Comparing the Use of Custom-built and Commercial Off-the-shelf Data Gathering Devices in IoT Systems

Mika Saari*, Janne Harjamäki*, Mikko Nurminen*, Petri Rantanen*
* Tampere University / Faculty of Information Technology and Communication Sciences, Pori, Finland
mika.saari@tuni.fi

***Abstract*—IoT refers to a wide range of devices that have Internet access, and collect and transmit information. Data have to be gathered from the most diverse environments. Custom data gathering devices are often necessary in situations where the data being collected are unique or need to be collected in a certain way. This is because commercial off-the-shelf (COTS) products may not have the necessary capabilities or features to meet the specific requirements of the data collection. This study focuses on COTS and self-built data gathering devices by comparing two data gathering prototype systems: The first designed and implemented from scratch and the second built with COTS components. Both systems are compared in terms of various features such as cost-effectiveness, time saving, and the usefulness of the output. The end goal of the study is to provide a proposal for when it is appropriate to use COTS components and when it is necessary to design and build devices yourself.**

Keywords*—IoT, Internet of Things, Off-the-shelf, Data Gathering

I. INTRODUCTION

There are already plenty of commercial Internet of Things (IoT) products for data collection on the market. This has also raised the issue of whether it is worth building sensor packages yourself if commercial products are available. This paper looks at how ready-to-use components have been utilized in IoT research.

This research was part of a research project, which focused on energy saving in Finnish apartments. The goal of project was to improve the energy efficiency of housing with the help of data collection. As a result, the objective was to produce information and tools for this purpose. The research project implemented prototype systems for data collection and analysis. This paper focuses on three different data collection device constructions: a prototype system based on self-made components and a prototype system based on commercial off-the-shelf components and their combination. Both systems partly use the same software components, so the information they provide to the user is very similar. We tried to keep the data storage and visualization applications unchanged; these are presented in [1] and [2].

This research does not aim to find out whether a self-built device is better than a commercial one, but to determine what kind of advantages are achieved by using commercial products as part of the system. Therefore, the research question is formulated as follows:

RQ: Is it reasonable to use off-the-shelf devices instead of self-made sensor nodes in IoT data gathering?

The research question will be resolved by presenting the general architecture of the data collecting system. The most important components, their purpose, and mode of operation are described in the model. In addition, the collected environmental data and how the collected data are presented are described.

For the actual comparison of the commercial and non-commercial systems, two implemented real prototype systems are used. The main parts and structure of the systems are explained. In addition, the data they collect (input), and the outgoing data (output) are described.

The systems are compared using the following criteria: cost-effectiveness, time saving, and the usefulness of the output. Also, additional features are considered. Finally, the conclusions are presented.

The structure of this paper is as follows: In Section II, we present the background including the related research. In Section III, we present the abstract data gathering model and three pilot cases of real world data gathering prototype system implementations. In the discussion section, Section IV, we expand on the findings of the comparison of the pilot cases and summarize the study in Section VI.

II. BACKGROUND

A survey [3] conducted in 2002 collected the eight basic design factors of a sensor network. For the present study, the most notable of these factors are scalability, production costs, hardware constraints, and transmission media. Design factors can be used as a guide when developing data gathering prototype systems such as those presented in this study.

Common properties of IoT systems include component-based design. From the software engineering area, one approach is Component-Based Software Engineering (CBSE), which contains features needed in IoT systems, i.e., composability, deployability, comprehensive documentation, independence, and standardization [4]. We have researched the usage of CBSE in relation to IoT in an earlier study [5].

Several studies compare COTS components with self-made components. One study [6] uses scalability, cost

efficiency, and performance as criteria when comparing smart home sensor devices with the developed low-power, fuzzy-based control method. Another study [7] also focuses on low-cost and low-power underwater prototype device development. Further, our earlier study [8] showed that the use of COTS devices is quite extensive.

This study is based on research conducted during the KIEMI project. The goal of the KIEMI project was to save energy, and we worked towards this goal by developing and constructing data gathering IoT sensor systems. During the project a total of 23 different types of pilot cases were carried out related to the energy efficiency and condition measurement of properties. The pilot cases (in chronological order: #03, #08, and #19) mentioned in this study were made in the KIEMI project. The overall results of the project are reported in [9].

