WOCC 24 IBIS An Infrastructure Management

Preprint · May 2024

DOI: 10.13140/RG.2.2.11758.22082

Alicia Esquivel Morel
University of Missouri
18 PUBLICATIONS **56** CITATIONS
SEE PROFILE

Kate Keahey
University of Chicago
147 PUBLICATIONS **5,848** CITATIONS
SEE PROFILE

Jianfeng Zhou
University of Missouri
107 PUBLICATIONS **1,546** CITATIONS
SEE PROFILE

Prasad Calyam
University of Missouri
232 PUBLICATIONS **2,805** CITATIONS
SEE PROFILE

IBIS — An Infrastructure Management Framework for Adaptable, Multi-Sensor Data Collection in Scientific Research ⋆

Alicia Esquivel Morel[1], Mark Powers[2], Kate Keahey[3], Zack Murry[1], Tomas Javier Sitzmann[4], Jianfeng Zhou[1], and Prasad Calyam[1]

[1] University of Missouri, Columbia MO 65201, USA
[2] Argonne National Laboratory, Lemont IL 60439, USA
[3] University of Chicago, Chicago IL 60637, USA
[4] University of Turin, Grugliasco TO 10095, Italy

Abstract. Multi-cloud environments integrated with various Internet of Things (IoT) have resulted in a diverse range of observational instruments that can be used and adapted for various use cases. An important role of these observational instruments in scientific research is to provide valuable data that can be understood and help solve practical problems. The evolution of `FLOTO`, an observational instrument initially designed for bandwidth measurement, has led to its adaptation for diverse multi-sensor applications. Leveraging its support for mainstream Single Board Computers (SBCs), it facilitates the deployment and operation of scientific instruments that enable data collection and sharing among different user groups. This paper explores the transition of `FLOTO` from its original purpose to `IBIS` an infrastructure management framework capable of integrating multi-sensor technologies, including environmental sensors and cameras for data acquisition and analysis. `IBIS`, inspired and named after the perceptive bird known for its keen eyesight, embodies the essence of a multi-sensor observational instrument designed for scientific discovery. We present a reference operation example that provides practical insights into implementing `IBIS`-based instruments for optimizing greenhouse environments with precision agriculture. Furthermore, we showcase how this application can be seamlessly integrated into the `IBIS` framework, allowing users to deploy and operate varied instruments in diverse environments. Lastly, this work provides guidelines for reproducibility, contributing to `IBIS'` documentation and fostering community accessibility.

Keywords: Multi-sensor data acquisition · Observational instrument · Edge computing · Reproducibility.

⋆ Results presented in this paper were obtained using the FLOTO project (Award 2213821) supported by the National Science Foundation. This material is based upon work supported by the U.S. Department of Energy, Office of Science, under contract number DE-AC02-06CH11357.

1 Introduction

The rapid development of the Internet of Things (IoT) has revolutionized how data is collected. IoT devices can be integrated with multi-sensor and Single Board Computers (SBCs) and transformed into observational instruments for scientific exploration. In addition, these observational instruments can be integrated within multi-cloud environments, aiming researchers to gather valuable and larger numbers of data streams from a wide range of sources. Thus, enabling the connectivity between these *physical things* and the *Internet* makes it possible to remotely access any data and control this physical environment [20]. The evolution of `FLOTO` [27, 18], a *Discovery Testbed and Observational Instrument*, initially designed for bandwidth research, has led to its adaptation for a diverse range of multi-sensor and multi-cloud applications.

`FLOTO` is an observational instrument that facilitates the deployment and management of widely used low-cost SBCs for large-scale data collection in field deployments. These scientific and observational instruments can be deployed for data acquisition and allow shared operation in remote areas without physical access or intervention. Furthermore, by enabling multi-tenant sharing among various applications and user groups, `FLOTO` aims for researchers to collaborate and unlock the full potential of the collected data. Real-time monitoring of air quality across urban landscapes, tracking movements of endangered species, or even measuring subtle fluctuations in atmospheric pressure exemplify how its capabilities can facilitate data collection for a nuanced understanding of diverse phenomena. A key question driving this research is whether `FLOTO` can be adapted to measure phenomena beyond its original focus on broadband. To achieve this, we propose adapting it into a flexible observational instrument. This adaptation involves seamless integration and deployment with a wider range of sensor peripherals. These peripherals could encompass environmental sensors, motion and position trackers, biometric monitors, imaging devices, and more. Additionally, it can be equipped to provide meaningful analysis of these new measurements, allowing researchers to analyze phenomena previously outside its scope.

