single module that compose the structure, and employing only open-source solutions	

In the future, further implementation could integrate other modules to real time online visualize and share monitoring results. The low-cost solution based on IoT-GIS integration represents an opportunity in the field of smart cities, where IoT integration is a crucial component [57]. In fact, imaging a large diffusion of IoT application scenarios for building of complex datasets and for implementation of big data-based algorithms, it's clear that, employing low-cost infrastructure and technology, it is possible to obtain a block of monitoring stations even more numerous than the network available using other monitoring solutions (considering the same economic expense).

The developed IoT platform represents a base for next steps where 2D and 3D web visualization and analysis could offer a smart solution in the field of digital twin systems.

To summarize, the key contributions of this work are threefold: (i) the design of a low-cost, open-source IoT-WebGIS framework; (ii) the development of a fully containerized, ready-to-use platform that removes the need for advanced programming or configuration skills; and (iii) the demonstration of a scalable architecture suitable for expanding urban and environmental monitoring networks. Outlining our future work, we plan to extend the platform's capabilities by integrating 3D WebGL visualizations to support digital twin applications, embedding Python-based machine learning modules for advanced real-time data analytics, and addressing the real-world deployment challenges previously discussed, such as cybersecurity enhancements and off-grid power management.

Recent experimentations carried out in the field of real time analysis and in digital representations of complex urban datasets [15,34,58,59] could be implemented with this kind of solution to integrate and visualize IoT real time analysis.

In fact, this solution could be employed in the future on several fields of research integrating further web visualization modules based on WebGL technology and real time operation modules based on Python language. Building energy monitoring, civil structure monitoring, hydrogeological risk analysis, vehicle traffic analysis, are only some of the possible applications where the developed framework could be used and integrated. Following this strategy, integrating information technology, geomatics and computer science, the potential of IoT technology will continue to grow, opening ever new fields of applications.

Author Contributions

Conceptualization, G.G., F.D., M.L.G. and A.S.; methodology, G.G., F.D., M.L.G. and A.S.; software, G.G., F.D., M.L.G. and A.S.; validation, G.G., F.D., M.L.G. and A.S.; formal analysis, G.G., F.D., M.L.G. and A.S.; investigation, G.G., F.D., M.L.G. and A.S.; resources, G.G., F.D., M.L.G. and A.S.; data curation, G.G., F.D., M.L.G. and A.S.; writing—original draft preparation, G.G., F.D., M.L.G. and A.S.; writing—review and editing, G.G., F.D., M.L.G. and A.S.; visualization, G.G., F.D., M.L.G. and A.S.; supervision, G.G., F.D., M.L.G. and A.S.;. All authors have read and agreed to the published version of the manuscript.

Funding

This research received no external funding.

Institutional Review Board Statement

Not applicable.

Informed Consent Statement

Not applicable.

Data Availability Statement

The data generated during the development and testing of the infrastructure are available from the corresponding author upon request.

Conflicts of Interest

The authors declare no conflict of interest. Given the role as a member of Editorial Board, Giovanni Garraffa had no involvement in the peer review of this paper and had no access to information regarding its peer-review process. Full responsibility for the editorial process of this paper was delegated to another editor of the journal.

Use of AI and AI-Assisted Technologies

During the preparation of this work, the authors used Gemini for grammatical and syntactic review. After using this tool, the authors reviewed and edited the content as needed and take full responsibility for the content of the published article.

References

1. Mell, P.; Grance, T. *The NIST Definition of Cloud Computing*; National Institute of Standards and Technology: Gaithersburg, MD, USA, 2011.

