FMDB *Transactions on Sustainable Computer Letters*

Towards Flexible Voice Assistants: Evaluating Privacy and Security Needs in IoT-enabled Smart Homes

Stefano Cirillo[1], Giuseppe Polese[2], Daniele Salerno[3], Benedetto Simone[4], Giandomenico Solimando[5,*]

[1,2,3,4,5] Department of Computer Science, University of Salerno, Fisciano, Salerno, Italy.
scirillo@unisa.it[1], gpolese@unisa.it[2], salerno@unisa.it[3], simone@unisa.it[4], gsolimando@unisa.it[5]

Abstract: The Internet of Things (IoT) is a fast-growing technology, especially for its application in smart homes. Through voice assistants, the user is provided with the ability to interact with devices without physical interaction, especially remotely. An issue that should not be underestimated is privacy and security, given that anything connected to the Internet is vulnerable. In addition to focusing on the risks of remote attacks, local risks must also be considered. Hence, access mechanisms are needed to limit the execution of critical functionality to only specific users. Since voice assistants on the market do not implement such functionality, adopting a voice assistant that provides maximum flexibility in adding new functionality arises. This paper discusses the feasibility and effectiveness of using the open-source Leon voice assistant to manage various home automation devices. Leveraging the unique flexibility offered by its open-source nature, we introduce and develop enhanced features to improve privacy and security measures within the smart home environment. Moreover, we perform an in-depth case study involving the integration of the Leon Voice Assistant with Home Assistant, specifically focusing on the control of smart bulbs. We developed a 'lights' module within the 'home assistant' package using the API provided by Home Assistant.

Keywords: Internet of Things; Smart Homes; Voice Assistants; Privacy and Security; Home Assistant; IoT-enabled Smart Homes; Use Access Mechanisms; Leon Voice Assistant.

Received on: 13/10/2022, **Revised on:** 20/12/2022, **Accepted on:** 22/01/2023, **Published on:** 14/02/2023

Cited by: S. Cirillo, G. Polese, D. Salerno, B. Simone, G. Solimando, "Towards Flexible Voice Assistants: Evaluating Privacy and Security Needs in IoT-enabled Smart Homes," *FMDB Transactions on Sustainable Computer Letters*., vol. 1, no. 1, pp. 25–32, 2023.

1. Introduction

The Internet of Things (IoT) describes a wide range of objects having sensing and actuating devices that collect, analyze, and share data among other objects, programs, and platforms. Since it is one of the most important technologies of this century, the use of IoT devices has grown from 8.7 billion to 50.1 billion [1]. One of the main applications of IoT concepts is the smart home, where objects are connected on a network to collaborate and be managed remotely via smartphones or voice assistants. The latter are devices that recognize people's natural language allowing them to receive information (e.g., about the weather, traffic, news of the day) and to give commands (e.g., turn on lights, adjust temperature). Some of the most widely used and popular consumer solutions include Amazon Alexa [2], Google Assistant [3], Apple Siri [4], and Microsoft Cortana [5].

The quick diffusion of voice assistants in the home environment simplifies everyday actions without physical interaction with the objects involved.

[*] Corresponding author.

IoT devices make it possible to perform actions remotely, such as checking surveillance cameras while on vacation. Examples of home devices that can be controlled include smart bulbs, air conditioners, smart plugs, and any other type of smart device. Through voice assistants, several actions can be performed, such as:

- the turning on and off of lights;
- the adjustment of home temperature and home thermostats;
- the startup of smart home appliances.

One of the issues that come with the use of Internet-connected devices is privacy and security [20]. Someone could access home cameras, listen to conversations, or remotely disable the home burglar alarm. Security risks can also be local, such as a burglar entering the home and disabling through voice assistant the home alarm instead of using the code [21]. This aspect has not been considered by the manufacturers of major solutions in the market [22].

Therefore, the need emerges to use mechanisms that can filter and block requests for actions from specific users or specific categories of users to avoid undesired and potentially dangerous actions for the individual and the collective [18]. By using these mechanisms, it is possible to avoid the problem of a burglar entering the home and easily disabling the alarm [19].

