Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 270 (2025) 387–396

Procedia Computer Science

www.elsevier.com/locate/procedia

29th International Conference on Knowledge-Based and Intelligent Information & Engineering Systems (KES 2025)

Design and Implementation of a Next-Generation Remote Lab for IoT and Industry 4.0

Piotr Czekalski[a,*], Krzysztof Tokarz[a], Godlove Suila Kuaban[b], Raivo Sell[c], Agris Nikitenko[d], Karlis Berkolds[d], Łukasz Lipka[e]

[a]*Faculty of Automatic Control, Electronics and Computer Science, Silesian University of Technology, Akademicka 16, 44-100, Gliwice, Poland*
[b]*Institute of Theoretical and Applied Informatics, Polish Academy of Sciences, Baltycka 5, 44-100, Gliwice, Poland*
[c]*Department of Mechanical and Industrial Engineering, Tallinn University of Technology, Ehitajate tee 5, Tallinn, Estonia*
[d]*Faculty of Computer Science, Information Technology and Energy, Riga Technical University, 6A Kipsalas Street, Riga, Latvia*
[e]*WEB Development Department, ITSilesia Łukasz Lipka, ul. Joachima Lelewela 10, Rydułtowy, Poland*

Abstract

The rapid adoption of the Internet of Things (IoT) across various industries, including manufacturing, healthcare, transportation, energy, and smart cities, has created a significant demand for skilled professionals in IoT hardware design, software development, deployment, and maintenance. However, the educational landscape has struggled to keep pace with this demand, as many universities lack dedicated IoT programs or fail to integrate IoT education into non-IT disciplines. This gap has led to a shortage of professionals skilled in IoT, posing a challenge for industries seeking to harness IoT technologies effectively. To address this challenge, next-generation remote laboratory infrastructures have emerged as a viable solution, enabling hands-on learning with IoT devices without the constraints of physical access to hardware. These infrastructures are particularly valuable for lifelong learners, students in remote areas, and institutions with limited resources, offering scalable and flexible access to IoT experimentation platforms via the Internet. Furthermore, the COVID-19 pandemic underscored the necessity of resilient and accessible online learning solutions, reinforcing the role of Virtual and Remote E-Laboratory (VREL) platforms in modern education. This paper presents the design, implementation, and operation of a next-generation remote laboratory infrastructure for IoT and Industry 4.0 education and research. Our approach builds upon the outcomes of two European Commission-funded projects—IOT-OPEN.EU (2016-2019) and IOT-OPEN.EU Reloaded (2022-2025) focuses on providing scalable, secure, and accessible remote IoT education solutions. We outline the architectural framework, key components, and technical challenges of deploying such infrastructures, including connectivity, security, scalability, and user management. Additionally, we discuss the educational impact of these platforms and their role in addressing the global IoT skills gap. Our findings demonstrate that remote IoT laboratories can be an effective and scalable alternative to traditional hands-on learning, supporting academic institutions and industry professionals in acquiring practical IoT skills. These infrastructures can enhance IoT education and workforce readiness by leveraging open-source technologies and modular architectures, ensuring broader accessibility and fostering innovation in the field.

Keywords: I4.0; industry 4.0; remote lab; distant lab; MQTT; CoAP; IoT; Industrial IoT;

* Corresponding author. Tel.: +48-32-237-2577

10.1016/j.procs.2025.09.157

1. Introduction

The Internet of Things (IoT) is transforming industries by enhancing productivity, efficiency, and automation across various sectors, including transportation, manufacturing, healthcare, education, agriculture, energy production and distribution, natural resource management, disaster monitoring, supply chain management, security, military defence, and smart cities. Tens of billions of IoT devices are being deployed, driving the demand for increased awareness of IoT technologies and a skilled workforce across the entire IoT ecosystem, encompassing hardware design and manufacturing, software development, deployment, operation, and maintenance.

The Fourth Industrial Revolution (Industry 4.0), driven by the Internet of Things (IoT), is reshaping industries and economies globally. However, IoT education remains limited. Many universities lack dedicated IoT programs or fail to integrate IoT courses into non-IT disciplines. This educational gap has led to a shortage of IoT-skilled professionals, presenting a significant challenge for industries seeking to leverage IoT technologies [1, 2, 3, 4].

To address this skill gap, IoT education is evolving in multiple ways. Universities are incorporating IoT into interdisciplinary study programs, and online self-study courses enable lifelong learners to gain IoT expertise either as a hobby or as a means to enhance their professional skills. Moreover, next-generation remote laboratory infrastructures are being deployed to provide hands-on learning opportunities for students and professionals who lack access to physical IoT hardware and software resources [5, 6, 7, 8, 9, 10, 11, 12].

