Chatbot-Based Querying of IoT Devices in EdgeX

Aalwahab Dhulfiqar[1], Norbert Pataki[1,*] and Máté Tejfel[1]

[1]*Department of Programmming Languages and Compilers, Faculty of Informatics, Eötvös Loránd University, 1/C Pázmány Péter st., Budapest, H-1117, Hungary*

Abstract

The increasing number of IoT devices connected to EdgeX makes it challenging to retrieve data from these devices efficiently. In this paper, we propose a chatbot-based solution for querying IoT devices connected to EdgeX. The chatbot utilizes natural language processing (NLP) techniques to understand user queries and retrieve relevant data from the EdgeX database. Our solution offers an easy-to-use interface for non-technical users to retrieve data from IoT devices, enabling them to quickly and easily access information about their devices. Our results demonstrate that our chatbot-based solution is efficient and effective in retrieving data from IoT devices, offering a more user-friendly approach for querying EdgeX databases. The proposed chatbot-based solution has the potential to improve the accessibility and efficiency of data retrieval from IoT devices in EdgeX.

Keywords

Open-Source Edge, EdgeX, Chatbot, RASA, IoT Devices, NLU

1. Introduction

With the increasing number of IoT devices connected to EdgeX, efficient data retrieval from these devices has become a challenging task. The EdgeX platform provides a standard architecture for integrating IoT devices and provides an open-source framework for data exchange and management [1]. However, querying the EdgeX database (influxdb) for specific device information can be a complex and time-consuming process, especially for non-technical users. To address this issue, we propose a chatbot-based solution for querying IoT devices connected to EdgeX.

As technology advances and consumers become more accustomed to convenient and user-friendly solutions, it is becoming increasingly clear that integrating a digital assistant within the IoT Service Portal is necessary. This will allow users to easily access and control their IoT devices through a single platform, streamlining the user experience and making it more efficient. By integrating a digital assistant, users can interact with their devices using natural language, making it easier for them to manage and monitor their devices and automate tasks. This trend is being observed in the industry and is expected to continue as consumers demand more seamless and enjoyable user experiences.

SQAMIA 2023: Workshop on Software Quality Analysis, Monitoring, Improvement, and Applications, September 10–13, 2023, Bratislava, Slovakia

*Corresponding author.

dolfi@inf.elte.hu (A. Dhulfiqar); patakino@elte.hu (N. Pataki); matej@inf.elte.hu (M. Tejfel)

0000-0002-7893-6250 (A. Dhulfiqar); 0000-0002-7519-3367 (N. Pataki); 0000-0001-8982-1398 (M. Tejfel)

Chatbots have emerged as a promising solution for improving the accessibility and efficiency of data retrieval from IoT devices. They provide a user-friendly interface for accessing information about IoT devices through natural language queries. The chatbot utilizes NLP techniques to understand user queries and retrieves relevant data from the EdgeX database.

In this paper, we propose a chatbot-based solution for querying IoT devices in EdgeX. Our solution is designed to offer an intuitive and accessible interface for non-technical users to retrieve data from IoT devices.

The remainder of the paper is organized as follows. In Section 2, we provide a detailed overview of related work in chatbot-based IoT data retrieval. Section 3 presents a brief review of the devices registration process in EdgeX; as an open-edge source solution. Section 4 explains how chatbot inquiry the EdgeX Database. Section 5 depicts in detail the architecture and design of our proposed chatbot-based solution for querying IoT devices in EdgeX. In Section 6, we evaluate the performance of our chatbot solution through dockerized images using docker-compose. Section 7 illustrates the integration process of the chatbot and RASA framework. Finally, we conclude the paper and discuss future directions for chatbot-based IoT data retrieval in EdgeX.

2. Related Work

Chatbots have gained popularity in the recent years due to their ability to provide an intuitive and accessible interface for accessing information through natural language queries. Several studies have investigated the advantages of chatbots for various applications, including IoT data retrieval. Here, we review some of the key advantages of chatbots in this context.

