



Dissertation on
“Voice Interfacing and Control of Smart Home Networks”

Submitted in partial fulfilment of the requirements for the award of degree of

Bachelor of Technology
in
Computer Science & Engineering

Submitted by
Shashank Prabhakar 01FB16ECS356
Shrey Tiwari 01FB16ECS368
Sumanth V Rao 01FB16ECS402

Under the guidance of
Prof. Prasad Honnavalli

January 2019 – May 2020

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

FACULTY OF ENGINEERING

PES UNIVERSITY

(Established under Karnataka Act No. 16 of 2013)

100ft Ring Road, Bengaluru – 560 085, Karnataka, India

CERTIFICATE

This is to certify that the dissertation entitled

‘Voice Interfacing and Control of Smart Home Networks’

is a bonafide work carried out by

Shashank Prabhakar 01FB16ECS356

Shrey Tiwari 01FB16ECS368

Sumanth V Rao 01FB16ECS402

In partial fulfilment for the completion of eighth semester project work in the Program of Study Bachelor of Technology in Computer Science and Engineering under rules and regulations of PES University, Bengaluru during the period Jan. 2019 – May. 2020. It is certified that all corrections / suggestions indicated for internal assessment have been incorporated in the report. The dissertation has been approved as it satisfies the 8th Semester academic requirements in respect of project work.

Signature

Prof. Prasad B Honnavalli

Professor

Signature

Dr. Shylaja S S

Chairperson

Signature

Dr. B K Keshavan

Dean of Faculty

External Viva

Name of the Examiners

Signature with Date

1. _____

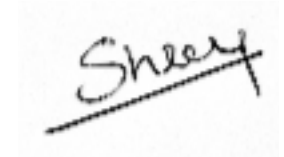
2. _____

DECLARATION

We hereby declare that the project entitled “**Voice Interfacing and Control of Smart Home Networks**” has been carried out by us under the guidance of Prof. Prasad Honnavalli, Professor, CSE and submitted in partial fulfilment of the course requirements for the award of degree of **Bachelor of Technology in Computer Science and Engineering of PES University, Bengaluru** during the academic semester January 2019 – May 2020. The matter embodied in this report has not been submitted to any other university or institution for the award of any degree.



Shashank Prabhakar 01FB16ECS356



Shrey Tiwari 01FB16ECS368



Sumanth V Rao 01FB16ECS402

ACKNOWLEDGEMENT

We would like to express our gratitude to Prof. Prasad Honnavalli, Dept. of Computer Science, PES University, for his continuous guidance, assistance and encouragement throughout the development of this project.

We are grateful to the project coordinators, Prof. Preet Kanwal and Prof. Sangeetha V, for organizing, managing and helping out with the entire process.

We take this opportunity to thank Dr. Shylaja S S, Chairperson, Department of Computer Science and Engineering, PES University, for all the knowledge and support I have received from the department. I would like to thank Dr. B.K. Keshavan, Dean of Faculty, PES University for his help.

We are deeply grateful to Dr. M. R. Doreswamy, Chancellor, PES University, Prof. Jawahar Doreswamy, Pro Chancellor – PES University, Dr. Suryaprasad J, Vice-Chancellor, PES University for providing to me various opportunities and enlightenment every step of the way. Finally, this project could not have been completed without the continual support and encouragement we have received from our parents.

As we write this report, the world is facing a global pandemic. We would like to extend our gratitude to the ones who are helping us fight the Covid-19 virus and are saving human lives in the front line.

ABSTRACT

Daily lives are being transformed by a plethora of new technologies that not only help devices perform household tasks, but also anticipate them. One of the greatest enablers in this transformation is the concept of the Internet of Things or IoT, where everyday appliances have the ability to communicate with each other over the internet. Smart Home Automation is a concept where these IoT enablers are used to make everyday tasks easy to carry out. It is not only a means of easy interaction with appliances and devices but is also a natural extension of human behaviour that infers the user's intentions and performs tasks without the user's intervention. The ability to use natural language to interact with the home network is key to realizing an easy-to-use and intuitive solution. In this project, the team attempts to unify the seemingly fragmented landscape and demonstrate a fully scalable, platform-agnostic home automation system that minimizes cost, allows for interoperability and is secure, by design.

TABLE OF CONTENTS

Chapter No.	Title	Page No.
1.	INTRODUCTION	9
	1.1 Current Landscape	10
	1.2 Motivation	11
	1.3 Relevance of Project	12
2.	PROBLEM DEFINITION	13
3.	LITERATURE SURVEY	14
	3.1 Voice as the New Interface - A New Era in Speech Processing	14
	3.2 Scalable Interconnected Home Automation System	14
	3.3 2019 Cyber Threat Outlook	15
	3.4 Skill Squatting Attacks on Amazon Alexa	15
	3.5 Smart Home is Where the Bot is	16
	3.6 Security Review of Consumer Home Internet of Things (IoT) Products	17
4.	SYSTEM REQUIREMENTS SPECIFICATION	18
	4.1 Functional Requirements	18
	4.2 Non-functional Requirements	20
	4.3 Hardware Requirements	22
5.	USE AND MISUSE CASES	25
6.	THREAT MODELLING	28
7.	SYSTEM ARCHITECTURE	30
	7.1 Initial Architecture	30
	7.2 Revised Architecture	33
	7.2.1 High Level View	33
	7.2.2 Detailed View	36

	7.2.3 Design Choices	39
8.	WORKFLOW	41
9.	IMPLEMENTATION	43
	9.1 Security by Design	43
	9.2 Cloud and Skills	45
	9.3 Abstraction Server	47
	9.4 Control Unit	49
	9.5 Supervision Module	50
10.	RESULTS AND CONCLUSION	54
11.	DISCUSSION	56
12.	REFERENCES	57
13.	BIBLIOGRAPHY	59
14.	APPENDIX	60

LIST OF TABLES

Table No.	Title	Page No.
6.1	The STRIDE Threat Model	29

LIST OF FIGURES

Figure No.	Title	Page No.
3.1	List of Squatable Skills	16
4.1	Raspberry Pi	23
4.2	4 Channel Relay Module	24
5.1	UML Diagram Representing Use and Misuse Cases	25
5.2	List of Use and Misuse Cases Identified for a Smart Home	27
7.1	Initial Architecture	30
7.2	High Level Architecture	34
7.3	Control Unit as the Basic Replicable Unit	37
7.4	Detailed Architecture	38
8.1	Workflow	41
9.1	Development of a Google Action	45
9.2	Development of an Alexa Skill	46
9.3	Error Handling in Google and Alexa	48
9.4	MQTT Topics Format	51
9.5	Node Red	51
9.6	UI for the Home Assistant	52
9.7	Website for the Project	53

1. INTRODUCTION

Home automation is the process of reducing human activity in trivial household tasks by either providing means of easy accessibility or by automating mundane tasks. A home automation system performs various tasks ranging from turning lights on and off to opening the garage door when a car pulls up. In the modern world, increase in the variety and complexity of home appliances has led to an increase in demand for home automation systems that are robust and practical. Today, a home automation system is expected to handle failures, integrate with a variety of devices seamlessly, be secure, cost effective and much more. Additionally, with wider internet coverage and more powerful edge compute devices, the concept of a Smart Home Automation System has surfaced. One can define a Smart Home as a living space that gives the user the freedom to choose from a variety of intuitive and convenient means of interfacing with home appliances.

Many companies today are providing a different means of accessing appliances that is different from a traditional switch, like a remote for controlling a fan, or a mobile application to switch on a light. However, this alone does not make a home smart.

“A Smart Home must interpret user actions effectively and must be a natural extension of human behavior.”

It should not just include smarter ways of controlling but should also provide for smarter ways of living. For instance, just being able to control a light via a smartphone application would *not* be considered ‘smart’. However, giving the user the ability to do the same, while being able to schedule toggling on or off of the light based on conditions like sunrise and sunset *would* be considered ‘smart’.

This project adopts a systematic and iterative approach to demonstrate a fully scalable, secure and platform-agnostic home automation system. Areas of focus include the factors that currently hinder the acceptance of smart home solutions in everyday life.

1.1 Current Landscape

Studies reveal that the adoption of Home Automation in homes is greatly driven by two factors - the cost of implementing such a system, and the effort it takes to turn a regular home into a usable Smart Home. Though access to automation is growing cheaper by the day, the industry is mostly fragmented and is only held together by freely available platforms that are open source. Limited capabilities and performance of the available software are among the main reasons holding back the growth of home automation systems.

Manageability of devices becomes a challenge as the number of entities that are part of the Home Automation System increases. The smart home must possess the capability to control and interact with a variety of physical appliances. Though the physical devices that make up a smart home are available easily, they come with an overhead of high installation and modification costs.

Along with these factors, the Voice Assistant platforms today - Google Assistant and Amazon Alexa - are itself very different in the way they work and integrate with various household devices. Both the platforms have their own advantages and disadvantages. While Google leads in its Natural Language Processing capabilities, Alexa is more compatible with third-party appliances and is more 'open' for configurability. There is no clear winner and this makes it difficult for a user to side with one of the two platforms. Thus, there is no real way of implementing a smart home cost effectively and have manageability, with such constraints.

1.2 Motivation

Historically, user interfaces consisted of hardware devices (keyboard and mouse), graphical interfaces and touchscreens. Considering this pattern, the most promising interfacing technology of the future is voice input as it is the next most intuitive form of interaction and has the added benefit of convenience. Though voice interfacing systems did exist previously, they were restricted to applications like call-center management, because of problems like inaccuracy and misinterpretation of speech detection. The recent developments in natural-language based voice recognition systems have helped bridge this gap and improve accuracy in speech detection and understanding. Due to these reasons, there has been a growth in the use of voice as a primary means of interaction with smart home devices.