III. Data gathering IoT prototype systems

The off-the-shelf hardware components can be presented as follows: For the lowest level we chose resistors, microprocessors, and other components, and built the circuit board ourselves. Also, the software was self-built. For the mid level we chose off-the-shelf devices such as Raspberry Pi or Arduino, and selected the necessary sensors and made the necessary connections between them. The software was partially ready-to-use, for example: Raspbian OS. Self-made software is also needed for connecting the selected hardware so that it works together. The highest level is to buy reasonable priced devices with reasonable software and use them.

The abstract architecture of the data gathering prototype system consists of three main parts, which are presented in Figure 1.

1) Data gathering device, dedicated to gathering data, which could be simple numerical data ("SN_n nodes" in Figure 1) or Big data chunks such as image data (marked Master & Sensor node(s)" in the figure).
2) Communication (data arrows in Figure 1) is chosen for data transfer from device to database. The appropriate communication channel should be selected according to the type of data.
3) Database ("DB" in Figure 1) is responsible for storing and presenting the collected data. The analyzing of the data could be also performed here.

These are the main parts of the systems; next we present three different implementations: The first one is mid level, and uses self-made hardware and software. The second one is at the highest level and uses more COTS components. The third one is a hybrid, where COTS sensors were used with a self-made master node. All three use case descriptions contain reflections about their advantages and disadvantages. The systems are considered using the following criteria: cost-effectiveness, time saving, and the usefulness of the output. Additional qualifications characteristics such as CBSE criteria are also considered.

A. Prototype with self-made sensor nodes

In pilot case #03 (Figure 2), a prototype system was constructed to measure and gather environmental data such as temperature, humidity, pressure (BME680 sensor), carbon monoxide (eCO2-sensor), and air quality (SGP30) from residential apartments with two sensor nodes. The sensor nodes sent data to the master node, which sent the data to the database. This system had no complex visualization; the data were presented on the worksheet in numerical form. The sensor nodes were based on the Sodaq ExpLoRer[1] to which the sensors were connected. Sodaq has the capability to use LoRaWan[2] radio technology. The master node was a Raspberry Pi with a LoRa shield and was connected to the Ethernet network using a 3G/4G base station. With this prototype, the commercial LoRaWAN provider was used as a test case, and therefore Sodaq did not need a master node, because all the data went through the commercial network. Sodaq and Raspberry Pi needed software development for data gathering, caching, and transferring, but the data structure was fully self-made.

The prototype system worked as planned and collected data at the customer's premises for a month. The main finding from this pilot case was quite high CO2 values when there were people in the room. More specific details about the results of this prototype system can be found in [10].

Advantages: Complete customizability in both areas, hardware and software, is the most important advantage. If the developers have the ability to design, configure, and build the prototype, the end result is a complete data gathering prototype system. The ability to choose the best components also affects the result.

Disadvantages: Time-consuming development process where all steps (design, configuration, and construction) have to be done in more depth than with COTS devices.

B. Prototype with commercial sensor nodes

Pilot case #19 (Figure 3), a data gathering prototype system consisting of COTS components, was designed to gather the temperature and humidity data. The sensor nodes (RS_n in the figure) were RuuviTags [3]. The master node was a Ruuvi Gateway [4], which received the data from RuuviTags and sent the data to the Ruuvi Cloud using a WiFi or Ethernet network. The Ruuvi Gateway made it possible to use commercial data storage and visualization, but it also provided an interface to fetch raw data, if needed. In the pilot case, raw data import as well as self-made data storage and visualization were used. The primary goal of the pilot case was to analyze and visualize thermal comfort [11] (Figure 4).

[1] https://support.sodaq.com/Boards/ExpLoRer/
[2] https://lora-alliance.org/about-lorawan/
[3] https://ruuvi.com/ruuvitag/
[4] https://ruuvi.com/fi/gateway/