Chatbots improve user experience by providing a conversational interface that is more natural and intuitive for users than traditional interfaces. It also can reduce the cost of data retrieval by eliminating the need for expensive hardware or software [2]. Chatbots reduce the need for technical expertise, allowing non-technical users to easily retrieve data from IoT devices. Additionally, it can improve data security by providing a secure and controlled interface for accessing IoT device data. [3]. Chatbots can improve the speed and efficiency of data retrieval by quickly and accurately retrieving relevant data from IoT devices [4]. Chatbots can support multiple languages, making them accessible to users with different language preferences [2]. Chatbots can be integrated with other systems, such as voice assistants, to offer a seamless user experience across different devices [5]. Chatbots can provide personalized recommendations based on user preferences, improving the relevance and usefulness of the retrieved data [6]. Chatbots can provide real-time alerts and notifications, keeping users informed about changes in the status of their IoT devices [6]. Chatbots can reduce the workload of human operators by automating repetitive tasks, such as data retrieval and analysis [4].

3. IoT Device Registration in EdgeX Solution

The device registration process in EdgeX is an essential step in managing IoT devices and sensors within the platform. This registration process involves several steps that must be completed to

ensure that the devices are correctly identified and can be effectively managed by the platform. In this section, we will discuss the process of registering devices in EdgeX in detail.

The first step in the device registration process is to define the device profile. This profile describes the type of device and the data it produces. The device profile includes information such as the device name, manufacturer, model, and any other relevant details. This profile is used to define the data model for the device, which specifies the format and structure of the data that will be collected from the device.

Once the device profile is defined, the next step is to create a device service. The device service is responsible for communicating with the device and collecting data from it. The device service is also responsible for translating the data collected from the device into a format that can be consumed by the EdgeX platform.

After the device service is created, the next step is to create a device object. The device object is a representation of the physical device within the EdgeX platform. It includes information such as the device profile, device service, and any other relevant details. The device object is used to manage the device within the platform.

Finally, the device must be registered with the EdgeX platform. This involves providing the platform with the necessary information to identify and communicate with the device. The registration process includes providing the device profile, device service, and device object to the platform. Once the device is registered, it can be managed and monitored within the platform.

4. Chatbot in EdgeX solution

In today's fast-paced world, users demand quick and easy access to information and services. A digital assistant can meet these needs by providing instant answers to user queries, offering a conversational interface for executing tasks, and simplifying access to essential information without requiring users to navigate complex settings or documentation. By introducing a digital assistant, businesses can improve customer satisfaction and foster growth by catering to the evolving demands of users in an interconnected world. The convenience and ease of use provided by a digital assistant can make a significant impact on the user experience, ultimately leading to increased engagement and loyalty. The integration of IoT devices with the EdgeX open-source platform (Section 3) is a complex task that typically requires technical expertise. Conversely, accessing device information and statistics should be made available to a wider range of users. To address this, the implementation of a chatbot is proposed as a solution to streamline device integration and enable easy data retrieval within the EdgeX platform.

Figure 1 provides an overview of the role of the proposed chatbot in EdgeX. As shown in the figure, the integration of a chatbot within the EdgeX platform involves leveraging device metadata to facilitate inquiries to the InfluxDB database. By utilizing information about the devices stored in metadata, the chatbot streamlines the process of interacting with the EdgeX platform and enables efficient querying of the InfluxDB database.

