Check for updates

SOFTWARE TOOL ARTICLE

LEAF: Lightweight equipment adapter framework

[version 1; peer review: 1 approved with reservations]

Matthew Crowther[1], Brett Metcalfe[2], Carlos Suarez[3], David Camilo Corrales[4], Jairo Alexander Astudillo Lagos[5], Anil Wipat[1], Timothy J. Rudge[1], Katherine James[1], Jasper J. Koehorst[2,6]

[1]Newcastle University, Newcastle upon Tyne, England, UK
[2]Laboratory of Systems and Synthetic Biology, Wageningen University & Research, Wageningen, Gelderland, The Netherlands
[3]Departamento de Telemática, Universidad del Cauca, Popayán, Colombia
[4]CNRS, INRAE, INSA, Université de Toulouse, Toulouse, France
[5]INRAE, Toulouse, UMS 1337 TWB, France
[6]UNLOCK, Wageningen, The Netherlands

v1 **First published:** 17 Jun 2026, **6**:210
https://doi.org/10.12688/openreseurope.24168.1
Latest published: 17 Jun 2026, **6**:210
https://doi.org/10.12688/openreseurope.24168.1

Abstract

Background

As our understanding of complex (biological) systems within the (life) sciences increases, so does the continuous demand for data to maintain that understanding. New approaches for knowledge generation, real-time prediction, and decision-making, such as machine learning, do not reduce this data deficit, instead requiring diverse and continuously updated data streams. Furthermore, in order to be reusable, machine-readable and computationally tractable, such diverse datasets must be seamlessly integrated and efficiently processed across different sources, necessitating a form of standardization. However, the amount and diversity of data required far exceed the capabilities of capturing and curating data manually, making manual approaches increasingly infeasible. One solution would be the automation of, or at least enhancement of, digitization of research data. Although automation of research environments and automated data extraction are not new ideas, the issue of vendor-specific software, processes, and data structures makes generic inter-equipment data extraction challenging. A “one-size-fits-all” approach can be difficult with such diverse and disparate systems. Results Here, the Lightweight Equipment Adapter Framework (LEAF) is introduced, a modular open-source system that can be adapted to monitor different equipment and extract data automatically. LEAF is flexible and scalable, enabling integration with various types of communication

Open Peer Review

Approval Status ?

	1
version 1 17 Jun 2026	? view

1. **Ruirui Zhong**, Zhejiang University, Hangzhou, China

Any reports and responses or comments on the article can be found at the end of the article.

protocols, instruments, and data formats. By leveraging reusable modules, the framework allows for rapid development of new adapters, minimizing the effort required to support additional equipment. Once deployed, LEAF operates with minimal human intervention, facilitating continuous and automated data collection. Standardisation of data is achieved through automated data collection, ensuring interoperability across different systems and enabling more efficient data analysis and downstream processing.

Conclusions

LEAF shifts compliance with data standards to the instrument level, generating better metadata, and allowing the integration of data from diverse sources. Ultimately, LEAF addresses the challenge of fragmented laboratory and research automation by providing a unified, adaptable solution for extracting and structuring experimental data.

Plain Language summary

Research often heavily depends on vast quantities of data generated by many different and diverse laboratory instruments and systems. At the same time, technologies such as artificial intelligence and machine learning require continuous, well-structured, and reusable data streams. However, research equipment from different manufacturers often stores data in incompatible formats, making it difficult to combine and analyze information efficiently. As datasets grow the manual collection and curation of these datasets is becoming increasingly impractical.

To address this challenge, we developed the Lightweight Equipment Adapter Framework (LEAF), an open-source system that automatically connects to equipment and extracts data. LEAF is flexible and modular, allowing it to work with many types of instruments, communication protocols, and data formats. New adapters can be added with limited effort, making the system scalable and reusable.

By automating data collection and standardization, LEAF improves interoperability, reduces manual workload, and supports more efficient analysis and integration of research data across diverse systems.

Keywords

Laboratory automation; Internet of things (IoT); Data Acquisition (DAQ); Lightweight Equipment Adapter Framework (LEAF).

HE This article is included in the Horizon Europe gateway.

Corresponding authors: Brett Metcalfe (brett.metcalfe@wur.nl), Jasper J. Koehorst (jasper.koehorst@wur.nl)

Author roles: Crowther M: Conceptualization, Formal Analysis, Investigation, Methodology, Software, Validation, Writing – Original Draft Preparation; **Metcalfe B**: Conceptualization, Formal Analysis, Funding Acquisition, Project Administration, Software, Supervision, Validation, Writing – Original Draft Preparation, Writing – Review & Editing; **Suarez C**: Methodology, Software, Validation; **Camilo Corrales D**: Conceptualization, Supervision, Validation, Writing – Original Draft Preparation, Writing – Review & Editing; **Astudillo Lagos JA**: Conceptualization, Software, Validation, Writing – Review & Editing; **Wipat A**: Conceptualization, Funding Acquisition, Project Administration, Supervision, Writing – Original Draft Preparation; **Rudge TJ**: Project Administration, Software, Supervision, Writing – Original Draft Preparation; **James K**: Conceptualization, Project Administration, Resources, Supervision; **Koehorst JJ**: Conceptualization, Data Curation, Formal Analysis, Funding Acquisition, Investigation, Methodology, Project Administration, Resources, Software, Supervision, Validation, Visualization, Writing – Original Draft Preparation, Writing – Review & Editing

Competing interests: No competing interests were disclosed.

