

# WoT-Server: Combining the Web-Of-Things and SensorThings standards to build an IoT platform

Joel Aumedes<sup>1</sup>, Xavier Celades<sup>1</sup>, Josep Pijuan<sup>1</sup>, and Robert Sanfeliu<sup>1</sup>

<sup>1</sup> Centre Tecnològic Eurecat, Catalonia, Spain

DOI: [10.xxxxxx/draft](https://doi.org/10.xxxxxx/draft)

## Software

- [Review](#)
- [Repository](#)
- [Archive](#)

Editor: [Daniel S. Katz](#)

## Reviewers:

- [@zeyus](#)
- [@Ahanaf-Aziz](#)
- [@shahchiragh](#)

Submitted: 04 November 2025

Published: unpublished

## License

Authors of papers retain copyright and release the work under a Creative Commons Attribution 4.0 International License ([CC BY 4.0](#)).

## Summary

This document presents WoT-Server, a JavaScript IoT server that combines the abstraction of the Web-Of-Things Standard (W3C, 2024) and the data model of the SensorThings Standard (Liang et al., 2024) to provide a simple interface for storing and requesting data in time series format. WoT-Server provides simplified access to the different instances required for organised storage of IoT time series data via easily accessible HTTP endpoints, which can also be adapted for access through different protocols such as CoAP and MQTT. This paper describes the concepts and techniques used in the development of this tool. It also demonstrates its use in a water management/irrigation application, presenting the original use case developed in the ACQUAOUNT project (Acquaount, 2024), a project funded by the European Union with the objective of providing tools for efficient use of water.

## Statement Of Need

WoT-Server was initially created as a centralised data management solution for the ACQUAOUNT Project. One of the objectives of the ACQUAOUNT Project is to provide smart irrigation recommendations to farmers using a water balance model. However, the challenges it addresses are common across many research domains that rely on IoT data, such as agriculture, environmental monitoring, hydrology, and smart cities. WoT-Server acts as a centralized, interoperable, and easily deployable API to allow easy access to sensor data for further model implementation and simplifies the upload, storage and retrieval procedures of measurements from sensors in the field. WoT-Server is capable of both receiving data via an HTTP endpoint or fetching the data itself from other API services, a feature useful for centralising data from multiple platforms.

The aim of WoT-Server is to provide a platform with minimal setup, aligned to standards, and with a centralized API that enables easy management of time series data for research-grade IoT data management, letting users focus on the tasks specific to their use case. With WoT-Server, developers creating IoT applications gain access to a platform where they can have their data storage up and running in minutes and use it to create AI models. By providing WoT-Server as a reference architecture and open-source implementation combining Web-Of-Things and SensorThings, into a lightweight data platform, this service is provided to scientific teams that require structured time-series storage but don't wish to commit to enterprise-scale IoT ecosystems.

## State Of The Art

Many implementations of the Web-Of-Things standard are available in open source format. However, these existing Web of Things implementations such as node-wot (Eclipse Thingweb, 2024) (the one used in this project) and Mozilla WebThings (Francis, 2019) primarily focus on

device interaction and scripting, and do not natively provide persistent, standardized storage mechanisms comparable to SensorThings-compliant servers. Other existing IoT platforms partially address this need but not in a single solution. Standard-compliant implementations such as FROST-Server (Fraunhofer Institute, 2024) provide robust support for the OGC SensorThings API but require substantial configuration and infrastructure knowledge, and do not provide both the simplicity of WoT and the SensorThings data model to store the data. Full-stack IoT platforms like ThingsBoard (ThingsBoard, 2026) and FIWARE (FIWARE Foundation, 2026) are overly complex solutions for the use cases that WoT-Server aims to service, such as research prototypes and short-term scientific deployments. To the authors' knowledge, WoT-Server is among the first open-source platforms to combine the Web of Things interaction model with a SensorThings-compliant data model and persistent storage.

## Design And Implementation

According to project requirements, the platform must consist of a WoT-compliant API implementation that manages the heterogeneity of the different sources of data found in the project, such as different types of in-situ sensors or different weather stations, by providing a common interface for data exchange. This solution addresses one of the main problems of IoT, the platform fragmentation, and defines how a sensor should communicate its data to the platform.

To provide all the required functionalities of the project in a clean and efficient manner, the platform is divided into multiple services, all of them virtualized and deployed using Docker (Docker, 2026). A basic diagram of the services and their connections can be seen in Figure 1.