This paper explores the evolution of `FLOTO` from its initial design and purpose to `IBIS` an adaptable platform capable of integrating multi-sensor technologies, including environmental sensors and cameras for data acquisition and analysis. `IBIS`, inspired and named after the perceptive bird known for its keen eyesight, embodies the essence of a multi-sensor observational instrument designed for scientific discovery. Just as the Ibis utilizes its senses to navigate its environment and gather valuable information, the `IBIS` instrument empowers researchers with critical insights through gathering and interpreting information from its environment. Prototype demonstration will highlight its ability to seamlessly integrate various sensor types, and effectively address diverse data collection scenarios. Finally, we illustrate `IBIS'` functionality in real-world settings and propose a methodology for reproducibility in "edge-to-cloud" experiments in a way that promotes community accessibility. The remainder of this paper is organized as follows: in Section 2, we discuss the approach and design of this edge-based observatory instrument. In Section 3, we present IBIS and its adaptation for an optimized greenhouse environment with precision agriculture use case. Section 4 presents application deployment and data collection. Section 5 presents the reproducibility aspect, and Section 6 the related work. Lastly, Section 7 concludes the paper and provide some insights for future work.

2 Approach

IBIS is an infrastructure management framework underlying the FLOTO project which deploys a thousand Raspberry Pi devices nationwide to measure the quality of broadband [18]. In this section, we first summarize how IBIS works, and then describe how it can be extended to support applications beyond broadband measurement, in particular applications that require combining compute capability at edge supplied by the SBCs with sensing abilities provided by a range of IoT peripherals.

2.1 IBIS: An Observational Instrument

The IBIS infrastructure [18, 8] implements a general *observational instrument* pattern where a large number of *observation points* can be deployed and managed to conduct observation, and then report data resulting from this observation to a central *aggregation point* where the data can be collected, combined, and processed. In the current IBIS implementation, the observation points are implemented as single board computers (SBCs), cost-effective solutions that are lightweight enough to support large deployment scales, yet powerful enough to provide sufficient cycles for observation. The infrastructure supports easy deployment of such observation points and organization into fleets composed of hundreds of devices that can be reliably managed over time without requiring physical access to any device.

The devices are monitored and managed via a dashboard that displays device information, including device profiles, performance statistics, which applications are running on a given device, and allows operators to execute device-specific actions. To provide the actual observation function, IBIS supports the deployment of containerized *observing applications* on selected groups of devices. These applications interact with the environment to capture and report on relevant phenomena. For example, the FLOTO project applications consist of different types of broadband tests that measure and report on broadband quality at the deployment site. Each observing application generates data that represents the result of its observation. This data is then uploaded to an aggregation point by a *data uploader application* at which point it can be stored, combined, or processed.

2.2 Adapting IBIS

We point out in [18] that, like any scientific instrument, IBIS has the potential to be adapted to answer different scientific questions by varying its deployment scope, adapting it to observe diverse phenomena by coupling it with appropriate sensors or running custom applications, and using tailored data aggregation techniques. For example, instead of measuring and reporting on the quality of broadband by running broadband tests, we can observe and report on wildlife sightings similarly to what was done in [32], or use distributed learning [14] to train models locally on protected biometric data and send those models to an aggregator that combines them rather than only preserving and managing access to data. `IBIS` supports such adaptation by allowing the user to customize three qualities. First, a user can equip the SBCs to support the desired observational function on a hardware level. This can be done, for instance, by establishing an Ethernet connection to a router to measure broadband or attaching a camera to enable visual analysis. Second, a user needs to run an application that implements the desired observational function, e.g., runs broadband tests, performs image recognition tasks, or trains a model based on observed images. Third, a suitable data aggregator can be created to process the reported data, which can be as simple as storing it or the aggregation can involve a data processing step as in averaging gradients in learning models. The sections below discuss how `IBIS` supports these adaptation actions.