Therefore, our work aims to demonstrate the feasibility of using the Leon voice assistant to manage some home automation devices. Since the latter is an open-source project, it provides us with maximum freedom in developing features related to privacy and security.

2. Related Work

Several IT companies are working hard to create the best voice assistant to satisfy users as much as possible. Cortana, Siri, Google Assistant, and Amazon Alexa are the most famous. A virtual assistant is software that understands natural language and, when properly trained, can communicate with human interlocutors to provide information or perform certain tasks. Among these, we will focus on the possibility of managing home automation devices of different types: smart bulbs, smart ovens, smart refrigerators, and many others.

There are three main types of voice communications in IoT environments:

- Bidirectional voice communication;
- Single-directional voice communication;
- Voice recognition.

Strategy Analytics [6] thinks that using voice commands is suitable for a variety of IoT applications for the following key reasons:

- Speech is the natural mode of communication for humans, and being intuitive makes it easier to transmit voice commands;
- Speech recognition is useful when the user is engaged in other actions. In some cases, there is an obligation to use voice commands; just think of them when driving;
- Telephony is an efficient medium for bidirectional voice communication. Virtual voice operators (e.g., telephony operators) are an efficient means of bidirectional communication: the voice assistant can listen and reply without the need for complex commands, simplifying identification procedures (e.g., requesting remaining credit on the sim being used) and limiting the use of operators, leaving only the most complex handling to them;
- Cost-saving factors: voice integration could make it possible to eliminate the touch screen on many devices, reducing costs for devices that will be dormant most of the time.

The operation of a Voice Assistant, shown in Figure 1, consists of the interaction of several components [7]:

- User: interacts vocally with the Smart Speaker, usually activating it with a keyword;
- Smart Speaker: equipped with a microphone and Internet connection, it records the user’s commands in audio format and sends them to the speech recognition service. It is also capable of communicating the outcome of the request vocally;
- Speech-recognition service: converts audio to text, interprets the user’s request, and sends the text in a processable format (e.g., JavaScript Object Notation, or "JSON": a format suitable for data interchange between client/server applications).

- Compute service: implements the computational logic of the requested service. It identifies which function is to be executed and, eventually, interacts with an external server to fulfill the user's request for handling a Smart Object.

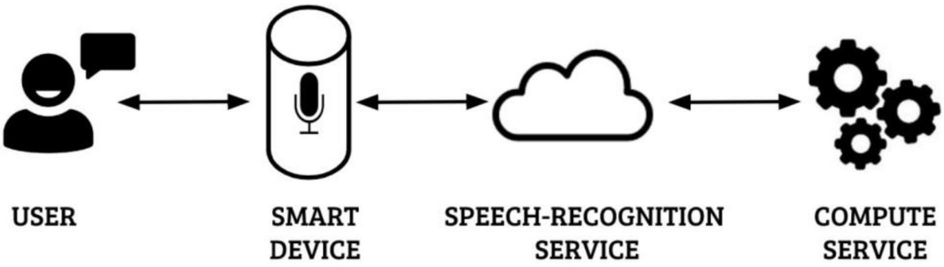


Figure 1: Voice Assistant architecture

Most Voice Assistants offer built-in functions: web search, home automation management, etc.

Alexa, for example, provides several features already built into the service, so-called "skills," to perform music playback, get weather information, search for content on Wikipedia, and so on.