The IOT-OPEN.EU ecosystem inherits from the two education-oriented EU-sponsored grants: IOT-OPEN.EU (2016-2019) and IOT-OPEN.EU Reloaded (2022-2025). The ecosystem naturally integrates various teaching and training resources to train new engineers in the IoT era, particularly in Industrial IoT, Industry 4.0, general IoT, and embedded systems. VREL's Next-Generation IoT laboratory is part of the ecosystem, enabling users to experiment with and interact with real hardware using only a web browser. An early version of the system, developed in 2016, was experimental in terms of technology and functionalities.

E-learning and distant labs are well-known formulas that recently experienced a boost during the COVID-19 pandemic. Many academic facilities were inaccessible due to ongoing lockdowns and epidemic-related limitations, such as the number of researchers, students, and tutors who could physically access the infrastructure. The old VREL system [13] was convenient during the pandemic outbreak and lockdowns, but the pandemic has ruthlessly exposed all its drawbacks.

Another factor driving the development of e-learning is mobility, as students, teachers, and researchers often live in different physical locations and require access to the necessary infrastructure and resources. Erasmus+ students typically arrive when the academic semester at the destination university is already underway, while activities are ongoing and sometimes already completed. This lack of synchronisation and inflexibility can be easily handled with self-paced learning, online modules, and remote laboratories. Such initiatives are vital for rural communities and developing countries, where access to physical laboratory facilities remains challenging [14, 15].

There is a need to address the rising demand for hands-on IoT education. Recent advancements in modular architectural frameworks for designing remote laboratory infrastructures have been discussed in [10]. These frameworks typically consist of components such as interface modules, experiment servers, learning management systems (LMS), and local networks, enabling seamless remote access to IoT resources. Several projects have successfully implemented VREL infrastructures, demonstrating their effectiveness in IoT education and STEM learning [5, 6, 7, 8, 9, 10, 11, 12].

The primary limitation of existing next-generation remote laboratory infrastructures is their lack of scalability, flexibility, reliability, and user-friendly interfaces, which makes it challenging to upgrade the infrastructure to accommodate more users and introduce new features. Additionally, users encounter difficulties using some platforms that are not user-friendly, which slows down the adoption rate of remote or blended learning using next-generation remote laboratory infrastructures. Another aspect of next-generation remote laboratory infrastructures that should be considered is security. A distributed denial of service (DDoS) can disrupt the operation of the infrastructure, and could even paralyse the entire infrastructure, rendering it inoperable. Additionally, improper access control and encryption could lead to data leaks, compromising users' data privacy.

E-mail address: piotr.czekalski@polsl.pl

In this paper, we designed and implemented a scalable, flexible, secure, reliable, and user-friendly next-generation remote laboratory infrastructure for IoT and Industry 4.0 research and education. This initiative was supported by two European Commission-funded projects: IOT-OPEN.EU (2016-2019) and IOT-OPEN.EU Reloaded (2022-2025). We also discuss the implementation, operation, administration, and provisioning of these infrastructures, providing insights into how remote IoT education and research can be effectively supported at scale.

The remainder of the paper is structured as follows: Section 2 presents the architecture of the VREL NextGen Laboratory—a next-generation remote laboratory infrastructure developed to support education and research in IoT and Industry 4.0. Section 3 details the implementation aspects of the VREL NextGen Lab, highlighting key components and technologies. Section 4 addresses critical security considerations related to the remote lab infrastructure. Finally, Section 5 concludes the paper.

2. The Architecture of VREL NextGen Laboratory Infrastructure

This section presents the architecture of the VREL NextGen Laboratory, a next-generation remote lab infrastructure designed for IoT and Industry 4.0 education and research, shown in Figs. 1 and 2. It comprises IoT hardware, software, and networking resources that can be accessed remotely for various IoT experiments. The laboratory resources are hosted by universities and companies participating in the IoT-Open EU project consortium, under which the infrastructure was developed. One of the VREL NextGen web applications, available at https://iot.aei.polsl.pl, manages laboratories in Poland, including those at the Silesian University of Technology (SUT) and itSilesia (ITS). It also oversees laboratory nodes in Estonia, hosted by Tallinn University of Technology (TalTech) and ITT Group (ITT). Meanwhile, Latvia operates a separate software instance to manage laboratory nodes at Riga Technical University (RTU) and RobotNEST.

2.1. Software Architecture

VREL NextGen remote access software has been implemented as a classical web application with a containerised structure. It runs on a Docker host, and it is comprised of the following components (Fig. 3):

- Frontend - a GUI that provides all components and features needed for remote software development.
- Backend - a REST API implementing all system functions but frontend and compilation, including persistent storage in SQL database (PostgreSQL) for credentials, source files, and configuration.
- Compiler Services - a set of parallelised containers with load balancing (at least one) that implements compilation services and communication with laboratory nodes.