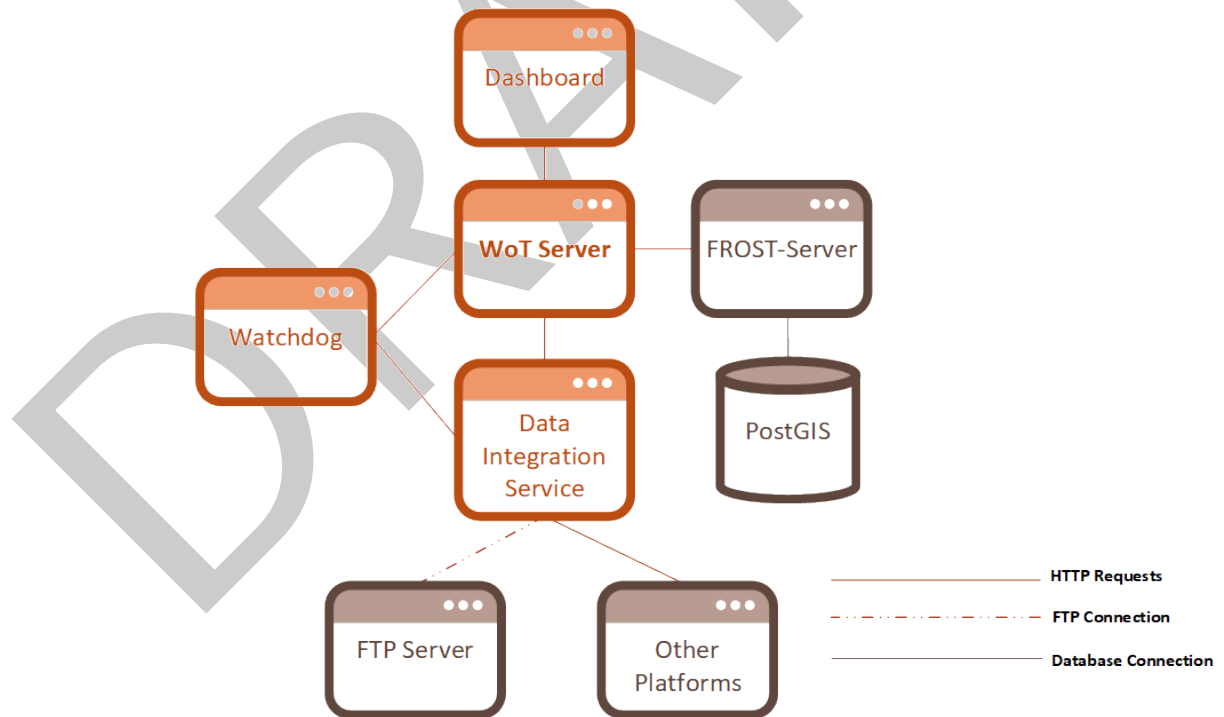


Figure 1: Diagram of the WoT platform and all related services.

The main service is the WoT server, developed using node-wot. This service provides all the controllers that manage the requests sent by the sensors and the gateways. To provide persistent storage of the data, a PostGIS database is used (PostGIS, 2026). However, to simplify the development process and to maintain consistency with the standards, FROST-Server, a SensorThings compliant server, is used to access the database. Then, the data

66 integration service imports data from external platforms into the WoT platform, either by  
67 directly requesting it to the external platform or by fetching it from an FTP server where it  
68 has been uploaded. In addition, the watchdog service monitors if the server and the data  
69 integration service are online and have not crashed and gone offline. If this happens, the  
70 watchdog sends an email to the server administrator to restart the service as soon as possible  
71 to avoid missing data being sent from the sensors. Finally, the monitoring platform provides a  
72 graphical way to check if data is correctly being uploaded to the server.

73 Inside the WoT server, Things are defined as the closest possible match to a physical location  
74 (Kaeibisch et al., 2024). Each physical location can be accessed through interaction affordances,  
75 described in the README.md file. These interaction affordances function as a bridge between  
76 the user and the data stored in the SensorThings database. Since the WoT model and the  
77 SensorThings model share the Thing as a common instance, they can collaborate to create a  
78 model that is complete and detailed but abstract and simple to interact with at the same time,  
79 as seen in Figure 2.

