

Bachelor Thesis

From Cloud to Open-Source: Evaluating Home Assistant's Potential for a Privacy-Safe and Sustainable Alternative to Cloud-Based Smart Home Systems

Raphael Reichel

Date of Birth: 19.10.2001

Student ID: 12118008

Subject Area: Information Business

Studienkennzahl: UJ 033 561

Supervisor: ao.Univ.Prof. Dr. Johann Mitlöhner

Date of Submission: September, 18th 2024

Department of Information Systems & Operations Management, Vienna University of Economics and Business, Welthandelsplatz 1, 1020 Vienna, Austria

Contents

1	Introduction	7
1.1	Research question	7
1.2	Research method	7
2	Smart Home Systems	9
2.1	What is a Smart Home?	9
2.2	Smart Home Platforms	11
2.3	Considerations when starting with Smart Home	12
3	Risks of Cloud-Based Smart Home Devices	14
3.1	Implications of Public Server Connections	14
3.2	Unauthorized Access to Smart Home Devices	16
3.3	Eavesdropping by Voice Assistants	17
4	Securing a Smart Home against Cloud-Risks	18
4.1	Avoiding Cloud Connections	18
4.2	IoT Network Security	19
4.3	Matter Protocol	19
5	Alternative to cloud-based Smart Home Systems: Home Assistant	20
5.1	Installation and Deployment	21
5.1.1	Home Assistant Green	22
5.1.2	Home Assistant on Raspberry Pi	23
5.1.3	Home Assistant Yellow	24
5.2	Concepts and Terminology	25
5.3	Device Support	28
5.4	Out-of-home Control	29
5.5	Dashboards	30
5.6	Automations	32
5.7	History	33
5.8	Voice Assistant	34
5.9	Advanced Features: Beyond traditional Smart Home	35
6	Comparison of Smart Home Platforms	37
6.1	Evaluation Criteria	37
6.2	Insight into the compared Platforms	40
6.2.1	Apple Home	40
6.2.2	Google Home	41
6.2.3	Samsung Smart Things	42

6.2.4	Amazon Alexa	43
6.2.5	Home Assistant	44
6.3	Evaluation	45
7	Experiment: Adoption of Home Assistant	47
7.1	Experiment Design	47
7.2	Experiment Evaluation	49
8	Conclusion	51
	Acknowledgements	52
	References	58

List of Figures

1	Revolv FAQ about their services [15]	15
2	Spotify FAQ for Car Thing [16]	15
3	Home Assistant Green front and back side	22
4	Raspberry Pi 5 [38]	23
5	Home Assistant Yellow [39]	24
6	Example of configured integrations in Home Assistant	25
7	Example of a motion sensor device shown in Home Assistant (Modified from [41])	26
8	Example of a service call in Home Assistant	27
9	Interactive HA dashboard demo [49]	31
10	Interactive HA dashboard demo - Editing of a dashboard [49] .	31
11	Interactive HA dashboard demo - Editing of a card on the dashboard [49]	32
12	Example of a temperature sensor in the History panel of Home Assistant	33
13	DIY pool controller running ESPHome	36
14	Participants categorised by their IT-Skills, Smart Home, and Raspberry PI experience	49
15	Perceived Difficulty of Participants during Experiment Phases	50
16	Time Spent by Participants in Experiment Phases	50

List of Tables

1	Specification for Comparison of Control Options	38
2	Specification for Comparison of Features	38
3	Specification for Comparison of Automations	39
4	Specification for Comparison of Voice Assistants	39
5	Comparison of Smart Home Platforms	46
6	Experiment Questionnaire	48

Abstract

Smart home systems have attracted more and more attention in recent years due to their huge potential to automate home management processes. However, the gained connectivity can potentially endanger the expected privacy within one's home - especially when accounting for the ubiquitous internet access of the modern age. This paper aims to address this cloud-based nature of smart home systems by assessing the current state of this technology, discussing the risk of cloud connections, and proposing a different approach to building a smart home that involves the usage of an open source and locally hosted smart home platform called *Home Assistant*. The paper will start with a brief introduction to the terminology of smart home systems and discuss selected terminologies while focussing on the most commonly adopted smart home platforms. Then, we will examine the imposing risks of smart home devices that are connected to cloud infrastructure and assess a handful of security measures to mitigate associated privacy risks. To emphasize the value proposition of a locally hosted smart home system, a major section of this paper will explore how Home Assistant operates and compare this system to the major competing smart home platforms, as well as highlight important topics for deploying Home Assistant within one's home.

1 Introduction

In an era marked by digital transformation, technological advancement did not hold from redefining how we interact with home appliances and thereby continued the trend of digitising every part of our daily life. In the case of the home, this meant integrating small computers into various devices, by which connectivity between previously unrelated appliances emerged, which led to the concept today known as a *Smart Home*.

While undoubtedly yielding many benefits and building unfathomable capabilities for automating home management processes, the gained connectivity can potentially endanger the expected privacy within one's home - especially when accounting for the ubiquitous internet access of the modern age. This risk is amplified by the cloud-based architecture of most traditional smart home systems, which expose home appliances to be controlled from the platform provider's servers.

This paper aims to address this cloud-based nature of smart home systems by assessing the current state of this technology, discussing the risk of cloud connections, and proposing a different approach to building a smart home that involves the usage of an open source and locally hosted smart home platform called *Home Assistant*.

1.1 Research question

The main research question of this paper is:

How does an alternative to cloud-based smart home systems work and can it improve the privacy of users without limiting their capabilities compared to traditional smart home platforms?

1.2 Research method

With a view to the aforementioned topics, this paper will start off with a brief introduction to smart home systems and discuss selected terminologies while focusing on the most commonly adopted smart home platforms. We will then later examine the imposing risks of smart home devices that are connected to cloud infrastructure based on literature and case-study reviews, and assess

a handful of security measures to mitigate associated privacy risks. The main proposal within this realm will be to deploy *Home Assistant* as a smart home platform and thereby omit the use of cloud connections for controlling smart home appliances altogether. To emphasize the value proposition of this approach, a major section of this paper will explore how Home Assistant operates and compare this system to competing smart home platforms, as well as highlight important topics for deploying Home Assistant within one's home.

After elaborating this proposal, Home Assistant will be compared to traditional smart home platforms based on its capabilities. Additionally, a small experiment with a few participants will be conducted in order to investigate the entrance barriers for new users who want to deploy Home Assistant on their own. For this the installation and on-boarding process for setting up Home Assistant on a Raspberry Pi will be assessed by the participants. Their experience and feedback will then be used to estimate if this approach of starting with Home Assistant would be appealing to a broader scope of users.

Open-Access: This paper, including the used materials, is available at this public github repository: <https://github.com/dir-rafa/Bachelor-thesis-Home-Assistant>

2 Smart Home Systems

A *Smart Home System* is the superset of connected devices and technologies that represent the physical and logical subtleties that make up the *Smart Home*, but is most commonly used as a synonym for the *Smart Home Platform* that handles communication between multiple devices and hosts the respective user interface. This terminologies do not really refer to the same concepts, since controlling devices via multiple platforms could have some benefits, e.g. different household residents want to use their respective Android or iOS native smart home platforms, but separating devices into multiple systems and therefore isolating them from communicating with another, will most often not be desirable.

A major point of differentiation of smart home systems is the medium that connects devices together [1]. While many alternatives exists for wireless systems [1], *KNX* dominates the market for wired systems [2]. Since the latter must be planned in advance and requires physical connections between devices, they are more expensive than wireless systems and are only deployable within the scope of a major renovation or construction of a new building [2]. Therefore, wireless systems are particularly interesting for homeowners who want to upgrade their existing home to a smart home. Moreover, this approach has the benefit that expanding the system is easier to accomplish as no new wires are required. For this reason we will focus on wireless systems when discussing smart home platforms, but first of all we will explore what a smart home is.

2.1 What is a Smart Home?

At its core a *Smart Home* is an interconnected network of home appliances that enables the home owner to perform actions that they would not be able to perform, in a comparative form, without the ability of the devices to communicate with each other. This does not necessarily require the user to perform manual actions, such as operating a light switch, but rather means that some trigger (like a switch or a sensor) can have an effect on a wider scope of appliances than itself is physically connected with. A common example for this is a self-regulating heating system, where temperature sensors control when a room needs to be heated, but the system is interrupted by smart contact sensors if the windows are opened [2]. The same system can be later expanded by setting a time trigger to warm up the bathroom before a resident gets up in the morning [3].

For this reason smart homes are often referred to as "intelligent living" [2], since they enable home electronics to be controlled from a central hub and automate processes to perform a set of actions that would otherwise require multiple interactions of the user [2]. This central hub is a core pillar of most smart homes and influences the set of capabilities the homeowner has to their disposal to create these action-sets. A core difference to conventional control systems is the aforementioned **expandability** of smart homes, meaning that additional devices or sensors can be easily added at any given time.

In addition to automating actions, this also enables the homeowner to repurpose their devices to perform tasks they were not supposed to perform initially. Given that some subset of devices, which typically are associated to be used only by residents (e.g. TVs or lights), communicate within this network, clever use of automations based on triggers like the current time can be used to create a simple presence simulation [2], which creates the illusion of activity within a home and, therefore, fools outsiders to believe that the home would be occupied at the moment.

This combination of capabilities shows the **emergent property** of smart homes, meaning that new utility can be created by re-contextualising the existing environment and leveraging the individual feature-sets of devices. Consumers perceive this property as the modularity of smart home systems, which has a great influence on the perception of the value of smart home [4].

In some sense a Smart Home can be viewed as the "[...] natural evolution of our homes. A smart home is not fundamentally different from a 'regular' home - it is just the improvement of one." [5]. In fact, smart homes are not a new technology or a sudden technological development. We have been putting computers into home appliances for more than a decade [5] and thus laying the foundation for the interconnected world in which we live today. What started with devices such as light bulbs, outlets, or thermostats led to the development of doorbells, locks, and vacuums, all controllable from a smartphone [5].

Like the uniqueness of physical houses, two distinct smart homes will always differ from each other. Although they may share certain properties, one should not imagine building a smart home as a normed operation. While a house must have an entrance and probably will have some windows, a smart home must have a common communication protocol and likely will have some form of user-interaction interface, but the physical location and material of that house windows will be determined by the preferences of the homeowner. Likewise the preferences for the visual representation of the user interface, or deployed technologies, will be different on a home to home

basis depending on the requirements and needs of the homeowner. Therefore, smart homes can fulfil different purposes and **operate at different levels of complexity** according to the preferences, interests, and technical proficiency of their owner.

2.2 Smart Home Platforms

The *Smart Home Platform* is a software framework that provides services to a smart home that enables the execution of actions with higher levels of complexity such as automations, serves as an organisational layer of abstraction (e.g. to group devices together), and acts as a central hub for managing smart devices. One of its core tasks is to handle communication between devices that use different communication protocols [1] or follow different specifications from different manufacturers [3]. Moreover, smart home platforms often provide a graphical user interface in the form of an smartphone-app or web interface for controlling devices and hosts some notification system to inform the user about changes or alerts. The organisation of devices is most often done by grouping them by their location into *rooms* that act as a digital counterpart of the physical house [2].

The most popular smart home platforms for consumers are developed by large tech companies [3] and are well known for their simple installation and easy integration of popular devices. Without particular order those *mainstream smart home platforms* are:

- **Amazon Alexa** - developed by Amazon
- **Apple Home** - developed by Apple
- **Google Home** - developed by Alphabet
- **Samsung SmartThings** - developed by Samsung

Most smart home platforms require a physical hardware component such as a *hub* [2] that can host these services, but many manufacturers combine this with some other type of appliance such as a smart speaker [3], but sometimes they are just small pucks that sit in an outlet somewhere in a home [6]. In any case, they act as the brain of the smart home platform and are essential for their operation.

In addition, hubs commonly incorporate radios for various wireless technologies that are required for local communication with devices [6]. Local

connectivity means that a smart home could be operated without a cloud connection and therefore stay operational in the event of an internet outage. However, not every smart home platform is equally independent from their manufacturers cloud. Especially voice assistants tend to be rendered useless without a cloud connection. Although some basic voice commands will most often work offline, more complex tasks require an active Internet connection. The importance of cloud connectivity for e.g. Alexa is apparent on their support site. The word "cloud" is mentioned 76 times in their roughly 18-page long FAQ-section about the operations of their voice assistant [7].

A common property of three of the four above-listed smart home platforms (with Alexa being the exception) is that they are developed by companies which also develop smartphones. They are also natively installed on those mobile devices. Although this in itself may not be significant for their success reasons, there might be an incentive for users to deploy the smart home platform that is developed by their phone manufacturer.

Besides aforementioned platforms, there are many other alternatives that are direct competitors or have a focus on specific use cases. *Home Assistant* is one of them and will be assessed in detail in section 5.

The choice of platform also determines the smart home community in which a user is likely to participate, as they may want to share their experiences with other consumers. This community influences the user's perceived value of smart home devices by creating social value through inter-consumer cooperation efforts [4].

2.3 Considerations when starting with Smart Home

It should be clear by now that building a smart home is not an easy task. Many decisions made along the way can influence the user's capabilities in the future, and likewise the environment in a home can change drastically. Even third parties can affect a smart home system. The devices deployed today will likely receive updates throughout the years, which can benefit the user, but can also change the software in such a way that the current way of operations has to be adapted.