For managing the various smart devices in a home, Alexa allows developers to build ad-hoc skills using Alexa Skill Kit (ASK), a collection of APIs, tools, documentation, and code examples. Depending on the type of functionality you want to implement, you must choose the most appropriate type of skill. ASK supports the creation of five different skill types:

Smart Home Skill: Allows controlling and managing Smart Home devices, such as lights, thermostats, and doors. The Smart Home Skill API manages the interaction model. "Intents," software mechanisms that allow users to coordinate the functions of different activities to achieve multiple goals, are processed by an AWS Lambda Function, a processing service offered by Amazon Web Services (AWS) [8];

Video Skill: Allows the control of video devices and streaming services. The implementation of this type of skill is very similar to Smart Home Skill, and it differs only from the interaction model, which in this case is defined by the Video Skill API. With such APIs, it is possible to develop skills that allow customers to control their entire video experience with voice, all achievable in a very simple and intuitive way [9];

Music and Radio Skill: Alexa Music and Radio Skill API is a set of interfaces that enable the selection and control of audio content streamed through an Alexa-enabled device. Interaction and usage are quite similar to the Video Skill API, making content available from the individual provider [10];

Custom Skill: A Custom Skill can handle any type of request, simple or complex. For example, purchasing on a website, reading e-mail, checking smart devices, etc. For the implementation, an interaction model needs to be defined, i.e., we need to indicate what requests the skill can handle (intent) and what expressions (utterance) the user needs to use to invoke those requests.

A cloud-based service will process those intents and subsequently return a response. In this case, the cloud-based service can be either a web service implemented ad-hoc for the skill or a function implemented on Amazon Web Service (AWS) Lambda. If you intend to create a new skill that can be used through Alexa, you must register and configure it on the Amazon Developer Portal.

Once this procedure is completed, the skill will be evaluated by Amazon, and if it meets the certification requirements, it will become available for download through the Alexa App; if not, it will be available only for the account through which it was developed, thus becoming a private skill [11].

In addition to the various Voice Assistants on the market, several are in the embryonic stage. Mycroft [12], Kalliope [13], Open Assistant [15], Jasper [14], and Leon are examples of open-source voice assistants. Leon [16], for example, currently provides a limited number of features called "modules," such as:

- Managing personal lists
- Running a speed test
- Grab the Github trend repository

- Download videos from Youtube

Since this is an open-source project, adding new features without limitations is possible. Creating new modules requires: Define the actions to be performed: the developer must define the source code related to the features he/she intends to implement:

- Define dataset: the developer must define the dataset in two parts.
- Expressions: are the data used to train Leon's understanding, i.e., the phrases to be spoken to perform a given action.
- Answers: these are the data Leon uses to provide your results bound with the modules outputs.

Each dataset can have different translations, for example, into English or French.

As mentioned in the Introduction, the key issue of Voice Assistants is privacy and security in using such devices. Consider the case of the Amazon Alexa voice assistant: this cannot be configured to recognize a particular voice in such a way as to allow specific users to execute certain commands. Thus, anyone can use all the device's functions, even the critical ones. Its only advantage is receiving customized responses based on the recognized user. To make up for the lack of limitations on certain features, Amazon Alexa users recommend changing the device's activation word, but this is not a solution to the problem since only four standard activation words are available. The same problem also affects Google Assistant, which only allows the creation of user profiles to receive personalized content based on the detected voice. On the opposite, an open-source Voice Assistant makes it easy to integrate such functionality.

3. Background

This paper will describe the technologies used to implement our project.

3.1. Leon [23]

Leon is an open-source personal assistant who can live on a personal server. To describeLeon's architecture, shown in Figure 2,
we will use the following scenario.

1. Client (web app, etc.) makes an HTTP request to GET some information aboutLeon;
2. HTTP API responds to information client
3. User talks with their microphone;
4. a) If hotword[1] server is launched, Leon listens (offline) if the user is callinghim by saying, Leon;
   b) If Leon understands the user is calling him, Leon emits a message to themain server via a WebSocket. Now Leon is listening (offline) to the user;
   c) The user said Hello! to Leon, and the client transformed the audio input into an audioblob;
5. ASR[2] transforms the audio blob into a wave file;
6. STT[3] parser transforms wave file to string (Hello);
7. a) User receives a string, and the string is forwarded to NLU[4];
   b) Or the user types Hello! with their keyboard (and ignores steps 1. to 7.a.).Hello! the string is forwarded to NLU;
8. NLU classifies string and picks up classification;
9. If collaborative logger[5] is enabled, classification is sent to collaborative logger;
10. Brain[6] creates a child process and executes the chosen module;
11. If synchronizer[7] is enabled and the module has this option, it synchronizes content;
12. Brain creates an answer[8] and forwards it to the TTS synthesizer;
13. The TTS9 synthesizer transforms text answers (and sends them to the user as text) to an audio buffer the client plays.