Grant information: This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No. 101094287 (RI services to promote deep digitalization of Industrial Biotechnology – towards smart biomanufacturing [BIOINDUSTRY 4.0]). J.J.K acknowledges the Dutch Research Council (NWO) and Wageningen University & Research for their financial contribution to the UNLOCK initiative (NWO: 184.035.007). LEAF is funded by the Dutch Research Council (NWO) and Open Science NL under grant agreement No. 500.040.2488 (Transforming Research Equipment Connectivity: An Open IoT Platform for Real-Time Data Collection and Digital Twinning). C.S. was funded through Call 823 – High-Level Training of Human Capital for the Cauca region. C.S. and D.C.C. acknowledge the support of the Ministry of Science, Technology, and Innovation (MinCiencias).
The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

Copyright: © 2026 Crowther M *et al*. This is an open access article distributed under the terms of the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

How to cite this article: Crowther M, Metcalfe B, Suarez C *et al.* **LEAF: Lightweight equipment adapter framework [version 1; peer review: 1 approved with reservations]** Open Research Europe 2026, **6**:210 https://doi.org/10.12688/openreseurope.24168.1

First published: 17 Jun 2026, **6**:210 https://doi.org/10.12688/openreseurope.24168.1

List of abbreviations

AI: Artificial Intelligence
DAQ: Data Acquisition
FAIR: Findable, Accessible, Interoperable, Reusable
IoT: Internet of Things
LEAF: Lightweight Equipment Adapter Framework
ML: Machine Learning
UTC: Coordinated Universal Time

Statement of need

The Third Industrial Revolution, characterized by the advancement of digital technologies, has transformed industries by integrating digital systems and automation.[1] Similarly, as in the wider world, innovation in digitalization has propagated into the day-to-day running of most research laboratories.[2] For example, most laboratories now benefit from equipment that is capable of some form of autonomous data generation (e.g., autosamplers), significantly improving the efficiency of data-driven research.[3] Others are developing "self-driving" laboratories.[4–7] As a result of this, experimental and observational methodologies are increasingly becoming sources of high velocity and high volume data. High throughput analytics, for instance, is responsible for tens to hundreds to thousands of measurements per day. However, despite this achievement and the open science movement gaining traction,[8,9] many laboratory instruments still rely on manufacturer-specific closed-source software to access, process, and transfer data in manufacturer-specific formats with manufacturer-specific terminology. Whereas, other instruments rely on manually transferring output files, colloquially referred to as the "Sneakernet".[10] Therefore, much of the resulting information and data remain locked within proprietary systems, creating research, or laboratory, equipment "data silos" that hinder the interoperability and integration of data.[11,12] This generates a paradox: laboratory automation is needed to accelerate scientific discovery (e.g.,[13,14]), but manual extraction, conversion, processing, and curation of such datasets coupled with a fragmented ecosystem of proprietary software, protocols, and data structures hinders it.

Unlike "manual" sampling (e.g.[15]), standardizing laboratory automation at the hardware and middleware layers would require significant vendor cooperation.[16] For a single laboratory or research team, integrating different devices and vendor specifics into a harmonious whole can be a large software project. Without direct vendor support, standardization efforts face substantial roadblocks. In an ideal world, equipment vendors would subscribe to agreed-upon standard protocols; naming conventions; data formats; and data structures allowing laboratory managers to quickly integrate new devices irrespective of manufacturer or date of production into existing laboratory data architecture. Although some manufacturers are agreeable to the potential of open source, many understandably retain tight control over their ecosystems, restricting third-party access to device firmware or proprietary middleware. Fragmentation can also occur when different vendors - where vendors also include in-house IT or R&D teams - design (bespoke) equipment with unique architectures, incompatible formats, and isolated workflows. Yet, if the issue of data acquisition and integration is left unresolved, it will create bottlenecks for open science[17,18] and FAIR'ification - the process of adapting datasets to the Findable, Accessible, Interoperable and Reusable (FAIR) principles.[19] Furthermore, it will also hinder progress towards achieving an interconnected laboratory landscape needed for the technologies of the next iteration of Industry Industry 4.0,[20,21] or 4IR. Industry 4.0 technologies rely on (near) real-time data transmission, such as machine learning (ML) and/or artificial intelligence (AI)[22,23] and digital twins,[24–26] the latter being digital replicas of physical systems.

Implementation

To address these challenges, here we present the Lightweight Equipment Adapter Framework (LEAF), an open source, modular solution designed to bridge the gap between various devices used in research environments and virtual client-side storage, monitoring, and analysis tools (Figure 1). In this context, the term adapter refers to the conversion of attributes (e.g., data, metadata, commands, etc.) from one format to another that is compatible with Internet of Things (IoT) architectures. Written in Python 3.12, LEAF provides a flexible adapter-based architecture that enables the seamless monitoring of equipment, allowing real-time data exchange, automated data processing, and integration of data from diverse equipment within 4IR frameworks. Provided that the equipment's data or metadata can be extracted programmatically any system can be connected.