One of the most important initial considerations will be to decide **which platform to use** [3], since this will determine *how one interacts* with the smart home and *what technologies it can support*.

At this point users have a wide array of options, but many tend to deploy

one of the four aforementioned mainstream platforms [3]. A deterministic for this decision could be that *users favor to build upon devices and technologies that they are currently using*. The already made argument that the users smartphone may influences their choice of smart home platform would support this hypothesis and leaning towards the platform of a manufacturer whose other devices (e.g. smart speakers, smart tv system, etc.) already provide value to the user, sounds plausible too. Therefore the owned devices may well be a good guide for what smart home platform is a good fit for starting a smart home [3].

In addition, the user has to ask themselves **what features they need** their smart home to have [3]. This ultimately influences the choice of their platform too. While Apple Home is known for their secure video platform (*Homekit Secure Video*) that processes footage of surveillance cameras locally before encrypting and storing them in the cloud [8], Samsung Smart Things offers an energy management system that can help users to cut down on their electricity usage [3][9]. Similarly, Amazon Alexa uses artificial intelligence to help users in their daily lives with suggestions based on their typical behaviour [10].

Above all **users have to trust a platform** and thus the manufacturer who is developing the platform of their choice, since they are likely dependent on the cloud infrastructure of the providers for e.g. processing voice requests (see section 2.2), store data like video footage [8] or analyse user data to give them insight into their behaviour [10] and energy consumption [9].

However, [11] shows that users seem to prioritize convenience and connectivity when selecting smart home devices. Privacy concerns are infrequent among users, as they typically just assume that their privacy is protected by the manufacturers. This decision is rarely based on evidence, but leads users to buy into potential privacy risks as machine learning can reveal sensitive information from usage data [11].

But what if trusting manufacturers with protecting privacy data in the cloud would not be necessary at all? The scope of this paper is to introduce an alternative to these cloud-based smart home platforms: *Home Assistant*. Therefore we will make the assumption that *users do not want to trust manufacturers with their data* and explore if smart home can work without a cloud that is operated by third parties. In order to examine how smart home devices can endanger the privacy and security of users, the next major section will assess associated risks that cloud-based devices may impose on smart home systems and their users.

3 Risks of Cloud-Based Smart Home Devices

Information security (InfoSec) is a common foundation in many it-systems to ensure basic security requirements. Likewise, smart home systems should comply with these principles and aim to ensure **Confidentiality**, **Integrity** and **Availability** of services and data within the smart home [12]. Those requirements are reasonable to satisfy since devices are likely to be exposed to various threats from inside and outside the home network given that they often will have internet access [12].

This section will focus on curated cases where cloud-based smart home devices endanger these foundational principles. Cloud connections undoubtedly can improve the convenience of smart home systems, but users have to consider potential drawbacks when deploying third-party services.

3.1 Implications of Public Server Connections

Granting smart appliances internet access enables them to establish a connection to external services, such as their manufacturers cloud. Although this communication may be desired by the user for receiving external information, these devices can also deliver their gathered information to outside services. Since some devices collect sensitive data, some users may therefore have concerns about the deployment of smart home devices within their home [13]. This perceived privacy encroachment is often enhanced by nonintuitive and nontransparent device configuration options. Only a few smart home devices allow one to customise the storage locations of collected data and frequently deploy *Zero-Conf configuration processes*, where the manufacturer enforces default security and privacy configurations [13]. Moreover all of this has to be configured on a device-on-device basis what makes configuring large deployments time consuming when a user wants to deviate from the default configuration.

An often overlooked risk of cloud-dependent smart home devices is the eventuality that the device’s manufacturer could turn off their servers any day. This exact scenario is what happened in 2016 to users of a smart home hub made by *Revolv* which had been acquired by *Nest*, a company operated by Google. They decided to drop their support of Revolv products which effectively turned their devices into e-waste overnight [14]. Revolv’s FAQ page declared that all services will be terminated, their app will no longer open, and the hub will no longer be operational (see figure 1) [15].

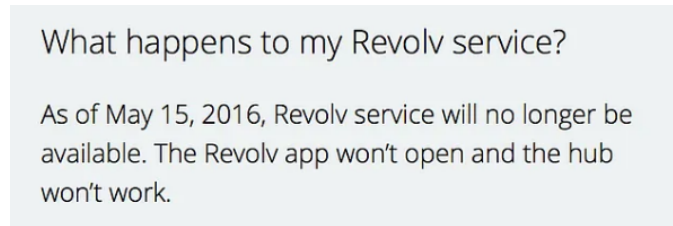


Figure 1: Revolv FAQ about their services [15]

A Revolv user named Arlo wrote an article about his experience on *Medium* and said:

On May 15th, my house will stop working. My landscape lighting will stop turning on and off, my security lights will stop reacting to motion, and my home made vacation burglar deterrent will stop working. This is a conscious intentional decision by Google/Nest.
[15]

This case shows the potential dangers of cloud dependency in smart home systems for availability and longevity of devices. Not every device will stop working once they can not make a connection with their manufacturers cloud, but the services related to it will terminate overnight once the company pulls the plug of their servers.

A more recent example of this happened in 2024 when *Spotify* decided to discontinue *Car Thing*, a product designed to control Spotify in the car that was produced in 2022. Their FAQ site now suggests to simply dispose of the product (see figure 2) [16].

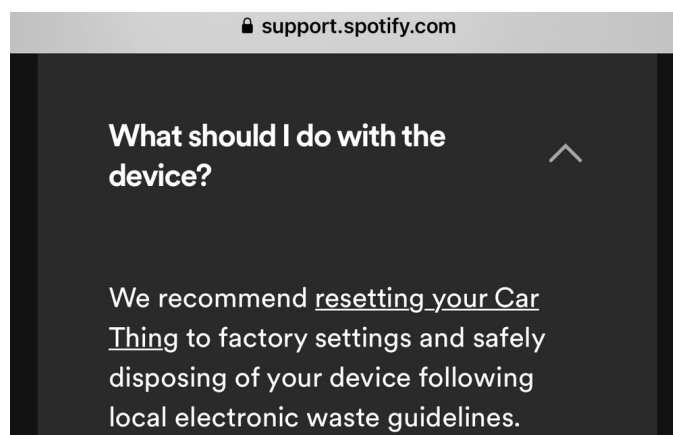


Figure 2: Spotify FAQ for Car Thing [16]

Additionally, the aforementioned consequences would also apply if the user's Internet connection goes down or the manufacturer faces temporary connection issues [2]. Therefore, users should carefully consider if they want to blindly trust any company they deploy devices from within their smart home.

3.2 Unauthorized Access to Smart Home Devices

Features like cloud storage might be perceived as convenient by smart home users, but imply that the collected data is not physically owned by them. Depending on the nature of the stored data and the technologies used to secure it this could have more or less severe consequences.

The video doorbell and surveillance camera company *Ring*, owned by Amazon, has been known for sharing video footage with law enforcement agencies in the United States. Ring introduced various systems where users were able to respond on public or private police requests where they could provide their recordings to help investigations of crimes near their location [17]. While this service was built on the consent of the product owner, Ring also provided video footage on their own in "rare occasions" if police indicated an emergency and therefore Ring disclosed the private recordings of affected users without the law enforcement agencies requiring a warrant [17]. This shows that depending on how a company stores their user data, they might be able to access this data on their own initiative. This case also shows that they might be able to give third parties access to a user's data without their knowledge or consent.

Ring is not the only security camera company that is known to comply with these types of emergency disclosure requests. Google, the company behind *Nest* stated within an interview with *The Verge* that based on their terms of service they would be allowed to send data to authorities without a warrant or consent of the user [18].

However, not every company can or will comply with this requests. Apple also offers to store user video footage on their servers, but uses end-to-end encryption with their *Homekit Secure Video* service [8] and therefore claims not even to be able to comply with these requests due to the technical implications of their encryption technique [18].

3.3 Eavesdropping by Voice Assistants

Smart home voice assistants are known to have many attack vectors and various privacy and security concerns as shown in [19]. While all of them should be taken seriously, this section will focus on the most immediate and user-centred privacy issue at hand that involves cloud-connectivity.

Voice Assistants offer a unique way of interacting with smart home devices with the capability to respond to voice commands, allowing the user a hands-free operation of their smart home [19]. To achieve this they wait for a predefined *wake-word* and record what the user says after that. This recording is then sent to a compute node which analysis the data using *Natural Language Processing* for identifying the users intent that eventually can be used as input for an artificial intelligence to issue following actions [19].

This process involves **two critical points of failure** that can endanger the privacy of users. First as stated in [19] the voice assistant is "*Always On, Always Listening*". The wake-word can be said accidentally or falsely recognised by the device due to any other phonetically similar word during an unrelated conversation. This would trigger the voice assistant to record the following conversation, which consequently *will then be uploaded to the Internet* [19]. This is because the second critical point of failure is the involvement of the manufacturers cloud. Almost all voice assistants are Internet-based systems that process most of their voice processing in their manufacturers cloud, where, moreover, they store transcripts or recording for improving their model. In fact, only the wake-word is recognised locally on the device [19][2].

The implications of this cloud communication and storage of user data were already discussed in the previous sections, and similar concerns arise when reviewing incidents related to voice assistants. In 2018 one specific case was covered widely in the media where Amazon's *Alexa* sent a recording of private conversations to a random contact of a user without being explicitly asked to do so [20]. A representative of Amazon explained: "Echo woke up due to a word in background conversation sounding like 'Alexa.' Then, the subsequent conversation was heard as a 'send message' request. [...]" [20]. The company then declared the case as a malfunction. Later in the same year, an Amazon customer received 1,700 audio files that contained recordings of strangers' voice requests made to Alexa. The information within the audio files together with a Facebook and Twitter search was enough to identify the stranger, who had not been contacted about the incident by Amazon before gaining public attention [21]. Although those two cases are rare incidents,

they show what data voice assistants collect and possibly can expose about their users.

4 Securing a Smart Home against Cloud-Risks

This section will discuss three approaches to reduce user exposure to some of the potential negative consequences of cloud-based smart home devices that were highlighted above.

4.1 Avoiding Cloud Connections

One straight forward approach how a user can reduce cloud connections within their smart home is to deactivate cloud services in the settings of their deployed devices. If it stays operational and the user experiences no drawbacks then the cloud service was optional to begin with and therefore can be easily disabled. This is the case for services like *out of home control*, where users can access their smart home from outside their residential home network. Deactivating those will only impact the user when they need to control the specific device remotely and have no alternative means of control through another smart home controller.

While some manufacturers, like *Bosch* [22] or *Shelly* [23], who both develop various smart home devices for controlling lights, shutters and indoor climate controller, will allow users to disconnect their devices from their cloud services, others like *Philips Hue* will have mandatory cloud connections [24]. This shows that combating cloud connections as an afterthought could potentially be troublesome, as not every device will allow the user to disconnect from all cloud services entirely.

Therefore, considering cloud connections before purchasing a fleet of smart home devices will yield better results to limit the overall amount of outside connections. Carefully selecting the smart home device manufacturers that the user trusts will also help avoid future frustrations in this domain. Moreover, it could be wise to consider if any device without a local API would even be worth deploying within a smart home, since local control will at least work for the entirety of the device's lifespan.

4.2 IoT Network Security

A common security approach to control the network traffic of devices is to establish VLANs that essentially operate as separated virtual networks and to implement powerful firewall rules to block unwanted traffic. Those techniques are also suitable for smart homes to protect against various threats [25]. Furthermore, network segmentation can be used to block cloud connections on the network layer by setting up an *IoT-VLAN* and configuring separate firewall rules for this segment. Smart home devices can be placed on this network, which then can be used not only to isolate devices from the Internet, but also to protect personal devices or sensitive data on the network from being accessed by IoT devices [26]. This can also be taken one step further by deploying *Private VLANs* that isolate devices completely from one another, so that all traffic must go through the router [25].

Some modern routers provide the option to easily configure a separate WiFi network to integrate only IoT devices [26]. Alternatively, the *Guest-WiFi* often serves a similar purpose. However, setting up firewall rules to block smart home devices from connecting to their cloud server requires some technical knowledge from the user. Therefore, network security can be difficult for smart home novices to configure.

Even so, blocking network traffic can be very effective when devices offer a local API but do not allow the user to deactivate all cloud connections.

4.3 Matter Protocol

The *Matter* Protocol is an open-source connectivity standard developed by the *connectivity standards alliance* for controlling smart home devices [27]. It is supported by major industry leaders such as Apple, Google, Amazon and Samsung with the goal of creating an ecosystem where smart home devices can communicate with each other regardless of their manufacturer [28]. It has been designed with state of the art security and privacy principles in mind which among others require Matter devices to be fully controllable locally [27].

While Matter does not rule out cloud connections [27] it ensures that its devices will still be operational in the event that a manufacturer shuts down their cloud services. Therefore, endorsing the Matter protocol elevates the potential danger to end with abandoned devices like in the case of Revolv or Spotify presented in section 3.1.