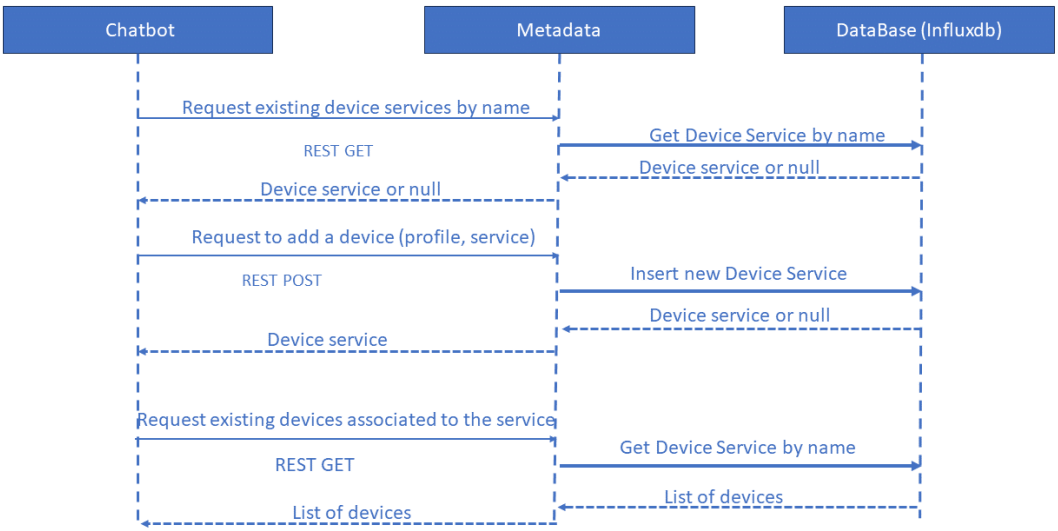

Figure 1: High Level Interaction Chatbot Diagrams

5. Enhancing EdgeX Platform with RASA-Powered Chatbot

The EdgeX Server Edge Solution is a comprehensive framework designed for edge computing, enabling the collection, processing, and analysis of IoT data closer to the source. While this system offers powerful capabilities, interacting with it solely through traditional user interfaces can be complex and time-consuming. To address this challenge, we propose the integration of a chatbot into the EdgeX ecosystem, leveraging the RASA framework.

5.1. RASA Framework Overview

The RASA framework [7] is an open-source toolset that provides developers with the necessary components to build conversational AI applications. It encompasses natural language understanding (NLU) and natural language generation (NLG) capabilities, as well as a dialogue management system that allows for context-aware and interactive conversations. With its flexibility, RASA enables the creation of sophisticated chatbot systems capable of understanding and responding to user queries effectively. RASA is a preferred choice for chatbot implementations due to its robust natural language understanding (NLU) capabilities, advanced dialog management features, open-source nature, customizability, integration capabilities, and strong community support. With RASA, developers can build chatbots that accurately understand user input, engage in meaningful conversations, adapt and learn from interactions, seamlessly integrate with existing systems, and benefit from an active community that contributes to ongoing improvement and expansion of the framework. Overall, RASA empowers developers to create intelligent and interactive chatbot solutions tailored to their specific needs.

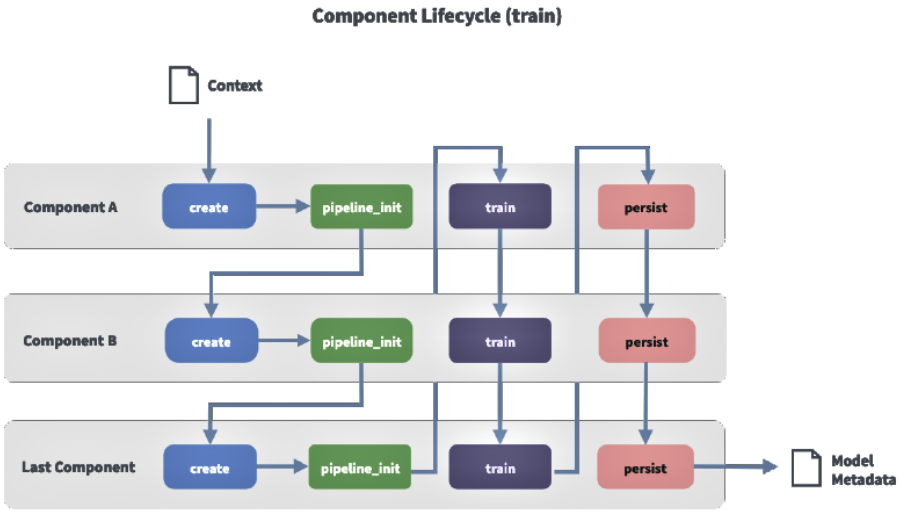

Figure 2: RASA Pipeline Training demonstration.