Design considerations

At present data acquisition limitations force researchers to rely on proprietary software (e.g., MATLAB and Simulink), vendor-specific solutions, or custom-built bespoke interfaces (e.g., Labview) that can lead to inefficiencies, high costs, and significant barriers to large-scale data-driven research. A number of open source, Python based packages do exist for data acquisition, including: PythonDAQ; PYDAQ; and PyMoDAQ. Our rationale for developing LEAF was to focus on providing a simplified and modular solution designed to bridge the gap between various devices used in scientific

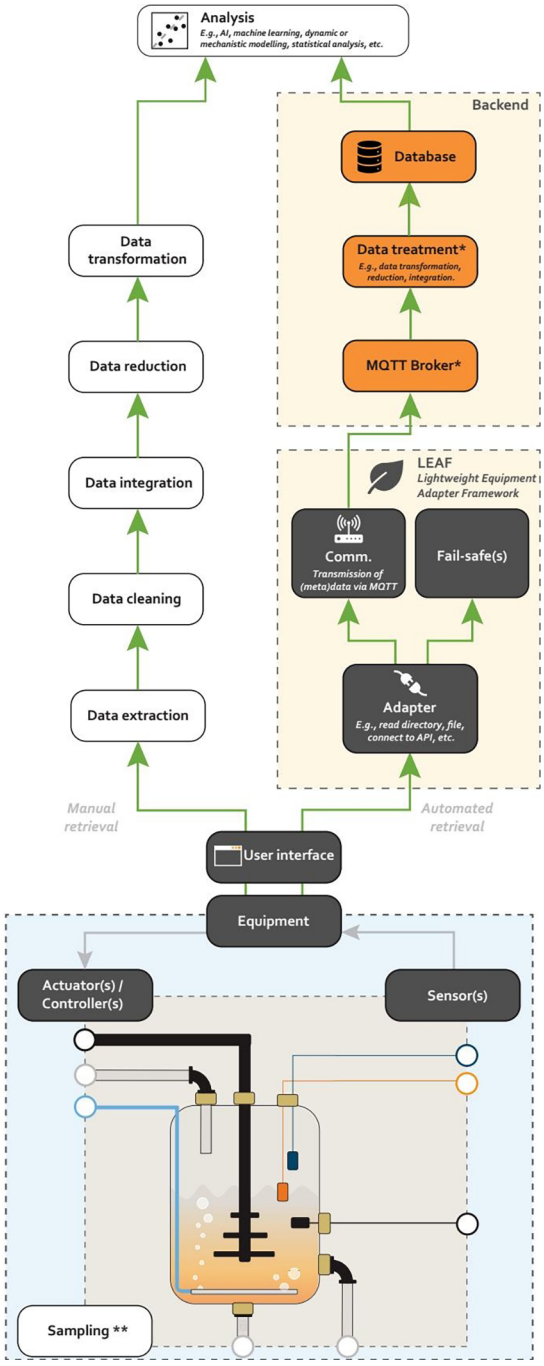

Figure 1. The overall goal of LEAF is to act as an intermediary between isolated laboratory equipment and client-side tools. LEAF should be laboratory equipment agnostic, providing a gateway by which data and metadata can be transmitted to client-side tools such as databases, machine learning, dynamic models, and other monitoring and analysis tools. The LEAF system transforms the disparate data from equipment into semi-standardized formats ready for consumption in the larger infrastructure.

research and virtual, client-side storage, monitoring and analysis tools (Figure 1). LEAF's design philosophy is based on five key principles:

- **Open** - To increase transparency, accessibility and allow for community collaboration without vendor lock-in.
- **Interoperable** - To encourage and to facilitate the system's use in as many pieces of laboratory equipment as possible, the system should facilitate interoperability. For instance, it should be operating system (OS) agnostic.

- **Modular** - The system should be composed of elements that perform a distinct function, reducing complexity when developing, debugging, and using.

- **Reusable** - Developed elements should be reusable for developing new elements to reduce effort and maximize returns through iterative improvements.

- **Persistent** - Once running, the system should persist until intentionally stopped irrespective of whether the system is shutdown (i.e., start on restart), no experiments are running, or if errors occur (i.e., experimentally or within the LEAF system).

Software design

LEAF achieves these design principles through a modular design with individual adapters performing a specific function with known inputs, outputs, and optional transformation steps. Equipment-specific modules (referred to as an EquipmentAdapter) acts as an interface composed of a "Controller" and "Interpreter". These modules facilitate the retrieval, transformation, and transmission of events and data (Figure 2). Depending on how the adapter has been configured, instrument and run (meta) data, as well domain-specific detail can be captured and terms standardised. The overall goal is to detect an event or state change on the equipment, such as an experiment starting or a measurement being taken, extract it, and then output this along with the required data to the larger infrastructure (outside of the adapter). This infrastructure may be an Internet of Things (IoT) architecture containing a time series database or a dashboard with live monitoring and data processing. Thereby, reducing the complexity surrounding the integration of individual devices. But also lowering the technical barrier for researchers and engineers to acquire data from laboratory systems.