Technologies like Machine Learning and Artificial Intelligence are witnessing explosive growth today. However, the same cannot be said for the Internet of Things and its applications. Thus, there is much room for innovation and creativity in this space. There are many gaps to fill before Home Automation becomes mainstream in today's world. Architecting a system that is easily manageable, scalable and interoperable, while also being cost-effective, is key. Another aspect that is crucial in this area is security. Most IoT enabled products that are available today are low on security, and security features are often added after a product has been made. Thus, incorporating security elements from the beginning would render a system more secure on the whole, and would make it more favourable for adoption by consumers. Lastly, the demand for customization in any product used by consumers, is high. Therefore, building a system that is flexible and easily customizable would increase the consumer base for IoT solutions.

1.3 Relevance of Project

The Internet of Things and its applications have not penetrated the Indian market as much as other technologies that are relevant today - such as E-Commerce or Machine Learning. It is still in its nascent stages of adoption and this is therefore an appropriate time for projects and startups to emerge in this domain.

Besides the Indian market, there are huge investments in the IoT and Home Automation sector, globally. The Global IoT Market is expected to grow at a CAGR of a healthy 10.2% during this decade, and the market is expected to grow from USD 916.9 Billion in 2020 to a staggering USD 2 Trillion in 2028. A favourable market is key for any project to see its role in the coming years.

Adoption of IoT in industrial applications and its subsequent advantages, has proven its effectiveness and has increased the trust among consumers to use IoT enabled products in their day-to-day lives. This can be seen in the increase in usage of smart watches and trackers for the purpose of fitness and health.

Today, there are easily available products like the Arduino microcontrollers and Raspberry Pi computers that assist with the automation. More specifically, in India, access to internet and cloud services is more affordable than ever and due to this, there has never been a better time for households to adopt IoT enabled solutions to make their homes 'smart'. Added to this, the cybercrime rate is at an all time high and it is therefore essential to make IoT-based products secure, reliable and resilient.

Considering all these factors, this project was chosen to be researched upon and executed. It delves deep into multiple facets of Computer Science like Data Structures, Algorithms, Computer Networks, Cybersecurity, Microprocessors, Web Technologies and Cloud Computing. In conjunction with this, its user centric nature and attention to topics such as user experience and practicality makes it a well-rounded Computer Science project.

2. PROBLEM DEFINITION

Existing technologies in the IoT domain are inflexible in terms of their functionality and usually deliver below-average performance. They are not scalable and are ineffective when it comes to managing an entire smart home. Limited interfacing options with the appliances and their dependence on an always-active internet connection for proper functioning has limited its penetration in the consumer market. More importantly, only expensive smart home devices integrate well with current home automation systems and there is no easy way to control existing home infrastructure using such solutions.

“In this light, this project aims at architecting a scalable home automation solution that is secure by design, with the voice as an additional means for interfacing with everyday home appliances.”

The solution can be implemented on existing infrastructure and does not require the usage of expensive ‘smart’ appliances. It must not be tied to one single platform and must be platform-agnostic. Lastly, the solution must incorporate other characteristics such as resilience to network failures and fast response to the given commands.

3. LITERATURE SURVEY

A variety of information sources were visited to understand the work that has been done in this domain. The Literature Survey has helped in understanding and identifying the areas that were not focused on and needed some innovation. All the key aspects from the papers are inculcated into the solution proposed.

3.1 Texas Instruments: Voice as the New Interface - A New Era in Speech Processing

This paper talks about the evolution of interaction methods between users and computers, from the traditional means such as keyboard and mouse to newer emerging technologies like gesture-based communication, brain-computer interfaces and voice-based interactions. The paper describes the key challenges related to voice interfacing and the current ways in which the industry is tackling them. Understanding the limitations and the working of voice-based systems can help us avoid common pitfalls and arrive at an optimal solution.

Gaining knowledge about trends in interaction systems and changing the way people prefer interfacing with devices helped in rethinking some of the project requirements. Emphasis on voice as a means of input was a result of reading this paper.

3.2 V. Stangaciu, V. Opârlescu, P. Csereoka, R. D. Cioargă, and M. V. Micea: Scalable Interconnected Home Automation System

In this paper, the authors describe the two main divisions of IoT automation technologies. The paper discusses a proposed architecture for scalability and versatility that wishes to blur the boundaries between both the divisions of the IoT landscape, namely IoT solutions to remote access/control and IoT solutions to promote interconnectivity and integration of devices. The paper presented a layered architecture to solve the problems of modern day IoT systems. From the analysis, the additional layers present in the architecture proposed

could account for higher latency levels and an overall censurable experience. Ideas were borrowed from this paper for architecting this project's proposed solution. Care is taken to avoid latency issues and boost performance while not compromising on security.

3.3 Booz Allen Hamilton: 2019 Cyber Threat Outlook

This report by Booz Allen Hamilton highlights the importance that needs to be given to offering protection against cyber crimes in the future and describes how the crime rate is predicted to grow. Connected devices like televisions, webcams, coffee machines and sensors are either targets of cyber attacks or are being leveraged to carry out cyber crime activities. The report describes the emergence of new attack surfaces and newer attacks being carried out on protocols like Bluetooth. It also provides certain safe practices that would be relevant for the study during the project. An important takeaway from this was that the design should try to use secure protocols as far as possible and block all unused ports on IoT devices and gateways at all times.

3.4 D. Kumar, R. Paccagnella, P. Murley, E. Hennenfent, J. Mason, A. Bates, and M. Bailey: Skill Squatting Attacks on Amazon Alexa

This is a report by Texas Instruments. It talks about a very important vulnerability in publicly available skills. Attackers misuse the weakness of smart home speakers' inability to distinguish between similar-sounding words and phrases. A skill squatting attack is when a malicious skill, which sounds similar to the user's intended skill, is called in place of the user's intended skill. This malicious skill can carry out undesirable activities, as according to the Alexa Skills Market, it is a legitimate skill being invoked by the user. This report suggests certification of skills as a means to prevent such attacks. However, it does not talk about what a developer can do while building a system to stop Skill Squatting attacks.

Skill	Squatted Skill
Boil an Egg	Boyle an Egg
Main Site Workout	Maine Site Workout
Quick Calm	Quick Com
Bean Stock	Been Stock
Test Your Luck	Test Your Lock
Comic Con Dates	Comic Khan Dates
Mill Valley Guide	No Valley Guide
Full Moon	Four Moon
Way Loud	Way Louder
Upstate Outdoors	Upstate Out
Rip Ride Rocket	Rap Ride Rocket

Malicious Skills

Figure 3.1: List of Squatable Skills

The Figure 3.1 shows normal skills along with similar-sounding, malicious skills that squat on the normal skill. Both normal and malicious skills are available in the Amazon Alexa Skills Market in the public domain.

Understanding these points are crucial for the design phase of any Voice based interaction system. During the implementation of the ‘proof of concept’, these points were kept in mind so as to not compromise security, while still giving the user the freedom of choosing between voice based input or an application based classical input.

3.5 J. Coumau, H. Furuhashi, and H. Sarrazin: Smart Home is Where the Bot is

The authors of this McKinsey article introduce a new idea - the concept of a Homebot. A Homebot is a managing entity that controls other bots and acts as a single point of contact for the users. It talks about a hierarchical structure in the architecture, where a master bot controls a service bot, which in turn controls a niche bot. The article highlights the importance of data and how data can be the largest source of revenue for this concept. It is mentioned that the integration of these entities would be the main challenge in the

implementation. Borrowing ideas from this article have helped improve the project to a great extent.

Apart from the new concept of a homebot, this article suggests a change in the mindset while envisioning the future of IoT. This new point of view puts things into perspective and helps come up with more creative solutions for problems that are more likely to become more relevant in the market. This project builds upon the same concepts and ideas while coming up with the solution for a fully manageable smart home.

3.6 M. Fagan, M. Yang, A. Tan, L. Randolph, and K. Scarfone: Security Review of Consumer Home Internet of Things (IoT) Products

This report by the National Institute of Standards and Technology provides security guidelines for an IoT setup. These guidelines have been categorized into six buckets - Device Identification, Device Configuration, Data Protection, Access to Interfaces, Software and Firmware Updates and Cybersecurity Event Logging. The points mentioned in this paper focus more on the manufacturing point of view and one should consider implementing some of these guidelines, as developers, for solutions using existing infrastructure. They serve as design patterns that are tried and tested by the industry. Suggestions from this paper are incorporated into the solution where ever possible.

4. SYSTEM REQUIREMENTS SPECIFICATION

In order for a system to work smoothly, it is essential to define what the system is expected to do. These requirements of the system can be categorized as Functional, Non-functional and Hardware requirements.

After extensive research and study of the domain, all the requirements in the domain of smart home systems can be reduced to a set of fundamental functional and nonfunctional requirements which when met, could solve most of the drawbacks of current-day solutions.

This project defines the functional and non-functional requirements below and also aims at achieving them as part of the proposed solution. This also sets the scope of implementation for the proof-of-concept that is developed as part of this project.

4.1 Functional Requirements

Functional requirements can be defined as a set of characteristics or features that the system is required to exhibit. Functional requirements help understand the behaviour of the system when certain inputs are provided and sets the output expectations.

This project presents some fundamental requirements of any smart home automation system along with additional requirements that increase security, practicality or both. Listed below are the functional requirements that are targeted by this project.

- **Ability to control and interface with the system using the web as well as a smart phone application**
 - The system must provide the user with the ability to control the smart home devices, interact with them and see their status via an online web platform and a smartphone application.
 - This requirement sets the basis for any Home automation system. It is necessary to provide an intuitive and feature rich interface to the user to increase the manageability of the smart home.