2.3 `IBIS` User Workflow

Hardware customization. Raspberry Pi, the principal SBC that `IBIS` currently supports, offers a flexible peripheral connection system. The USB ports provide compatibility with familiar peripherals, while a dedicated ribbon cable port is available for dedicated camera modules. However, the true versatility lies in the General-Purpose Input/Output header (GPIO) — a 40-pin connector on all recent Raspberry Pi models. This header provides voltage and ground pins for powering circuits and general-purpose pins for two-way communication. Users can collect data from a wide range of sensors by interfacing with these pins.

Some pins even support specialized protocols like I2C, SPI, UART, and PCM, which are crucial for certain sensors to communicate effectively with the Raspberry Pi. These require a secure connection to the correct pins for proper function. The pins can be configured for peripheral-specific protocols in the OS hardware configuration file (`/boot/config.txt`). To simplify complex GPIO connections, Raspberry Pi offers Hardware Attached on Top (HATs), self-contained modules that stack directly onto the device, utilizing all GPIO pins. This modular approach allows for easy expansion of the Raspberry Pi's capabilities.

Adding Devices with IoT peripherals to IBIS. The initial step in peripheral installation involves following the manufacturer's instructions for installing the device on a Raspberry Pi, e.g., plugging it into a USB port, connecting a camera cable to the camera serial port, attaching a HAT to the GPIO pins, or connecting wires to the corresponding GPIO pins. Next, the peripheral must be registered with `IBIS`. In principle, `IBIS` can support any Raspberry Pi-compatible peripherals that can be accessed via Linux device interfaces at the software level. In practice, with IoT experiencing rapid innovation and new peripherals becoming available every day, we can only provide out-of-the-box support for a limited

set of such peripherals — the most common ones, covering the typical use cases — and provide instructions for users to develop their custom support to cover remaining and emergent use cases using existing drivers.
Supported interfaces currently include the Raspberry Pi camera module, access to the analog and digital I/O pins of the GPIO subsystem, and the SPI and I2C serial interfaces. To enable access via those interfaces, device operators must submit at least enough metadata so that application developers know how to access a peripheral, e.g., "device /dev/i2c-1 must be mounted into the container, and the sensor accessed at address 0x78, using a documentation link or library". To extend the scope of the system beyond these use cases, we provide documentation that allows users to extend this support. Briefly, the OS uses "device trees" — a data structure that describes the hardware components to map physical interfaces to Linux devices under `/dev` and load relevant kernel modules; if necessary the user can use the meta-data to specify additional device trees or parameters to load at boot (`IBIS` propagates this to the device's config.txt file), in addition to the information for application developers on how to access the interface once presented via the OS.

Application Development. Developing a new application consists of developing a Docker container containing an application capable of interacting with the peripheral using the information provided in the meta-data. Since `IBIS` itself supports primarily production capability, we recommend that this development is done on CHI@Edge [16], an edge testbed of the Chameleon project [17, 4] which supports the same device and peripheral model and uses container deployment to reconfigure devices in a way similar to `IBIS`. After deploying the container, the user mounts the relevant interfaces from `/dev` (created by the OS loading device trees and kernel modules on boot) and then communicates to those interfaces using the information in meta-data, usually by leveraging standardized libraries already built into the Raspberry Pi such as the `libcamera` driver and Sense HAT libraries.