It is almost impossible to forecast the load demand from the users. Thus, there is no direct performance benchmarking data. The following challenges have been identified in this area:

- Number of consecutive compilations - refers directly to the user's programming skills.
- Source code complexity - users may compose virtually any code, from simple, 1-page "Hello World" apps to complex communication IoT applications using a variety of protocols.
- Amount of libraries used - because of the use of e.g. PlatformIO, users may freely use virtually any software library that needs to be pulled from the Internet repository during compilation time. A local caching mechanism is employed (at both the compiler and file system levels) to provide faster access for consecutive compilations, thereby reducing the demand for time and resources.

The upper limit case was identified as a theoretical (but low-probability) scenario where all laboratory nodes are booked and compilation occurs synchronously. In practical work, it never happens, but a good approach is to provide at least one CPU core per compiling activity. For this reason, scalability mechanisms are ensured on two levels: Docker-based containerisation, which easily enables the offloading of the three main components into different hardware resources (even different servers), and parallelisation of the stateless compiler, which has been identified as the system's most resource-demanding component. Each container is implemented in a multithreaded, asynchronous

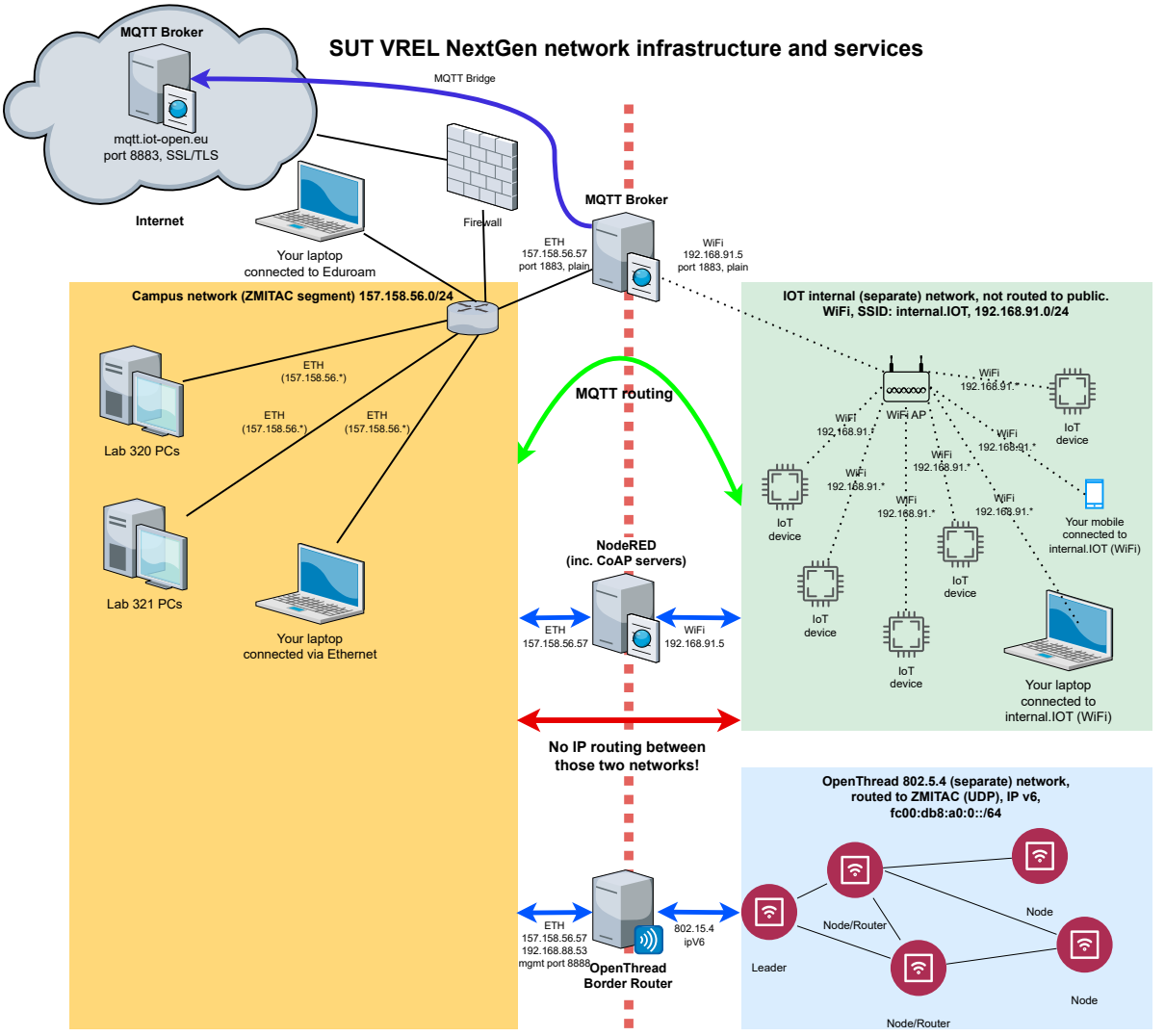

Fig. 1. Local VREL NextGen Network Infrastructure in SUT

model. Thus, it can use features of the modern multicore CPUs.