5 Alternative to cloud-based Smart Home Systems: Home Assistant

Home Assistant is a community-driven **open source** smart home platform that puts **local control** and **privacy first**. It has been designed to run on local servers and mini computers like the [Raspberry PI](https://www.raspberrypi.com/)¹ without depending on the cloud [29].

The project was started by Paulus Schoutsen in 2013 with the initial aim to take greater control of the Philips Hue devices that he had purchased at that time. Over the course of the coming years and with the help of mainly volunteer contributors, Home Assistant has grown to become one of the leading smart home platforms [30] and the most widely used home automation system for DIY enthusiasts.

Home Assistants fundamental vision is to build a smart home that protects the values of what they call **The Open Home**:

The Open Home is about **privacy, choice** and **sustainability**. [...]

Privacy for the Open Home means that devices need to work locally. No one else needs to know if you turn on a light bulb or change the thermostat.

It is okay for a product to offer a cloud connection, but it should be extra and opt-in. [...]

Choice for the Open Home means that devices need to make the gathered data available through local APIs. This avoids vendor lock-in and allows users to create their own smart home with devices from different manufacturers. [...]

Sustainability for the Open Home means that devices are designed and built to keep working and use as little energy as possible. Not just this year, but for the next decade. If they outlive their original purpose, they should be able to be reused or repurposed for something else. [...]

— Paulus Schoutsen about *The Open Home* [31]

In 2024 the core group of developers decided to protect and codify their ideals of the Open Home and created the *Open Home Foundation*, a non-

¹<https://www.raspberrypi.com/>

profit organisation that acts as a formal home for open source projects so that they are protected from future buyouts and abandonment. The Open Home Foundation holds ownership of more than 240 smart home projects, standards, drivers, and libraries. Home Assistant being one of them [30].

All of this is funded by the for-profit organisation *Nabu Casa*, which was also founded by the creators of Home Assistant. Nabu Casa is the legal entity that employs the developers that work full-time on Home Assistant. Furthermore, the company hosts *Home Assistant Cloud*, which provides easy access to services like end-to-end encrypted out-of-home control, Google Assistant or Alexa connection (more on that later), and develops special hardware made for Home Assistant [32]. However, all these services can also be configured without Home Assistant Cloud, but require technical know-how of the user.

The information in this section applies to Home Assistant version 2024.7.

5.1 Installation and Deployment

Home Assistant can be installed in various ways and on almost every system. Nabu Casa even developed two distinct hubs that come with Home Assistant preinstalled. The official documentation distinguishes *five levels of difficulty* [33]:

- *Easiest*: [Home Assistant Green](#)² (a Nabu Casa device)
- *Easy*: Raspberry Pi
- *Intermediate*: [Home Assistant Yellow](#)³ (a Nabu Casa device)
- *Hard*: Other Hardware - ARM or x86-64 machines
- *Expert*: Advanced Installation Methods like [Docker](#)⁴, [Proxmox](#)⁵ or [Python virtual environment](#)⁶

²<https://www.home-assistant.io/green>

³<https://www.home-assistant.io/yellow>

⁴<https://www.docker.com/>

⁵<https://www.proxmox.com/de/>

⁶<https://docs.python.org/3/library/venv.html>

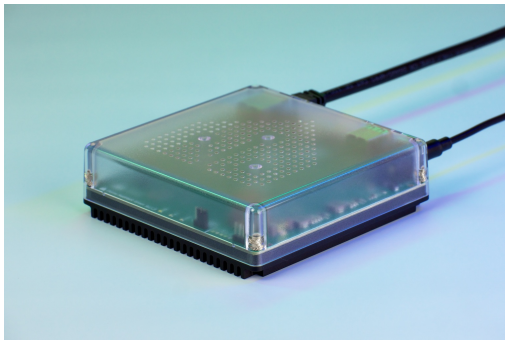
We will go into further detail on installing Home Assistant on Home Assistant Green, Home Assistant Yellow, and on a Raspberry Pi.

The recommended installation methods advice to install the *Home Assistant Operating System* (HA OS) [34], which is a Linux based operating system that uses Docker as a container engine. Further installation options for virtualized environments are available [33], but out of scope for this section. Refer to the [Home Assistant installation documentation](https://www.home-assistant.io/installation/)⁷ for further details on advanced installation methods.

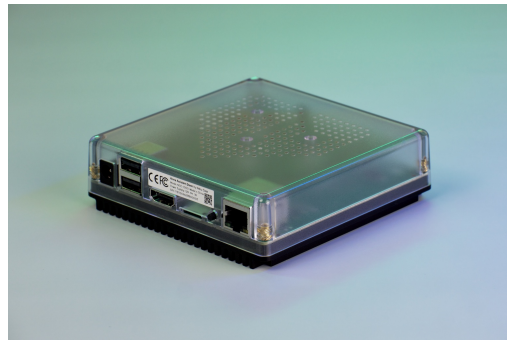
5.1.1 Home Assistant Green

Home Assistant Green (HAG) is the easiest way to get started with Home Assistant. It is a plug-and-play solution that comes with Home Assistant preinstalled. The HAG is a beginner-friendly device with decent compute power for running Home Assistant. It is available at a recommended MSRP of \$ 99 and can be bought from various vendors [35]. The german vendor *mediarath* [36] sells the HAG for around € 105.

Once the device is connected to power and the home network via an ethernet cable, it can be set up using either the Home Assistant smartphone app or the web browser by visiting the url `http://homeassistant.local:8123`, which is a mDNS record pointer that redirects the user to the IP address of the HAG [35]. Once connected, the user is greeted with the onboarding process and asked to create a local admin account.



(a) front of HAG [35]



(b) back of HAG [35]

Figure 3: Home Assistant Green front and back side

⁷<https://www.home-assistant.io/installation/>

5.1.2 Home Assistant on Raspberry Pi

Installing Home Assistant on a Raspberry Pi is a DIY version of buying a Home Assistant Green. Assuming that the user wants to install the HA OS, the only additional tasks that they are required to perform is to flash the operating system to a micro SD card and assemble the Pi. The easiest way to flash the OS is to use the official [Raspberry Pi Imager](https://www.raspberrypi.com/software/)⁸ that supports Home Assistant as a pre-configured operating system [37]. Apart from flashing the SD card, the installation is identical to HAG.

For the Pi itself Home Assistant recommends to either use an [Raspberry Pi 5](https://www.raspberrypi.com/products/raspberry-pi-5/)⁹ or [Raspberry Pi 4](https://www.raspberrypi.com/products/raspberry-pi-4-model-b/)¹⁰, but for most cases even a [Raspberry Pi 3 B](https://www.raspberrypi.com/products/raspberry-pi-3-model-b/)¹¹ should be fine [37], although the system could become slower once Home Assistant gets used more heavily.

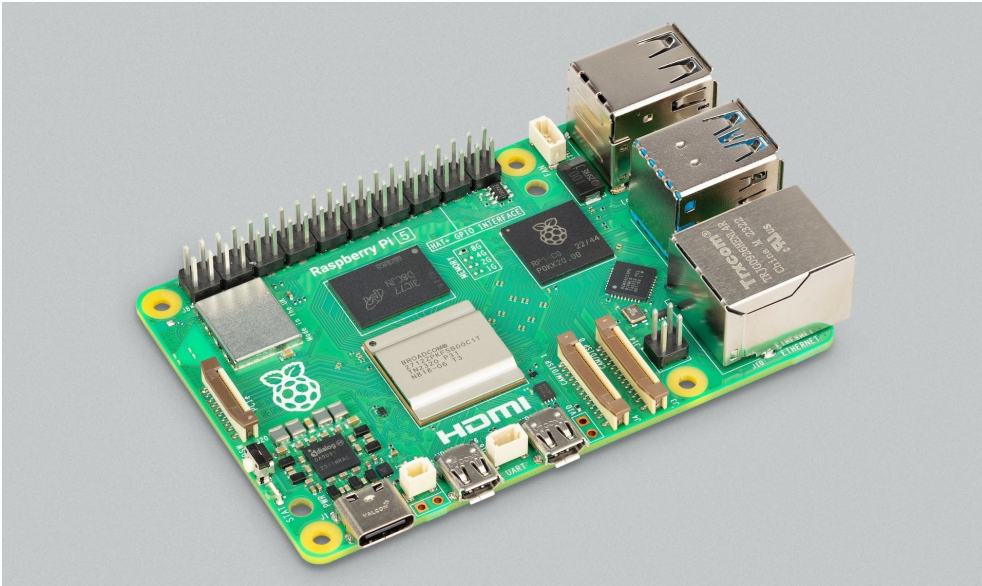


Figure 4: Raspberry Pi 5 [38]

⁸<https://www.raspberrypi.com/software/>

⁹<https://www.raspberrypi.com/products/raspberry-pi-5/>

¹⁰<https://www.raspberrypi.com/products/raspberry-pi-4-model-b/>

¹¹<https://www.raspberrypi.com/products/raspberry-pi-3-model-b/>

5.1.3 Home Assistant Yellow

The Home Assistant Yellow (HAY) is the advanced option of Home Assistant Green. It was designed to be an upgradable and extendable smart home hub that provides advanced features like a M.2 expansion slot and a [Zigbee](#)¹² & [Thread](#)¹³ multi-protocol radio, which are both wireless protocols for IoT-devices. The core of the HAY is a [Raspberr Pi Compute Module](#)¹⁴ that can be swapped out in the future to increase the compute power of the hub [39]. The HAY is not as widely available as the HAG and is sold at a price roughly around € 150 [40] (depending on the vendor and configuration).

The HAY can be bought in two flavours as either a preassembled device with Home Assistant already installed, or as a standalone kit without the Raspberry Pi Compute Module. However, the kit has the advantage that it is also available as a power-over-ethernet option. The preassembled device offers the same easy starting experience as the Home Assistant Green, where just a power and network cable must be plugged in. The kit requires the user to not only assemble the device, but also flash the operating system similarly to the Raspberry Pi [39].

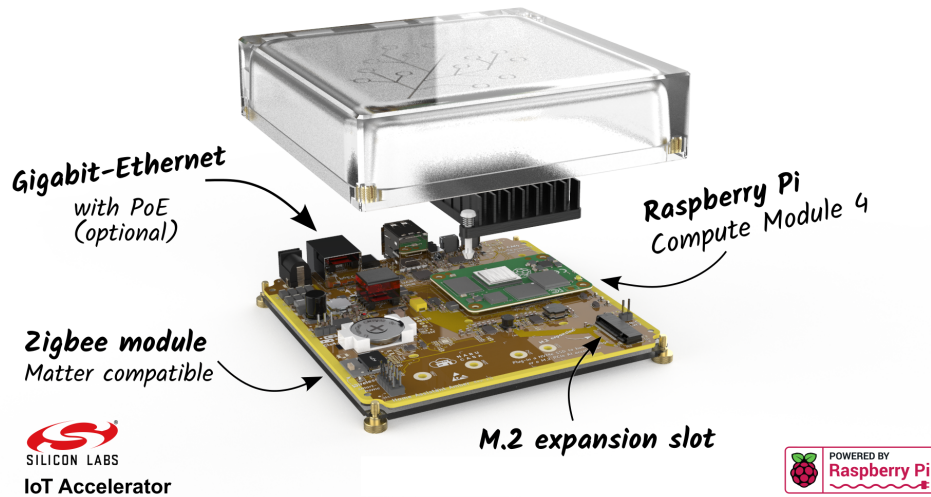


Figure 5: Home Assistant Yellow [39]

¹²<https://csa-iot.org/all-solutions/zigbee/>

¹³<https://www.threadgroup.org/>

¹⁴<https://www.raspberrypi.com/products/compute-module-4/>

5.2 Concepts and Terminology

Home Assistant generally uses a similar terminology to other smart home platforms, but there is more depth to commonly used concepts. For example, most smart home platforms just know *devices* themselves, but in HA devices consist of multiple *entities*. To establish an understanding of those concepts in HA, the following paragraphs provide a brief overview of the most important ones.

Integrations To bring devices into HA, software modules called *integrations* are required that allow HA to connect to external hardware or software controllers. This could be a bridge (e.g. Philips Hue Bridge) or devices themselves (e.g. Shelly relays). Integrations enable third party devices to be controlled by Home Assistant [41]. They provide connections to devices with proprietary vendor-specific protocols, connectivity standards such as [Matter](https://csa-iot.org/all-solutions/matter/)¹⁵ or Zigbee, but also to cloud services.