5.2. Architecture of the Chatbot for EdgeX Server Edge Solution

Our chatbot implementation follows a client-server architecture, where the RASA framework serves as the backend and the EdgeX Server Edge Solution acts as the frontend for interacting with IoT devices. The chatbot receives user queries via a user interface, processes them using RASA's NLU engine, and generates appropriate responses based on the extracted intent and entities. The backend interacts with the EdgeX Server through APIs, enabling seamless communication and device management.

5.2.1. Dialogue Management

Once the user query has been understood through NLU, the dialogue management component takes over to generate appropriate responses. Dialogue management rules and policies are defined to guide the flow of the conversation and determine the chatbot's behavior in various scenarios. For instance, if a user asks to retrieve sensor data from a specific device, the dialogue management system can orchestrate the interaction with the EdgeX Server APIs to retrieve and present the requested information. The RASA configuration serves as a systematic process, or pipeline, for training the RASA chatbot, focusing primarily on two key components: Natural Language Understanding (NLU) and the Dialogue Manager. This configuration sequence effectively tailors the chatbot to accurately interpret user input and manages conversations based on the data it has been trained on, thus fostering more sophisticated and efficient user interactions. Figure 2 shows the training process.

The training data and model development phase of implementing the chatbot for the EdgeX Server Edge Solution using the RASA framework involves gathering a diverse dataset of user queries, annotating and labeling the data with intents and entities, preprocessing the data to ensure consistency, and applying data augmentation techniques to enhance diversity. This

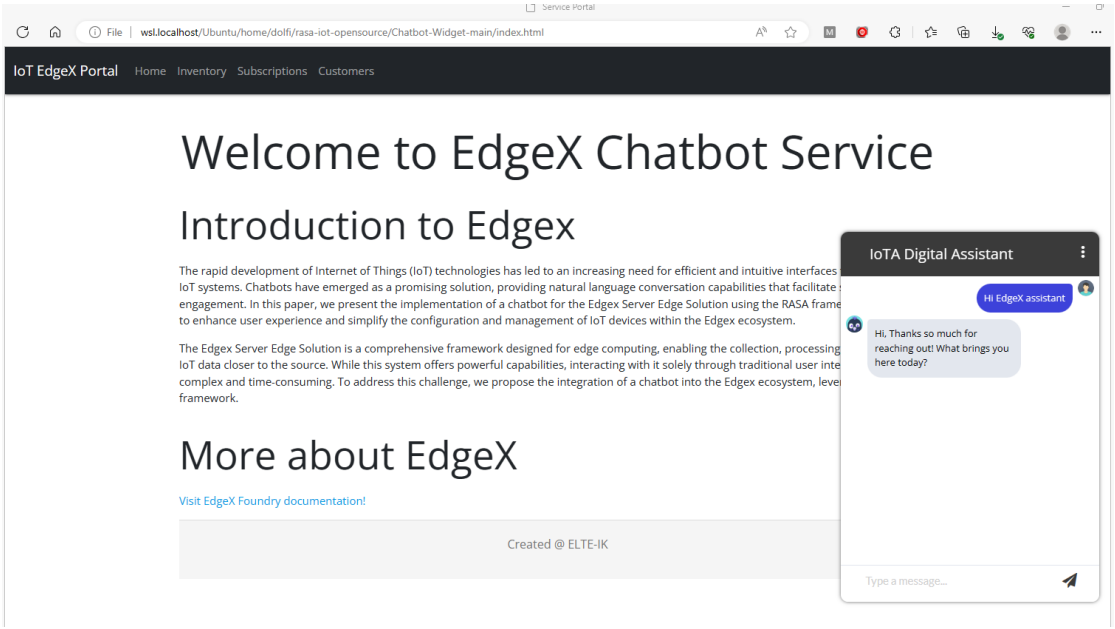

Figure 3: EdgeX service portal (GUI).