Reusable modules

Reusability is achieved through two mechanisms: Modules and Core Adapters. Modules handle specific tasks, such as monitoring databases or outputting data via MQTT, while Core Adapters provide predefined structures and mappings for common equipment behaviours without being tied to specific devices, abstracting functions to enable reuse. These reusable modules reduce the need for developers to build everything from scratch because many types of equipment report events similarly, often writing measurements to files or databases or transmitting them via standard protocols such as Bluetooth, Modbus, OPC UA or Protobuf. LEAF prevents redundant implementation and accelerates development by offering pre-built modules that handle these common functions. Adapters are structured with functional slots, such as InputModules and OutputModules, allowing developers to search an existing repository for suitable modules. Once a match is found, they can integrate the module directly into the adapter, ensuring consistency and reducing effort.

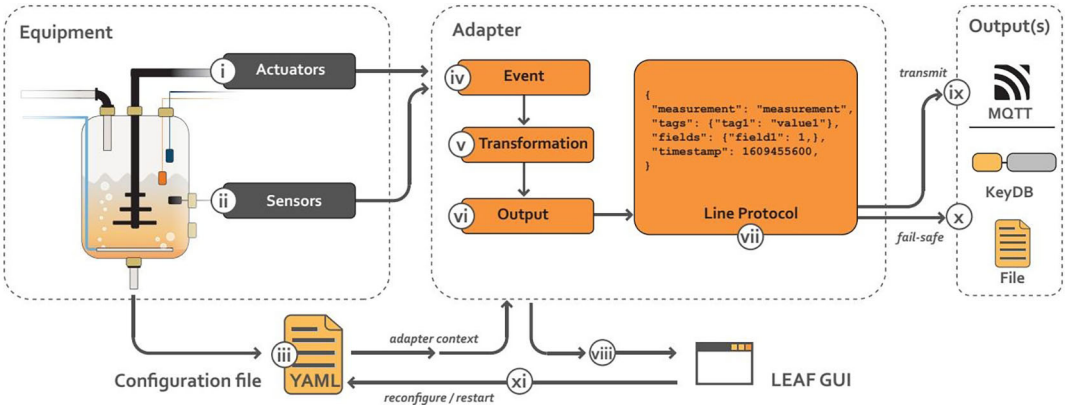

Figure 2. An overview of the LEAF adapter framework. LEAF can be installed on equipment representing either a collection of (i) actuators, (ii) sensors, or a combination of both. Users provide (iii) a configuration file to provide LEAF with both the "context" and any mandatory information. When (iv) events (e.g., measurements) occur on the equipment, these are detected and route the data through an appropriate process or phase can contain multiple processes, each with multiple phases, depending on the equipment type and requirements. Depending on the event, data may be (v) transformed before being (vi) transmitted externally in (vii) line protocol format. If LEAF is run in GUI mode the output will also been shown in the web interface. The output can be stored locally (x) in a KeyDB or File and/or (ix) transmitted externally using MQTT. If required (xi) the adapter can be reconfigured and restarted.

Mapping events to processes

Research equipment can be broadly separated into devices that continuously perform sensing and actions and those that have discrete experimental windows (i.e., the device starts and stops). For the latter, these devices can provide experimental metadata, identify when errors have occurred, perform multiple actions, and measure many factors. Capturing the metadata around the experiments, including the control aspects, provides a better context for the experiment. With a better grasp of what occurred during an experiment, experimental and methodological protocols can include details that allow for increased replication. Therefore, because the LEAF system is designed to extract data from many pieces of equipment and each piece of equipment may perform multiple actions and be in various states, there needs to be a mechanism to route different events into appropriate workflows. For example, a measurement should not be processed in the same way as an error message.

When an input event is received from the equipment, such as a measurement or an error occurring, the adapter must determine how to handle it. This is done by mapping the input to a specific internal process (Figure 2) with the knowledge that the given data type represents a broad event on the equipment (e.g., an experiment run). Within each process, there are one or more 'phases", representing a discrete step or action, such as initialization, measurement, or shutdown. These phases contain the logic for interpreting and transforming the event data. By breaking down equipment behaviour into modular components and mapping events to the correct context, the adapter can cleanly separate concerns, accurately reflect the equipment's state, and make sense of the asynchronous, varied events produced by complex laboratory systems—ensuring downstream systems receive timely, well-categorized, and actionable data. Furthermore, because some processes are common (most equipment have runs that can be considered "discrete" as opposed to "continuous"), this modularity enables reuse across adapters, i.e., once a process is defined, it can be reused in similar functioning adapters.

Output format

The output of LEAF[27] follows the format of InfluxDB's line protocol for time series databases.[28] Line protocol consists of four components: measurement, tags, fields, and a timestamp. Where, the "measurement" is the identity of the measurement set. The "tag" and "fields" sets are key-value pair(s) that hold the data and metadata, respectively. Finally, the "timestamp" is Unix time, i.e., the number of non-leap seconds since 1st January 1970 00:00:00 UTC.