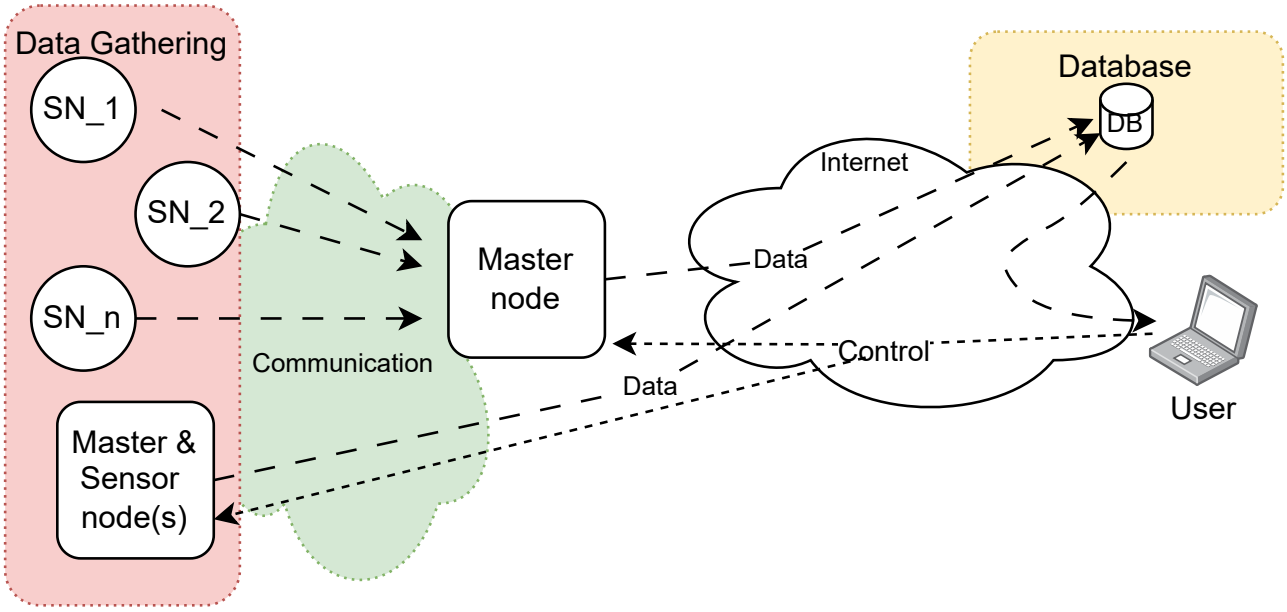

Fig. 1: Abstract data gathering system architecture.

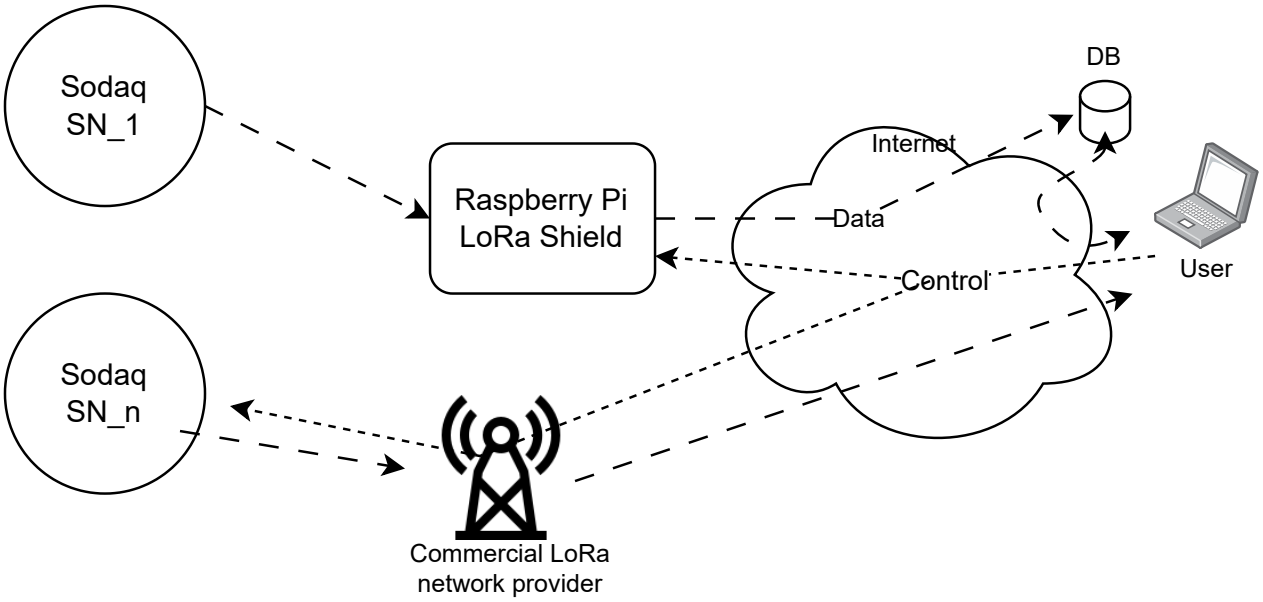

Fig. 2: Mid level data gathering system architecture of partially self-made components.