annotated and preprocessed training data is then used to train the chatbot model, which learns patterns and associations between user inputs, intents, and entities. The model is evaluated using an evaluation dataset, and necessary adjustments and iterations are made to improve its performance. Transfer learning techniques, utilizing pretrained language models, can be applied for faster training and adaptation to the specific EdgeX domain. Continuous improvement and maintenance involve monitoring user feedback, updating the training data and model, and ensuring the chatbot stays up-to-date and provides accurate responses. Overall, this process enables the chatbot to accurately understand user queries, predict intents, and extract entities, leading to an enhanced user experience and improved device management capabilities within the EdgeX ecosystem.

5.3. GUI Implementation of EdgeX Chatbot

Figure 3 shows the frontend implementation of the proposed chatbot. The chatbot introduces a user-friendly graphical user interface (GUI) implementation of an EdgeX chatbot, enabling normal users to effortlessly retrieve information about devices integrated into the edge environment. The GUI design follows user-centric principles, emphasizing simplicity, clarity, and intuitive navigation. It incorporates interactive elements, such as real-time device status updates and advanced search options, enhancing the user experience and providing timely access to device-related data. With its intuitive design and contextual guidance, the GUI implementation empowers non-technical users to easily interact with the EdgeX platform, facilitating informed decision-making and promoting broader adoption in diverse IoT scenarios.

6. Docker Compose Deployment

Deploying the proposed chatbot, which is highly compatible with the EdgeX solution, can be achieved efficiently using Docker Compose. Docker Compose allows for the seamless orchestration of multiple containers that comprise the chatbot's various components [8]. The deployment configuration includes the following port numbers:

- RASA Action Server is running on Port: 50552: The RASA Action Server is responsible for handling custom actions and external API integrations. This ensures smooth communication between the chatbot and external services.
- RASA Server is running on Port: 50051: The RASA Server is the core component of the chatbot framework. It processes natural language understanding (NLU) requests, dialogues, and generates appropriate responses based on the trained models and rules.
- Mongo Server is running on Port: 27017: The Mongo Server stores the necessary data for the chatbot's operation. This server facilitates data storage and retrieval, providing persistent storage for user information, conversation history, and any other required data.
- Node Server is running on Port: 31313: The Node Server acts as the user interface or web-based application for interacting with the chatbot. It provides users with a convenient and intuitive interface to communicate with the chatbot and access its functionality.

By utilizing Docker Compose, these components can be deployed as separate containers, each mapped to the corresponding port numbers mentioned above. Docker Compose simplifies the deployment process by managing container creation, networking, and configuration, allowing for a streamlined and scalable deployment of the chatbot solution within the EdgeX ecosystem. To seamlessly integrate the proposed chatbot with EdgeX, a `docker-compose.yaml` file can be utilized to orchestrate the deployment of the necessary components. The docker-compose file outlines the configuration and dependencies of the chatbot system within the EdgeX ecosystem. The chatbot service is configured to use the "chatbot-image" as its Docker image. It exposes port 8080, allowing users to interact with the chatbot via the User Interface Portal. The environment variables "EDGEX-HOST" and "EDGEX-PORT" are set to specify the hostname and port number of the EdgeX core services that the chatbot will interact with. The "depends_on" section ensures that the chatbot service starts after the edgex-core service.

The edgex-core service is configured with the "edgex-core-image" Docker image. It exposes several ports (48081, 48082, 48087, and 5563) to allow communication between the EdgeX core services and other components. The "volumes" section creates a named volume "edgex-data" for persisting data used by the EdgeX core services.

To deploy the chatbot and Edgex services, navigate to the directory containing the `docker-compose.yaml` file and run the following command:

```
docker-compose up -d
```

This command will start the containers in detached mode, allowing the chatbot and EdgeX services to run in the background.

By integrating the chatbot with Edgex using the `docker-compose.yaml` file, the chatbot system can seamlessly communicate with the EdgeX core services. This integration facilitates efficient device management, enabling the chatbot to retrieve data from the EdgeX ecosystem and provide users with valuable insights and assistance.

