Sebastian Stadler, Eduardo Romero Borrero, Johannes Zauner, Christian Hanshans*

Development and Implementation of an OpenSource IoT Platform, Network and Data Warehouse for Privacy-Compliant Applications in Research and Industry

A fast-growing, multi-user, hardware and cloud agnostic IoT data solution. Easy to deploy, privacy compliant, cost-effective, and OpenSource.

Abstract: In the scientific field, data acquisition using commercial or self-developed sensors is a necessity for many research activities. Data security and data privacy are important requirements in all types of IoT applications, especially in the medical context. *IoTree42* is a powerful, OpenSource platform that closes the gap between cost-efficiency, the ease of use and digital sovereignty. The flexible and user-friendly design makes the platform ideal for the originally conceived research context. Nevertheless, it can be used for smaller budget-oriented setups and industrial applications with thousands of sensors and actuators likewise. The platform was developed with security in mind by minimizing interfaces and the use of stable software components and transport protocols. *IoTree42* utilizes lightweight OpenSource software cast into a flexible backend, allowing the deployment not only on full-fledged servers but also on single board computers like the *Raspberry Pi*. Data is transmitted with low overhead via *MQTT*, a robust protocol optimized for machine to machine communication. The transmission technology between all components is freely selectable. Besides data acquisition, *IoTree42* enables control of actuators and automation through interfaces with visual programming tools. This allows for remote interventions without requiring prior programming experience. The incoming data can conveniently be visualized in dashboards or easily be exported for further analysis. The network consists of a central server and satellites (gateways) arranged in a star topology. Sensors and actuators connect to the gateway. The initial setup of the server and the gateways is automated, well

documented, and can be done within minutes. Thanks to the increasing usage and the contribution of the community, the pool of code examples for sensors is growing and lowers the entry hurdle for users without programming background. *IoTree42* is a fast-growing, multi-user, hardware and cloud agnostic, easy to deploy, privacy compliant and competitive solution compared to commercial IoT alternatives.

Keywords: *IoT Platform, MQTT, OpenSource, Internet of Things, data privacy, digital sovereignty, pseudonymization, connected sensors, sensor application, scientific automation, Arduino, ESP, Raspberry Pi, DIY, laboratory automation, Linux, visual programming*

<https://doi.org/10.1515/cdbme-2021-2129>

1 Introduction

In the scientific field, data acquisition and secure data handling of commercial and self-developed sensors is a necessity for many research activities. Proprietary systems can handle these tasks but are expensive, take sovereignty of the data, and are hard to adapt to non-commercial hardware. Individually programmed solutions seem to be a cheap alternative, but they require time and expertise to develop, test, and implement. Additionally, they are prone to failure leading to data loss. OpenSource systems offer a way out of these issues [1]–[3]. The *IoTree42* platform was developed out of the needs of our researchers in field studies with wearable devices and in the research facility, e.g., with experimental setups and equipment monitoring. How to reliably collect data from multiple sensors and sensor types into a stable database? How to query this data for easy visualization, automation, and sophisticated external analysis? While these problems are peripheral to the main research question, they take much of a researcher's time to solve and troubleshoot. We conceived *IoTree42* as a flexible

*Prof. Dr. med. Dipl.-Ing. Christian Hanshans: Hochschule München University of Applied Sciences, Lothstraße 34, Munich, Germany, christian.hanshans@hm.edu

Sebastian Stadler B.Eng., Eduardo Romero Borrero B.Eng., Johannes Zauner M.Sc.: Hochschule München University of Applied Sciences, Munich, Germany

and cost-effective solution, open for any researcher or interested parties. Through the OpenSource community, future research can profit from an ever-growing repository of code samples and templates for sensors, platform extensions and automation scenarios.

2 Materials and Methods

2.1.1 Network Architecture

Figure 1: Schematic of a single network branch with one gateway. The connection technology between gateway and sensors or actuators is application dependent. The HTTP/s client can be a user (web browser) or 3rd party software.

The *IoTree42* network is structured in a star topology, where communication nodes (gateways) are connected to a central server (see Figure 1). Gateways are placed in a local network with internet access. The server provides a suitable installation package for the gateway setup on *Linux* based operating systems. Once the gateway is installed, a secure connection to the server is automatically established and gateways can be controlled through the server.