Advantages: The development process of the system is quite simple, when the connected devices work together. The software of the selected devices was ready-to-use, so there was no need for complicated configuration procedures. Mainly, these aspects were due to one commercial ecosystem, consisting of RuuviTag sensor nodes and a Ruuvi Gateway. The sensor nodes have the capability of using self-made firmware software as well as the Ruuvi Gateway.

Disadvantages: If the selected devices target ordinary consumers, the configuration possibilities are most probably limited, i.e., vendor lock. Also, even though the devices used have several possible features available, the documentation was limited. Ecosystem thinking might also limit the use of different manufacturers' devices together.

C. Hybrid prototype system

Pilot case #08 was a hybrid prototype with 12 RuuviTag sensors. The master node for RuuviTags was a self-configured Raspberry Pi. The purpose of the prototype was similar to pilot case #19, i.e., to collect humidity, pressure, and temperature sensor values. These data were cached on the Rasperry Pi database. Pilot case #08 was an early phase KIEMI pilot and therefore the main goal was to test RuuviTag sensors and their features.

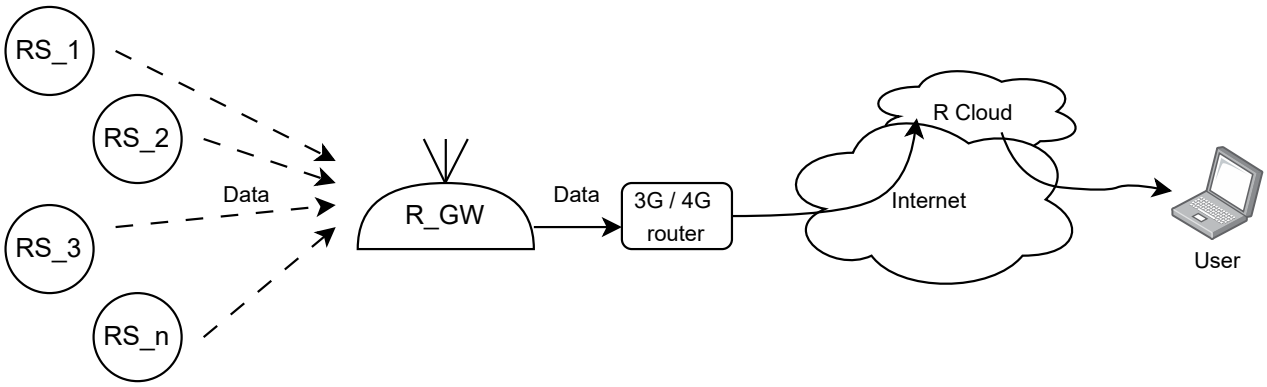

Fig. 3: Data gathering system architecture of COTS devices.

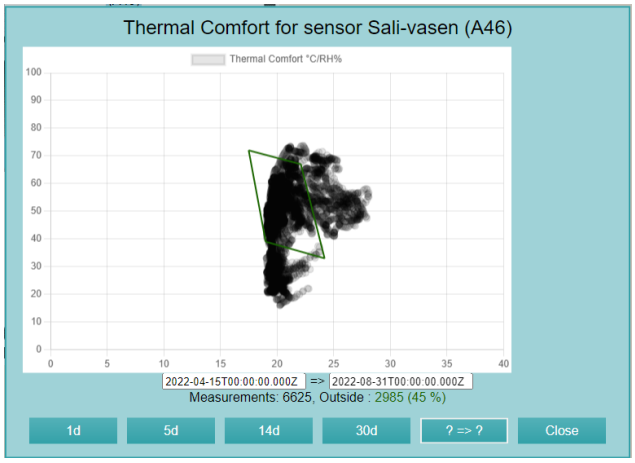

Fig. 4: Thermal comfort can be estimated using time, humidity, and temperature. The results are from pilot case #19. The green parallelogram marks "good" values.

Advantages: Sensor devices are supported for use in a specific interface definition. The sensor node manufacturer supports the open source development of the master node software. Usability of COTS devices depends on whether they are do-it-yourself or consumer products.

Disadvantages: Interoperability issues with specific hardware or the documentation of interfaces is limited. Ecosystem thinking limits the usage of different manufacturers' devices together.

IV. Discussion

The following research question was set for this study: Is it reasonable to use off-the-shelf devices instead of self-made sensor nodes in IoT data gathering?