Video streams can be reduced to 720p and 1 fps to lower the network load and ensure comfortable operation for distant users working on low-bandwidth networks. Note, however, that it is up to the node implementation to ensure proper video streaming, not the VREL NextGen software itself, due to how the video stream is delivered to the end user (Fig. 8) and specific laboratory nodes (e.g. those with rapidly moving mechanical components) may require higher video resolution and fps.

A typical use-case scenario for bare-metal code development for laboratory nodes is presented in Fig. 4. As is known from many online websites, regular users (non-administrators) register with the website using an e-mail-based acknowledgement.

2.1.1. Common Use-case Scenario

Users can then select a device from the list of available devices and book it for exclusive use for a limited time. It is possible to book multiple devices. Booking time limits are set per laboratory node. Usually, this limit corresponds

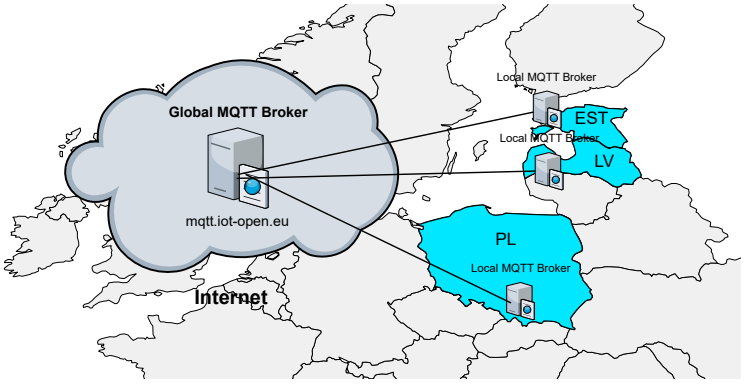

Fig. 2. Global VREL NextGen Network Infrastructure - MQTT bridging

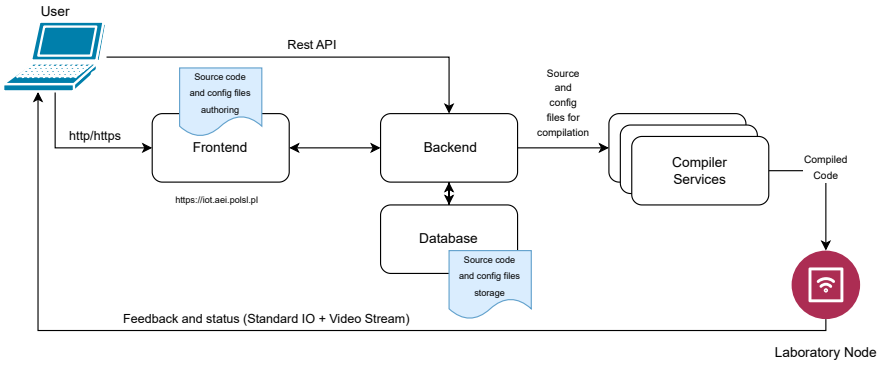

Fig. 3. VREL NextGen Structure

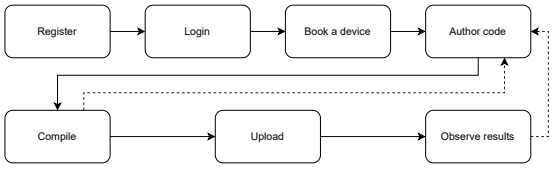

Fig. 4. Typical use-case scenario

with the study model, e.g., 45 minutes (one laboratory unit) or 90 minutes (two units in a row). Publicly available devices are usually set for a two-hour limit.

Once booked, users can develop code in the programming language predefined in the laboratory node's configuration, e.g., C++ or MicroPython. The system offers a flexible configuration with the free definition of Linux shell (bash) commands for any activity, allowing virtually unlimited use of various compilers and interpreters. The default configuration utilises the PlatformIO [16] and Arduino framework [17], as it is the most common scenario for bachelor-level students and hobbyists. Source code and other programming components are developed solely in the web browser (Frontside container) and later stored in the database (Backend container). Source code is commonly composed of one source file, but is not limited to.