- **Ability to control and interface with the system using voice as an alternative mode of input**
 - The system must provide the user with the ability to control and interact with the Home Automation System using voice as well. The system must understand the user's intention and be capable of providing him with meaningful replies.
 - It is also necessary that the system integrates well with the popular voice assistants giving the user the freedom to pick the voice assistant of his choice.
- **Ability to add and remove devices from the smart home**
 - It must be possible for the user to add and remove devices from the smart home network seamlessly and securely.
 - The solution must provide options to the users that enable them to carry out such tasks effortlessly.
 - For added convenience, the user should be able to perform such operations using voice commands too. There must be no need to perform manual configuration changes on code files for successful execution of such tasks.
- **Presence of Failsafe Mechanisms**
 - The solution should be architected in such a way that failsafe mechanisms are built into the home automation system from start.
 - The system must be able to perform and retain most of its functionality even in case of network failures.
 - It must be possible for the user to control the entire smart home even in the case of failure of the entire home automation system.
- **Ability to monitor the smart home**
 - It must be possible for the user to monitor the state of the devices that are part of the smart home system, access the readings from sensors and monitor other subcomponents of the smart home.

- The home automation system must provide options to help the user take appropriate actions for devices that are in undesirable states.
- **Ability to access the smart home network remotely**
 - The user must be able to access the smart home network even when not at home and do so in a secure manner.
 - The solution must incorporate the right design choices so as to provide the user with the option to invoke most of the features of the home automation system remotely and securely.
- **Access Control**
 - The system must provide mechanisms to impose access control rights over users of the home automation system
 - A 'Role Based Access Control' system is incorporated in the proposed solution to meet this requirement and increase security.
- **Compatibility with a variety of devices**
 - The home automation system must be able to interface with and integrate well with a variety of home appliances ranging from normal household devices to popular smart home devices available in the markets.
 - All the devices that are a part of the smart home system must be visible and controllable from a single uniform interface.

4.2 Non-functional Requirements

Non-functional requirements can be defined as those requirements that help in the overall evaluation of the system and support the functional requirements. These mostly describe the quality attributes of the system and the software.

As part of this project, the choice of non functional requirements has been made based on the extensive study, identified drawbacks of existing offerings and effect on likelihood of adoption. The nonfunctional requirements that this project targets are presented below.

- **Security by design**

- Great emphasis is put on developing a system that is secure by design. In the domain of security, It is common knowledge that a system is only as strong as its weakest link. This concept can be applied to a product/solution as well - a solution is only as good as its weakest feature.
- Security carries great value in the modern day and age. Building a solution that packs all the bells and whistles but lacks in the domain of security would be just another lab experiment and not a practical solution to real life problems.
- This project aims to dedicate efforts towards pressing problems and develop a solution that incorporates the right set of features, hence making it usable in the practical sense.

- **Cost effectiveness**

- Research in the domain of smart home systems reveal that most of the smart home devices are expensive and non interoperable. Cost is one for the biggest barrier in the growth of smart home systems.
- The solution proposed should be feature rich while being cost effective. Cost effectiveness can be considered as one of the unique selling points (USP) of this project.

- **Manageability**

- A solution is practical only if it is manageable. Most of the present day software solutions can scale up to accommodate all the devices in a smart home but trade off manageability and user experience in doing so.
- As part of this project, manageability and scalability as considered key requirements for any successful implementation of a home automation system.

- **Performance**

- The solution must be performant in order to provide a great user experience. The solution must focus on reducing the latency in the system and increase the responsiveness as far as possible.

- **Ease of use**

- The system must be user friendly, intuitive to use and must possess an easy to use user interface. This requirement is important to the system as it forms the foundations for other requirements such as manageability and practicality.

- **Configurability**

- The system must be designed in such a way that it must be easy for the user to configure new devices, modify existing devices and remove devices from the smart home network.
- Configurability of the system plays a major role when more and more devices are added to the system and hence this project focuses on this requirement from the design phase itself.

- **Robustness**

- The system must be robust in nature. There must be error handling present in the right places and fail-safes built into the system.
- It is necessary for the solution to be resilient and be able to recover from any undesirable states with minimal user interaction.

4.3 Hardware Requirements

A combination of Hardware and Software components is essential for a computer system to meet the end-users' needs. Here, a variety of Hardware modules were used to make the system functional. Thorough comparison between different modules providing similar functionality were made. Informed choices were taken for the same.

- **Microcontrollers**

A microcontroller in simple terms is a computer system on a chip that is capable of performing tasks. A microcontroller contains a microprocessor, memory for program data, control pins and input/output ports. A Wi-Fi enabled microcontroller is one that is capable of sending and receiving commands wirelessly. It is a development board that can interface with a hardware controller using software signals. These boards are usually not powerful in terms of processing capabilities and this helps keep their cost low. For the purposes of this research project, the NodeMCU (ESP8266) development board was used as the control module. It provides a sufficient number of GPIO (General Purpose Input Output) pins that can be used to interface with the lower layer hardware devices.

- **Raspberry Pi**

The Raspberry Pi is a low cost, palm-sized computer. It runs Linux and has General Purpose Input Output Pins for the sake of controlling electronic components and IoT-enabled devices. Here, the Raspberry Pi was used to run the Home Assistant server that serves as the Supervision Module, the heart and brain of the home automation system.

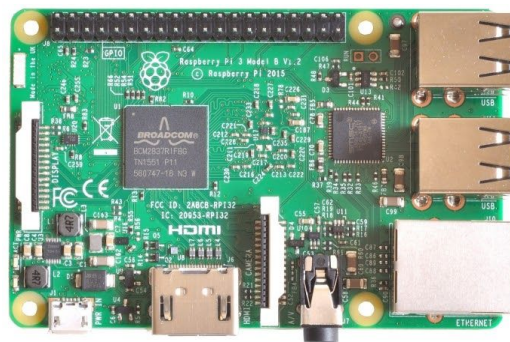


Figure 4.1: Raspberry Pi

- **Router**

A router's main purpose here is to route traffic to and from the Internet from the home network. It is being used in the project for the sake of Network Segregation between the smart home network and general home network.

- **Relays**

A relay is a switch that uses electromagnetism to toggle a circuit on or off. It contains input and output terminals. Its basic function here is to send signals to the electronic and electrical devices during a state change.

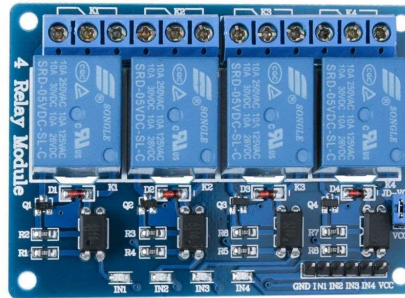


Figure 4.2: 4 Channel Relay Module

- **Sensors**

Sensors help collect information about the environment. This helps build intelligence into the system. Data collected by the sensors are relayed directly to an endpoint for further analysis.

- **AC Voltage Controllers**

These modules are used to supply variable voltage Alternating Current Output. This is used when the end device can take a range of values for a particular attribute. For instance, the brightness of a light bulb, or the speed of a fan.

- **Amazon Alexa or Google Assistant Speaker**

This is the voice assistant speaker that the end user interacts with using voice. It can simply be a Alexa or Google Assistant smartphone application that the user voices their commands to. It can also be a smart speaker like Amazon Echo or Google Home.

The Figure 5.1 depicts the use cases and misuse cases for a home automation system in the UML format. Analyzing the misuse cases for the system described, potential threats have been identified. Suitable countermeasures to mitigate the risk have also been provided.

- **Skill Squatting**

Invocation of a malicious skill that is present on a public domain in place of a user intended skill that sounds similar. This misuse case highlights the vulnerabilities related to the publicly available skills. The countermeasure for this is the implementation of certification systems and verification of skills before public launch.

- **Unauthorized access and control of an IoT device using voice commands**

Incidents when an unauthorized entity eavesdrops on voice commands and replays them to carry out undesirable activities. The inherent nature of voice-based communication makes these misuse cases feasible. Voice recognition and unique passphrases that change periodically can help mitigate the risk posed by such an attack.

- **Information leakage from IoT Home Monitor to outsiders or guests**

Outsiders and guests can gather intricate information about the home network from the monitor that is publicly accessible. Implementation of access control and mechanisms of authentication can help reduce such risks.

- **Use home IoT network as Botnet to launch DDoS attack**

Infecting the home network to use the IoT devices as a launchpad for a denial of service attack on a targeted victim network, without the user's knowledge. The weak security measures on cheap IoT devices enable such attacks. Timely firmware updates of the IoT devices and imposition of rate-limiting on the number of requests allowed per device can help mitigate such attacks. The presence of perimeter defence systems like 'firewalls' and 'intrusion prevention systems' reduce the risk of network compromise.

- **Security breach due to malicious third-party IoT device**

The attacker is able to gain access to the network by planting malicious IoT devices into the network. White- listing trusted third party devices and implementing a mutual authentication system helps widen the defences.

- **Gain access to the IoT network via voice commands when not at home**

The attacker can use predefined, legitimate voice commands to gain access to the home network when the owner is not home. Disabling voice interfacing subsystems of the smart home network in the absence of the owner can prevent such attacks.