[1] The hot word node is an independent Node.js process that allows you to listen for the Leonhot word. Once Leon hears his name, he listens to your request.

[2] ASR, or Automatic Speech Recognition, uses computer hardware and software-basedtechniques to identify and process human voices.

[3] STT, or Speech-To-Text, transforms an audio stream (speech) into a string (text).

[4] NLU (Natural Language Understanding) helps computers understand human language.

[5] The collaborative logger helps to improve Leon's understanding. For each query you will submit to Leon if the collaborative logger is enabled, it sends an HTTP request to an external Leon'sserver.

[6] Leon's brain is a major part of his core. This is where he executes his modules, talks, picks up sentences, etc.

[7] The synchronizer allows you to synchronize your content via different methods (Google Drive,on your current device, etc.) restricted by the requested module’s offerings.
[8] Answers are Leon’s responses. Each package has its own set of answers with different translations.
[9] TTS or (Text-To-Speech) transforms a string (text) into an audio stream (speech) [23].

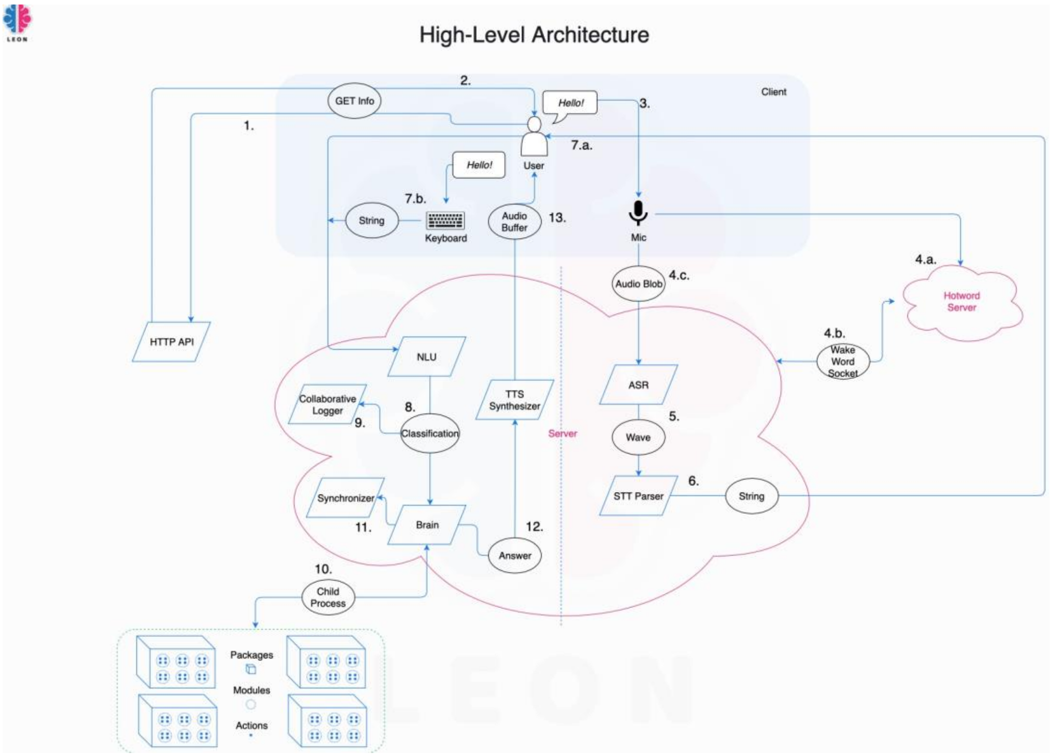


Figure 2: Leon's architecture [23]

3.2. Home Assistant

Home Assistant is a home automation platform that allows us to add and manage all the smart devices in our home. It is software that aims to act as a central control system for smart home devices and can be accessed through a Web-based user interface or complementary apps developed for Android and iOS.