Sample screen with live device's view and integrated documentation is present in Fig. 5, where 1 is source code development area, 2 is a section for live streams presenting the device's work in close-to-real-time (usually a web camera but not limited to), 3 is an output section for browsing code compilation results and results of interfacing with end node device via proxy (e.g. firmware upload) and section 4 is integrated documentation (both technical node documentation and hand-on-labs scenarios).

Fig. 5. Developer's View

The code compilation process provides textual feedback. Before compilation, the Backend copies all source and configuration files to the Compiler Service according to the file transfer pattern defined by the administrator. Then, if previous compilation results (binaries) exist, the Backend cleans them up and executes the necessary tools to compile, interpret, or verify the code. When the code is finished and compiled, it can be uploaded to the device to flash the memory, reconfigure it, or perform any other necessary actions to execute the algorithm on the laboratory node. A standard IO output is provided to the users to inform them about successes and failures. In the case of laboratory nodes that have been implemented so far, a fog-class IoT device (Raspberry Pi 4 and 5, abbreviated as RPI) was used for this task. VREL NextGen Compiler Service communicates remotely with laboratory nodes, i.e., using SSH exposed by fog-class devices, which work as proxies for the end-node IoT devices. Depending on the type of IoT hardware, the proxy's role is to upload, reset, flash, erase, or reconfigure the end-node IoT device.

These fog-class devices also work as webcam servers, providing users with close-to-real-time feedback via video streaming. Video streams are delivered directly to the end user's browser as an embedded IFRAME, significantly lowering processing demand and distributing load across various devices.

2.1.2. Users, Laboratories and Administration

The newly registered user is added to the "public" group, which has limited access to laboratory nodes. This approach has zero administration costs for instances. Devices are organised into "laboratories" (groups of IoT devices), and each laboratory can have at least one (usually more) administrator. One device can be shared across multiple laboratories, which is a rare occurrence. Integration services are necessary to facilitate the collaboration of various devices that are spatially distributed and differ in technology, as discussed in Section 2.3.

It is possible to configure a variety of features for every laboratory node, including sets of Linux bash commands executed: When copying prior compilation starts, e.g. copying sources from the database to the Compiler Service and composing an appropriate project folder structure; prior compilation, e.g. removal of old binaries remaining after previous compilations; to check if compilation was successful - a script returning 1 or 0; for compilation task, linking, syntax checking - framework depending; to upload the project to the device.

The device's configuration also includes definitions for the documentation section in the form of URLs that appear in the lower right corner of the user's GUI (Fig. 5 section 4) as tabs and for remote inspection of the laboratory node's hardware (e.g. webcam, Fig. 5 section 2) to observe development results. The system supports multiple cameras per node.

Users can be assigned to multiple user groups. Groups can then be assigned laboratories, thus providing access to the devices. This makes it easy to organise and manage various virtual laboratory rooms and share resources among student cohorts.

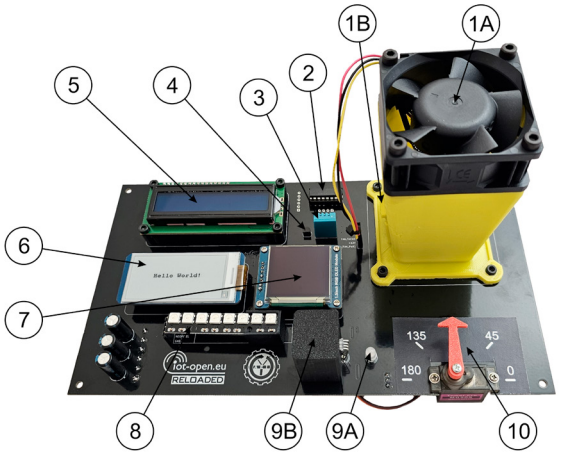

Fig. 6. Sample IoT end node, edge-class device design, implemented in SUT and RTU

2.2. VREL Hardware Design

Laboratory nodes implemented in existing laboratories are designed as zero-administration devices. Due to purely remote access, there are no physical inputs, such as buttons or knobs; instead, there are sensors and actuators optimised for video streaming with web cameras. Fig. 6 presents one of the laboratory node designs. It includes a motherboard with sensors (1B, 2, 3, 4, 9B), actuators, and displays utilising various technologies (1A, 5, 6, 7, 8, 9A, 10), as well as power lines [18].