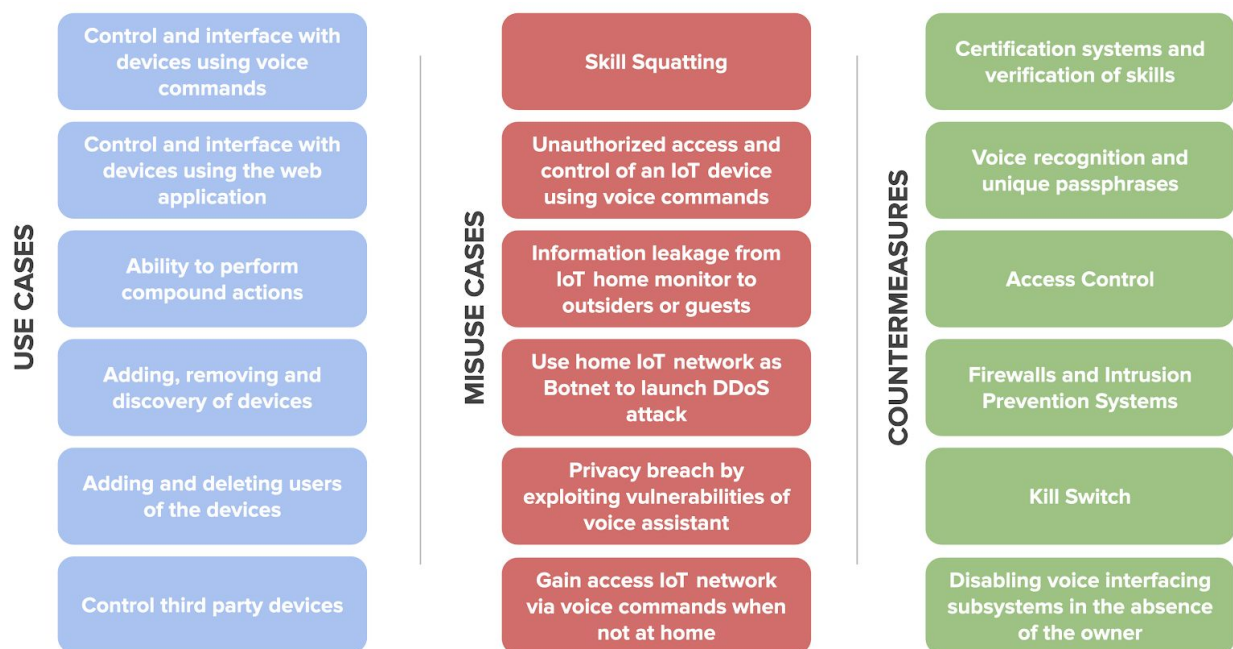


Figure 5.2: List of use and misuse cases identified for a smart home

The Figure 5.2 provides a summary of the detailed use case and misuse case analysis that was performed on the project. All the use cases are enumerated with their corresponding misuse cases and possible countermeasures.

It must be noted that the use case and misuse case study has been performed even before the architecting phase of the project. This has enabled informed decision making during the architecture phase in turn leading enhanced security of the solution.

6. THREAT MODELLING

Threat Modelling is the process of analysing the system, identifying potential structural vulnerabilities (design flaws) that could lead to threats and actively looking for security measures and appropriate safeguards to mitigate risk. It is best to apply this process during all phases of the software development lifecycle.

The STRIDE model has been incorporated for threat modelling for the architecture and designing phase. STRIDE is a model that helps categorize threats, reason out the vulnerabilities and find suitable countermeasures to implement in the system.

The table depicted in Table 6.1 captures the STRIDE threat modelling performed as part of this project. The table describes the top threats pertaining to the domain of large IoT networks and also highlights potential countermeasures that can help in the mitigation of such threats. The countermeasures described provide various means of risk mitigation ranging from architectural/design choices to finer software security modules.

The Threat Modelling provides some significant insights on the security posture of the architecture and also suggests ways to improve it. This process along with the use case and misuse case study help the project deliver on the requirement of security by design. The project has followed an iterative approach to all the phases of development. All the learnings from each iteration and exercises such as these have been incorporated in the successive iterations to help arrive at an optimal solution that fulfills all the requirements, is secure and has made the right tradeoffs.

Category	Threat	Countermeasures
Spoofing	An attacker can reuse a password	<ul style="list-style-type: none"> Periodic changing of passwords
	An attacker can anonymously connect to the network	<ul style="list-style-type: none"> Secure communication protocols (WPA2 - PSK) Session cookies
	Response spoofing from the server	<ul style="list-style-type: none"> Security by Obscurity - Reverse Proxy Nonce Encryption
	System ships with default passwords	<ul style="list-style-type: none"> Software authentication architecture - force change default passwords
Tampering	Distributed ' Access Control ' rules	<ul style="list-style-type: none"> Centralized Home Hub
	An attacker can replay data without detection	<ul style="list-style-type: none"> Nonce - timestamps and sequence numbers
	An attacker can directly modify or write to a data store	<ul style="list-style-type: none"> Secure communication protocols Access control
Repudiation	The system has no logs	<ul style="list-style-type: none"> Logging software
	An attacker can alter log messages on the network	<ul style="list-style-type: none"> Heartbeat option for logging system
	An attacker can edit logs and there's no way to tell	<ul style="list-style-type: none"> Secure communication Access control
Information Disclosure	An attacker can see error messages with security-sensitive content	<ul style="list-style-type: none"> Default error messages
	An attacker can act as the man in the middle	<ul style="list-style-type: none"> Encryption Certification
	The attacker can discover the fixed key being used for encryption	<ul style="list-style-type: none"> Periodic change of keys Secure storage of keys
Denial of Service	An attacker can render your authentication system unusable	<ul style="list-style-type: none"> Security by obscurity
	An attacker can make your network unstable	<ul style="list-style-type: none"> Rate limiting on requests per device
	An attacker can block functionality	<ul style="list-style-type: none"> Rate limiting on requests per device Intrusion Detection Systems
Elevation of Privilege	You include user-generated content within your page, possibly including the content of random URLs (XSS)	<ul style="list-style-type: none"> Handled by protocols and structured communication standards.
	An attacker can inject a command that will run at a higher privilege level	<ul style="list-style-type: none"> Code filtering Network segmentation

Table 6.1: The STRIDE Threat Model

7. SYSTEM ARCHITECTURE

Defining an architecture for a computer system is crucial. It helps in making design choices and provides for a stable framework to work on. It gives clarity on the project and helps divide responsibilities among different components. In this project, an iterative approach was followed to arrive at the most optimal architecture.

7.1 Initial Architecture

The first version of the architecture that was created for this project was straightforward and simple. The architecture presented in this section strived towards unifying existing technologies to achieve completeness and an easy to use solution that could be operated using voice as well as a mobile application.

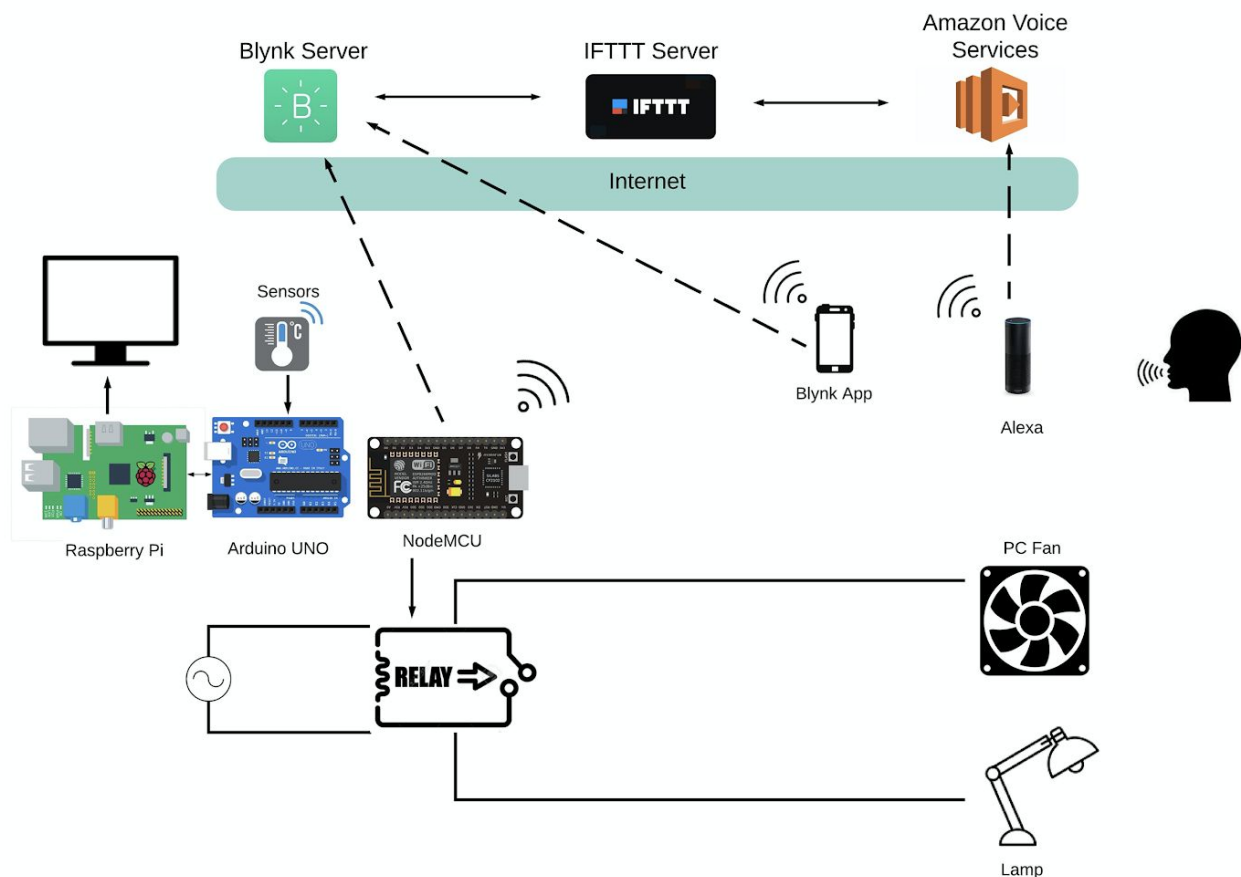


Figure 7.1: Initial Architecture

This architecture consisted of three layers:

- **Hardware Layer**

This layer contains the physical devices of the system. It included a wifi-controlled System-on-a-chip (NodeMCU), a Raspberry Pi, Relay boards, electrical appliances and sensors. The main function of this layer was to interact with the physical devices.