All IoT devices, technologies, software, applications, and services are supported by modular integration components, which not only include native integrations for local connectivity protocols such as Bluetooth, MQTT, Zigbee, and Z-Wave but also support control of proprietary ecosystems if they provide public access via an open API for third-party integrations.

Home Assistant integrates with over a thousand different devices and services. Upon startup, it will automatically scan the network for known devices and enable easy configuration. It also offers the ability to easily install other applications to implement home management. Being a cross-platform software, it is possible to use the official Home Assistant apps to quickly control all devices.

After integrating all the devices in the home, you can create various custom automation, such as turning on the lights when the sun goes down. The major advantage of Home Assistant is that since it is not cloud-based like other similar platforms, it has been designed with a greater emphasis on security and privacy. Since Home Assistant communicates with all devices locally, all Smart Home data remains local.

4. Case Study

In this paper, we present our case study in which we perform integration between Home Assistant, using only smart bulbs, and Leon Voice Assistant.

Since we used smart bulbs, we focused on the actions related to those devices. To do that, we implemented, using the API made available by Home Assistant [17], a "lights" module, within the "home assistant" package, which is responsible for managing all functionalities related to smart bulbs.

4.1. Turn-on and Turn-off

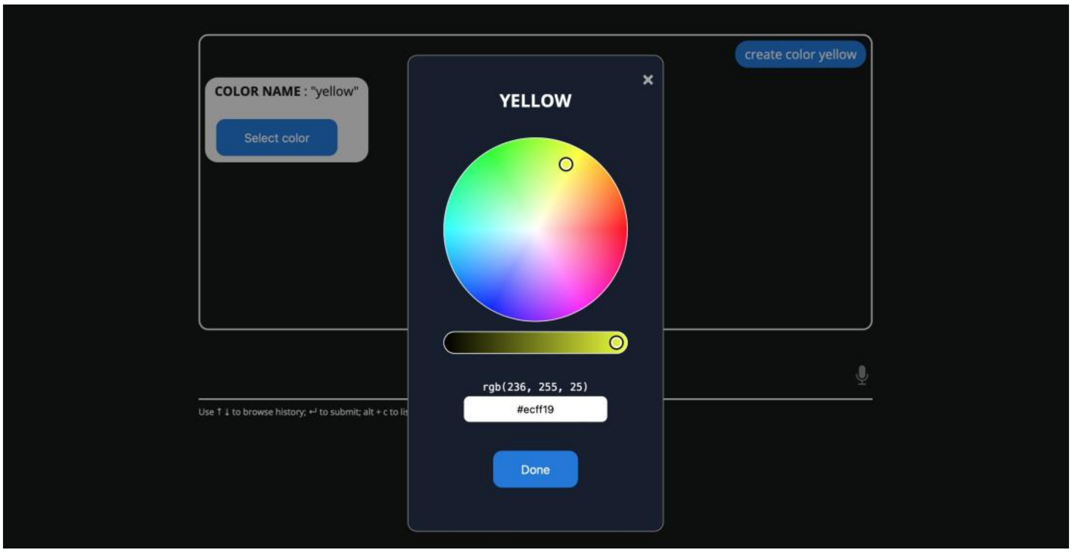
The first feature developed was to turn a bulb, or a group of bulbs, on and off using the following voice commands:

- "Turn on/off the x light/group"
- "Can you turn on/off the x light/group?
- "Please turn on/off the x light/group."