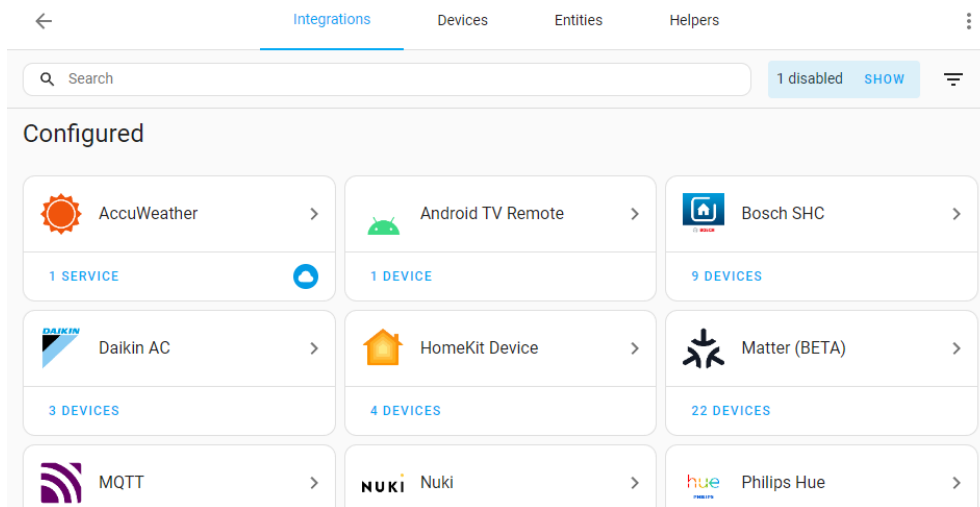


Figure 6: Example of configured integrations in Home Assistant

Entities Capabilities of devices are encapsulated in *entities* that each house or control a single physical property or functionality of a device. The stored information is exposed as the entity’s state. A state can only be one value at a time, but further information related to the current state can be stored in the entities attributes [41]. Moreover, a entity is uniquely distinguishable

¹⁵<https://csa-iot.org/all-solutions/matter/>

by its *entity ID* and is a member of a entity domain. For example, a light facet (a device) would have an entity of the "light" domain with either a state of "on" or "off" that controls the output of a single light source. The brightness and color values of this light could then be stored as attributes that update as the state changes. If the facet has multiple independent light sources, they each would be exposed as their own entities.

Devices One *device* typically consists of multiple entities and is associated with an integration. Devices can therefore be viewed as a logical grouping of entities that represents a physical or logical unit of some type but most commonly refer to physical devices/objects [41]. For example, a motion sensor often monitors additional attributes of a rooms like brightness or humidity, but also has internal attributes like the battery state or device temperature. Each of these sensors would be exposed as a single entity with its own state, but all are grouped *in* the device and shown as one physical object.

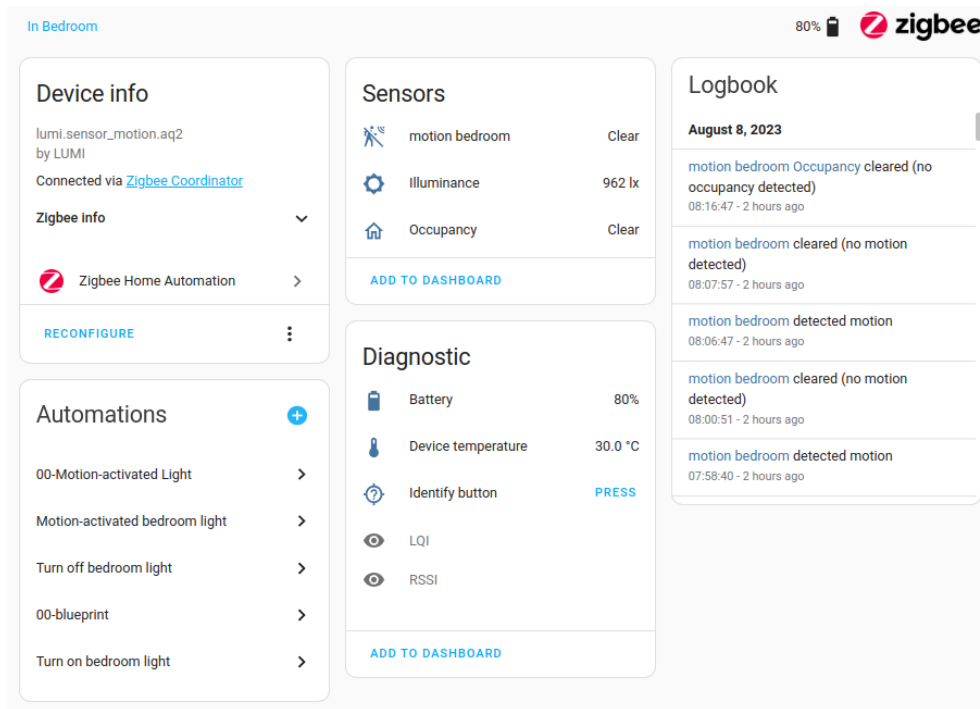


Figure 7: Example of a motion sensor device shown in Home Assistant (Modified from [41])

Areas *Areas* are another level of logical grouping for devices and behave similar to *rooms* in other smart home platforms. They are meant to represent physical rooms of the user’s home and can also be assigned to a floor [41].

Service calls *Service calls* roughly compare to an API interface for a smart home. A service performs one specific action like turning on a light and is registered to a domain of entities that it can be called on. Services are a powerful tool for automations and scripts, but can also be used for controlling many devices at once. A service can target specific entities, devices, areas and labels (another user-defined level of grouping) [42]. While services are not meant to be manually called on a daily basis, they are vital for automating tasks. See Figure 8 for an usage example.

The screenshot shows the 'Service' configuration page in Home Assistant for the 'Light: Toggle' service. The page has a header with the service name and a close button. Below the header is a description: 'Toggles one or more lights, from on to off, or, off to on, based on their current state.' There is a 'Targets' section with a list of selected targets: 'Living Room', 'Desk', and 'Terrace Light'. Below the targets are four buttons: '+ Choose area', '+ Choose device', '+ Choose entity', and '+ Choose label'. The 'Advanced options' section is expanded, showing four options: 'Transition' (checkbox), 'Color' (checkbox), 'Color temperature' (checkbox), and 'Brightness' (checkbox). Each option has a description and a slider. The 'Transition' slider is set to 1 second. The 'Color' slider is set to black. The 'Color temperature' slider is set to 2700 Kelvin. The 'Brightness' slider is set to 100%. At the bottom of the page, there is a 'GO TO YAML MODE' link and a 'CALL SERVICE' button.

Figure 8: Example of a service call in Home Assistant

Automations At the surface, *automations* in HA work and behave identical to other smart home platforms. However, they offer unparalleled possibilities and flexibility for automating even the most complex tasks by leveraging logical operators, loops, blocks for waiting on additional triggers, variables and so much more. Automations are assessed in further detail in section 5.6.

Scripts A *script* is similar to an automation, but is not run automatically by a trigger. Instead it is manually run by the user or by another automation. Scripts can also be exposed on the dashboard of HA [41].

Scenes *Scenes* are a common concept in smart home and in HA they work identically to other platforms. They set an environment to a predefined state, without having to control every device individually [41].

Add-ons If HA was installed using one of the methods described in section 5.1, the system supports the installation of *add-ons*, which are containerised third-party applications that offer additional functionality. In contrast to integrations that connect HA to other apps/bridges, add-ons host new applications [41]. More about add-ons in section 5.9.

5.3 Device Support

One of the main strengths of Home Assistants is the vast support they receive from the open source community. This particularly shows in the number of integrations that are available for HA. In total, the [official integration store](#)¹⁶ offers **2831 unique integrations** [43] (as of July 8, 2024).

However, this is only the set of *officially* supported integration. Home Assistant can be extended by anyone at any time and one of this extensions is [HACS](#)¹⁷ (Home Assistant Community Store) [44], which houses many further integrations. Therefore, chances are that if a device is supported by any smart home platform, it would be Home Assistant.

¹⁶<https://www.home-assistant.io/integrations>

¹⁷<https://hacs.xyz/>

5.4 Out-of-home Control

Since everything in Home Assistant runs locally, remote access is more difficult than on other smart home platforms. In general, there are three approaches for setting up out-of-home control [45]:

- **Home Assistant Cloud** - the HA subscription service that provides, amongst other things, end-to-end encryption for remote access
- **VPN** - secure remote connection into the own home network
- **Port forwarding** - exposing Home Assistant to the internet

Although all of these options are possible to deploy securely, port forwarding requires some technical proficiency and is not suitable for every installation. Therefore, it is not recommended for the average user. To give some basic understanding of how the three options are deployed in Home Assistant, the following paragraphs will briefly touch on the matter.

Home Assistant Cloud This is the easiest option for remote access, but requires a subscription at Nabu Casa. Home Assistant Cloud provides the user with a unique remote URL and a certificate to encrypt the traffic between the user's devices and the HA instance. The subscription itself costs € 7.50 per month, or € 75 annually [46] and directly funds the development of Home Assistant. Besides creating an account and logging in, this option does not require further setup [45].

VPN A Virtual Private Network (VPN) enables the user to remotely access the resources on their home network as long as they are connected via the VPN service. Home Assistant offers a few options for hosting a VPN server by installing an application via the add-on store. HA features two VPN services on their documentation page that are easy to setup: [Tailscale](https://tailscale.com/)¹⁸ and [ZeroTier One](https://www.zerotier.com/)¹⁹ [45]. Both of their free-tier services are sufficient for most use cases.

¹⁸<https://tailscale.com/>

¹⁹<https://www.zerotier.com/>

Port forwarding Setting up port forwarding means exposing the Home Assistant instance publicly. Therefore, in no circumstances should this be done without setting up proper end-to-end encryption, which requires a *public domain* and a *certificate*. Since some Internet Service Providers only offer dynamic IP addresses, this is easiest done by using a *Dynamic DNS service* [45] like [DuckDNS](https://www.duckdns.org/)²⁰ and a *certificate service* like [Let's Encrypt](https://letsencrypt.org/)²¹, which both are completely free of charge. For setting this up with Home Assistant the user can install the [DuckDNS add-on](https://github.com/home-assistant/addons/blob/master/duckdns/DOCS.md)²² from the add-on store and follow the instructions from their documentation. In essence, after installation, the user is required to register an account at DuckDNS, select a free domain name, copy an API token & domain to the configuration of the add-on, and accept the Let's Encrypt terms of service. Everything else, including renewal of the certificate, is done by the application [47]. After exposing a port on the router the HA instance should be remotely accessible. However, access is of course still protected by user credentials and two-factor authentication.

5.5 Dashboards

Home Assistants' approach on dashboards is somewhat different to most smart home platforms. While e.g. Apple Home exposes all of the users devices on the dashboard as soon as they are added to the smart home system, HA wants the user to *create and layout* their dashboard according to their personal liking and handle device management on a separate device panel in the settings. Moreover, Home Assistant allows the user to create as many dashboards as they want.

This approach offers the advantage that dashboards can be more refined, personalised, and specialised than in most smart home systems, but also means that devices have to be explicitly added to the dashboard, which can be time consuming at first.

Home Assistant provides a live [interactive dashboard demo](https://demo.home-assistant.io/)²³ (also visible in Figure 9) so that everyone can form their personal opinion about this system without requiring HA to be installed in advance. On that website, one can also view different dashboard designs that were made by the community.

A further advantage of dashboards in HA is that everything can be cus-

²⁰<https://www.duckdns.org/>

²¹<https://letsencrypt.org/>

²²<https://github.com/home-assistant/addons/blob/master/duckdns/DOCS.md>

²³<https://demo.home-assistant.io/>

tomised by either the UI or CSS styling. Alternatively, users can also download or create custom themes [48] to style their whole interface.

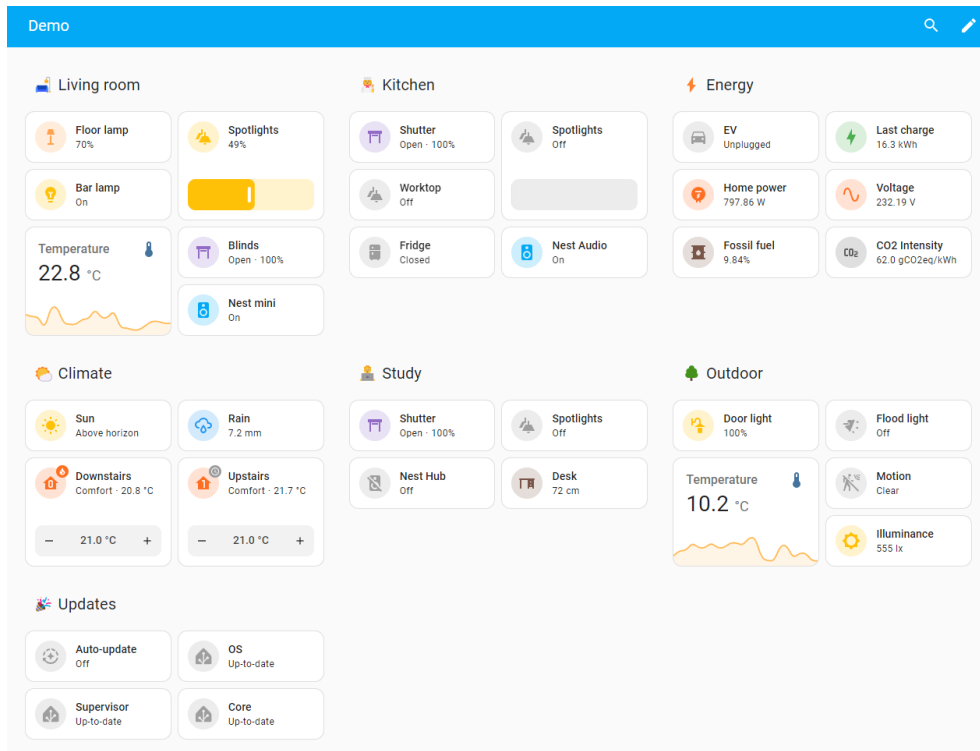


Figure 9: Interactive HA dashboard demo [49]

The dashboard can be adapted by entering the edit mode (pencil icon at the top right corner). In this mode cards can be rearranged by drag and drop, and created by clicking on the marked empty spaces in their section. The edit UI is shown in Figure 10.