2. Biswas, A.R.; Giaffreda, R. IoT and cloud convergence: Opportunities and challenges. In Proceedings of the 2014 IEEE World Forum on Internet of Things (WF-IoT), Seoul, Korea, 6–8 March 2014. https://doi.org/10.1109/WF-IoT.2014.6803194.
3. Wan, S.; Zhao, K.; Lu, Z.; et al. A modularized IoT monitoring system with edge-computing for aquaponics. *Sensors* **2022**, *22*, 9260. https://doi.org/10.3390/s22239260.
4. Wu, T.-Y.; Kong, F.; Wang, L.; et al. Toward smart home authentication using PUF and edge-computing paradigm. *Sensors* **2022**, *22*, 9174. https://doi.org/10.3390/s22239174.
5. Wortmann, F.; Flüchter, K. Internet of things. *Bus. Inf. Syst. Eng.* **2015**, *57*, 221–224. https://doi.org/10.1007/s12599-015-0383-3.
6. Lee, I.; Lee, K. The internet of things (IoT): Applications, investments, and challenges for enterprises. *Bus. Horiz.* **2015**, *58*, 431–440. https://doi.org/10.1016/j.bushor.2015.03.008.
7. Phillips. Hue Personal Wireless Lighting. Available online: http://www2.meethue.com/ (accessed on 22 December 2025).
8. Nest. Nest Smoke Alarm. Available online: https://nest.com/ (accessed on 23 December 2025).
9. Dubey, Y.K.; Damke, S. Baby monitoring system using image processing and IoT. *Int. J. Eng. Adv. Technol.* **2019**, 8, 4961–4964. https://doi.org/10.35940/ijeat.F9254.088619.
10. Sivanathan, A.; Sherratt, D.; Gharakheili, H.H.; et al. Low-cost flow-based security solutions for smart-home IoT devices. In Proceedings of the 2016 IEEE International Conference on Advanced Networks and Telecommunications Systems (ANTS), Bangalore, India, 6–9 November 2016. https://doi.org/10.1109/ANTS.2016.7947781.
11. Aslam, M.; Khan Abbasi, M.A.; Khalid, T.; et al. Getting smarter about smart cities: Improving data security and privacy through compliance. *Sensors* **2022**, *22*, 9338. https://doi.org/10.3390/s22239338.
12. Riaz, S.; Latif, S.; Usman, S.M.; et al. Malware detection in internet of things (IoT) devices using deep learning. *Sensors* **2022**, *22*, 9305. https://doi.org/10.3390/s22239305.
13. Khan, N.; Khattak, K.S.; Ullah, S.; et al. A low-cost IoT based system for environmental monitoring. In Proceedings of the 2019 International Conference on Frontiers of Information Technology (FIT), Islamabad, Pakistan, 16–18 December 2019. https://doi.org/10.1109/FIT47737.2019.00041.
14. Nawandar, N.K.; Satpute, V.R. IoT based low cost and intelligent module for smart irrigation system. *Comput. Electron. Agric.* **2019**, *162*, 979–990. https://doi.org/10.1016/j.compag.2019.05.027.
15. Scianna, A.; Gaglio, G.F.; La Guardia, M. Structure monitoring with BIM and IoT: The case study of a bridge beam model. *ISPRS Int. J. Geo-Inf.* **2022**, *11*, 173. https://doi.org/10.3390/ijgi11030173.
16. Lee, G.; Cho, J.; Ham, S.; et al. A BIM- and sensor-based tower crane navigation system for blind lifts. *Autom. Constr.* **2012**, *26*, 1–10. https://doi.org/10.1016/j.autcon.2012.05.002.
17. Ding, L.Y.; Zhou, C.; Deng, Q.X.; et al. Real-time safety early warning system for cross passage construction in Yangtze Riverbed Metro Tunnel based on the internet of things. *Autom. Constr.* **2013**, *36*, 25–37. https://doi.org/10.1016/j.autcon.2013.08.017.
18. Kensek, K.M. Integración de sensores medioambientales con BIM: casos de estudio usando Arduino, Dynamo, y Revit API. *Inf. De La Construcción* **2014**, *66*, e044. https://doi.org/10.3989/ic.13.151.
19. Rabiee, R.; Karlsson, J. Multi-Bernoulli tracking approach for occupancy monitoring of smart buildings using low-resolution infrared sensor array. *Remote Sens.* **2021**, *13*, 3127. https://doi.org/10.3390/rs13163127.
20. Schuldt, C.; Shoushtari, H.; Hellweg, N.; et al. L5IN: Overview of an indoor navigation pilot project. *Remote Sens.* **2021**, *13*, 624. https://doi.org/10.3390/rs13040624.
21. Goodchild, M.F. Geographic information systems and science: Today and tomorrow. *Procedia Earth Planet. Sci.* **2009**, *1*, 1037–1043. https://doi.org/10.1016/j.proeps.2009.09.160.
22. Zhan, F.B. Three fastest shortest path algorithms on real road networks: Data structures and procedures. *J. Geogr. Inf. Decis. Anal.* **1997**, *1*, 69–82.
23. Neteler, M.; Bowman, M.H.; Landa, M.; et al. GRASS GIS: A multi-purpose open source GIS. *Environ. Model. Softw.* **2012**, *31*, 124–130. https://doi.org/10.1016/j.envsoft.2011.11.014.
24. Eddie, W.L.C.; Li, H.; Yu, L. A GIS approach to shopping mall location selection. *Build. Environ.* **2007**, *42*, 884–892. https://doi.org/10.1016/j.buildenv.2005.10.010.
25. Chainey, S.; Tompson, L.; Uhlig, S. The utility of hotspot mapping for predicting. *Secur. J.* **2008**, *21*, 4–28. https://doi.org/10.1057/palgrave.sj.8350066.
26. Yang, C.P.; Wong, D.W.; Yang, R.; et al. Performance-improving techniques in web-based GIS. *Int. J. Geogr. Inf. Sci.* **2005**, *19*, 319–342. https://doi.org/10.1080/13658810412331280202.
27. Su, Y.; Slottow, J.; Mozes, A. Distributing proprietary geographic data on the World Wide Web—UCLA GIS database and map server. *Comput. Geosci.* **2000**, *26*, 741–749. https://doi.org/10.1016/S0098-3004(99)00130-2.