The user must replace the letter x with the name of the light bulb he/she intends to turn on or off. Upon receiving the voice command, the name that the user has associated with the device will be derived from it, and after searching its id, the following APIs will be used to turn the device on or off, respectively:

- http://homeassistant.local:8123/api/services/light/turn_on
- http://homeassistant.local:8123/api/services/light/turn_off

Several checks will also be made to see if the device exists and is available to avoid abnormal behavior by the system. The same on/off procedure will also be carried out in the case of a group of bulbs.



create color yellow

COLOR NAME : "yellow"

rgb(236, 255, 25)

#ecff19

Figure 3: Colour picker to select the custom colour yellow

4.2. Colour management

Most lighting devices allow the colour of the light to be changed according to the user’s preferences or needs, and it is generally possible to use a set of colours preprogrammed by the device manufacturers. In the proposed system, it is possible to visualize the list of available colours using the command:

- "Show all possible colours for the lights."

At this point, Leon will return the list of available colour names in the output. Typically, such devices work using colours in RGB format, so it is possible to create a custom colour. Creating a custom colour and choosing its name is possible within the system we have developed. The user can use the following command, making sure to replace the x with the name of the colour he/she wants to create:

- "Create the colour x"

At this point, Leon will show the user a colour picker in Figure 3, where he/she can choose the colour and later save it.

These colours can later be applied to the smart bulbs using the following command:

- "Change the colour of x in y colour"

The user must replace the letter x with the name of the smart bulb whose colour he/she intends to change and y with the name of the colour he/she intends to select. Upon receiving the voice command, the user's name associated with the device and the colour identifier will be derived from it. The following API, previously used to turn on lights, can also be used to change the colour of a device by specifying the RGB code of the colour that he/she intends to use:

- http://homeassistant.local:8123/api/services/light/turn_on

Several checks will also be made to see if the device exists if the device is available, and if the specified colour name exists to avoid strange behavior by the system.

4.3. Change brightness

In addition to changing the colour of smart bulbs, it is possible to change their brightness with the command:

- "Change the brightness of x in y”

The user must replace the letter x with the name of the bulb or group and the letter y with the desired brightness value from 0 to 100. Upon receiving the voice command, the name that the user has associated with the device will be derived from it, and after searching for its id, the following API will be used, sending as an additional parameter the brightness value:

- http://homeassistant.local:8123/api/services/light/turn_on

Several checks will also be made to see if the device exists if the device is available, and if the specified brightness value is between 0 and 100 to avoid strange behavior by the system.

4.4. Turn on all, turn off all

Suppose the user wants to turn all the lights on or off simultaneously. He/She can use the following voice command:

- "Turn x all the light of my house";
- "Please turn x all the lights in my house";
- "Can you turn x all the lights in my house?"

The user should replace the letter x with the command "on" or "off" upon receiving the voice command; all available bulbs will be searched and will be respectively turned on or off with the following APIs:

- http://homeassistant.local:8123/api/services/light/turn_on;
- http://homeassistant.local:8123/api/services/light/turn_off

Several checks will also be made to see if devices are available to avoid errors within the system.

5. Conclusion and Future Work

This paper demonstrated the feasibility of using an open-source voice assistant to manage smart devices using Home Assistant. The final system obtained can manage smart bulbs efficiently, and the various tests performed were successful. Thus, the work has achieved its goal: Leon can be used to manage home automation devices. Currently, it can only manage smart bulbs, but in future implementations, it will be possible to manage any type of smart device supported by Home Assistant by creating new modules and taking advantage of the available APIs. In addition, it will be possible to implement various privacy- and security-related features, such as an authentication mechanism to restrict critical functionality. To do this, a new module could be created to perform role-based user profiling, using voice, and insert limitations to certain functionality through appropriate controls. It would then be the administrator user who would specify the list of users enabled to perform certain actions.

Acknowledgment: This work was partially supported by project SERICS (PE00000014) under the NRRP MUR program funded by the EU - NGEU.

Data Availability Statement: This study uses media and company information data. This is a new study conducted by the authors.

Funding Statement: No funding has been obtained to assist in preparing this manuscript and research paper.

Conflicts of Interest Statement: The authors declare no conflicts of interest. This is a new work by the authors. Citations and references are cited according to the information used.

Ethics and Consent Statement: Consent from the company's public information and media during data collection and Ethical Approval and Consent of Participants has been received.