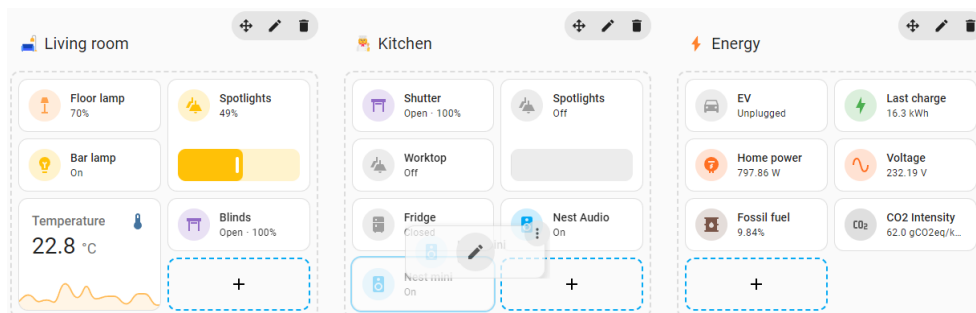


Figure 10: Interactive HA dashboard demo - Editing of a dashboard [49]

Cards can also be configured to the users liking by adding additional control options (called *Features* in the UI), changing the default tap actions when clicking either the icon or the card background and changing the whole size of the card in the grid. Furthermore, they can be shown conditionally when, e.g. an entity has a state or on a per-user basis.

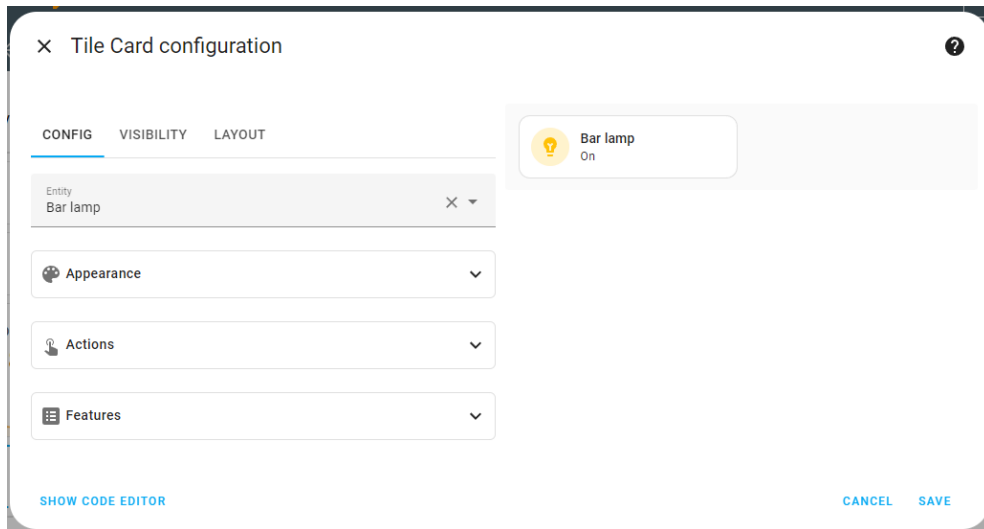


Figure 11: Interactive HA dashboard demo - Editing of a card on the dashboard [49]

In essence, everything on the HA dashboards can be configured and extended, especially when considering adding custom cards from the aforementioned Home Assistant Community Store.

5.6 Automations

In their basic form, automations are made up of three components [42]:

1. **Triggers** - the event that starts the automation
2. **Conditions** - an optional test that has to pass before actions run
3. **Actions** - the tasks the automation should perform

Automations can be triggered by any event in Home Assistant. This includes state changes of devices/entities, on a specific time or time pattern,

location events, scanning NFC Tags or even by triggering webhooks and many more. The conditions are similar to triggers, but relate to the current state, rather than a state change of some entity.

Actions are almost limitless in their capabilities for automation. In addition to controlling devices and calling services, actions include various building blocks that represent loops and logic known from traditional programming like *If-then*, *Repeat* (while-loop), *Variables*, *Choose* (switch statement) and *delays*. Furthermore, there are complex building blocks for waiting on additional triggers or checking for conditions further down in the action sequence, and for running actions in sequence or in parallel.

An easy way to get started with setting up automations is to use [community made blueprints](#)²⁴ which can be imported into HA with a single button from the forums. Later, these blueprints can be configured and edited by the user [50].

5.7 History

Since devices in a smart home generate a lot of data, exploring historic data can be valuable. The *History* panel in Home Assistant allows the user to view the state changes of an entity over time. By default, HA stores the full historic data for ten days (can be modified). After that period, long term statistical data is generated which averages one data point per hour [51]. However, these long term statistics are only generated for entities with continuous variables, like temperature, as storing and condensing discrete states, such as switch statuses, is generally less useful and more challenging.

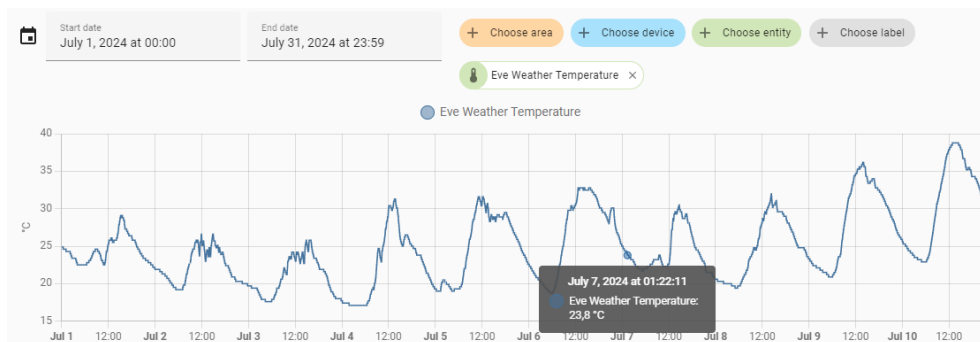


Figure 12: Example of a temperature sensor in the History panel of Home Assistant

²⁴https://www.home-assistant.io/docs/automation/using_blueprints/

5.8 Voice Assistant

Home Assistant offers the possibility to either integrate a third-party voice assistant into HA, or use their own implementation called *Assist*.

Either Google Assistant or Amazon Alexa are supported as third-party voice assistants [52]. They can be integrated manually (DIY option) [53][54] or by using Home Assistant Cloud [32], but the later is the recommended simpler solution.

Assist is the voice assistant interface developed by Home Assistant, but it is rather different to setup than traditional voice assistants. Assist is composed of three components that form the **Assist pipeline**:

- **Conversation agent** - the brain of the voice assistant that processes incoming text commands
- **Speech-to-text** - engine for converting voice to text
- **Text-to-speech** - engine for converting text to voice

Each of these components can be configured or swapped out by the user, and only the conversation agent comes pre-installed with Home Assistant [52]. However, without the other two components Assist can only process incoming text commands and answer with text output.

HA provides two options to solve this. Either subscribe to Home Assistant Cloud and use the compute power of their cloud servers [55], or **host them locally** [56]. Home Assistant provides users the possibility to have full agency over their voice assistant, a feature that no traditional smart home platform offers.

For setting this up locally, the user must install and configure [faster-whisper](https://github.com/SYSTRAN/faster-whisper)²⁵ for the speech-to-text and [Piper](https://github.com/rhasspy/piper)²⁶ for the text-to-speech component [56]. Both are available in the add-on store inside of Home Assistant. Afterwards, the Assist pipeline can be re-configured to use the two components. Although this configuration can even be hosted on a Raspberry Pi, more powerful hardware is advised for achieving comfortable response times. For example on a Raspberry Pi 4 faster-whisper takes eight seconds to process incoming voice commands. On a Intel NUC, this can be done in under one second [56].

²⁵<https://github.com/SYSTRAN/faster-whisper>

²⁶<https://github.com/rhasspy/piper>

Counterparts to stationary devices similarly to e.g. an Amazon Echo can also be installed for using the Assist as a traditional voice assistant. These voice satellites can be build with, amongst others, an ESP-based device [52] like the [ESP-S3-BOX-3](#)²⁷ [57], but even tiny boards like the [ATOM Echo](#)²⁸ that is available for as little as \$ 13 work with Assist [58].

For these voice satellites Home Assistant supports wake-word detection for triggering Assist. Depending on the computing power of the device this can either be done by the HA server with the [openWakeWord](#)²⁹ add-on, or on-device with [microWakeWord](#)³⁰ [59]. Furthermore, the model can be trained by the user to use custom wake-words [60].

5.9 Advanced Features: Beyond traditional Smart Home

Home Assistant is built by the open source community, which is infamous for creating customisable and open software. On the past few pages, many different features were described, but HA has so much more to offer. Especially for users that want to deeper engage with their smart home. In the following few paragraphs, some of those advances topics will be briefly touched on.

Add-ons Various add-ons such as Tailscale, DuckDNS, openWakeWord, and HACS were already mentioned previously and all show what additional capabilities third-party applications can add to Home Assistant. The add-on store enables these to be up and running in seconds, and they often do not require complex configuration. Further interesting add-ons to play with are, for example [Grafana](#)³¹, which enables advanced data visualisation of the locally stored sensor data, or [Grocy](#) that is a self-hosted grocery and household management solution similar to an enterprise resource program. All of these are open source software that can be used together with HA, which in itself is a major advantage of using Home Assistant as a smart home platform.

²⁷https://www.home-assistant.io/voice_control/s3_box_voice_assistant/

²⁸https://www.home-assistant.io/voice_control/thirteen-usd-voice-remote/

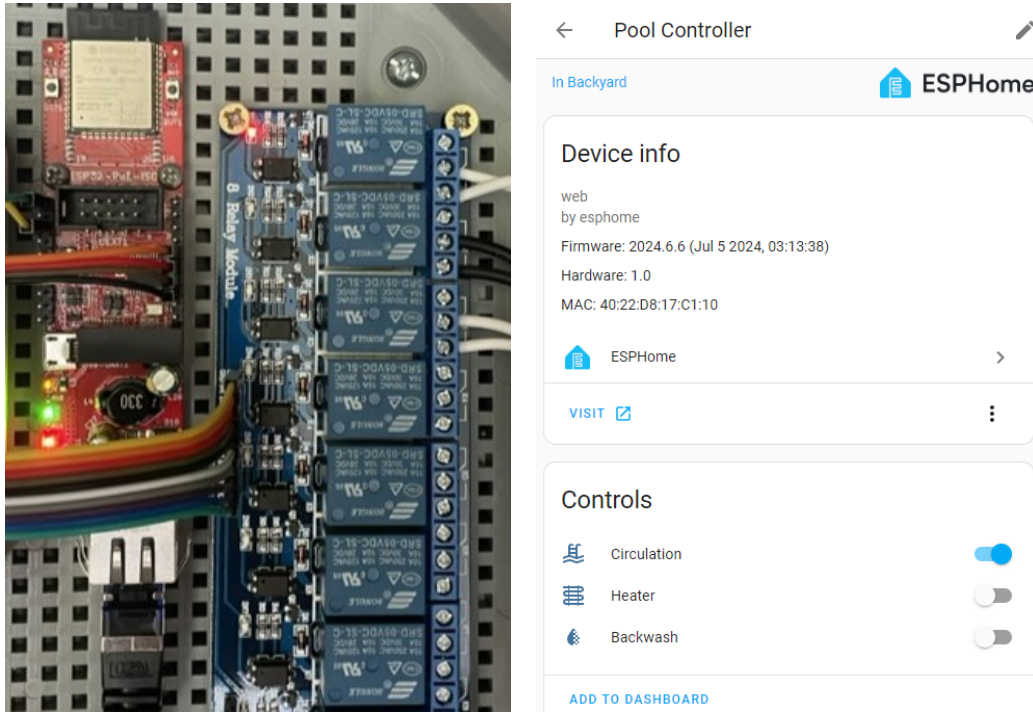
²⁹<https://github.com/dscripka/openWakeWord>

³⁰<https://github.com/kahrendt/microWakeWord>

³¹<https://grafana.com/>

Data Science A smart home generates large amounts of data, which can be explored using data science in Home Assistant. For example, [JupyterLab](https://github.com/hassio-addons/addon-jupyterlab)³² can be installed as an add-on in HA, enabling jupyter notebooks to access the HA database. Users interested in data science can follow the introduction and quick-start guide at the [Home Assistant Data Science Portal](https://data.home-assistant.io/docs/quick-start)³³ to learn how to experiment with their own smart home data.

ESPHome The creators of Home Assistant also developed [ESPHome](https://esphome.io/index.html)³⁴, a system to control microcontrollers by writing simple configuration files. Furthermore, the devices integrate seamlessly into Home Assistant. This project enables DIY enthusiasts to control various appliances with e.g. ESP32 boards, expose them in HA, and update them easily [61]. For users interested in DIY projects, Home Assistant is a viable tool to integrate these projects into their smart home. An example of this is shown in figure 13, where an ESP32-POE board is exposed in Home Assistant using the ESPHome project.