Data Collection. Lastly, the user can pair the application with a data uploader to transfer the collected data from the device to a central collection and processing point. To do this, `IBIS` offers an `rclone`-based default uploader application that connects to a data aggregation service currently running on the Chameleon Cloud [17] (An open experimental testbed for Computer Science funded by the National Science Foundation) and collects data from all `IBIS` deployments by default. Alternatively, the user can clone or modify the data uploader to connect to their own data collection facilities, such as a public cloud, or provide an upload capability more suitable to their deployment conditions (e.g., weighing power efficiency against transfer efficiency). For portability, the application is configured to honor environment variables passed to it at runtime, these variables are used to customize the operation for a particular user, such as to specify which S3 bucket to upload data to, or how often to take samples. When a job is executed on `IBIS`, these variables are packaged along with the containers to which they should be applied.

3 Case Study, IBIS to the Test: Optimizing Greenhouse Environments with Precision Agriculture

In this section, we showcase a use case study where we have adapted IBIS as an edge-based instrument for a specific research scenario, i.e., *precision agriculture*. This case study will explore the different IBIS components used and how they were configured to address a real-world scientific challenge. This use case study showcases IBIS by focusing on its applicability in optimizing greenhouse environments through precision agriculture.

3.1 Precise Environmental Control

Precision agriculture optimizes crop yield and growth in greenhouse production systems. These techniques, consisting of multi-sensor data collection, analysis, and decision-making systems, enable holistic control and management of crops according to the dynamics of environments, like temperature, humidity, CO_2 levels, light intensity, soil moisture, and air quality. All these factors are essential for optimal plant growth [6] and, traditionally, monitoring these parameters has been labor-intensive [5], and subject to visual observations. IBIS can simplify greenhouse management by integrating sensors that can continuously collect data on various aspects of plant health and the surrounding environments, further enhancing this application with remote sensing, and providing real-time, high-resolution data on plant parameters.

The measured plant characteristics can include plant height, leaf area index, chlorophyll content, and stress levels [24]. With the comprehensive data of plants and environments from sensors, farmers can precisely monitor and analyze plant development and health conditions across the entire greenhouse, and make optimal management decisions [29]. This enhances timely interventions to maximize yield and quality while reducing resource input. For instance, accurate measurements of soil moisture, temperature, and light intensity allow for adjustments to irrigation schedules, optimization of nutrient delivery, and mitigation of environmental stressors. Thus, these combined approaches lead to improved crop performance and resource efficiency.

3.2 Transformation to Edge and Leveraging Remote Sensing

By leveraging the IBIS infrastructure management framework as an edge-based observatory instrument, farmers can analyze data in real time and make informed decisions about irrigation, ventilation, and other environmental controls. This data-driven approach can aid farmers in cultivating diverse crops across different climates and seasons, ultimately leading to improved yields and resource efficiency. **Figure 1** depicts the general architecture for leveraging sensing in greenhouse environments with precision agriculture. Adapting IBIS to leverage remote sensing involves deploying a strategic network of sensors throughout

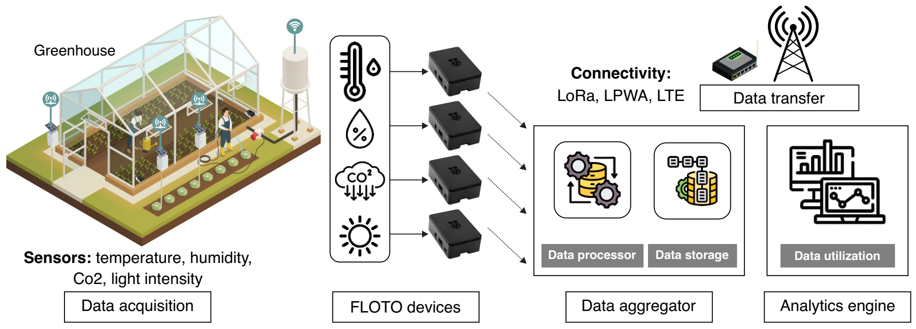

Fig. 1. General architecture for leveraging remote sensing in greenhouse environments with precision agriculture.

the greenhouse. The number and distribution of these sensors depend on several factors including greenhouse size, required spatial resolution, environmental variation, and types of measurements [25, 31].