The motherboard's back side contains a universal slot for connecting various IoT microcontrollers that implement network connectivity. Lab nodes use universal development boards that are available on the market. They are proxied via custom-made adapters that reroute the GPIOs of the microcontroller development boards, ensure proper powering (e.g., DC-DC adapters), and add port expanders in the case of pin-constrained microcontrollers. Adapters also implement additional connectivity features, such as 2nd radio coprocessor. A sample adapter for STM32WB55RG development boards is presented in Fig. 7. The motherboard is designed to allow users to experiment with almost all industrial standard embedded protocols, such as I2C (TWI), SPI, UART, PWM, and 1-Wire, as well as visualisation technologies including e-Paper, RGB LED, Smart LED stripes, OLED, and LCD. The motherboard does not contain any network interfaces - it is solely implemented by the MCU part plugged into the motherboard.

Each device contains an edge (end-node) IoT device and a proxy RPI; this kind of laboratory node architecture is presented in Fig. 8. Each IoT device is represented here by a programmable IoT-class MCU and standard motherboards that compose the IoT edge (end-node); the programmable device is connected to the RPI via USB. Compiler Service connects to the proxy and sends commands and firmware packs to be flashed in the edge device for bare-metal programming or uploads source code if using middleware interpreters.

An integrated web camera observes the front of the motherboards with sensors and actuators. A video stream is created with a USB camera and a Motion server. The video stream is then provided as a secure MJPEG stream that can be accessed via HTTPS and embedded into the development software as presented in Fig. 5, Section 2.

2.3. Integration Services

Integration services for IoT labs enable two-way communication between devices within the same IoT network and across the global Internet. This raises a variety of challenges, mainly related to cybersecurity. Exposing all IoT devices to the public Internet is not reasonable from a security perspective. If devices with weak security are publicly available, they can be easily compromised and used as DDoS attack endpoints.

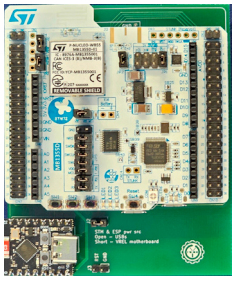

Fig. 7. Adapter board for 802.15.4-based STM32 WB55RG MCU with 802.11 radio coprocessor

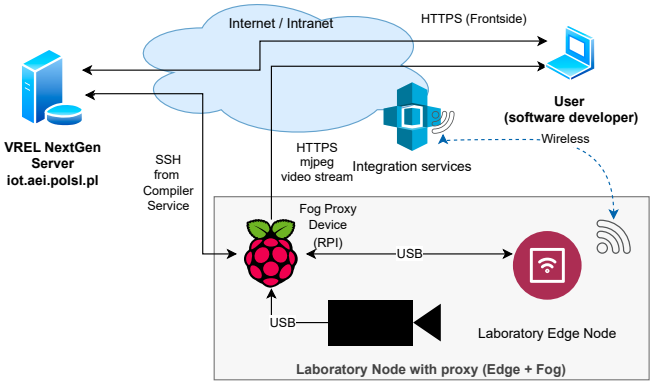

Fig. 8. Laboratory architecture with proxy (fog) and end-node (edge) IoT devices

2.3.1. Spatial Integration and Its Challenges

A single VREL NextGen management software instance can manage multiple laboratories across multiple physical and logical locations. While this feature seems quite simple in its logic, its implementation is challenging, primarily due to latency and security requirements. Most IoT end-nodes are constrained devices and, therefore, are protected and hidden behind a firewall, using a separate network for connectivity. Also, it is a typical security policy that the infrastructure is not fully or even partially available in the public IP addressing space. Thus, connectivity from the software instance to the remote devices requires port forwarding techniques, VPN connectivity, and reverse proxying.

Another challenge is integrating services among different spatial locations: IoT devices commonly use communication protocols that are not straightforwardly compatible with the Internet, e.g. Thread or Zigbee. Moreover, even those devices capable of implementing a full Internet stack (TCP/UDP over IP) are commonly limited in Internet access due to the lack of security or the security policies that exclude experimentation on Internet-opened devices.

2.3.2. Proposed Implementation

The VREL NextGen ecosystem currently implements selected integration services that, on the one hand, are considered secure enough to be accessible on the public Internet, and on the other, enable interconnection among different locations, such as Poland, Latvia, and Estonia. Details of those services are presented in Figs. 1 and 2, and include:

- local MQTT-brokers implemented specifically for integration of local IoT network and local Internet connections (e.g. campus network) - enable secure relay of the messages from the IP-based IoT networks to and from the local Internet, behind the firewall,
- global MQTT-broker (specifically mqtt.io-open.eu, accessible in public Internet, still secured) that enables interconnection among local MQTT brokers, fully enabling cross-border integration of the spatially distributed laboratories, enabling truly global connectivity,
- local OpenThread Border Routers that bind local IoT 802.15.4 Thread mesh networks with IP-based local Internet, including CoAP to MQTT protocol translation thanks to the use of NodeRED.