(a) ESP32-POE with relay board

(b) ESPHome device exposed in HA

Figure 13: DIY pool controller running ESPHome

³²<https://github.com/hassio-addons/addon-jupyterlab>

³³<https://data.home-assistant.io/docs/quick-start>

³⁴<https://esphome.io/index.html>

6 Comparison of Smart Home Platforms

The aim of this chapter is to examine whether Home Assistant, as a locally hosted service, delivers a comparable set of capabilities in relation to traditional smart home platforms. For this the platforms were tested in an experiment and compared using a utility analysis. The traditional smart home platforms tested in this comparison are Apple Home, Google Home, Samsung Smart Things and Amazon Alexa. Capabilities were evaluated by comparing the available **control options**, the offered **feature set**, the potential for configuring powerful **automations** and the abilities of the **voice assistant** associated with the respective smart home platform. Control options and Features represent *active tools* for users to interact with their smart home (e.g. controlling, monitoring, or gathering information), while Automations and Voice Assistant represent *passive tools*, where the smart home executes actions on behalf of the user. Although this framework does not cover all dimensions of smart home technology, such as supported protocols, reliability or system cost, its aim is to cover a basic smart home usage scenario. The mentioned attributes were furthermore deliberately ignored for this comparison, since they are heavily influenced by the protocols in use as shown in [62].

All smart home platforms were tested with at the time of writing (24th of June 2024) officially supported (not deprecated) devices with the latest firmware installed. Except of the device used for Amazon Alexa, all devices are still sold by their respective companies and therefore could be newly deployed by a user who starts with smart home today.

6.1 Evaluation Criteria

The comparison is split in the aforementioned dimensions, which criteria are further specified in

- Table 1: Specification for Comparison of Control Options
- Table 2: Specification for Comparison of Features
- Table 3: Specification for Comparison of Automations
- Table 4: Specification for Comparison of Voice Assistants

Additional insight on the distinct smart home platforms can be found in section 6.2.

Table 1: Specification for Comparison of Control Options

Control Options	
Criteria	Requirement
Smartphone App	Offer a dedicated smartphone app
Cross-Platform Apps	Apps are available for iOS and Android phones
Web Interface	Smart home can be controlled via a web browser
Voice Assistant	Platform offers its own distinct voice assistant
Offline-Control	The interface can be controlled locally and therefore without an active internet connection
Out-of-Home	The interface is available while the user is outside their residential network

Table 2: Specification for Comparison of Features

Features	
Criteria	Requirement
Scenes	Change the state of multiple devices of different device-families simultaneously
Automations	Perform event-driven actions based on triggers
Combine Accessories	Group devices so that they can be controlled together
Combine Actions (Scripts/Flows)	Execute a sequence of actions with conditional logic
Locate Devices/People	Assist in finding phones or other tracked objects
Third-Party Extensions	Extend the base application with capabilities from third parties
Device State History	Show a record of state changes of specific devices over time
Event Logs	Show a record of activities, actions, or changes within the system
Energy Monitoring	Measure, record, and visualise the energy usage of devices over time

Table 3: Specification for Comparison of Automations

Automations	
Criteria	Requirement
Time based trigger	Turn on the garden lights at 8pm.
Location based trigger	Lock the door when the user leaves the house.
Send notifications based on events	Send a notification if the door gets opened after 10pm.
Simple Conditional Logic	Turn a light on if motion gets detected after 10pm.

Table 4: Specification for Comparison of Voice Assistants

Voice Assistants	
Criteria	Requirement
Obtain external information	“How is the weather today?” – provide forecast
Perform actions	“Turn off the lights in this room.” – turn off lights
Set a reminder	“Remind me to go grocery shopping tomorrow 10am.” – add a reminder
Set a timer	“Set a timer for 10 minutes.” – sound an alarm in 10 minutes
Identify voice of household members	“Who am I?” – enable personal requests by distinguishing voices

6.2 Insight into the compared Platforms

This section will briefly describe the most relevant key-facts of each smart home platform and point out important information for the final comparison.

6.2.1 Apple Home

Apple Home (AH) has been tested using a Apple HomePod mini with the firmware 17.5 (21L569) and an iPhone 12 Pro with the latest operating system version iOS 17.5.1 installed.

Control Options AH exclusively offers apps for Apple devices and therefore excludes household members that use Android smartphones or Windows PCs. Its voice assistant is called *Siri* which can execute some voice commands locally and communicates with iPhones on the same network to receive personal information, but depends on external servers for more complex tasks. Out-of-home control is available through Apple’s iCloud Service.

Features In terms of features, AH focusses mainly on basic capabilities. Devices of the same type can be grouped to display as a single entity, for example, the individual lights of a chandelier, and scenes can be created to control multiple devices simultaneously. An event history is available, but this is limited to include only smart locks and garage doors. However, more complex features, such as viewing the state history of devices or energy monitoring, are currently missing.

Automations AH supports the most common event triggers for automations like people arriving/leaving a place, accessories changing their state (being controlled or sensors detecting something) and time-based events. The action flow that can be executed within an automation is limited to setting the state of devices. Logical operations like if-then, conditionally waiting on another event or for time to pass are not supported. The only exception are lights, which can be turned off again after a set amount of time.

Voice Assistant Although Siri is commonly perceived as a rather weak voice assistant, it satisfies all the criteria of this test.

6.2.2 Google Home

Google Home (GH) has been tested using a Google Nest Mini with the firmware 2.57.375114.

Control Options GH is mainly focused on controlling smart home devices with their voice assistant *Google Assistant*. Out-of-home control is available through the Google Cloud. However, most smart home devices stop working once the hub loses its connection to the Internet. Presumably only devices that directly connect to the phone, such as some smart TVs, remain controllable. Google recently confirmed that they are working on implementing some form of local control [63].

Features Scenes are not directly integrated into the GH app. External applications can provide scenes to the Google Assistant API [64], but it is not supported to create new scenes within the GH app. Third-party integrations are available with Google Cloud. Energy monitoring is only available for thermostats, but not for power plugs. This means that heat can be monitored, but electricity can not.

Automations GH distinguishes between *Household Routines* and *Personal Routines*, but for this smart home comparison only the household routines are relevant. Automation triggers can be voice commands, a specific time or an alarm on a smart speaker. Noteworthy about GHs automation is that they provide a web interface for configuring advanced routines.

Voice Assistant GH is heavily focused on using its voice assistant as the primary interaction interface and had no shortcomings in the comparison.

6.2.3 Samsung Smart Things

Samsung Smart Things (SST) has been tested using a Samsung The Frame TV with firmware 1455.3.

Control Options SST does not have distinct voice assistant satellites like, for example, an Amazon Echo Dot, but allows users to connect external voice services like Samsung Bixby, Google Assistant or Amazon Alexa. SST also provides a web interface to control devices from the browser. In the event of an Internet outage, SST behaves inconsistently. Some automations work offline on the hub, but others require a cloud server [65].

Features SST provides every feature defined in this comparison. The event history allows to filter for specific devices and therefore its implementation counts as a device state history, although it is not explicitly labeled as such.

Automations Automations can be triggered based on manual actions, time, device state, location, weather condition and its integrated security mode (which mimics an alarm system). SST allows to set triggers as preconditions that have to be fulfilled for other triggers to execute the automation.

Voice Assistant As mentioned above, SST does not provide dedicated voice assistant satellites and Samsung Bixby is only available on mobile devices such as smartphones or built into Samsung's own home appliance devices such as refrigerators or soundbars that can not be as easily placed in every room of a house as typical voice satellites. Since it lacks this characteristics, Bixby will not be treated as SSTs voice assistant. Instead, it will be assumed that integrating Google Assistant or Alexa with SST is the more common approach a user would take when building a smart home with this platform. For this reason, Google Assistant will be used together with Samsung Smart Things in this comparison.

6.2.4 Amazon Alexa

Amazon Alexa (AA) has been tested using a Amazon Echo (2nd Generation) with the firmware 10369674116

Control Options AAs primary control interface is its voice assistant *Alexa*. A major point of concern of this platform is that it stops working completely if the device loses Internet connection. Even the smartphone app becomes unresponsive and not even local devices can be controlled through the app in this event. Out-of-home control is available through the Amazon Cloud.

Features Out of the box AA offers many features and can even be heavily extended further with third-party integrations called *Alexa-Skills*. AA shows an activity history within its app that combines the event history with the device state history. However, it lacks the ability to filter those events by devices, and therefore the view can clutter up rather quickly. This implementation serves more as an event log than as a full-backed device state history.

Automations Automations are called *Routines* which offer the ability to be triggered by custom voice commands in addition to traditional time based triggers or device state changes. Based on various user reports on the amazon forums, AA cannot perform location-based automations in various European countries, or apparently only in the USA [66]. The action flow of automations is similarly limited as in Apple Home, but AA requires a second automation for turning a light off again after it has been turned on by an automation.

Voice Assistant AA is primary developed as a voice assistant, and therefore it unsurprisingly excels at those tasks.

6.2.5 Home Assistant

Home Assistant (HA) has been tested with version 2024.7 on a Raspberry Pi 4. Detailed information on HA can be found in section 5.

Control Options Home Assistant provides all control options of this comparison. While out-of-home control requires some additional configuration (see section 5.4), it is not too difficult to set up.

Features HA also supports every feature of this comparison. The features of Home Assistant were elaborated thoroughly in its dedicated section.

Automations In addition to basic automation capabilities HA offers complex conditional logical and loops known from most programming languages. Despite that, automations can be configured using the UI using building blocks. See section 5.6 for further details.

Voice Assistant The voice assistant of HA (*Assist*) has been tested using a fully local installation of the required components that were described in section 5.8. Overall it is the least capable of the voice assistants in this comparison. However, just like SST, HA can integrate with Google Assistant and Amazon Alexa, which can be a viable substitute if local processing is not a priority.

6.3 Evaluation

The evaluation is split into four subscores, one for each of the aforementioned specifications explained in Table 1 (C-Score), Table 2 (F-Score), Table 3 (A-Score) and Table 4 (V-Score). One point is credited for each criteria that the smart home platform meets (marked by ✓).

The total score for this comparison is then calculated as the total sum:

$$\textbf{Total Score} = \text{C-Score} + \text{F-Score} + \text{A-Score} + \text{V-Score}$$

The results of this comparison can be found in table 5 on the next page. However, the total score should not be viewed as an indicator for how "good" a smart home platform is. The score only shows how rich in capabilities they are, given a basic usage scenario. Furthermore, the evaluation does not weight the importance of each feature, nor is every criteria equally distinct. Some of them are more similar to another than others and some features could be more important to user A than to user B. Moreover, it should be noted that the criteria were selected in such a way that subtle differences between the platforms can be observed, while unique features were deliberately omitted, so that no platform would artificially score higher than others due to supporting niche use cases. The goal of the comparison is to compare Home Assistant against the traditional smart home platform and not to determine the smart home platform with the most capabilities. Therefore, this should not be viewed as a ranking system.

Furthermore, Samsung Smart Things and Home Assistant can each use Google Assistant and Amazon Alexa as their voice assistant, which would alter the score of the configured system. However, for this comparison Home Assistant will entirely be evaluated with an fully local setup, since the main proposal of this paper is to deploy Home Assistant as an alternative to cloud-based smart home systems. All other platforms have access to their entire cloud infrastructure.

Note: *AH = Apple Home, GH = Google Home, SST = Samsung Smart Things, AA = Amazon Alexa, HA = Home Assistant*

Table 5: Comparison of Smart Home Platforms

		Smart Home Platforms				
		AH	GH	SST	AA	HA
Active Tools	Control Options					
	Smartphone App	✓	✓	✓	✓	✓
	Cross Platform Apps		✓	✓	✓	✓
	Web Interface			✓		✓
	Voice Assistant	✓	✓	✓*	✓	✓
	Offline-Control	✓		✓		✓
	Out-of-Home	✓	✓	✓	✓	✓
	C-Score	4	4	6	4	6
	Features					
	Scenes	✓		✓	✓	✓
	Automations	✓	✓	✓	✓	✓
	Combine Accessories	✓	✓	✓	✓	✓
	Combine Actions (Scripts/Flows)			✓		✓
	Locate Devices/People	✓	✓	✓	✓	✓
	Third-Party Extensions		✓	✓	✓	✓
	Device State History			✓		✓
	Event Logs		✓	✓	✓	✓
	Energy Monitoring			✓	✓	✓
	F-Score	4	5	9	7	9
Passive Tools	Automations					
	Time based trigger	✓	✓	✓	✓	✓
	Location based trigger	✓	✓	✓		✓
	Send notifications based on events	✓	✓	✓	✓	✓
	Simple Conditional Logic	✓	✓	✓		✓
	A-Score	4	4	4	2	4
	Voice Assistants					
	Obtain external information	✓	✓	✓*	✓	
	Perform actions	✓	✓	✓*	✓	✓
	Set a reminder	✓	✓	✓*	✓	✓
	Set a timer	✓	✓	✓*	✓	✓
	Identify voice of household members	✓	✓	✓*	✓	
	V-Score	5	5	5	5	3
Total score		17	18	24	18	22

* using Google Assistant

In relation to the research question of this paper, this comparison shows that Home Assistant can indeed be a viable alternative to cloud-based smart home systems based on its capabilities. The primary weakness of Home Assistant at the moment is its voice assistant. Although, comparing a locally hosted service with a cloud service may not be entirely fair. Users who are open to using a cloud-based voice assistant can still integrate services like Google Assistant into Home Assistant, thus addressing this weakness. In this configuration, Home Assistant would then show no shortcomings compared to the cloud-based smart home platforms.