- **Interaction Layer**

This layer is where the user interacts with voice assistant services. It consists of the Alexa enabled voice assistant speaker. It can also be a smartphone. An open source software application called Blynk was also used which facilitated interaction between this layer and the below Hardware Layer.

- **Cloud Layer**

The third and final layer in this architecture comprises the different cloud services required for the functioning of this system. These services run on the internet. They include:

- **The IFTTT Server**

The If This Then That is a free-to-use service that performs conditional actions. 'Applets' are defined on IFTTT, such that when they receive a certain input command from the Amazon voice service, they send a webhook to the Blynk server which in turn triggers a output sequence on the hardware layer.

- **The Blynk Server**

This receives commands from the IFTTT server that is running and actively polling for requests made. The Blynk server then sends the response to the Blynk Client running on a NodeMCU to perform the required action.

- **Amazon Voice Services**

The AVS processes voice input and converts it to text. It is an Amazon proprietary service and serves primarily as the Natural Processing Unit in the system. Its sole function is to interpret the commands given by the user and

forward the keywords extracted to the IFTTT server that is actively running. It forms the foundation for voice-based input

Though functional, this architecture above has its own drawbacks.

- **Rigidity**

The above architecture is rigid in terms of the invocation of voice commands. It only supports very specific commands and a slight deviation would throw errors. Also, it supports only one of the voice assistants, Amazon Alexa, and there is no support for Google Assistant.

- **Limited Capability**

This does not support compound actions and actions can be performed only one at a time.

- **Low Performance**

Due to the heavy dependence on open source services like IFTTT and Blynk that run on the internet, there were large latencies in the movement of messages among components. Coupled with this, was also longer response times due to internet delays. This led to an overall slow performance of the system as the time between invocation and execution of commands was large.

- **Low Configurability**

Commands and intents were hard-coded and this made the system less intelligent and less open to a wide range of actions. This reduced the configurability of the system.

All in all, the above factors led to limited functionality of the system. Thus, a revised architecture had to be devised from the ground up.

7.2 Revised Architecture

After considering the drawbacks of the previous architecture, a new architecture had to be devised.

7.2.1 High Level View

After gaining hands-on experience and in-depth knowledge about the IoT technologies available, the focus was turned towards security and functionality. During the development of the new architecture there were a few key goals that had to be achieved. They are as follows:

- **Cost-Effectiveness**

The system had to be built using tools and technologies to keep the cost as low as possible.

- **Security**

The system must avoid the common pitfalls and mitigate the security vulnerabilities present in IoT networks.

- **Scalability**

The architecture must support scalability of the smart home network. It should allow for adding more devices as and when needed.

- **Manageability**

Interfacing with the network should be seamless and intuitive. Hassle-free modification of network topology must be supported.

Before architecting a solution, it is necessary to understand important factors that could affect the usability of the solution. To arrive at an optimal as well as practically feasible solution, other aspects of the problem domain such as user requirements, security challenges, existing solutions and standard design patterns were carefully looked into. Having performed the misuse case study and threat modelling prior to the development of the new architecture, the knowledge from these activities and information gathered from research were inculcated into the new architecture. Building the new architecture from the ground up, a high-level hierarchical structure of the solution was first designed. This can be seen in Figure 7.2. There are four layers, each with specific roles and responsibilities.

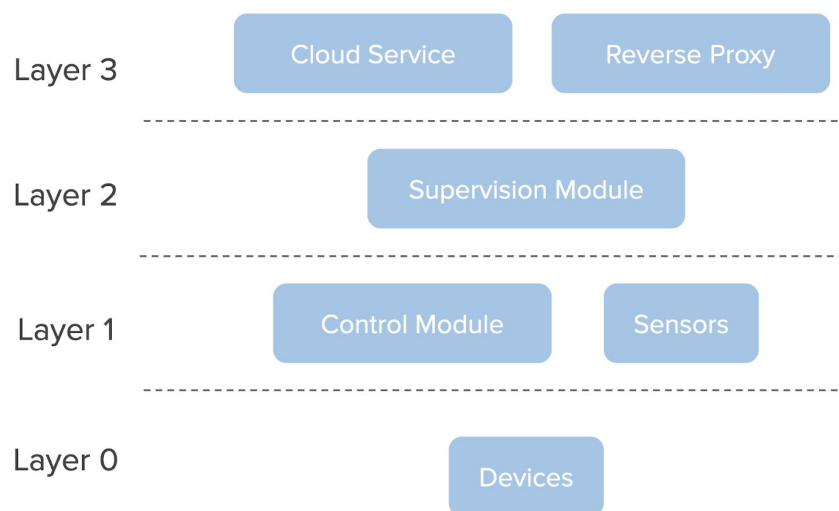


Figure 7.2: High Level Architecture

A. Layer 0

This layer comprises all the hardware tools and devices that need to be integrated into the smart home network. The main purpose and basic functionality that this layer offers is the capability to turn on and off home appliances. The higher layer builds on top of this to realize more sophisticated tasks. The relay module is used to fulfil this requirement.

B. Layer 1

This layer comprises two main components. The control module and the sensors.

- **Control Module**

The control module is any wifi-enabled development board that can interface with a hardware controller using software signals. These boards are usually not powerful in terms of processing capabilities and this helps keep their cost low. For the purpose of this project, the NodeMCU (ESP8266) development board was used as the control module. It provides a sufficient number of GPIO (General Purpose Input Output) pins that can be used to interface with the lower layer hardware devices.

- **Sensors**

To build intelligence into the system, various sensors that can measure features about the environment and relay the results directly to an endpoint were used.

C. Layer 2

This layer solely comprises one component, namely, The Supervision Module. The supervision module is the heart and brain of the entire system. It performs a variety of tasks ranging from Command Processing to Data Logging. This module is responsible for instructing the control module when there is a request to change the state of a particular home appliance. The supervision module monitors the state of all the devices connected to the network and has the access rights to modify any network section on command. In this project, the 'Home Assistant' software has been made use of as the supervision module and it runs on the Raspberry Pi microprocessor. It is an operating system that is light-weight and has the capability to integrate different smart-home related features.

D. Layer 3

This layer represents all the internet-related technologies and cloud services that will be used by the smart home network. The two main components of this layer are:

- **Cloud Voice Assistant services**

These are the services that are required by the voice assistant to function properly and execute smart home commands/requests successfully. The voice assistants that will be used in this project are Google Assistant and Amazon Alexa.

- **Reverse Proxy**

The reverse proxy is a publicly visible web service that acts as an endpoint for the results generated by the cloud voice assistant services. It also carries out any platform-dependent processing of data as it is responsible for providing the layer of abstraction needed to make the solution platform agnostic. The 'Amazon Web Services' platform was used in this project. All the layers work in tandem, communicating with the immediately adjacent layers to deliver all the intended functionalities of the system.

7.2.2 Detailed View

As one can see in the architecture depicted in Figure 7.3, the Control Module along with the devices and hardware tools from Layer 0 form a single unit. This is the basic unit of replication of the system that can help achieve scalability. The communication between this basic unit and the Supervision Module follows the 'Publisher-Subscriber' model. This model of communication supports scalability while using low network bandwidth. It also increases the network flexibility and reduces cost as whenever a change occurs, no rewiring is needed or no software / firmware changes are required. The protocol that is used to implement this publisher-subscriber model is the "Message Queuing Telemetry Transport" (MQTT) protocol. This protocol provides additional benefits such as state monitoring for connected devices and options for authentication.

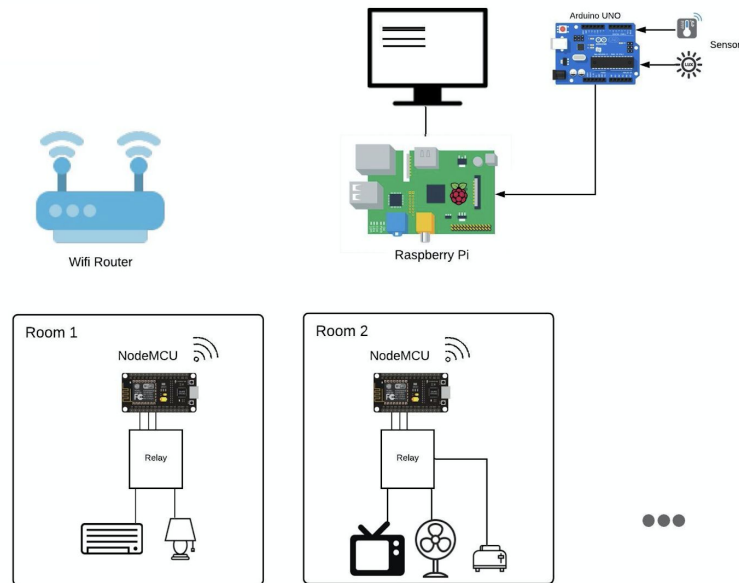


Figure 7.3: Control Unit as the basic replicable unit

There is also a ‘Local Area Network’ in the architecture proposed. The local area network that is created helps us achieve the following goals:

- **Reliability**

Developing a local area network ensures that there is less dependence on external services and system functioning can be guaranteed when during internet shortages.