For instance, larger greenhouses may require more sensors for fine-grained measurements in different areas. Additionally, the crop type must be considered, as some crops may have specific data requirements, and sensor parameters must be selected to capture specific variables that are crucial to the crop's health and growth [23]. It is also important to capture soil moisture variations across the greenhouse, considering different locations and depths within the soil profile to run tests that can measure the moisture at different depths. This can help to correlate temperature and moisture to evaluate the utility of a sensor at one unique depth. This can aim to create plant trial setups for testing biowaste materials in organo-mineral fertilizers [26]. These sensors are connected to the `IBIS` devices equipped with apps for data processing and communication.

Reliable communication plays a critical role in real-time data processing and analysis. The sensor data can be transmitted through wireless communication technologies like WiFi or Zigbee, or long-range ones, like LPWAN, NB-IoT, LoRa, or LoRaWAN [2]. The application is designed to continuously collect and transmit data from the sensors to the `IBIS` devices and, consequently, to the data aggregator. The collected data requires filtering to remove unnecessary data points for efficient analysis, in addition to performing basic data analytics to prepare it for further processing. Sensor readings must be filtered to eliminate potentially erroneous readings. Once processed, the data is transmitted to a central analytics engine or cloud platform for in-depth analysis and storage. For example, the collected imagery data can be processed to build orthomosaic images and point cloud data of target plans, through a more holistic high-resolution spatial and spectral reflectance information of each plant, and its organs [30]. In addition, various analytics and machine learning models can be applied to extract meaningful insights and patterns. For instance, inference models can be developed to correlate plant growth parameters with environmental factors and weather predictions, optimizing irrigation scheduling, nutrient management, and pest control strategies [1]. Leveraging advanced analytics allows for further enhancement of crop productivity, resource efficiency, and sustainability.

4 Application Deployment and Data Collection

Adaptations to the hardware. The chosen application for this prototype deployment focuses on monitoring several key environmental factors that are crucial for optimal plant growth. The initial hardware setup needs to be adapted based on the application's needs. For our prototype, illustrated in **Figure 2**, we rely on the Google Coral Environmental Sensor Board board, an add-on board with sensing capabilities.

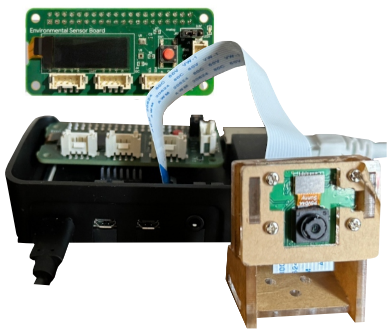

Fig. 2. `IBIS` hardware adaptation.

This board provides atmospheric data such as light level, barometric pressure, temperature, and humidity, and it is designed to work with the 40-pin GPIO header. In addition, we added a Raspberry Pi mini camera module with a fixed focus and the provided mount. The camera communicates through the Camera Serial Interface (CSI) with the libcamera library and Picamera2 to capture images. Continuous data collection from the Google Coral Environmental Sensor Board provides real-time updates on the environmental metrics mentioned above. Image capture with the camera is triggered at a preset interval, capturing images every 30 minutes. These captured images can be used to for example, to measure plants' health. To power the device and sensors reliably, they were connected to an AC power source, with surge protectors implemented for each connection to ensure uninterrupted operation.

Challenges in Hardware Adaptation:

1. More complex peripherals for the Raspberry Pi make assumptions about the OS's inclusion and auto-detection of device trees and kernel modules, such as the "Camera Module 3" needing a very new kernel for auto-detection, or the "Coral sense HAT" which includes a custom device tree onboard in flash memory and expects the Raspberry Pi to load it automatically. Since `IBIS` devices do not run the Raspberry Pi OS distribution, supporting the firmware for these devices can be complex and error-prone.
2. `IBIS` currently uses a "one size fits all" networking model, where containers have access to a private, per-device network, and reach the outside world through NAT and the device's routing table. This makes it difficult to support experiments such as comparing Wi-Fi and Ethernet performance on the same device or running an experiment across two devices, using one as a source, and the other as a sink for the traffic. More control of how container

network traffic is mapped to the physical network interface of a device is an important feature to add.