Complex network structures with sub-networks and load distribution are just as feasible as a locally separated operation with multiple gateways. The encrypted data between the gateway and the server is transmitted via *MQTT* v5 (Message Queuing Telemetry Transport), which has been known for decades as a reliable and low-overhead protocol for communication [4], [5]. The transmission technology between the gateway and devices such as sensors and actuators is freely selectable and only limited by the physical inputs and outputs of the components. It can range from serial interfaces like *I²C*, *Can-Bus*, or *rs485*, to radio standards like *Bluetooth*, *Wi-Fi*, *LoRa*, or *ZigBee*. Many of them are used today in healthcare applications [2]. Sensors and actuators can be commercial or a custom design. Any data source can be integrated into the database, from a simple temperature sensor to third party APIs. The gateway acts as a bridge between the sensors/actuators and the platform. MQTT messages are sent with a so-called

topic. Topics, in MQTT terminology, are the way to exchange information in a hierarchical structure that can be compared to a path structure known from computer file systems. Storing (or receiving) temperature data from a temperature sensor will result in a MQTT like "universe/DHT22/temp_sensor". The prefix "universe" indicates that the message will be forwarded to the platform and "temp_sensor" is a container for the temperature data received from a DHT22 temperature and humidity sensor connected to an ESP32 microcontroller wired to a ESP32 microcontroller that communicates with the Raspberry Pi via WiFi and acts as the gateway. The topic structure is user selectable and forms an interactive tree structure to assign data. This tree structure was the namesake for the project title "IoTree42".

2.1.2 Software Architecture

Software & Functions: *IoTree42* utilizes a suite of lean OpenSource software for efficiency and security in conjunction with a flexible and powerful backend for controlling and monitoring data streams. The message flow between the network participants is managed via a *Mosquitto MQTT* broker. Tags and rules can be defined to filter messages before storing them in an *InfluxDB* time series database. This is especially useful when a sensor's data string must be modified before being interpretable. For each user, there is a separate, password-protected database that cannot be accessed directly. The frontend of the multi-user platform was arranged with the *Django* framework and completed with the *Grafana* visualization tool. The latter allows users to create dashboards as well as explore, filter, and export stored data as *CSV* (Comma-Separated Values) files or perform simple calculations. Charts, bar graphs, gauges, text panels, etc., are available and can be created without prior programming knowledge. All web applications are hosted behind *NGINX*, a reverse proxy and web server that provides an additional layer of security along with load-balancing capabilities. To minimize the hurdle of integration into an existing infrastructure and to simplify further processing of the collected data, an application programming interface (*REST-API* [6]) was implemented. This allows the stored data to be queried by programs such as *Python*, *MATLAB*, *R*, etc.

Pseudonymization: Data, that is associated with human subjects or patients can even be pseudonymized at the gateway level via a preconfigured *Python* script. The metadata will be hashed with a *SHA256* algorithm and is replaced before being passed on to the server [7]

Installation & requirements: The *IoTree42* platform requires little operational power and works on dedicated or virtualized

servers with a common *Linux* distribution, but also on a small single board computers like the *Raspberry Pi*. The installation package from the *GitHub* repository contains a setup with several security-level deployment options [8].

On the device side, the user defines an MQTT topic under which the data will be published. If a protocol other than MQTT is used between the devices and gateways, a conversion might be required.

3 Results and Discussion

An exemplary deployment scenario: The *IoTree42* platform is currently being implemented as a control and monitoring system for our research group within several laboratories in the Munich University of Applied Sciences (MUAS). The following example is a brief overview of this scenario, highlighting the main use cases and devices.

The IoT platform is in use at the laboratories of microbiology, tissue engineering, and biomedical engineering. All laboratories monitor experimental data, environmental parameters, as well as the usage of shared resources. These laboratories are located at different buildings and floors of the MUAS and are thus physically separated. They constantly work together and share resources.

IoTree42 reliably assumes tasks for these laboratories like data acquisition, big data management, interconnectivity, and automation. These measures reduce the workload on the researchers, solve some of the daily challenges, and reduce errors. Microbiological laboratories require seamless monitoring of fridges, freezers, and incubators, all of which contain sensitive cell cultures and biological samples. The labs of tissue engineering are currently carrying out experiments investigating light induced effects on living cells. The experimental setup needs to be controlled and monitored.

IoTree42 is deployed on a virtualized server accessible from the internet. Each lab has an independent gateway installed on a *Raspberry PiB3+*. Gateways, sensors, and actuators are connected via *Wi-Fi*. The installed sensors and actuators are a mixture of commercially available devices, custom built sensors based on *ESP32/ESP8266* or other *MQTT* capable microcontrollers, or even more complex machines, such as 3-D printers or industrial equipment. For example, cell irradiation in tissue engineering is controlled through *ESP32* controlled relays and automated by routines easily set up through the API. The resulting thermal influence is measured by *PT1000* sensors embedded in the apparatus of the experiment. Once the devices and the routines for automation are set up, the system requires little maintenance and data acquisition runs seamlessly.

The microbiological laboratories benefit from continuous monitoring of sensible samples, recognizing and even predicting issues, e.g., when the N_2 tanks run low. The tissue engineering labs save valuable research time by handling experimental data acquisition and automated setup control, which also prevents human error. This automation is executed through *NodeRed*. Users create routines with visual programming (like *LabView*) to send, read, or modify incoming or outgoing data streams in real time. In the biomedical lab, the IoT platform monitors the use of the machines and helps the staff to efficiently allocate print jobs. Machine deployment is thus handled efficiently with little downtimes.