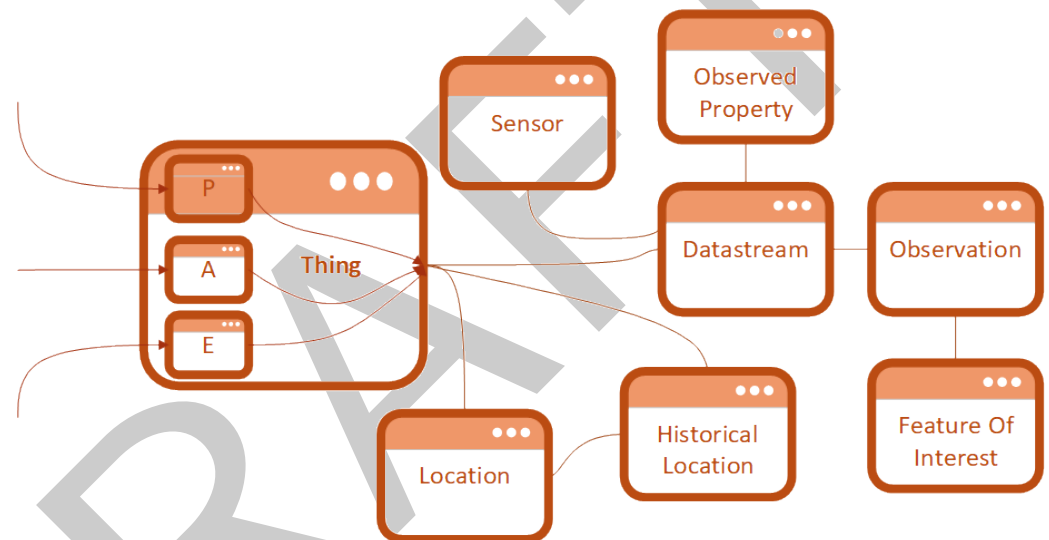


Figure 2: Diagram of WoT being used as the interface to access the SensorThings data model.

## 80 Use Case

81 WoT-Server is one of the core tools of the ACQUAOUNT Project, a project funded by the  
82 European Union via the PRIMA program and supported by Horizon 2020, with the objective  
83 of developing a variety of innovative tools focused on making the use of water at farm level  
84 and basin level as efficient as possible. These technologies are developed and evaluated in  
85 four areas of the Mediterranean area (Sardinia, Italy; Bekaa Valley, Lebanon; Al-Jifārah plain,  
86 Tunisia; Central Jordan Valley, Jordan), seen in the map in Figure 3.



Figure 3: Map of the Pilot Locations of the ACQUAOUNT Project.

The ACQUAOUNT Project has many lines of research and development, such as sensor installation, LoRaWAN networks, or basin modelling. WoT-Server acts as the link between the sensors and platforms and the models, storing data and providing functionalities in a standardized way. WoT-Server stores the data received from the data sources and provides it to services that calculate recommendations based on this data. Once the services finish the calculations, the results are also stored in WoT-Server to be consumed by the dashboard and frontend services to be displayed to the final users in a graphical and understandable way.

## Acknowledgements

Financial support has been provided by the European Union via the PRIMA programme and supported by the Horizon 2020 financial instrument.

## References

- Acquaount. (2024, January 9). *ACQUAOUNT*. <https://www.acquaount.eu/#>
- Docker. (2026, January 16). *Docker presentation website*. <https://www.docker.com/>
- Eclipse Thingweb. (2024, May 14). *Eclipse thingweb GitHub repository*. <https://github.com/eclipse-thingweb/node-wot>
- FIWARE Foundation. (2026, February 23). *FIWARE website*. <https://www.fiware.org/>
- Francis, B. T. (2019, April 24). *Mozilla WebThings: An open source implementation of the web of things*. <https://w3c.github.io/wot/workshop/ws2/Papers/29%20-%20Francis,%20Mozilla%20-%20Mozilla%20Web%20Things.pdf>
- Fraunhofer Institute. (2024, May 28). *FROST-server documentation*. <https://fraunhoferiosb.github.io/FROST-Server/>
- Kaebisch, S., McCool, M., Korkan, E., Kamiya, T., Charpenay, V., & Kovatsch, M. (2024, March 19). *Web of things (WoT) thing description 1.1*. W3C. <https://www.w3.org/TR/wot-thing-description/#abstract>
- Liang, S., Khalafbeigi, T., Schaaf, H. van der, Miles, B., Schleidt, K., Grellet, S., & Alzona, M. (2024, March 19). *OGC SensorThings API part 1: Sensing version 1.1*. Open Geospatial Consortium (OGC). <https://docs.ogc.org/is/18-088/18-088.html>
- PostGIS. (2026, January 16). *PostGIS presentation website*. <https://postgis.net/>

- 115 ThingsBoard. (2026, February 23). *ThingsBoard website*. <https://thingsboard.io/>
- 116 W3C. (2024, March 19). *Documentation - web of things (WoT)*. [https://www.w3.org/WoT/](https://www.w3.org/WoT/documentation/)
- 117 [documentation/](https://www.w3.org/WoT/documentation/)

DRAFT