Precision Agriculture Application. At the edge level, the application performs minimal data processing on the sensor data. Sensor readings like temperature, ambient light, and humidity can be averaged over a specific window for noise reduction before transmission. Captured images might undergo a basic resizing process to reduce storage requirements. The collected sensor data (including averaged readings for temperature and humidity) and captured images are periodically transmitted from the IBIS devices to a central server hosted on a cloud platform. On the server, the data is further analyzed using a data visualization dashboard. This dashboard displays real-time and historical trends for all collected metrics, allowing for comprehensive monitoring of environmental conditions. Captured images are stored on the server and can be manually inspected for visual signs of plant stress, such as discoloration. Additionally, software-based image analysis techniques could be explored in the future for automated stress detection. The data can be analyzed and visualized and generic analysis tools, including open-source and commercial options e.g. Grafana [3], can offer user-friendly interfaces for building dashboards.
Data Collection. The IBIS devices are programmed to collect sensor data at a user-defined frequency. A high-frequency collection rate (e.g., every minute) allows for detailed monitoring of environmental fluctuations within the greenhouse. IBIS provides a customizable data uploader container [9] that can be used in such multi-container applications. This built-in customizable data uploader container allows any data placed in this container to be uploaded to a designated cloud storage location. For this prototype, we rely on Chameleon Cloud [17] and its object store [19]. It is also important to highlight that this process can be adapted to store any other generic data from other types of sensors, and it can be easily configured to target commercial cloud back-ends.

5 Reproducibility

Our work prioritizes artifact reproducibility by employing the IBIS application store, a dedicated repository for sharing research workflows. We also consider *practical reproducibility* [15], supporting accessible, integrated, and reusable experiments, represented as a combination of hardware, experimental environment, experimental body, and data analysis. The example prototype described in this paper (gather environmental data situated within a greenhouse environment) is publicly available on the IBIS website [10]. We consider three levels of experiment reproduction within the IBIS infrastructure; first, users can achieve an **exact replication** by deploying the application provided in the public listing of our containers and applications on the IBIS dashboard, utilizing the same sensor setup (Raspberry Pi model and sensor types) within the IBIS environment and following the instructions provided in the getting started documentation [11]. This approach guarantees the most comparable results to the original experiment. In addition, the IBIS infrastructure's flexibility allows users to leverage applications while employing different compatible sensors to collect environmental data. While the core functionalities (data collection and analytics) remain the same, the specific sensor data might differ due to varying sensor characteristics. This **variation approach** can provide valuable insights into how the experiment behaves with different sensor types. Lastly, **further exploration** would be beyond replicating the experiment. Users can potentially modify the application's source code to capture additional sensor data or implement different data processing techniques within the `IBIS` framework. While `IBIS` promotes reproducibility, one potential challenge users might face is access to a Raspberry Pi and compatible sensors for an exact replication. This challenge can be overcome by emphasizing the importance of detailed and comprehensive documentation for applications and their deployment. Clear instructions, including a thorough explanation of the application's purpose and functionality, ensure clarity for users even with potentially different sensor setups. Additionally, the documentation should provide detailed descriptions of the sensor data collected and the averaging process. Guidance on how to interpret the results and potential considerations for variations in sensor setups should also be included.

6 Related Work

The convergence of multi-cloud environments and the Internet of Things (IoT) has led to a rise in observational instruments capable of collecting diverse data for various applications [12, 28]. This aligns with the growing emphasis on leveraging sensor data to address real-world challenges and advance scientific research[22]. Several existing technologies address the challenge of managing large-scale deployments of devices and multi-sensors for data collection and analysis. Cloud-based platforms like AWS IoT Greengrass [13], and Azure IoT Edge [7], offer user-friendly interfaces and robust functionalities for device management, application deployment, and data integration with cloud services. However, these platforms can lead to vendor lock-in and ongoing costs. Open-source alternatives like OpenBalena [21] fleet management provide flexibility and control but do not support multi-tenancy and do not handle scalability and back-end services. We leverage this open-source fleet management and extend it with features for scaling and user experience: multi-tenant usage, ad-hoc shell commands, device collections, and a device dashboard. `IBIS` exemplifies a implementation of an "observational instrument", based on zero-touch installation, automated management, cloud integration, and data collection from multi-sensors.