- **Performance**

On a local area network, the communication speed and bandwidth are quite high. This helps achieve faster response times and low command processing latency.

- **Security**

The local area network can function even when not connected to the internet. This reduces the attack surfaces by a great extent thereby increasing the security of the system.

The Home Assistant that is the brains of the system, acts as the central point of contact for both the user as well as the reverse proxy. The MQTT Server and one of the endpoints for the websocket are running on the home assistant. The entire Local Area Network exists on a separate subnet, different from the one on which the primary router operates. This network segregation helps manage the devices easily and also provides added security.

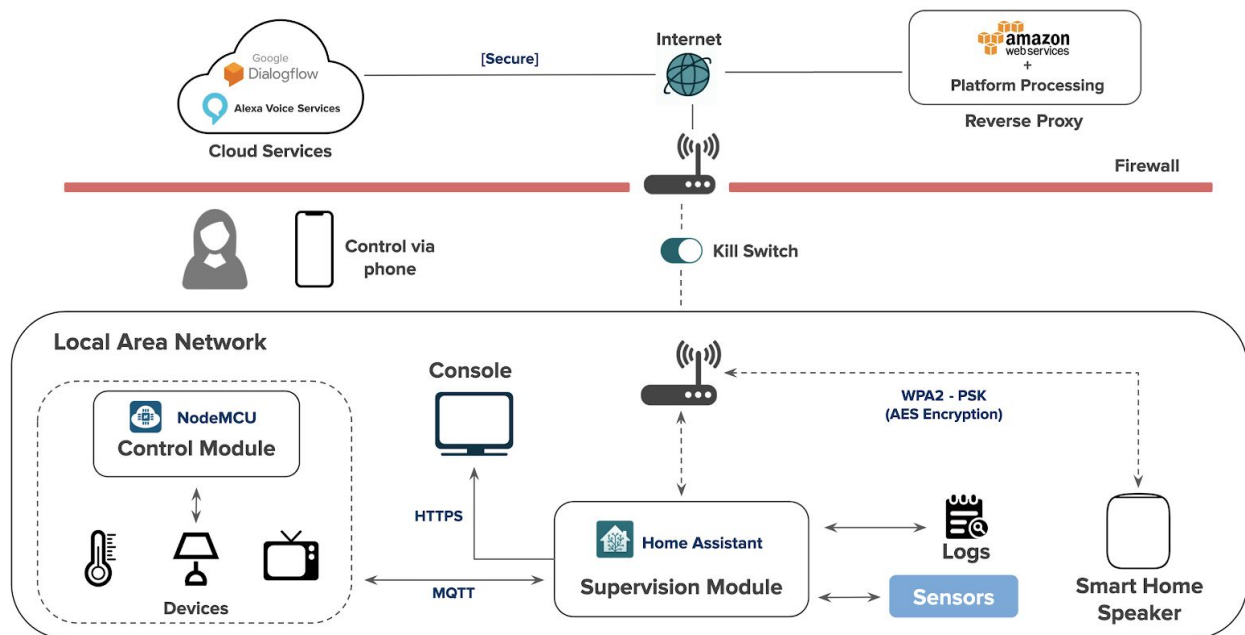


Figure 7.4: Detailed Architecture

There is a presence of a kill switch and a firewall on the primary router. These are features introduced to strengthen the security posture of the solution proposed. The firewall protects all the devices NATTED behind it from requests that are of unknown origin. The kill switch is a safety feature that is built into the architecture for the user. If the user finds the need to cut out the voice services or completely isolate his/her smart home from any kind of internet access, then the kill switch can be used. The smart home will still continue to function and support all of its features except for the ones dependent on the internet.

The cloud services like the voice backends of the voice assistants are located in remote servers and these servers, upon successful processing of the voice commands, send webhook requests along with appropriate data to the abstraction server running the

reverse proxy. The abstraction server performs all the platform specific processing on the data and finally hands the data to the home assistant over a websocket.

Thus, after understanding the architecture, one can see that the proposed architecture accounts for all the challenges and meets the functional as well as the non-functional requirements.

7.2.3 Design Choices

After the architectural phase, there were a few major design choices that were made as part of the project to deliver upon the non functional requirements. From the fig <no> depicting the detailed architecture of the system, the following points can be noted.

- **Reverse Proxy**

A reverse proxy in conjunction with websockets has been used in the system to provide greater security to the entire solution. The use of this design pattern helps in establishing communication with the smart home without revealing or ever storing the true IP address of the client system.

- **Websockets**

This protocol for communication is used between the reverse proxy and home assistant software. Websockets are protocols optimised for event based communication where the server notifies the client whenever an event occurs. This way the client needs to continually poll the server, thereby reducing the processing load on the client side.

- **Home Assistant**

This is an open source operating system that is well optimised for the IoT scenario and event based processing. There is availability to integrate entities into the software, track, monitor as well as control them. It also provides an intuitive user interface and a variety of customizability options.

- **Node-RED**

This is a piece of software that is used as the middle ware to connect different moving parts of the system. It links together the control module, supervision module, the MQTT server and the reverse proxy to provide. It has a user friendly UI, easy customization options and a variety of plugins to support various features and functionality.

The above mentioned design choices were made after a good amount of research and critical evaluation between the trade-offs and gains of choosing one option over the other.

8. WORKFLOW

A workflow in a Computer System describes the sequence of tasks performed to obtain a particular outcome. In this project, the workflow has gone through multiple iterations to arrive at the final one.

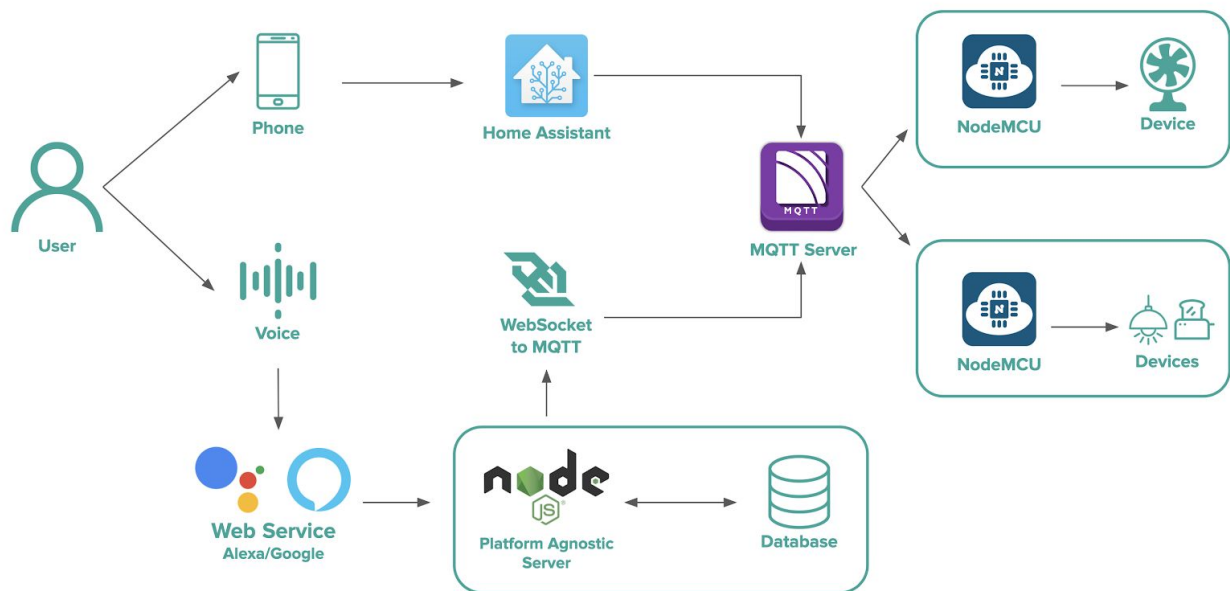


Figure 8.1: Workflow

As seen in Figure 8.1 depicting the workflow, the user has two ways of interacting with the home automation system - one is via voice, and the other is through the smartphone application.

A user can issue a voice-based command to either an Amazon Alexa or Google Assistant supporting speaker. The command, once sent to the voice assistant's web service, is forwarded to a Node.js abstraction server. This server is what makes the system Platform-Agnostic. The server checks whether the command has been issued from a Google Assistant or Amazon Alexa device. Along with this, the server checks the correctness of the command by comparing the keywords in command to data already available in the Database, like the presence of correct entities and invocation of the right intents. This error handling function is very crucial to the system as it checks whether the user has given sufficient data or not. If not, he is prompted by the respective web service to provide the missing data. After validation of the command, the server forwards the

appropriate details present in the command like device name and task to be performed to the MQTT Server. This transfer happens via a WebSocket, a globally-used secure communication protocol. The MQTT server then publishes this data to appropriate command topics. The control modules subscribed to this topic are notified of this, and they take the appropriate action. It can be to toggle a device on or off, or change certain attributes in a device, like brightness or speed.

The second means of interaction with the system is on a smartphone. The user can log in to the smart home using the Home Assistant application. The user can use the appropriate entity cards that have been created to access any device. Based on the user's choices, appropriate data is sent to the MQTT Server which then publishes it to command topics. The control modules then perform the required action. This mode of interaction works even in the absence of an active internet connection as it does not involve voice command processing by the web services.

The two forms of interaction ensure ease-of use and provide flexible ways of accessing the smart home. Providing one without the other would lessen the capabilities of a smart home system.