- NodeRED-based solution to bind local IoT, IP-based network with local Internet, virtually enabling the implementation of any communication and translation scenarios, also providing dashboard services accessible from local Internet to control IoT devices working in a dedicated IoT network.

3. The implemented VREL next-generation laboratory scenarios

Regarding specific needs for remote access hardware, there is technical documentation for each node, as well as hands-on scenarios for both embedded and IoT tasks. Embedded scenarios serve as starting points for new users who need to understand how to use the node and visualise its state, such as communication status. Once they get familiar with the technical details of accessing sensor data and controlling actuators (including displays to visualise data for remote view via webcam), users can switch to more sophisticated scenarios requiring multiple devices, external (integration) services, and IoT communication. In the book [18], there are currently 12 embedded type scenarios, one embedded advanced scenario and 9 IoT communication scenarios.

This builds the concrete basis for real-life use cases related to Industrial IoT, Smart Homes, and other IoT application domains.

4. Security in VREL NexgGen

VREL NextGen software shares common security implementations with many other websites. In short, those are: encrypted data transmission, hidden components behind a reverse proxy (utilising NGINX reverse proxies), hidden access to the proxy devices behind a NAT with a firewall, and encrypted connections among software components using an internal, hidden, container-dedicated network.

As VREL NExtGen IoT laboratory nodes are freely programmable, edge-class end-node devices should not directly connect to the Internet. They may, however, use integration services (e.g., local MQTT Brokers) that require authorisation, provide secure connection mechanisms and enable logical relaying of the messages and communication with the Internet and other segments of the spatially distributed infrastructure.

4.1. Global Secure MQTT Infrastructure

A special use case is the MQTT messaging infrastructure that binds laboratories spatially distributed across networks and countries. Local MQTT brokers enable interconnectivity among devices within the same network, typically located in the same physical space. Those brokers are frequently located behind the firewall of academic institutions. A part of the VREL NextGen infrastructure is a global MQTT broker. Local MQTT brokers forward selected messages to and subscribe to selected messages from the global MQTT broker. This technique is known as MQTT bridging and utilises a secure connection with certificate-based authentication and strong encryption for communication. Architecture is presented in Fig. 2.

5. System Validation

The VREL NextGen system is currently installed in two locations: the Silesian University of Technology (public instance), available online at https://iot.aei.polsl.pl, and at Riga Technical University (private instance). The total number of students participating in the system evaluation is over 50 for the public instance. Students perform their regular laboratory duties for subjects such as: general Internet of Things (Bachelor's), IoT Security and IoT Networking (Master's).

6. Conclusion

The rapid expansion of IoT applications across various industries has created an urgent need for skilled IoT professionals. The current education landscape struggles to meet this demand, with many universities lacking dedicated IoT programs or failing to integrate IoT courses into non-IT disciplines.

We designed and implemented a next-generation remote laboratory for IoT and Industry 4.0 education and research to address this challenge. This infrastructure provides a scalable, flexible, and secure remote approach, enabling students, professionals, and lifelong learners to acquire practical IoT skills regardless of their location or institutional resources.

Our approach was developed through the IOT-OPEN.EU (2016-2019) and IOT-OPEN.EU Reloaded (2022-2025) projects demonstrate that remote IoT laboratories can be an effective alternative to traditional learning.

Moving forward, we anticipate that remote IoT laboratories will play a critical role in supporting interdisciplinary learning and fostering innovation. Further research should explore AI-driven automation, improved security mechanisms, and real-time collaboration features to enhance the effectiveness and scalability of such platforms.

Acknowledgements

This publication was supported by the Department of Computer Graphics, Vision, and Digital Systems under the statute research project (Rau6, 2025), Silesian University of Technology (Gliwice, Poland). This paper was partially supported by the IOT-OPEN.EU Reloaded Erasmus+ KA2 HED grant no 2022-1-PL01-KA220-HED-000085090.

References

[1] I. Delgado, E. Sancristobal, S. Martin, and A. Robles-Gómez, "Exploring iot vulnerabilities in a comprehensive remote cybersecurity laboratory," *Sensors*, vol. 23, no. 22, p. 9279, 2023.

[2] G. S. Kuaban, M. Nowak, P. Czekalski, K. Tokarz, J. K. Tangka, K. Siggursson, A. Nikitenko, K. Berkolds, and R. Sell, "An iot course program to foster the adoption of iot driven food and agriculture in sub-saharan africa (ssa)," in *2022 International Conference on Electrical, Computer and Energy Technologies (ICECET)*. IEEE, 2022, pp. 1–7.