7 Conclusions and Future Work

The Internet of Things (IoT) has revolutionized how data can be collected and processed. In addition, Single Board Computers (SBC) can be integrated with multi-sensors and transformed into observational instruments for scientific exploration. This paper presents the transition of `FLOTO` from its original purpose to IBIS an infrastructure management framework capable of integrating multi-sensor technologies. The seamless integration with various sensors and the ability to analyze this new data addresses a key research question: Can FLOTO be adapted to measure phenomena beyond broadband? This paper demonstrates IBIS's adaptability and effectiveness through a real-world precision agriculture application, showcasing IBIS's potential to enhance data collection methodologies. Lastly, the proposed reproducibility methodologies are based on practical reproducibility, promoting open access within the research community. Future work includes more advance sensor integration, and broaden the application scope of this work. It can include implementing advanced data analysis like Machine Learning and visualization tools. In addition, a more flexible approach that can allow control over container network traffic will be considered.

References

1. Abioye, E.A., Hensel, O., Esau, T.J., Elijah, O., Abidin, M.S.Z., Ayobami, A.S., Yerima, O., Nasirahmadi, A.: Precision irrigation management using machine learning and digital farming solutions. AgriEngineering **4**(1), 70–103 (2022)
2. Ali, A., Hussain, T., Tantashutikun, N., Hussain, N., Cocetta, G.: Application of smart techniques, internet of things and data mining for resource use efficient and sustainable crop production. Agriculture **13**(2), 397 (2023)
3. Chakraborty, M., Kundan, A.P.: Grafana. In: Monitoring cloud-native applications: Lead agile operations confidently using open source software, pp. 187–240. Springer (2021)
4. Chameleon: Chameleon. https://www.chameleoncloud.org/, accessed: 2024-03-07
5. Charania, I., Li, X.: Smart farming: Agriculture's shift from a labor intensive to technology native industry. Internet of Things **9**, 100142 (2020)
6. Chaudhary, D., Nayse, S., Waghmare, L.: Application of wireless sensor networks for greenhouse parameter control in precision agriculture. International Journal of Wireless & Mobile Networks (IJWMN) **3**(1), 140–149 (2011)
7. Edge, A.I.: Build the intelligent edge. https://azure.microsoft.com/en-us/products/iot-edge, accessed: 2024-01-13
8. FLOTO: Floto. https://floto.cs.uchicago.edu/, accessed: 2024-02-14
9. FLOTO: Floto. https://github.com/UChicago-FLOTO/data_uploader, accessed: 2024-02-14
10. FLOTO: Floto application. https://floto.cs.uchicago.edu/applications/, accessed: 2024-03-07
11. FLOTO: Floto dashboard. https://portal.floto.science/dashboard/, accessed: 2024-02-14
12. Hassan, R., Qamar, F., Hasan, M.K., Aman, A.H.M., Ahmed, A.S.: Internet of things and its applications: A comprehensive survey. Symmetry **12**(10), 1674 (2020)
13. IoT, A.: Build intelligent iot devices faster. https://aws.amazon.com/greengrass/, accessed: 2024-01-13
14. Jiang, J.C., Kantarci, B., Oktug, S., Soyata, T.: Federated learning in smart city sensing: Challenges and opportunities. Sensors **20**(21), 6230 (2020)
15. Keahey, K., Anderson, J., Powers, M., Cooper, A.: Three pillars of practical reproducibility. In: 2023 IEEE 19th International Conference on e-Science (e-Science). pp. 1–6. IEEE (2023)