28. Karnatak, H.C.; Shukla, R.; Sharma, V.K.; et al. Spatial mashup technology and real time data integration in geo-web application using open source GIS—A case study for disaster management. *Geocarto Int.* **2012**, *27*, 499–514. https://doi.org/10.1080/10106049.2011.650651.
29. Agrawal, S.; Gupta, R.D. Web GIS and its architecture: A review. *Arab. J. Geosci.* **2017**, *10*, 518. https://doi.org/10.1007/s12517-017-3296-2.
30. Alonge, F.; Cusumano, P.; D’Ippolito, F.; et al. Localization in structured environments with UWB devices without acceleration measurements, and velocity estimation using a Kalman–Bucy filter. *Sensors* **2022**, *22*, 6308. https://doi.org/10.3390/s22166308.
31. Garraffa, G.; Sferlazza, A.; D’Ippolito, F.; et al. Localization based on parallel robots kinematics as an alternative to trilateration. *IEEE Trans. Ind. Electron.* **2022**, *69*, 999–1010. https://doi.org/10.1109/TIE.2021.3050354.
32. Vacca, G.; Fiorino, D.R.; Pili, D. A spatial information system (SIS) for the architectural and cultural heritage of Sardinia (Italy). *ISPRS Int. J. Geo-Inf.* **2018**, *7*, 49. https://doi.org/10.3390/ijgi7020049.
33. Scianna, A.; Gaglio, G.F.; La Guardia, M.; et al. Development of a virtual CH path on WEB: Integration of a GIS, VR, and other multimedia data. In *Digital Heritage. Progress in Cultural Heritage: Documentation, Preservation, and Protection*; Lecture Notes in Computer Science; Springer: Cham, Switzerland, 2021; Volume 12642, pp. 178–189.
34. La Guardia, M.; D’Ippolito, F.; Cellura, M. Construction of a WebGIS tool based on a GIS semiautomated processing for the localization of P2G plants in Sicily (Italy). *ISPRS Int. J. Geo-Inf.* **2021**, *10*, 671. https://doi.org/10.3390/ijgi10100671.
35. Kieu, Q.; Nguyen, T.; Hoang, A. GIS and remote sensing: A review of applications to the study of the Covid-19 pandemic. *Geogr. Environ. Sustain.* **2021**, *14*, 117–124. https://doi.org/10.24057/2071-9388-2021-054.
36. Pisani, D.; Pazienza, P.; Perrino, E.V.; et al. The economic valuation of ecosystem services of biodiversity components in protected areas: A review for a framework of analysis for the Gargano National Park. *Sustainability* **2021**, *13*, 11726. https://doi.org/10.3390/su132111726.
37. Toro Herrera, J.F.; Carrion, D.; Brovelli, M.A. A collaborative platform for water quality monitoring: Simile WebGIS. *ISPRS Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **2021**, *XLIII-B4-2021*, 201–207. https://doi.org/10.5194/isprs-archives-XLIII-B4-2021-201-2021.
38. Miloudi, L.; Rezeg, K. Leveraging the power of integrated solutions of IoT and GIS. In Proceedings of the 2018 3rd International Conference on Pattern Analysis and Intelligent Systems (PAIS), Tebessa, Algeria, 24–25 October 2018. https://doi.org/10.1109/PAIS.2018.8598500.