Using LEAF

LEAF is distributed via the Python Package Index (PyPI)[29] and available via Gitlab.[30] Individual adapters can be installed through the LEAF Marketplace,[31] where community-developed extensions can be shared. The LEAF Marketplace is currently populated with adapters developed during the development and testing phase, including adapters for the BioLector XT Microbioreactor from Beckman, MinKnow from Oxford Nanopore Technologies, the MAQ weather observatory in the Netherlands, and iRODS (Integrated Rule-Oriented Data System) for storage monitoring. As laboratory equipment is highly heterogeneous, multiple adapters have already been developed and new ones can be created as needed to support specific devices or data files. Developers who wish to build new adapters can use the provided template at.[30] Comprehensive documentation for installation, usage, and adapter development is available.[32]

After the installation of LEAF and the required adapter(s) - these adapters are automatically detected by LEAF - a user must compose the necessary configuration file (Figure 2). This configuration file defines the adapter to be used, associated metadata (e.g., device ID, institute, and department), operational parameters (e.g., time intervals, file paths, ports, hosts, and API tokens), and the output (i.e., MQTT, KeyDB database, or file-based storage) that LEAF will export to. Each adapter provides an example configuration file to assist the user. Once this configuration file has been completed, LEAF can be started in two modes: (i) Headless mode via the command line with a configuration file specifying the adapters and output settings required to communicate with the laboratory equipment; or (ii) in GUI mode with a web interface (Figure 2).

When using the configuration file mode, the application establishes a connection to the laboratory equipment during startup, and the resulting data stream is forwarded to the MQTT server for further processing. In the web interface, a configuration file can be modified via the GUI's configuration section, which triggers the (re) start of the appropriate adapters. In addition, the web interface allows users to discover and install adapters directly from the adapter section on demand. During an adapter's run, it can be stopped, restarted, or the configuration file modified. This can be achieved by modifying the configuration file directly, restarting LEAF or through the web interface, and adjusting the configuration file accordingly.

Discussion

Bridging the gap with LEAF

Here, we have presented LEAF, a tool that unifies research equipment, providing a real-time connection for subsequent data analysis, processing, and (Figure 1). By automatically capturing data from heterogeneous devices, LEAF overcomes the challenges posed by vendor specific software and manual data handling, streamlining the resulting research workflows. The key benefits of such a system include equipment and laboratory monitoring (i.e., knowing device availability or usage), experimental monitoring and optimization (i.e., knowing what occurred during an experiment), and enhanced reproducibility (i.e., transferring experimental information as it occurs). In the following, we will expand upon these benefits.

Laboratory and equipment monitoring

For laboratory managers and technical staff, LEAF can reduces manual data collection and collation facilitating laboratory and research automation. LEAF can also enhance laboratory oversight by unifying data from various devices. Thus collecting real-time performance metrics that can potentially allow early detection of likely problems before they escalate into costly failures. In addition, LEAF can be used to offer real-time visibility of available and active equipment and its experimental or measurement status. This enables efficient scheduling, identification of underutilized resources, and automatic logging of experiment timelines and parameters. By centralizing these data sources, LEAF supports a proactive, data-driven approach to equipment management.

Experiment monitoring and optimisation

For researchers, research and laboratory staff, LEAF also facilitates real-time monitoring, allowing researchers to track ongoing experiments with greater granularity and adjust when deviations arise. This not only accelerates the iterative process and enhances experimental efficiency but also shifts decision-making from a 'post-experimental' phase to a more practical 'during-experiment' phase. For example, researchers and technicians can terminate an experiment or measurement run whose data set may be low quality or unreliable before exhausting time, money, and samples. Moreover, LEAF's continuous data streams enable the development of digital shadows[33–35] and ultimately digital twins[24,26,33–36] — virtual models that optimize and control physical equipment in real-time by using real-time measurements to update the state parameters of the virtual model. By bridging the gap between simulation and laboratory operations, digital twins reduce manual trial-and-error, supporting adaptive and data-driven experimental design.

Enhancing reproducibility

For researchers, the real-time autonomous collection of data from research devices and its subsequent transmission can help reduce errors associated with manual data extraction, such as file or data copying and conversion. By providing the data in a standardized format, simple but common errors, such as switching around the month and date in timestamps, can be easily avoided. Furthermore, scientific research often suffers from reproducibility issues due to missing contextual details, as biological systems are highly sensitive to even minor variations. LEAF mitigates this by enabling comprehensive metadata capture, including experimental parameters, hardware changes, and deviations from the intended (experimental) setup. Preserving these details ensures researchers can accurately reconstruct experiments while assessing unforeseen variables, increasing confidence in results and improving the reliability of replication efforts.

Cloud if you want to, local if you don't

Overall, LEAF establishes a platform that meets the need for continuous, integrated data flow. As laboratories increasingly adopt automation and computational analysis, LEAF can be a critical bridge between experimental workflows and modern digital infrastructure. By breaking down data silos and promoting interoperability, LEAF lays the foundation for more scalable, adaptive, and intelligent laboratory systems.