9. IMPLEMENTATION

After clearly defining the use cases as well as modelling misuse cases and threats, the implementation do's and don'ts have been well established. The below points adhere to the scope of implementation which have been set while defining the project specific functional and non-functional requirements.

9.1 Security by design

Security by design in Software engineering means that the software has been built from the foundation to be secure. A robust architectural design is at the very core of this technique and the security is incorporated into the system from the group up.

This project incorporated *Security by design* through the following steps

- **Network Segregation**

- The design incorporates a two router setup. The internal router is the point of access for all the end devices within the home while the external router acts as an entry point for requests from the internet.
- The two routers are on different networks. Any device scanning the external network would be unable to detect the devices connected to the internal network. All the devices connected to the internal network can access the internet as well.
- Furthermore, static IP has been assigned to the internal router and routing rules have been set up in the external router.

- **Web Sockets**

- The communication between the Reverse Proxy and the Local Area Network (LAN) needs to be instantaneous and secure. Furthermore, a long lasting session is required as a home command can be invoked by the end device at any time.
- Creating a new TCP connection for every request from the Reverse proxy to the LAN would be a huge overhead. The time to set up every new connection

would increase latency and if unregulated, it could be vulnerable to Denial of Service attacks.

- In order to overcome these challenges Web Sockets were used which allow for quick, full duplex communication over a single TCP channel.

- **Access Control in home assistants**

- Home Assistant runs on the Local Area Network by default on port 8123. However in order to access any setting or feature the user needs to sign in using their credentials which had been setup before. This ensures that any user who gains access to the home network will be unable to change the settings.

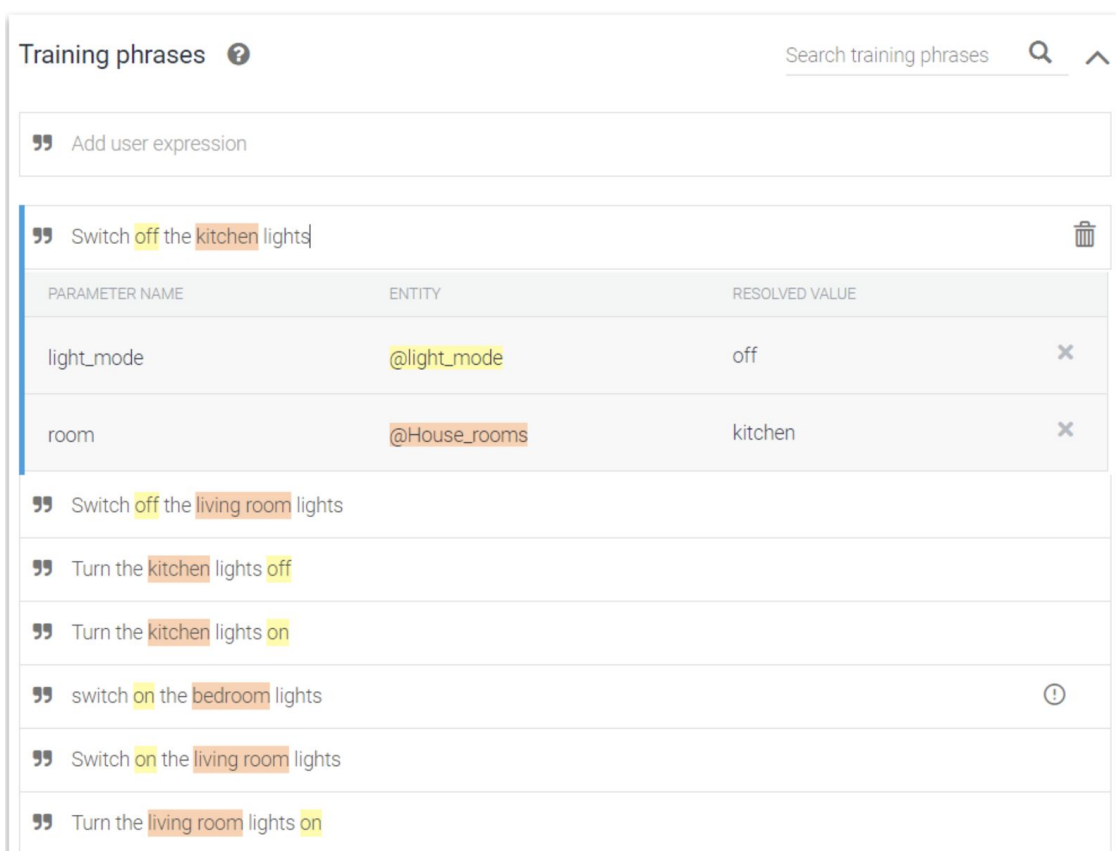
- **Secure protocols**

- The security and integrity of data transmitted over a network is entirely dependent on the security of its underlying network protocol. A secure network protocol prevents any unauthorized user, application or device from accessing the network.
- The communication between the Web Service running on the reverse proxy and the backend of the cloud voice assistant takes place over https protocol.
- The communication between the Reverse Proxy and the Local Area Network takes place through web sockets. The WSS protocol works over SSL/TLS and is encrypted thus protecting it from man-in-the-middle attacks.
- The publisher-subscriber model used is based on the MQTT protocol (Message Queuing Telemetry Transport). The MQTT protocol is not inherently secure. However, plenty of solutions which help make MQTT more secure can be commonly found in IoT applications. Further, in this project the MQTT protocol runs only internally to our Local Area Network and not on any device exposed to the internet. This reduces the attack surface to a great degree.

9.2 Cloud and Skills

The first step in performing any smart home command involves voice-based invocation of the different commands that have to be supported. The voice invocation can be done using Google Home, Amazon Echo or using any of their variants (Google Home mini, Amazon Echo dot, etc). The voice commands can also be invoked using their respective mobile apps. The only prerequisite for this is to ensure that the devices are signed into the appropriate user accounts.

When a command is invoked the platform matches the voice request with a set of statements. These statements are called intents. Each intent may contain 'entities'. These are intent parameters which dictate the type of data present. Multiple such intents and custom entities types were defined for intuitive home commands which a user can invoke. Google provides the Dialogflow platform to help the developers build intents while Amazon provides the Alexa Developer console for the same. The Figure 9.1 demonstrates the intents and entities for a toggle device intent on Google Dialogflow platform.



The screenshot shows the 'Training phrases' section of the Google Dialogflow console. It includes a search bar and a list of training phrases. The first phrase, 'Switch off the kitchen lights', is selected, and its parameters are displayed in a table below it.

PARAMETER NAME	ENTITY	RESOLVED VALUE
light_mode	@light_mode	off
room	@House_rooms	kitchen

Below the table, several other training phrases are listed, including 'Switch off the living room lights', 'Turn the kitchen lights off', 'Turn the kitchen lights on', 'switch on the bedroom lights', 'Switch on the living room lights', and 'Turn the living room lights on'.

Figure 9.1: Development of a Google Action

When an invoked request is matched with an intent, the parameter extraction is carried out implicitly by each of the platforms. Each of these extracted parameters can then be sent to a backend web service (Node.js abstraction server) termed as a ‘fulfillment’. Once the desired processes are performed a response is sent back from the web service to the platform and is converted to speech output. The Figure 9.2 demonstrates the intents which are setup for an appliance toggle request on alexa developer console platform.

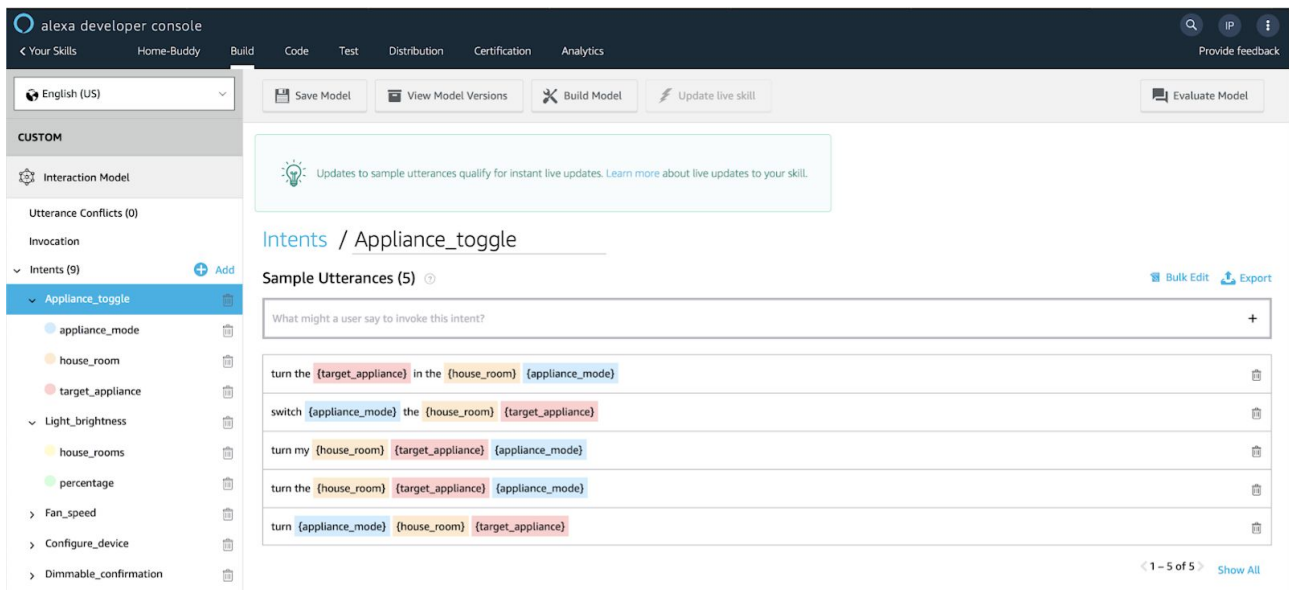


Figure 9.2: Development of an Alexa Skill

A voice based communication needs to handle the numerous nuances in human speech, at the same time keep the ease-of-use as simple as possible. The intents have been designed such that in case a necessary entity is missing the platform automatically prompts the user to specify the missing information. Only after sufficient data has been collected to process the intent, is the intent request sent to the fulfillment. This gives a more natural feel to the conversation between the user and the voice assistant. Furthermore, error handling has been incorporated ensuring robustness in the system. A good example of this would be when the user tries to toggle an appliance, say a bed lamp in the bedroom, which has not yet been setup. The fulfillment processes this requests and queries the database to realize that such a device does not exist in the said room. It returns an appropriate error message and prompts the user to set up a device before use.