Data can be queried from the database through a web browser. Measurements can be grouped by time, sensor, and gateway. Data sources are easily traceable. Researchers and staff can access and visualize the information effectively through dashboards. Quick visualization of data and an alert system is particularly important for researchers to ensure the correct execution of procedures and to be able to react promptly in

Figure 2: Example dashboard for the deployment scenario. Data is grouped by room and displays the monitored laboratory equipment, experiments, and job status on the 3D printers.

case of a malfunction. The server automatically records sensor downtimes and provides visual feedback in the dashboard. Data analysis is simplified through the platform as well. Researchers may access and analyse incoming or historical data with MATLAB, R, or any other preferred statistical tool through data export or the well-documented API.

Implementing *IoTree42* in our laboratories improved the workflow and provides effective research data management by allowing the relevant measurements to be directly streamed into a single database under full control of the research group. *Discussion & Limitations:* *IoTree42* has been tested and is in production for many different applications. It has demonstrated to be sufficient for many sensing applications in laboratories, such as an in vitro bioimpedance analyzer, radon detector, mobile light spectrometer, and many more. The laboratory use case outlined above is already aiding our researchers in their experiments. In the near future, we will deploy a wearable light spectrometry device and commercial actimetry sensors connected to the *IoTree42* platform. Measurements with these low-cost sensors will be a vital part of a field study.

Nevertheless, qualified personnel for the installation and maintenance of the platform is required at certain points. This includes server setup and administration, ensuring security throughout the system, and minimizing unnecessary load on the network and server. The repository also contains a growing selection of sensor and actuator examples for quick plug-and-play integration. At present, *IoTree42* does not provide as many ready to use code examples as most commercial platforms do, but scores with a flexible use and application-oriented features. With a growing and active community, these limitations might be overcome in the near future. *IoTree42* is published under the *MIT License*. (www.github.com/IoTree/IoTree42).

4 Conclusion

IoTree42 is a flexible and powerful IoT platform that is an OpenSource alternative to self-developed and isolated or commercial cloud-based solutions. It was developed out of the needs of researchers and reflects expansibility, cost efficiency, and data privacy. Once the central server is installed, it supports the researcher with a fast setup of distributed sensors and actors that can be connected wired or wirelessly. The transport of the data is done via the robust *MQTT* protocol. This design decision allows an easy integration of commercial or self-developed devices, based on common microcontrollers as the *Arduino* or *ESP* family. The data streams can further be picked up by third-party applications, allowing laboratory

automation or data processing in real time. *IoTree42* seamlessly connects to visual programming environments as *NodeRed* or *LabView* and lowers the hurdle of automating lab experiments for users without programming experience. The platform stores the data in a timeseries database and comes with a graphical user interface and dashboards. Dashboards are customizable to the individual needs and deliver basic data visualizations. The freedom to run *IoTree42* on a dedicated server as well as a single board computer such as the *Raspberry Pi*, depending on the needs, makes the platform attractive for scientific and industrial applications, or even for home use. The platform is continuously under development and benefits from the contribution of code for sensors, actors, or features contributed by the rapidly growing community.

Author Statement

Research funding: The authors state no funding involved. Conflict of interest: Authors state no conflict of interest. Informed consent: Informed consent has been obtained from all individuals included in this study.

References

- [1] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash, "Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications," *IEEE Communications Surveys Tutorials*, vol. 17, no. 4, pp. 2347–2376, 2015
- [2] G. Marques, R. Pitarma, N. M. Garcia, and N. Pombo, "Internet of Things Architectures, Technologies, Applications, Challenges, and Future Directions for Enhanced Living Environments and Healthcare Systems: A Review," *Electronics*, vol. 8, no. 10, Art. no. 10, Oct. 2019
- [3] S. P. Amaraweera and M. N. Halgamuge, "Internet of Things in the Healthcare Sector: Overview of Security and Privacy Issues," in *Security, Privacy and Trust in the IoT Environment*, Z. Mahmood, Ed. Cham: Springer International Publishing, 2019, pp. 153–179. doi: 10.1007/978-3-030-18075-1_8.
- [4] M. B. Yassein, M. Q. Shatnawi, S. Aljwarneh, and R. Al-Hatmi, "Internet of Things: Survey and open issues of MQTT protocol," in *2017 International Conference on Engineering MIS (ICEMIS)*, May 2017, pp. 1–6. doi: 10.1109/ICEMIS.2017.8273112.
- [5] B. Mishra and A. Kertesz, "The Use of MQTT in M2M and IoT Systems: A Survey," *IEEE Access*, vol. 8, pp. 201071–201086, 2020
- [6] R. T. Fielding, "Architectural styles and the design of network-based software architectures," phd, University of California, Irvine, 2000.
- [7] R. Schwartmann and S. Weiß, "Requirements for the use of pseudonymisation solutions in compliance with data protection regulations," p. 17, 2018
- [8] S. Stadler, *IoTree/IoTree42*. 2021. Accessed: Jul. 05, 2021. [Online]. Available: <https://github.com/IoTree/IoTree42>