Future work

Community

Although the LEAF system, as presented, is a fully functional approach to automatic data retrieval from various pieces of laboratory and research equipment, presently there are only a small number of adapters. These adapters represent the equipment available to the authors, but of course, there is a multitude of devices and equipment used in research that could conceivably be included in LEAF. In fact, if the LEAF system is to be successful, new adapters must be developed by the community—ideally through an open, collaborative model that encourages broad participation and contribution.[17,37] It is partly for this reason that the LEAF system is modular, allowing individual users to openly develop individual adapters for pieces of equipment for the wider scientific and research community without having to re-develop the core functionality (e.g. communication). Our aim is to foster a community spirit for devices. If a sufficient number of adapters are developed, especially for common data types or common equipment, the cost of laboratory automation will be reduced, as the wider community will bear it. To enable reuse of developed adapters, we have developed a marketplace. At,[31] a central repository contains all publicly available adapters that anyone can explore, install, and integrate into their

own LEAF instance. Within the LEAF web interface, a tab within the GUI has been added allowing users, especially those less technically minded, to quickly install adapters of interest. Making it easy for users to quickly find and get an adapter running. To further facilitate users each adapter comes with an example configuration to help users understand the requirements needed for each adapter. Through this interface they can also be removed if needed.

Closing the loop with control

The current version of LEAF is one-directional; data is transmitted from the device to a broker that can then be transmitted onwards to either back-end or front-end tools and services (i.e., data processing, storage, etc.). The implementation of a data exchange adapter for the transmission of simulation results from models (e.g., AI/ML, Digital Twins) to physical bioreactors, facilitating real-time corrective actions. This will require the design of standardized communication protocols, ensuring compatibility with control systems, and establishing a feedback loop to update simulations continuously. This approach can potentially improve process efficiency and robustness by enabling bioreactors to adjust parameters dynamically based on predictive simulations. In some instances and for some equipment, there is the potential to 'close the loop', allowing the transmission to send a prediction and the corrective action to be performed.

Conclusions

There is increasing demand for real-time data capture and generation to understand complex systems better, leading to scientific and laboratory equipment that benefits from automation. However, though they may be capable of generating data autonomously, these datasets may be effectively 'stuck' on such equipment, given limitations in interoperability and communication. To remove this impediment to real-time data integration here, the Laboratory Equipment Adapter Framework (LEAF) is presented. LEAF is a lightweight, open source, modular data acquisition solution designed to bridge the gap between diverse laboratory devices and virtual, client-side storage, monitoring and analysis tools (e.g., time series databases, dashboards, analytical workflows, etc.).

Availability of source code and requirements

- Project name: Lightweight Equipment Adapter Framework
- Home page: https://leaf-framework.org/
- Software available from: https://pypi.org/project/leaf-framework/
- Source code available from: https://gitlab.com/leaf-framework/leaf
- Archived source code at time of publication: doi.org/10.5281/zenodo.20510989
- Documentation: https://docs.leaf-framework.org/
- Operating system(s): Platform independent
- Programming language: Python
- Package management: (Python > 3.12, < 3.15) with aioredis, dateparser, httpx, influxobject, nicegui, paho-mqtt, pyyaml, python, redis, requests, watchdog, cairosvg, pillow
- License: Apache License, Version 2.0

Ethics and consent

Ethical approval and consent were not required.

Data availability

No public datasets were generated or used in this study, all described resources are publicly available.

Acknowledgements

Members of BIOINDUSTRY 4.0 are thanked for their discussions and input. BM thanks the Department of Earth Sciences, Vrije Universiteit Amsterdam for the use of their facilities as a Visiting Fellow. In addition we would like to thank Jöern Viell and Alexandros Giannakopoulos of RWTH Aachen University personally for testing adapters in a semi-industrial setting and Ariane Bize (INRAE), Arnau Gasset Franch (UAB), Pau Ferrer Alegre (UAB) for access to equipment for testing and prototyping. The MQTT community is thanked for providing the MQTT logo.[38]

References

1. Maynard AD: **Navigating the fourth industrial revolution.** *Nat. Nanotechnol.* 2015; **10**(12): 1005–1006.
Publisher Full Text

2. Rihm SD, Bai J, Kondinski A, *et al.*: **Transforming research laboratories with connected digital twins.** *Nexus.* 2024; **1**(1): 100004.
Publisher Full Text

3. Holland I, Davies JA: **Automation in the Life Science Research Laboratory.** *Front. Bioeng. Biotechnol.* 2020; **8**: 571777.
PubMed Abstract | **Publisher Full Text** | **Free Full Text**

4. Brazil R: **Inside the 'self-driving' lab revolution.** *Nature.* 2026; **652**: 262–264.
PubMed Abstract | **Publisher Full Text**

5. Arnold C: **Can robotic lab assistants speed up your work?** *Nature.* 2024.
PubMed Abstract | **Publisher Full Text**

6. Callaway E: **Will self-driving 'robot labs' replace biologists? Paper sparks debate.** *Nature.* 2026; **650**: 809–810.
PubMed Abstract | **Publisher Full Text**