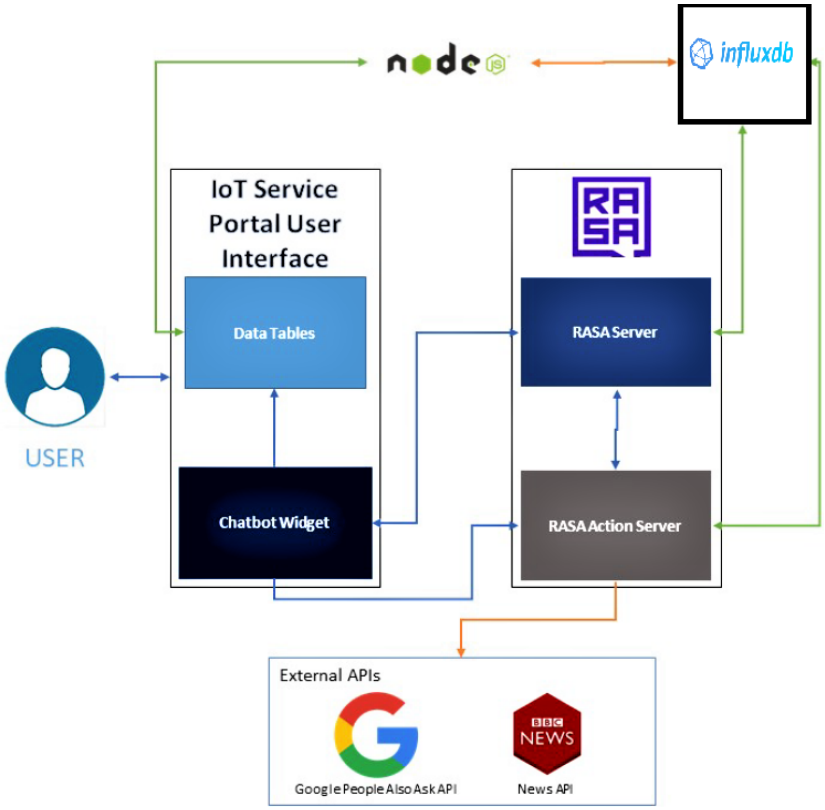

Figure 4: Chatbot Integration Sketch.

7. Implementation Sketch and Integration

The implementation of the chatbot system involves a well-structured architecture that integrates various components to provide a seamless user experience. Figure 4 illustrates the key elements and their connections within the system.

The user interacts with the chatbot system through a User Interface Portal specifically designed for IoT applications. This portal serves as the entry point for users to communicate with the chatbot and access its functionalities related to device management within the EdgeX ecosystem.

The RASA Server acts as the central component of the chatbot system, responsible for processing natural language understanding (NLU) requests and generating appropriate responses. It is connected to the Chatbot Widget, which enables the display of chatbot interactions within the User Interface Portal.

The RASA Server is also connected to the RASA Action Server, which handles custom actions and external API integrations. This connection allows the chatbot to perform specific actions based on user requests, such as querying the database or interacting with external APIs.

For data storage and retrieval, the RASA Server can access the InfluxDB database. This integration enables the chatbot to retrieve real-time device information and historical data for

user queries related to the EdgeX ecosystem.

Additionally, the RASA Action Server can be utilized by external APIs such as BBC or Google. This integration expands the chatbot's capabilities by allowing it to fetch news updates from the BBC News API or access contact information from the Google People API.

Overall, this implementation sketch showcases the seamless integration of components within the chatbot system. It highlights the interactions between the User Interface Portal, RASA Server, RASA Action Server, InfluxDB database, and external APIs, enabling the chatbot to provide efficient device management and extended functionalities for users within the EdgeX ecosystem.