39. Ye, J.; Chen, B.; Liu, Q.; et al. A precision agriculture management system based on internet of things and WebGIS.In Proceedings of the 2013 21st International Conference on Geoinformatics, Kaifeng, China, 20–22 June 2013. https://doi.org/10.1109/Geoinformatics.2013.6626173.
40. Du, K.; Chu, J.; Sun, Z.; et al. Design and implementation of monitoring system for agricultural environment based on WebGIS with internet of things. *Trans. Chin. Soc. Agric. Eng.* **2016**, *32*, 171–178. https://doi.org/10.11975/j.issn.1002-6819.2016.04.024.
41. Adão, T.; Soares, A.; Padua, L.; et al. Mysense-Webgis: A graphical map layering-based decision support tool for agriculture. In Proceedings of the IGARSS 2020—2020 IEEE International Geoscience and Remote Sensing Symposium, Waikoloa, HI, USA, 26 September–2 October 2020. https://doi.org/10.1109/IGARSS39084.2020.9323885.
42. Abdennabi, M.; Azar, S.; Haris, M.K.; et al. Holistic IoT and cloud-based telemetry architecture for proactive fire monitoring in smart agriculture. *Sci. Rep.* **2026**, *16*, 8669. https://doi.org/10.1038/s41598-026-43538-0.
43. Lopes, S.I.; Moreira, P.M.; Cruz, A.M.; et al. RnMonitor: A WebGIS-based platform for expedite in situ deployment of IoT edge devices and effective radon risk management. In Proceedings of the 2019 IEEE International Smart Cities Conference (ISC2), Casablanca, Morocco, 14–17 October 2019. https://doi.org/10.1109/ISC246665.2019.9071789.
44. Du, P.; Chen, J.; Sun, Z.; et al. Design of an IoT-GIS emergency management system for public road transport networks. In Proceedings of the 1st ACM SIGSPATIAL International Workshop on the Use of GIS in Emergency Management, Bellevue, WA, USA, 3–6 November 2015. https://doi.org/10.1145/2835596.2835611.
45. Kim, T.; Youn, J.; Kim, H.; et al. Development of a IT-based volcanic disasters response system. In Proceedings of the Geospatial World Forum, Geneva, Switzerland, 5–9 May 2014.
46. Kolios, S.; Karveli, P.; Stylios, C. Web-based geographical information system for real-time flood monitoring of the river arachthos in Epirus region Greece. In Proceedings of the 10th International Conference on Advances in Satellite and Space Communications (SPACOMM 2018), Athens, Greece, 22–26 April 2018.
47. Safari Bazargani, J.; Sadeghi-Niaraki, A.; Choi, S.-M. A survey of GIS and IoT integration: Applications and architecture. *Appl. Sci.* **2021**, *11*, 10365. https://doi.org/10.3390/app112110365.
48. Pachoulas, G.; Petsios, S.; Spyrou, E.D.; et al. An adaptable Web GIS platform for monitoring port air quality. In Proceedings of the 2021 29th Mediterranean Conference on Control and Automation (MED), Puglia, Italy, 22–25 June 2021. https://doi.org/10.1109/MED51440.2021.9480193.