In terms of the defined capabilities, a fully local installation of Home Assistant is therefore rivaled only by Samsung Smart Things. However, this comparison excludes human-focused attributes such as user-experience, which would be subject to future research.

7 Experiment: Adoption of Home Assistant

In this final section, an experiment is conducted to evaluate the adoption process of Home Assistant. The aim is to investigate if the proposed method of deploying an instance of Home Assistant on a Raspberry Pi, as elaborated in section 5.1, is suited for potential smart home users. This would offer an alternative to purchasing a hub as it is necessary for the deployment of other traditional smart home systems. However, this experiment will focus on the experience of a small group of participants, and the results should be interpreted as a collection of first-time user experiences rather than generalised conclusions.

7.1 Experiment Design

Participants will be asked to complete a questionnaire before and after the experiment. The pre-experiment questionnaire will gather information on their general IT skills, as well as their experience with smart home technology and Raspberry Pi. The post-experiment questionnaire will focus on their experience during the setup of Home Assistant. The experiment itself consists of assembling a Raspberry Pi into its case, flashing a micro-SD card with the Home Assistant operating system, setting up the HA instance on the Raspberry Pi and finally connecting and controlling one smart home device using Home Assistant.

A brief overview of the questionnaire can be found in Table 6. The top section contains the pre-experiment questions, while the bottom section holds the post-experiment questions. Additionally, each question provides extra space (not displayed in this table) for participants to describe their experiences further.

Table 6: Experiment Questionnaire

IT Skills:
How would you describe your IT skills?
☐Beginner ☐Intermediate ☐Advanced ☐Expert

Smart Home Technology Experience:
What is your previous experience with smart home technology?
☐None ☐Beginner ☐Intermediate ☐Advanced

Raspberry Pi Experience:
Have you ever used or assembled a Raspberry Pi before?
☐Yes ☐No

Assembly Process:
How would you describe your experience of the assembly process?
☐Very Easy ☐Easy ☐Moderate ☐Difficult ☐Very Difficult

Installation Process:
How would you describe your experience of the installation process?
☐Very Easy ☐Easy ☐Moderate ☐Difficult ☐Very Difficult

Configuration Process:
How would you describe your experience of adding a device to HA?
☐Very Easy ☐Easy ☐Moderate ☐Difficult ☐Very Difficult

Overall Process:
How would you describe your overall experience?
☐Very Easy ☐Easy ☐Moderate ☐Difficult ☐Very Difficult

Note: The full questionnaire as well as the participants answers are available at this papers' github repository (see section 1.2).

7.2 Experiment Evaluation

The experiment involved six participants with varying IT skills and experience with smart home technologies. In particular, two participants had never used a Raspberry Pi, while others briefly used one during their school education. A visual self-categorisation of participants' skills can be found in Figure 14.

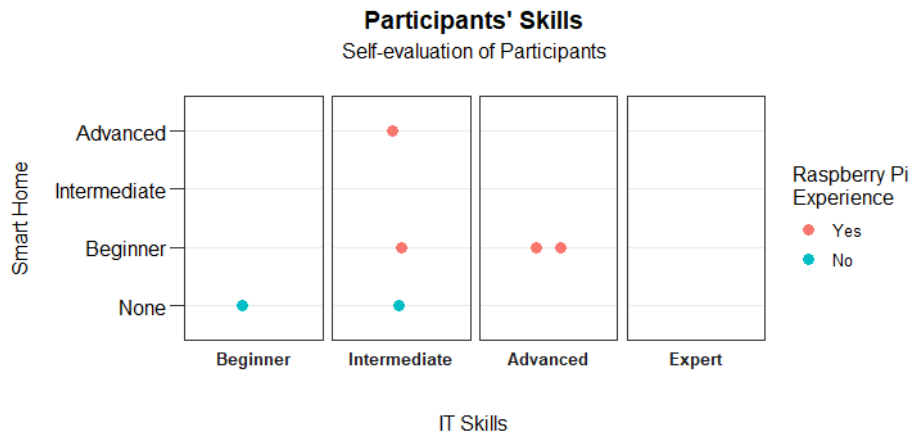


Figure 14: Participants categorised by their IT-Skills, Smart Home, and Raspberry PI experience

Overall, participants found setting up a Home Assistant instance on a Raspberry Pi to be rather easy throughout all phases of the experiment. Only one participant encountered challenges during the configuration process, which included the onboarding process and connecting devices to the instance. They stated that this was mostly due to the overwhelming user interface of Home Assistant. A breakdown of the perceived difficulty of the participants is visualised in Figure 15.

Furthermore, two patterns emerged when comparing the time required to complete the experiment. Participants who had used a Raspberry Pi before took around 12 minutes to perform all tasks, while those new to Raspberry Pi needed a few extra minutes. However, this was mainly due to the extra time that they needed to flash the Home Assistant image on the micro-SD card. The Raspberry Pi assembly process did not present a particularly challenging task. Further information on the time required by the participants for each phase can be found in Figure 16. It should be noted that these times do not

include the preparation time that is needed by the Raspberry Pi to download all dependencies from the linux package repository (up to 20 minutes) since this is idle time for the user and not relevant for this experiment

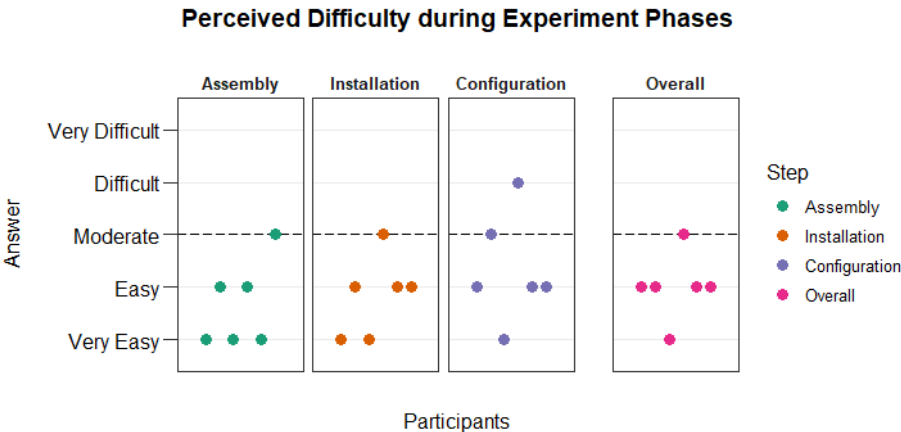


Figure 15: Perceived Difficulty of Participants during Experiment Phases

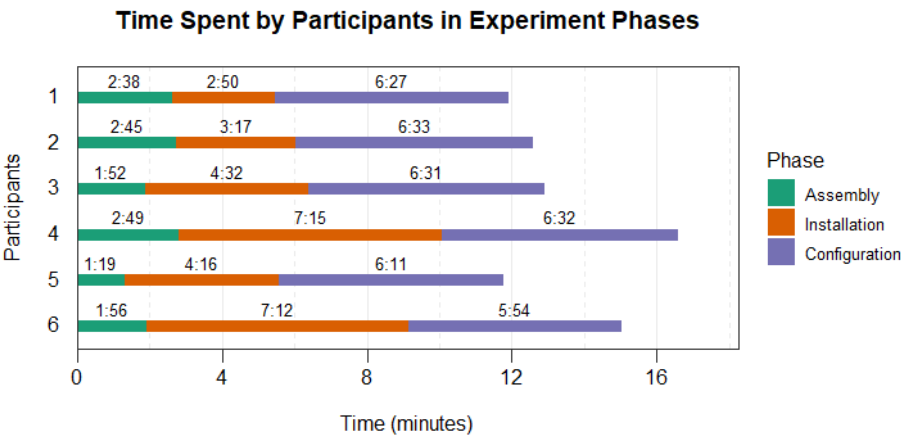


Figure 16: Time Spent by Participants in Experiment Phases

Although the conclusions from this experiment cannot be generalised, the results suggest that installing Home Assistant on a Raspberry Pi is worth trying even for those unfamiliar with Raspberry Pi or other smart home platforms. In particular, users who already own a Pi have the opportunity to

try out Home Assistant at no additional cost with a negligible investment of time and effort. Considering the confusion many participants faced when first encountering the Home Assistant user interface post-onboarding, it is recommended that new users spent some time reviewing the basic documentation pages in order to get familiar with Home Assistants' structure and features.

8 Conclusion

This paper highlighted concerns associated with the cloud-based nature of smart home technologies and devices that have been developed in recent years, which jeopardise the users' privacy and agency over their personal and smart home data. In some cases, revoking access to cloud infrastructure even renders otherwise functional devices as electronic waste. While cloud technology can be very helpful to users, the development of cloud dependencies must be critically assessed by users who expect their devices to remain functional over many years.

With the aim of mitigating the highlighted concerns, this paper analysed the open source project Home Assistant as an alternative to cloud-based smart home technologies. Their system is designed around the vision of *The Open Home* that focuses on privacy, choice, and sustainability by ensuring full local operation and providing full ownership of users' smart home data. This approach not only helps to protect the privacy of users, but self-hosting this system also ensures user autonomy over smart home devices and data.

The functionality of Home Assistant has been thoroughly assessed and investigated how traditionally cloud dependent features can be set up fully locally. This includes out-of-home control and a voice assistant.

In order to serve as a viable alternative, this system has to offer similar capabilities compared to traditional smart home platforms. Therefore, this paper evaluated Home Assistant against Apple Home, Google Home, Samsung Smart Things, and Amazon Alexa, and demonstrated that it is a competitive alternative in terms of its capabilities, while providing a wide range of advanced features that allow users to further enhance their smart home experience and extend their system with community-made or self-developed modules. The only feature that Home Assistant currently cannot easily fully substitute is a voice assistant, where competitors still maintain an advantage due to the resource-intensive nature of voice processing that is most often

done in the cloud.

Furthermore, an experiment was conducted to assess the user experience of adopting Home Assistant as a smart home platform by self-hosting the system on a Raspberry Pi. Regardless of IT proficiency all participants perceived this process to be rather easy than difficult, therefore suggesting that Home Assistant is a viable smart home solution for novice users. Therefore, any user interested in deploying a locally hosted smart home can start using Home Assistant on their own hardware or on a power-efficient minicomputer.

Further Research To evaluate the feasibility of using Home Assistant as an alternative to cloud-based smart home systems in daily life, additional research is required to assess the user experience of its interface and mobile application. This paper focused on the technical capabilities of smart home platforms and therefore deliberately excluded user experience as a evaluation factor. After determining the technical competitiveness of the smart home platform the groundwork for further user-centered research design has been established.

Open-Access: This paper, including the used materials, is available at this public github repository: <https://github.com/dir-rafa/Bachelor-thesis-Home-Assistant>

Acknowledgements

The author of this paper would like to sincerely thank Mr. ao.Univ.Prof. Dr. Johann Mitlöhner for his unwavering support and invaluable feedback throughout the writing of this thesis. Furthermore, the author wants to extend his thanks the open-source community, whose collective efforts help developing and improving Home Assistant and similar projects.

References

- [1] Muhammad Raisul Alam, Mamun Bin Ibne Reaz, and Mohd Alauddin Mohd Ali. “A Review of Smart Homes—Past, Present, and Future”. In: *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)* 42.6 (2012), pp. 1190–1203. DOI: [10.1109/TSMCC.2012.2189204](https://doi.org/10.1109/TSMCC.2012.2189204).
- [2] Chris Bertko. *Home, Smart Home : Der praktische Einstieg in die Hausautomation. Inkl. Marktüberblick: AVM, Belkin, Fibaro, Gigaset, HomeMatic, SCHWAIGER u.v.m.* ger. München: Hanser Verlag, 2017, 1 Online-Ressource (349 Seiten). ISBN: 9783446454248. DOI: [10.3139/9783446454248](https://doi.org/10.3139/9783446454248).
- [3] Jennifer Pattison Tuohy. *How to pick a smart home platform*. 2023. URL: <https://www.theverge.com/23751295/smart-home-platform-google-amazon-apple-samsung> (visited on 05/12/2024).
- [4] Ruiyang Tang and Yuki Inoue. “Services on Platform Ecosystems in the Smart Home 2.0 Era: Elements Influencing Consumers’ Value Perception for Smart Home Products”. eng. In: *Sensors (Basel, Switzerland)* 21.21 (2021), p. 7391. ISSN: 1424-8220. DOI: [10.3390/s21217391](https://doi.org/10.3390/s21217391).
- [5] Jennifer Pattison Tuohy. *What is a smart home, and do you need one?* 2023. URL: <https://www.theverge.com/23749376/smart-home-explained-voice-assistant-tv-gadgets> (visited on 04/22/2024).
- [6] Jennifer Pattison Tuohy. *Smart home hubs: what they are and why you need one*. 2024. URL: <https://www.theverge.com/24087882/smart-home-hub-apple-google-alexa> (visited on 05/16/2024).
- [7] Amazon. *Alexa and Alexa Device FAQs*. URL: <https://www.amazon.com/gp/help/customer/display.html?nodeId=201602230> (visited on 05/16/2024).
- [8] Jacob Kastrenakes and Dan Seifert. *Apple enables HomeKit support for home security cameras and routers*. 2019. URL: <https://www.theverge.com/2019/6/3/18646453/apple-homekit-support-smart-home-security-routers-wwdc-2019> (visited on 05/18/2024).
- [9] Samsung. *The New Face of Homer Energy Monitoring*. URL: <https://partners.smartthings.com/smartthings-energy> (visited on 05/16/2024).