The study shows that all three approaches have advantages. With self-made systems, there are more opportunities to modify the features. The hardware can be selected as required and software can be produced with the features that are needed. The disadvantages of self-made systems are their time-consuming development, as both software and hardware require definition before production can start. The use of COTS devices is less time-consuming, but their modifiability is limited. Also, the configuration can be challenging depending on the state of the product - whether it is a consumer product or a do-it-yourself product. A hybrid system where developers can select the best devices using selected criteria was a good combination of cost-effectiveness, time- saving, and the usefulness of the output.

However, it cannot be said which is the best because its application depends on the particular use case. We have earlier research results [9] where several prototype systems were built for data gathering. That project and study showed that reusability increased when similar prototype systems were built. Reusability refers to software and hardware, but as shown by the research, software is developed more from self-made with several iteration rounds. On the other hand, hardware more commonly uses COTS devices because their development is more time-consuming. In addition, the KIEMI project made extensive efforts to collect data without device limitations. Further, one reason for custom software development was that the research group had more knowledge about software development than hardware development.

The study showed the need for further research on how to find and select the necessary components, both software and hardware. A possible approach here could be connecting and networking with other partners in industry.

V. Conclusions

This study presented three real world pilot cases where custom data had been collected. This was done by using fully or partly COTS products that had the necessary capabilities or features to meet the specific requirements of the data collection. This study focused on COTS and self-built data gathering devices by comparing three data gathering prototype systems: The first was designed and implemented from scratch. The second was built with COTS components, and the last one was a combination of both. These systems were compared in terms of various features such as cost-effectiveness, time saving, and the usefulness of the output. The study presents examples of the usage of the different devices, showing the advantages and disadvantages at different points of usage. Finally, the

conclusion is that there are situations where the use of a ready-made commercial component makes more sense than using a self-built one.

ACKNOWLEDGMENTS

This work was funded by the European Regional Development Fund and the Regional Council of Satakunta.

REFERENCES

[1] M. Nurminen, A. Lindstedt, M. Saari, and P. Rantanen, "The Requirements and Challenges of Visualizing Building Data," in *2021 44th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*. IEEE, 2021.

[2] M. Nurminen, M. Saari, and P. Rantanen, "DataSites: a simple solution for providing building data to client devices," in *2021 44th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*. IEEE, 2021.

[3] I. Akyildiz, Y. Sankarasubramaniam, and E. Cayirci, "A survey on sensor networks," *IEEE Communications Magazine*, vol. 40, no. 8, pp. 102–114, 2002.

[4] I. Sommerville, *Software engineering (10th edition)*, 10th ed. Pearson Education Limited, 2016.

[5] M. Saari, M. Nurminen, and P. Rantanen, "Survey of Component-Based Software Engineering within IoT Development," in *2022 45th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*. IEEE, 2022.

[6] G. Pau and V. M. Salerno, "Wireless sensor networks for smart homes: A fuzzy-based solution for an energy-effective duty cycle," *Electronics*, vol. 8, no. 2, 2019. [Online]. Available: https://www.mdpi.com/2079-9292/8/2/131

[7] B.-C. Renner, J. Heitmann, and F. Steinmetz, "Ahoi: Inexpensive, low-power communication and localization for underwater sensor networks and uauvs," *ACM Trans. Sen. Netw.*, vol. 16, no. 2, jan 2020.

[8] M. Saari, A. M. bin Baharudin, and S. Hyrynsalmi, "Survey of prototyping solutions utilizing Raspberry Pi," in *2017 40th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*. IEEE, 2017, pp. 991–994. [Online]. Available: http://ieeexplore.ieee.org/document/7973568/

[9] J. Harjamäki, M. Saari, M. Nurminen, P. Rantanen, J. Soini, and D. Hästbacka, "Lessons learned from collaborative prototype development between university and enterprises," in *Information Modelling and Knowledge Bases n*, ser. Frontiers in Artificial Intelligence, M. Tropmann-Frick, H. Jaakkola, and N. Yoshida, Eds. IOS Press, 2023, accepted for publication.

[10] *Towards the utilization of cost-effective off-the-shelf devices for achieving energy savings in existing buildings*. IEEE, 2020.

[11] N. Langner and M. Illner, "Thermische behaglichkeit nach din en iso 7730 – ein ansatz zur vereinfachten datenaufnahme und berechnung für die bewertung von bürogebäuden," *Bauphysik*, vol. 37, no. 3, pp. 159–168, 2015. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/bapi.201510019