7. Olsen ML, Copeland CE, Sundberg CA, *et al.*: **Design-driven optimization of low-cost reagent formulations for reproducible and high-yielding cell-free gene expression.** *Nat. Commun.* 2026; **17**.
PubMed Abstract | **Publisher Full Text** | **Free Full Text**

8. McKiernan EC, Bourne PE, Brown CT, *et al.*: **Point of View: How open science helps researchers succeed.** *elife.* 2016; **5**: e16800.
PubMed Abstract | **Publisher Full Text** | **Free Full Text**

9. Stirling J: **Open instrumentation, like open data, is key to reproducible science. Yet, without incentives it won't thrive.** *Philos. Transact. A Math. Phys. Eng. Sci.* 2024 July; **382**: 20230215.
PubMed Abstract | **Publisher Full Text** | **Free Full Text**

10. Jaya AC, Safitri C, Mandala R: **Sneakernet: A Technological Overview and Improvement.** *2020 IEEE International Conference on Sustainable Engineering and Creative Computing (ICSECC).* 2020; pp. 287–291.
Publisher Full Text

11. Denton N, Molloy M, Charleston S, *et al.*: **Data silos are undermining drug development and failing rare disease patients.** *Orphanet J. Rare Dis.* 2021 April; **16**(1): 161.
PubMed Abstract | **Publisher Full Text** | **Free Full Text**

12. Kleerebezem R, Stouten G, Koehorst J, *et al.*: **Experimental infrastructure requirements for quantitative research on microbial communities.** *Curr. Opin. Biotechnol.* 2021; **67**: 158–165.
PubMed Abstract | **Publisher Full Text**

13. Naugler C, Church DL: **Automation and artificial intelligence in the clinical laboratory.** *Crit. Rev. Clin. Lab. Sci.* 2019; **56**(2): 98–110.
Publisher Full Text

14. Wenzel T: **Open hardware: From DIY trend to global transformation in access to laboratory equipment.** *PLoS Biol.* 2023; **21**(1): e3001931.
PubMed Abstract | **Publisher Full Text** | **Free Full Text**

15. Nijsse B, Schaap PJ, Koehorst JJ: **FAIR data station for lightweight metadata management and validation of omics studies.** *GigaScience.* 2023; **12**(12): giad014.
PubMed Abstract | **Publisher Full Text** | **Free Full Text**

16. Bonvoisin J, Molloy J, Häuer M, *et al.*: **Standardisation of Practices in Open Source Hardware.** *Journal of Open Hardware.* 2020; **4**(1).
Publisher Full Text

17. Woelfle M, Olliaro P, Todd MH: **Open science is a research accelerator.** *Nat. Chem.* 2011; **3**(10): 745–748.
Publisher Full Text

18. Ramachandran R, Bugbee K, Murphy K: **From Open Data to Open Science.** *Earth and Space Science.* 2021; **8**(5): e2020EA001562.
Publisher Full Text

19. Wilkinson MD, Dumontier M, Aalbersberg IJ, *et al.*: **The FAIR Guiding Principles for scientific data management and stewardship.** *Sci. Data.* 2016; **3**(1): 160018.
PubMed Abstract | **Publisher Full Text** | **Free Full Text**

20. Morisson A, Pattinson M: *Industry 4.0. : A Policy Brief from the Policy Learning Platform on Research and innovation.* Lille, France: Interreg Europe Policy Learning Platform; 2019.

21. Pfeiffer S: **The Vision of Industrie 4.0 in the Making a Case of Future Told, Tamed, and Traded.** *NanoEthics.* 2017; **11**: 107–121.
PubMed Abstract | **Publisher Full Text** | **Free Full Text**

22. Acosta-Pavas JC, Robles-Rodriguez CE, Griol D, *et al.*: **Soft sensors based on interpretable learners for industrial-scale fed-batch fermentation: Learning from simulations.** *Comput. Chem. Eng.* 2024; **187**: 108736.
Publisher Full Text

23. Metcalfe B, Acosta-Pavas JC, Robles-Rodriguez CE, *et al.*: **Towards a machine learning operations (MLOps) soft sensor for real-time predictions in industrial-scale fed-batch fermentation.** *Comput. Chem. Eng.* 2025; **194**: 108991.
Publisher Full Text

24. Knibbe WJ, Afman L, Boersma S, *et al.*: **Digital twins in the green life sciences.** *NJAS: Impact in Agricultural and Life Sciences.* 2022; **94**(1): 249–279.
Publisher Full Text

25. Bauer P, Stevens B, Hazeleger W: **A digital twin of Earth for the green transition.** *Nat. Clim. Chang.* 2021 Feb; **11**(2): 80–83.
Publisher Full Text

26. Grieves M: *Digital twin: manufacturing excellence through virtual factory replication.* Florida, US: Florida Institute of Technology; 2014.