References

1. Y.-J. Lee and H.-J. Choi, "Comparative study of emotion annotation approaches in Korean dialogue," in 2021 IEEE International Conference on Big Data and Smart Computing (BigComp), Jeju Island, Korea (South), pp. 354-357, 2021.
2. Amazon.com. [Online]. Available: https://developer.amazon.com/it-IT/alexa. [Accessed: 11-Oct-2022].
3. "Google Assistant, your own personal Google," Assistant. [Online]. Available: https://assistant.google.com/. [Accessed: 31-Oct-2022].
4. "Siri," Apple (Italia). [Online]. Available: https://www.apple.com/it/siri/. [Accessed: 31- Oct-2022].
5. "Cortana - Your personal productivity assistant," Cortana - Your personal productivity assistant. [Online]. Available: https://www.microsoft.com/en-us/cortana. [Accessed: 11- Oct-2022].
6. A. Koohang, C. S. Sargent, J. H. Nord, and J. Paliszkiewicz, "Internet of Things (IoT): From awareness to continued use," Int. J. Inf. Manage., vol. 62, no. 102442, p. 102442, 2022.
7. A. Brown, "The role of voice in IoT applications," in Whitepaper, Strategy Analytics, Newton, 2016.
8. V. Lavecchia, "Caratteristiche, a cosa serve e fasi dell'Analisi dei Requisiti per lo sviluppo software," Informatica e Ingegneria Online, 10-Mar-2020. [Online]. Available: https://vitolavecchia.altervista.org/caratteristiche-. [Accessed: 15-Oct-2022].
9. Amazon.com. [Online]. Available: https://developer.amazon.com/en-. [Accessed: 31-Oct-2022].
10. US/docs/alexa/smarthome/understand-the-smart-home-skill-api.html [Accessed: 31-Oct-2022].
11. Amazon.com. [Online]. Available: https://developer.amazon.com/en-US/docs/alexa/video/understand-. [Accessed: 30-Nov-2023].
12. Amazon.com. [Online]. Available: https://developer.amazon.com/en-US/docs/alexa/music-. [Accessed: 15-Sep-2023].
13. Amazon.com. [Online]. Available: https://developer.amazon.com/en-US/docs/alexa/design/design-your-skill.html. [Accessed: 22-Jul-2022].
14. mycroft-core: Mycroft Core, the Mycroft Artificial Intelligence platform. [Accessed: 22-Jul-2022].
15. C. Weise and A. M. Karimi, Eds., Kalliope: Zeitschrift für Literatur und Kunst, 1st ed. Siegburg, Germany: Bernstein-Verlag, 2009.
16. "Jasper," Github.io. [Online]. Available: https://jasperproject.github.io/. [Accessed: 25-Jul-2022].
17. "Open assistant – open source voice assistant," Openassistant.org. [Online]. Available: https://openassistant.org/wp/. [Accessed: 31-Nov-2022].
18. Leon: Leon is your open-source personal assistant. Leon: https://github.com/leon-ai/leon, [Accessed: 25-Jul-2022].
19. "Rest api," Home-assistant.io. [Online]. Available: https://developers.home-assistant.io/docs/api/rest/. [Accessed: 06-Jul-2022].
20. Breve, Bernardo, Stefano Cirillo, and Vincenzo Deufemia. "An Intrusion Detection Framework for Non-expert Users (S)." DMSVIVA 2020-Proceedings of the 26th International DMS Conference on Visualization and Visual Languages. Vol. 2020.
21. P. Moh, "Characterizing everyday misuse of smart home devices," in Proceedings of the 2023 IEEE Symposium on Security and Privacy, 2023.
22. B. Breve, L. Caruccio, S. Cirillo, D. Desiato, V. Deufemia, and G. Polese, "Enhancing user awareness during internet browsing," in ITASEC, 2020, pp. 71–81.
23. "Architecture," Getleon.ai. [Online]. Available: https://docs.getleon.ai/architecture. [Accessed: 31-Dec-2022].