16. Keahey, K., Anderson, J., Sherman, M., Zhen, Z., Powers, M., Brunkan, I., Cooper, A.: Chameleon@ edge community workshop report (2021)
17. Keahey, K., Anderson, J., Zhen, Z., Riteau, P., Ruth, P., Stanzione, D., Cevik, M., Colleran, J., Gunawi, H.S., Hammock, C., Mambretti, J., Barnes, A., Halbach, F., Rocha, A., Stubbs, J.: Lessons learned from the chameleon testbed. In: Proceedings of the 2020 USENIX Annual Technical Conference (USENIX ATC '20). USENIX Association (July 2020)
18. Keahey, K., Feamster, N., Martins, G., Powers, M., Richardson, M., Schrubbe, A., Sherman, M.: Discovery testbed: An observational instrument for broadband research. In: 2023 IEEE 19th International Conference on e-Science (e-Science). pp. 1–4. IEEE (2023)
19. Keahey, K., Riteau, P., Stanzione, D., Cockerill, T., Mambretti, J., Rad, P., Ruth, P.: Chameleon: a scalable production testbed for computer science research. In: Contemporary High Performance Computing, pp. 123–148. CRC Press (2019)
20. Kopetz, H., Steiner, W.: Internet of things. In: Real-time systems: design principles for distributed embedded applications, pp. 325–341. Springer (2022)
21. OpenBalena: Open source software to manage connected iot devices at scale. https://www.balena.io/open, accessed: 2024-01-13
22. Qiu, S., Zhao, H., Jiang, N., Wang, Z., Liu, L., An, Y., Zhao, H., Miao, X., Liu, R., Fortino, G.: Multi-sensor information fusion based on machine learning for real applications in human activity recognition: State-of-the-art and research challenges. Information Fusion **80**, 241–265 (2022)
23. Rayhana, R., Xiao, G., Liu, Z.: Internet of things empowered smart greenhouse farming. IEEE journal of radio frequency identification **4**(3), 195–211 (2020)
24. Ru, C., Hu, X., Wang, W., Ran, H., Song, T., Guo, Y.: Evaluation of the crop water stress index as an indicator for the diagnosis of grapevine water deficiency in greenhouses. Horticulturae **6**(4), 86
25. Rustia, D.J.A., Lin, C.E., Chung, J.Y., Zhuang, Y.J., Hsu, J.C., Lin, T.T.: Application of an image and environmental sensor network for automated greenhouse insect pest monitoring. Journal of Asia-Pacific Entomology **23**(1), 17–28 (2020)
26. Sitzmann, T.J., Sica, P., Grignani, C., Magid, J.: Testing biowaste materials as peat replacement in organo-mineral fertilizers. Frontiers in Sustainable Food Systems **8**, 1330843
27. Sundaresan, S., Burnett, S., Feamster, N., De Donato, W.: {BISmark}: A testbed for deploying measurements and applications in broadband access networks. In: 2014 USENIX Annual Technical Conference (USENIX ATC 14). pp. 383–394 (2014)
28. Vermesan, O., Friess, P., Guillemin, P., Sundmaeker, H., Eisenhauer, M., Moessner, K., Le Gall, F., Cousin, P.: Internet of things strategic research and innovation agenda. In: Internet of things, pp. 7–151. River Publishers (2022)
29. Zhang, J., Huang, Y., Pu, R., Gonzalez-Moreno, P., Yuan, L., Wu, K., Huang, W.: Monitoring plant diseases and pests through remote sensing technology: A review. Computers and Electronics in Agriculture **165**, 104943 (2019)
30. Zhou, J., Nguyen, H.T.: High-Throughput Crop Phenotyping. Springer (2021)
31. Zhou, S., Mou, H., Zhou, J., Zhou, J., Ye, H., Nguyen, H.T.: Development of an automated plant phenotyping system for evaluation of salt tolerance in soybean. Computers and Electronics in Agriculture **182**, 106001 (2021)
32. Zualkernan, I., Dhou, S., Judas, J., Sajun, A.R., Gomez, B.R., Hussain, L.A.: An iot system using deep learning to classify camera trap images on the edge. Computers **11**(1), 13 (2022)