27. PyPi: influxobject: 2024. accessed: 2026-04-02.
Reference Source

28. InfluxData Inc: **Write data to InfluxDB 3 Core: Line protocol.** 2026. accessed: 2026-04-02.
Reference Source

29. PyPi: Lightweight Adapter Framework (LEAF): 2025. accessed: 2026-04-02.
Reference Source

30. Gitlab: LEAF: 2026. accessed: 2026-04-02.
Reference Source

31. LEAF - Lightweight Equipment Adapter Framework Marketplace: 2026. accessed: 2026-04-02.
Reference Source

32. LEAF - Lightweight Equipment Adapter Framework: 2026. accessed: 2026-04-02.
Reference Source

33. Tekinerdogan B, Verdouw C: **Systems Architecture Design Pattern Catalog for Developing Digital Twins.** *Sensors.* 2020; **20**: 5103.
PubMed Abstract | **Publisher Full Text** | **Free Full Text**

34. Metcalfe B, Boshuizen HC, Bulens J, *et al.*: **Digital twin maturity levels: a theoretical framework for defining capabilities and goals in the life and environmental sciences.** *F1000Res.* 2023; **12**: 961.
Publisher Full Text

35. Kritzinger W, Karner M, Traar G, *et al.*: **Digital Twin in manufacturing: A categorical literature review and classification.** *IFAC-PapersOnLine.* 2018; **51**: 1016–1022.
Publisher Full Text

36. Grieves M, Vickers J: **Digital Twin: Mitigating Unpredictable, Undesirable Emergent Behavior in Complex Systems.** Kahlen FJ, Flumerfelt S, Alves A, editors. *Transdisciplinary Perspectives on Complex Systems: New Findings and Approaches.* Cham: Springer International Publishing; 2017; pp. 85–113.
Publisher Full Text

37. Kim SM, Choi HS, Rhee WS: **IoT home gateway for auto-configuration and management of MQTT devices.** *2015 IEEE Conference on Wireless Sensors (ICWiSe).* 2015; pp. 12–17.
Publisher Full Text

38. MQTT org Graphics: accessed: 2026-04-02.
Reference Source

Open Peer Review

Current Peer Review Status: ?

Version 1

Reviewer Report 18 August 2026

https://doi.org/10.21956/openreseurope.26166.r79212

© **2026 Zhong R.** This is an open access peer review report distributed under the terms of the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

?

Ruirui Zhong
Zhejiang University, Hangzhou, China

This manuscript presents LEAF, an open-source, modular Python framework designed to connect heterogeneous laboratory equipment with downstream storage, monitoring, and analysis infrastructures. The motivation is timely because laboratory automation increasingly requires interoperable, continuously collected, machine-readable data streams. The manuscript clearly describes LEAF's high-level architecture, including EquipmentAdapters, reusable modules, event-to-process mapping, configuration mechanisms, MQTT/KeyDB/file outputs, and InfluxDB line protocol.

The software appears practically relevant and potentially useful for laboratory digitalization. The availability of source code, documentation, PyPI distribution, licensing information, and archived source code is also a significant strength. However, the present article is considerably stronger as a software description and vision paper than as a scientifically validated software-tool study. In particular, quantitative evaluation, reproducible demonstrations, comparative benchmarks, failure analysis, and representative output datasets are insufficiently developed.

Therefore, I consider the manuscript promising but requiring major revision before it can be regarded as scientifically sound and sufficiently reproducible.

1. The manuscript should include quantitative validation across representative devices, reporting acquisition latency, throughput, data-loss rates, CPU/memory consumption, uptime, and recovery performance under realistic laboratory operating conditions.
2. LEAF is positioned relative to proprietary tools and several Python DAQ packages, but no systematic comparison is provided. Benchmark functionality, extensibility, supported protocols, deployment complexity, performance, and interoperability against representative alternatives.
3. Please provide one complete reproducible example covering equipment connection, configuration, event detection, transformation, transmission, storage, and resulting output records, including exact software versions, commands, configuration files, and expected outputs.
4. The manuscript states that LEAF produces semi-standardized outputs, but the semantic

mapping strategy remains insufficiently formalized. Define mandatory fields, metadata conventions, units, naming rules, validation procedures, and adapter-specific extensibility mechanisms.

5. The authors describe LEAF as flexible and scalable, yet scalability is not measured. Please evaluate concurrent adapters, increasing event frequencies, multiple instruments, broker loads, and resource consumption to determine practical deployment limits.
6. Several Discussion statements regarding failure prediction, optimization, reproducibility, digital twins, and adaptive control are plausible but not experimentally demonstrated. Clearly separate validated capabilities, potential applications, and planned future developments.
7. The literature review identifies relevant automation, FAIR, IoT, and digital-twin studies, but should more critically compare LEAF with existing middleware, laboratory integration frameworks, device abstraction approaches, and industrial communication architectures.

Is the rationale for developing the new software tool clearly explained?
Yes

Is the description of the software tool technically sound?
Partly

Are sufficient details of the code, methods and analysis (if applicable) provided to allow replication of the software development and its use by others?
Partly

Is sufficient information provided to allow interpretation of the expected output datasets and any results generated using the tool?
Partly

Competing Interests: No competing interests were disclosed.

Reviewer Expertise: My research expertise is primarily in intelligent systems, automation, and IoT for automated systems. I am well positioned to assess the software framework, system architecture, interoperability, methodology, and validation aspects of this article.

I confirm that I have read this submission and believe that I have an appropriate level of expertise to confirm that it is of an acceptable scientific standard, however I have significant reservations, as outlined above.