[3] S. Coşkun, Y. Kayıkcı, and E. Gençay, "Adapting engineering education to industry 4.0 vision," *Technologies*, vol. 7, no. 1, p. 10, 2019.

[4] O. Bongomin, G. Gilibrays Ocen, E. Oyondi Nganyi, A. Musinguzi, and T. Omara, "Exponential disruptive technologies and the required skills of industry 4.0," *Journal of Engineering*, vol. 2020, no. 1, p. 4280156, 2020.

[5] A. Kalashnikov, H. Zhang, J. Jennings, and M.-M. Abramriuk, "Remote laboratory: using internet-of-things (iot) for e-learning," 2017.

[6] P. Jacko, M. Bereš, I. Kováčová, J. Molnár, T. Vince, J. Dziak, B. Fecko, Š. Gans, and D. Kováč, "Remote iot education laboratory for microcontrollers based on the stm32 chips," *Sensors*, vol. 22, no. 4, p. 1440, 2022.

[7] M. Garefalakis and S. Panagiotakis, "Integration of a remote lab with a learning system for training on microcontroller programming," in *Proceedings of the 27th Pan-Hellenic Conference on Progress in Computing and Informatics*, 2023, pp. 224–231.

[8] S. Amador Nelke, D. Kohen-Vacs, M. Khomyakov, M. Rosienkiewicz, J. Helman, M. Cholewa, M. Molasy, A. Górecka, J.-F. Gómez-González, M. Bourgain *et al.*, "Enhancing lessons on the internet of things in science, technology, engineering, and medical education with a remote lab," *Sensors*, vol. 24, no. 19, p. 6424, 2024.

[9] S. Viswanadh Kandala, A. Gureja, N. Walchatwar, R. Agrawal, S. Sinha, S. Chaudhari, K. Vaidhyanathan, V. Choppella, P. Bhimalapuram, H. Kandath, and A. Hussain, "Engineering end-to-end remote labs using iot-based retrofitting," *IEEE Access*, vol. 13, pp. 1106–1132, 2025.

[10] S. A. Nelke, M. Khomyakov, S. Mhmad, M. Winokur, and A. Benis, "Designing and developing a remote iot lab for enhanced lab classes," in *Proceedings of the Designing and Developing a Remote IoT Lab 2023 Pre-ICIS SIGDSA Symposium on Addressing Global and Grand Challenges with Analytics*, 2023, pp. 1–4.

[11] A. R. Rao and A. Elias-Medina, "Designing an internet of things laboratory to improve student understanding of secure iot systems," *Internet of Things and Cyber-Physical Systems*, vol. 4, pp. 154–166, 2024.

[12] M. Ramya, G. Purushothama, and K. Prakash, "Design and implementation of iot based remote laboratory for sensor experiments," 2020.

[13] K. Tokarz, P. Czekalski, G. Drabik, J. Paduch, S. Distefano, R. Di Pietro, G. Merlino, C. Scaffidi, R. Sell, and G. S. Kuaban, "Internet of things network infrastructure for the educational purpose," in *2020 IEEE Frontiers in Education Conference (FIE)*. IEEE Press, 2020, p. 1–9. [Online]. Available: https://doi.org/10.1109/FIE44824.2020.9274040

[14] G. S. Kuaban, V. Nkemeni, O. J. Nwobodo, P. Czekalski, and F. Mieyeville, "Internet of things adoption in technology ecosystems within the central african region: The case of silicon mountain," *Future Internet*, vol. 16, no. 10, p. 376, 2024.

[15] G. S. Kuaban, P. Czekalski, E. L. Molua, and K. Grochla, "An architectural framework proposal for iot driven agriculture," in *Computer Networks: 26th International Conference, CN 2019, Kamien Slaski, Poland, June 25-27, 2019, Proceedings 26*. Springer, 2019, pp. 18–33.

[16] P. Labs, "Platformio: Your gateway to embedded software development excellence," https://platformio.org/, 2025, [Online; accessed 18 March 2025].

[17] M. Banzi and M. Shiloh, *Getting started with Arduino: the open source electronics prototyping platform*. Maker Media, Inc., 2022.

[18] P. Czekalski, T. Krzysztof, S. Ingmar, S. Raivo, N. Agris, B. Karlis, and L. Łukasz, "The yellow book 2nd ed. - vrel nextgen laboratory guide," https://www.roboticlab.eu/book/pdf/iot-open2ndedpractical.pdf, 2025, [Online; accessed 18 March 2025].