- [10] Shannon Liao. *Alexa Hunches can now suggest actions based on daily behavior*. 2018. URL: <https://www.theverge.com/2018/9/20/17883396/amazon-alexa-hunches-prediction-smart-home-ai> (visited on 05/18/2024).
- [11] Serena Zheng et al. “User Perceptions of Smart Home IoT Privacy”. eng. In: *Proceedings of the ACM on human-computer interaction* 2.CSCW (2018), pp. 1–20. ISSN: 2573-0142. DOI: [10.1145/3274469](https://doi.org/10.1145/3274469).
- [12] Jin-Hee Han, YongSung Jeon, and JeongNyeo Kim. “Security considerations for secure and trustworthy smart home system in the IoT environment”. In: *2015 International Conference on Information and Communication Technology Convergence (ICTC)*. 2015, pp. 1116–1118. DOI: [10.1109/ICTC.2015.7354752](https://doi.org/10.1109/ICTC.2015.7354752).
- [13] Mahsa Keshavarz and Mohd Anwar. “Towards Improving Privacy Control for Smart Homes: A Privacy Decision Framework”. In: *2018 16th Annual Conference on Privacy, Security and Trust (PST)*. 2018, pp. 1–3. DOI: [10.1109/PST.2018.8514198](https://doi.org/10.1109/PST.2018.8514198).
- [14] Nathan Proctor. *Bricked house: How obsolescence looms over our “smart” home devices*. 2020. URL: <https://pirg.org/articles/bricked-house-how-obsolescence-looms-over-our-smart-home-devices/> (visited on 05/20/2024).
- [15] Arlo Gilbert. *The time that Tony Fadell sold me a container of hummus*. 2016. URL: <https://arlogilbert.com/the-time-that-tony-fadell-sold-me-a-container-of-hummus-cb0941c762c1#.tsud8t7hs> (visited on 05/20/2024).
- [16] The Open Home Foundation. *All cloud-based products will come to an end*. 2024. URL: <https://newsletter.openhomefoundation.org/all-cloud-based-products-will-come-to-an-end/> (visited on 07/22/2024).
- [17] Anumita Kaur. *Ring camera app won’t share footage with law enforcement anymore*. 2024. URL: <https://www.washingtonpost.com/technology/2024/01/25/ring-camera-footage-sharing-police/> (visited on 05/20/2024).
- [18] Mitchell Clark. *Google, like Amazon, may let police see your video without a warrant*. 2022. URL: <https://www.theverge.com/2022/7/26/23279562/arlo-apple-wyze-eufy-google-ring-security-camera-footage-warrant> (visited on 05/20/2024).

- [19] Jide S. Edu, Jose M. Such, and Guillermo Suarez-Tangil. “Smart Home Personal Assistants: A Security and Privacy Review”. In: *ACM Comput. Surv.* 53.6 (Dec. 2020). ISSN: 0360-0300. DOI: [10.1145/3412383](https://doi.org/10.1145/3412383). URL: <https://doi.org/10.1145/3412383>.
- [20] Helena Horton. *Amazon Alexa recorded owner’s conversation and sent to ‘random’ contact, couple complains*. 2018. URL: <https://www.telegraph.co.uk/news/2018/05/25/amazon-alexa-recorded-owners-conversation-sent-random-contact/> (visited on 05/23/2024).
- [21] Sasha Ingber. *Amazon Customer Receives 1,700 Audio Files Of A Stranger Who Used Alexa*. 2018. URL: <https://www.npr.org/2018/12/20/678631013/amazon-customer-receives-1-700-audio-files-of-a-stranger-who-used-alexa> (visited on 05/23/2024).
- [22] Bosch. *Hilfe zum Smart Home Controller*. URL: <https://www.bosch-smarthome.com/de/de/service/hilfe/hilfe-zum-produkt/hilfe-zum-smart-home-controller/> (visited on 05/26/2024).
- [23] Shelly. *Can I use the Shelly devices without Shelly Smart Control App?* 2024. URL: <https://support.shelly.cloud/en/support/solutions/articles/103000044277-can-i-use-the-shelly-devices-without-shelly-smart-control-app> (visited on 05/26/2024).
- [24] Jennifer Pattison Tuohy. *Philips Hue will soon require an account to use its app — here’s what that means*. 2023. URL: <https://www.theverge.com/2023/9/28/23892761/philips-hue-app-account-changes> (visited on 05/23/2024).
- [25] Ali Maki AlAali, Abdulla AlAteeq, and Wael Elmedany. “Cybersecurity Threats and Solutions of IoT Network Layer”. In: *2022 International Conference on Innovation and Intelligence for Informatics, Computing, and Technologies (3ICT)*. 2022, pp. 250–257. DOI: [10.1109/3ICT56508.2022.9990734](https://doi.org/10.1109/3ICT56508.2022.9990734).
- [26] Federal Office for Information Security of Germany. *Smart home: connecting your home securely*. URL: <https://www.bsi.bund.de/EN/Themen/Verbraucherinnen-und-Verbraucher/Informationen-und-Empfehlungen/Internet-der-Dinge-Smart-leben/Smart-Home/smart-home.html> (visited on 07/23/2024).
- [27] Naama Bak et al. *Matter Security and Privacy Fundamentals*. 2022. URL: https://csa-iot.org/wp-content/uploads/2022/03/Matter_Security_and_Privacy_WP_March-2022.pdf (visited on 07/23/2024).

- [28] Kumar Shashwat et al. “Security Analysis of Trust on the Controller in the Matter Protocol Specification”. In: *2023 IEEE Conference on Communications and Network Security (CNS)*. 2023, pp. 1–6. DOI: [10.1109/CNS59707.2023.10288747](https://doi.org/10.1109/CNS59707.2023.10288747).
- [29] Home Assistant. *Awaken your home*. URL: <https://www.home-assistant.io/> (visited on 06/28/2024).
- [30] Open Home Foundation. *About the Open Home Foundation – Open Home Foundation*. URL: <https://www.openhomefoundation.org/about/> (visited on 06/28/2024).
- [31] Paulus Schoutsen. *The Open Home*. 2021. URL: <https://www.home-assistant.io/blog/2021/12/23/the-open-home/> (visited on 06/28/2024).
- [32] Nabu Casa. *Home Assistant Cloud*. URL: <https://www.nabucasa.com/> (visited on 07/01/2024).
- [33] Home Assistant. *Installation*. URL: <https://www.home-assistant.io/installation> (visited on 07/01/2024).
- [34] @home-assistant. *Home Assistant Operating System*. URL: <https://github.com/home-assistant/operating-system> (visited on 07/05/2024).
- [35] Home Assistant. *UPGRADE YOUR SMART HOME*. URL: <https://www.home-assistant.io/green> (visited on 07/05/2024).
- [36] mediarath. *Home Assistant Green Smart Home Hub, 2X USB, Gigabit LAN, HDMI, MicroSD Slot*. URL: <https://mediarath.de/products/home-assistant-green-smart-home-hub-2x-usb-gigabit-lan-hmdi-microsd-slot> (visited on 07/05/2024).
- [37] Home Assistant. *Raspberry Pi*. URL: <https://www.home-assistant.io/installation/raspberrypi> (visited on 07/05/2024).
- [38] Raspberry Pi. *Raspberry Pi hardware*. URL: <https://www.raspberrypi.com/documentation/computers/raspberry-pi.html> (visited on 07/05/2024).
- [39] Home Assistant. *Home Assistant Yellow*. URL: <https://www.home-assistant.io/yellow> (visited on 07/05/2024).
- [40] Mauser. *Hub de Casa Inteligente para Raspberry Pi CM4 (CM4 vendido separado) - Home Assistant Yellow Standard Kit*. URL: https://mauser.pt/catalog/product_info.php?products_id=095-3150 (visited on 07/05/2024).

- [41] Home Assistant. *Concepts and terminology*. URL: <https://www.home-assistant.io/getting-started/concepts-terminology/> (visited on 07/06/2024).
- [42] Home Assistant. *Service calls*. URL: <https://www.home-assistant.io/docs/scripts/service-calls/> (visited on 07/08/2024).
- [43] Home Assistant. *Integrations*. URL: <https://www.home-assistant.io/integrations> (visited on 07/08/2024).
- [44] HACS. *HACS*. URL: <https://hacs.xyz/> (visited on 07/08/2024).
- [45] Home Assistant. *Remote access*. URL: <https://www.home-assistant.io/docs/configuration/remote/> (visited on 07/08/2024).
- [46] Nabu Casa. *Pricing*. URL: <https://www.nabucasa.com/pricing/> (visited on 07/08/2024).
- [47] @home-assistant. *Home Assistant Add-on: DuckDNS*. URL: <https://www.home-assistant.io/docs/configuration/remote/> (visited on 07/08/2024).
- [48] Home Assistant. *Home Assistant frontend*. URL: <https://www.home-assistant.io/integrations/frontend/> (visited on 07/09/2024).
- [49] Home Assistant. *Overview - Home Assistant Dashboard*. URL: <https://demo.home-assistant.io/> (visited on 07/09/2024).
- [50] Home Assistant. *Using automation blueprints*. URL: https://www.home-assistant.io/docs/automation/using_blueprints/ (visited on 07/09/2024).
- [51] Home Assistant. *History*. URL: <https://www.home-assistant.io/integrations/history/> (visited on 07/10/2024).
- [52] Home Assistant. *Assist - Talking to Home Assistant*. URL: https://www.home-assistant.io/voice_control/ (visited on 07/09/2024).
- [53] Home Assistant. *Google Assistant*. URL: https://www.home-assistant.io/integrations/google_assistant/ (visited on 07/09/2024).
- [54] Home Assistant. *Amazon Alexa*. URL: <https://www.home-assistant.io/integrations/alexa/> (visited on 07/09/2024).
- [55] Home Assistant. *Creating a cloud Assist pipeline*. URL: https://www.home-assistant.io/voice_control/voice_remote_cloud_assistant/ (visited on 07/09/2024).
- [56] Home Assistant. *Installing a local Assist pipeline*. URL: https://www.home-assistant.io/voice_control/voice_remote_local_assistant/ (visited on 07/09/2024).

- [57] Home Assistant. *ESP32-S3-BOX voice assistant*. URL: https://www.home-assistant.io/voice_control/s3_box_voice_assistant/ (visited on 07/09/2024).
- [58] Home Assistant. *\$13 voice assistant for Home Assistant*. URL: https://www.home-assistant.io/voice_control/thirteen-usd-voice-remote/ (visited on 07/09/2024).
- [59] Home Assistant. *About wake words*. URL: https://www.home-assistant.io/voice_control/about_wake_word/ (visited on 07/09/2024).
- [60] Home Assistant. *Create your own wake word*. URL: https://www.home-assistant.io/voice_control/create_wake_word/ (visited on 07/09/2024).
- [61] ESPHome. *ESPHome*. URL: <https://esphome.io/index.html> (visited on 07/10/2024).
- [62] Mehedi Hasan et al. "Smart Home Systems: Overview and Comparative Analysis". In: *2018 Fourth International Conference on Research in Computational Intelligence and Communication Networks (ICRCICN)*. 2018, pp. 264–268. DOI: [10.1109/ICRCICN.2018.8718722](https://doi.org/10.1109/ICRCICN.2018.8718722).
- [63] @kelanfromgoogle. *We're the team behind the latest updates to Nest devices and Google Home for web - ask us anything!* 2023. URL: https://www.reddit.com/r/googlehome/comments/1bpxro1/comment/kwztl8z/?utm_source=share&utm_medium=web3x&utm_name=web3xcss&utm_term=1&utm_content=share_button (visited on 06/27/2024).
- [64] Google Developer Center. *Smart Home Scene Guide*. 2023. URL: <https://developers.home.google.com/cloud-to-cloud/guides/scene?hl=en> (visited on 06/27/2024).
- [65] @moelet. *Run hub offline?* 2023. URL: https://www.reddit.com/r/SmartThings/comments/1cp9twl/run_hub_offline/ (visited on 06/27/2024).
- [66] @vgeldbr. *Location not showing as a "when this happens" option on routines*. 2022. URL: <https://uk.amazonforum.com/s/question/OD56Q0000ANS5MFSQ1/location-not-showing-as-a-when-this-happens-option-on-routines> (visited on 06/27/2024).