49. Xu, M. Enhancing GIS models for sustainable development in human settlements using intelligent IoT infrastructure and human-machine interaction. *Results Eng.* **2025**, *27*, 106713. https://doi.org/10.1016/j.rineng.2025.106713.
50. Sejati, A.W.; Buchori, I.; Rudiarto, I.; et al. Open-source Web GIS framework in monitoring urban land use planning: Participatory solutions for developing countries. *J. Urban Reg. Anal.* **2020**, *12*, 19–33. https://doi.org/10.37043/JURA.2020.12.1.2.
51. Yuchen, J.; Shen, Y.; Kuan, L.; et al. Industrial applications of digital twins. *Philos. Trans. R. Soc. A* **2021**, *379*, 20200360. https://doi.org/10.1098/rsta.2020.0360.
52. Pylianidis, C.; Osinga, S.; Athanasiadis, I.N. Introducing digital twins to agriculture. *Comput. Electron. Agric.* **2021**, *184*, 105942. https://doi.org/10.1016/j.compag.2020.105942.
53. Ketzler, B.; Naserentin, V.; Latino, F.; et al. Digital twins for cities: A state of the art review. *Built Environ.* **2020**, *46*, 547–573. https://doi.org/10.2148/benv.46.4.547.
54. Ying, Y.; Koeva, M.N.; Kuffer, M.; et al. Urban 3D modelling methods: A state-of-the-art review. *ISPRS Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **2020**, *XLIII-B4-2020*, 699–706. https://doi.org/10.5194/isprs-archives-XLIII-B4-2020-699-2020.
55. Cao, H.; Wachowicz, M. The design of an IoT-GIS platform for performing automated analytical tasks. *Comput. Environ. Urban Syst.* **2019**, *74*, 23–40. https://doi.org/10.1016/j.compenvurbsys.2018.11.004.
56. Kychkin, A.; Gorshkov, O.; Kukarkin, M. IoT-platform for ML-based industrial air emissions data processing. In Proceedings of the 2022 International Conference on Industrial Engineering, Applications and Manufacturing (ICIEAM), Sochi, Russian, 16–20 May 2022. https://doi.org/10.1109/ICIEAM54945.2022.9787190.
57. Whaiduzzaman, M.; Barros, A.; Chanda, M.; et al. A review of emerging technologies for IoT-based smart cities. *Sensors* **2022**, *22*, 9271. https://doi.org/10.3390/s22239271.
58. La Guardia, M.; D’Ippolito, F.; Cellura, M. A GIS-based optimization model finalized to the localization of new power-to-gas plants: The case study of Sicily (Italy). *Renew. Energy* **2022**, *197*, 828–835. https://doi.org/10.1016/j.renene.2022.07.120.
59. La Guardia, M.; Koeva, M.; D’Ippolito, F.; et al. 3D data integration for web based open source WebGL interactive visualisation. *ISPRS Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **2022**, *XLVIII-4/W4-2022*, 89–94. https://doi.org/10.5194/isprs-archives-XLVIII-4-W4-2022-89-2022.