8. Conclusion

The process of establishing connectivity between IoT devices and the EdgeX open-source platform poses considerable challenges and demands specialized technical knowledge for successful execution. Conversely, acquiring comprehensive information regarding added devices and their corresponding statistics should be accessible to individuals with varying levels of expertise. To address this requirement, the integration of a chatbot emerges as a crucial solution to facilitate seamless interactions, ensuring efficient device integration and simplified access to relevant data within the EdgeX platform.

The proposed chatbot-based solution provides an efficient and user-friendly approach to retrieve data from IoT devices connected to EdgeX. This solution has the potential to significantly improve the accessibility and efficiency of data retrieval for EdgeX users. Future work for the EdgeX chatbot includes incorporating device control functionality, enabling users to remotely manage devices (e.g., turning them on/off) through the chatbot interface. This expansion will require implementing secure authentication and authorization measures, integrating with the EdgeX infrastructure, and providing real-time feedback on device control actions. In addition to its core functionality within the EdgeX ecosystem, the proposed chatbot has the potential for future enhancements and expansion through the integration of external APIs. By leveraging APIs such as the BBC News API and Google People API, the chatbot can offer users access to a broader range of information and services.

Integration with the BBC News API enables the chatbot to deliver real-time news updates, tailored to the user's preferences. Users can inquire about the latest headlines, specific news topics, or receive personalized news recommendations. This integration enriches the chatbot's capabilities, providing users with valuable and up-to-date information.

Furthermore, integrating the Google People API enables the chatbot to access and manage user contacts and information. Users can interact with the chatbot to retrieve contact details, schedule appointments, or perform tasks related to personal information management. This integration enhances the chatbot's functionality and empowers users to conveniently handle their personal and professional contacts.

By extending the chatbot's capabilities with these external APIs, it becomes a versatile assistant that not only assists with EdgeX-related tasks but also provides a wider range of services and information. This extensibility ensures that the chatbot remains adaptable and can evolve to meet the changing needs and preferences of its users.

Acknowledgments

Supported by the NKP-22-4 New National Excellence Program of the Ministry for Culture and Innovation from the source of the National Research, Development and Innovation Fund. Also, the project was executed with the support of Ericsson.

References

[1] D.-H. Kwon, H.-K. Lim, Y.-H. Han, M.-s. Kim, Y.-G. Hong, Implementation of IoT control system based on EdgeX Foundry, in: Proceedings of the Korea Information Processing Society Conference, Korea Information Processing Society, 2018, pp. 995–997.

[2] K.-J. Chung, Y. Kwon, K.-J. Kim, Chatbot for Internet of Things: A case study, Journal of Ambient Intelligence and Humanized Computing 9 (2018) 479–485.

[3] H. Yan, X. Sun, L. Guo, Chatbot-based IoT device management, in: 2018 IEEE International Conference on Communications (ICC), IEEE, 2018, pp. 1–6.

[4] S. Borah, S. Dutta, N. Das, Chatbot based IoT interface for smart agriculture, in: 2019 International Conference on Electrical, Electronics and Computer Engineering (ICEECE), IEEE, 2019, pp. 1–4.

[5] Y. Zhao, X. Peng, J. Hu, Building an IoT-based chatbot system with natural language processing, in: 2019 IEEE 15th International Conference on Control and Automation (ICCA), IEEE, 2019, pp. 1766–1771.

[6] C. Chen, C. Li, L. D. Xu, Chatbot-based recommendation system for smart homes, IEEE Internet of Things Journal 8 (2021) 7587–7595.

[7] A. Singh, K. Ramasubramanian, S. Shivam, Introduction to Microsoft Bot, RASA, and Google Dialogflow, Apress, Berkeley, CA, 2019, pp. 281–302. URL: https://doi.org/10.1007/978-1-4842-5034-1_7. doi:10.1007/978-1-4842-5034-1_7.

[8] Á. Révész, N. Pataki, Containerized A/B testing, in: Z. Budimac (Ed.), Proceedings of the Sixth Workshop on Software Quality Analysis, Monitoring, Improvement, and Applications, CEUR-WS.org, 2017, pp. 14:1–14:8. URL: http://ceur-ws.org/Vol-1938/paper-rev.pdf.