9.3 Abstraction Server

One of the main requirements of the project included a platform agnostic way of processing requests. This meant a level of abstraction was required to handle and respond to requests from both Amazon and Google platforms. The best place to handle this would be on the Reverse Proxy which acts as the endpoint for all the requests from the cloud voice assistant services.

The web service running on the reverse proxy has been written in Node.js. No additional sdks for processing the requests have been used in order to minimize dependencies and improve maintainability. The request first gets classified as originating from Google Dialogflow or Alexa Developer skills by reading the structure of data present in the packet body. This distinguishing is required since parameter extraction and response generation would be different for each platform.

When a device needs to be set up, the invocation triggers a “setup device” intent. The setup device intent needs to know the device name, room in which to set up, the node ID and the pin on the node. This data is stored on the Abstraction server in a database. Such a database is required since further invocations to toggle devices like ‘bed lamp in bedroom’ do not need to ask for the Node ID and pin number each time. The toggle intent would only need the device name, the room and the status (On/Off). The abstraction server parses the toggle invocation and queries the database to find the setup details of this device (which have been setup previously). These details are sent to the Supervision module running on the Local Area Network through the Web Socket protocol.

One important thing to consider would be how the different error cases are handled by the system, as intuitively as possible. For instance, if a user tries to set up a device which already exists in a particular room, then they are informed of this clash and prompted to rename the device. Further, intents to remove a device which has already been setup or to modify the details about a device (pin number, node ID, room) are provided to make the task of configurability and manageability as easy as possible.

For the purpose of this project, the Abstraction Server has been set up on an t2.micro Amazon EC2 instance. An Amazon API Gateway is set up to redirect requests to the web

service on the EC2 instance. The advantage of doing this is that the API Gateway gives us a https endpoint without having to go through the hassle of creating certificates.

The Figure 9.3 describes the response received when a duplicate device with the same name is found in the room in which it is being set up. Responses from both the platforms can be seen here.

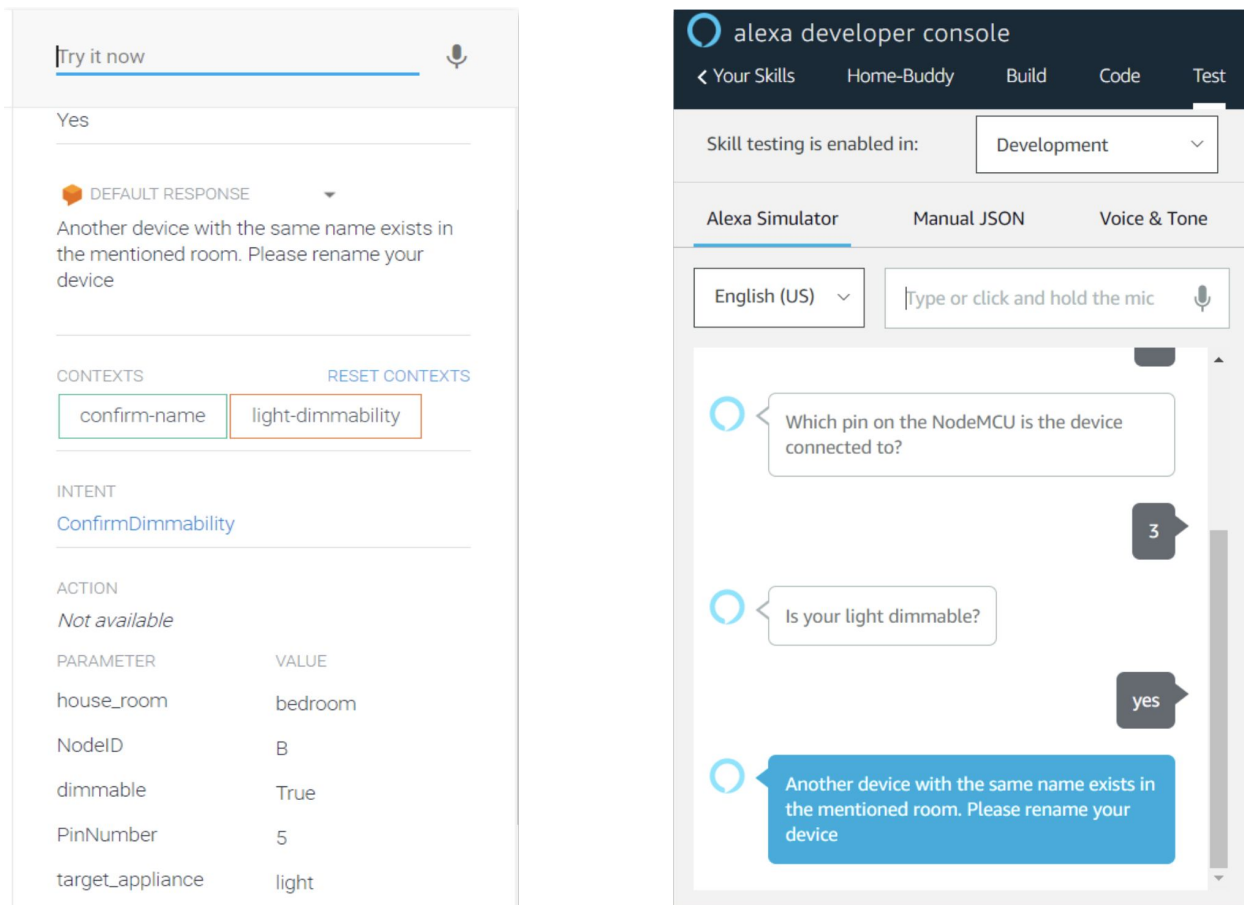


Figure 9.3: Error Handling in Google and Alexa

9.4 Control Unit

A basic block or unit of replication needs to be clearly defined from a hardware perspective. This is the minimum necessary hardware required to enable voice automated control of the particular device. In this case, the control module along with the devices and hardware tools would serve as the basic block. The control module is NodeMCU (ESP8266). It is a wifi enabled board that can interface with hardware controllers using software signals.

One important requirement in order to achieve easy configurability of a new device is to minimise the hardware changes required during device set up. Thus, the code was restructured in such a way that all the control units that belong to the smart home automation system can run a uniform agent on them. Creating such a uniform agent running on all the control units increases the configurability of the system tremendously as the processing logic can be moved to the supervision module. The software running on the control units need not be changed whenever a new device is added or when an existing device is removed or modified.

The agent, on start up, attempts to establish an internet connection by logging into the local area network via the router. Once it is successfully connected, it establishes a connection to the MQTT server that is running on the LAN, and subscribes to all the required topics. The agent is now ready and is waiting for commands from the supervision module. When the agent receives a request, it parses the data received and triggers the appropriate pin action, thereby controlling the required device. As a uniform agent running on all the control units is listening to the supervision module, changes to the smart home network can be performed easily. For instance, if there is a smart room light that needs to be moved from the bed room to the living room, there is no need to change the code on the control units, all that is required to be done is to redirect the request to the control unit present in the living room rather than forwarding it to the one present in the bedroom.

The smart home comprises a variety of devices like fans, heaters, lights, television sets, music systems and much more. The way the control unit is able to interact with such a diverse device base is through a neat trick. After some observation, one realises that all kinds of electrical devices have only two ways of interaction

- **Current Control**

This is where the current supply to an electrical device is toggled, thus turning on or turning off the device.

- **Voltage Control**

This is where the voltage being supplied to the device is varied to change some behaviour of the device. For example, changing the voltage supplied to a light bulb changes its brightness.

These concepts were inculcated into the software of the control unit and hence the control units (that contain the microcontrollers, relays and voltage controllers) are able to interact with almost all the devices of the smart home.

9.5 Supervision Module

The supervision module is the brains of the system and acts as the central hub receiving, forwarding and processing all the requests. Home Assistant was the choice for the supervision module due to reasons already mentioned. There are three main components that are a part of the supervision module

- **MQTT Server**

This is the communication server that manages all the topics, subscribers and publishers. The MQTT server is chosen to run locally on the supervision module to reduce the latency of communication and also avoid the risk of packets being sniffed over the internet. Hence, this helps achieve better performance and security.

The topics to which the smart home devices subscribed/published had to be restructured too in order to support the uniformity of control unit agents and stick to the promise of configurability. As part of implementation, there were a lot of topic formats that were tested before arriving at the final one. The fig <no> depicts the final design iteration for the topics format.

Earlier: `/smarthome/bedroom/light/brightness`

Now: `/ smarthome / bedroom / A / 5 / analog / state`

Smart Home

NodeMCU
ID

Adjustable

Room Name

Pin Number

Figure 9.4: MQTT topics format

• Node-RED

This is the middleware used to connect various sources of information and the different moving parts of the system. The figure <no> depicts an example of how the Node-RED software can be used to connect the websocket from the abstraction server to the MQTT server running locally and also monitor the messages being exchanged as part of normal operation.

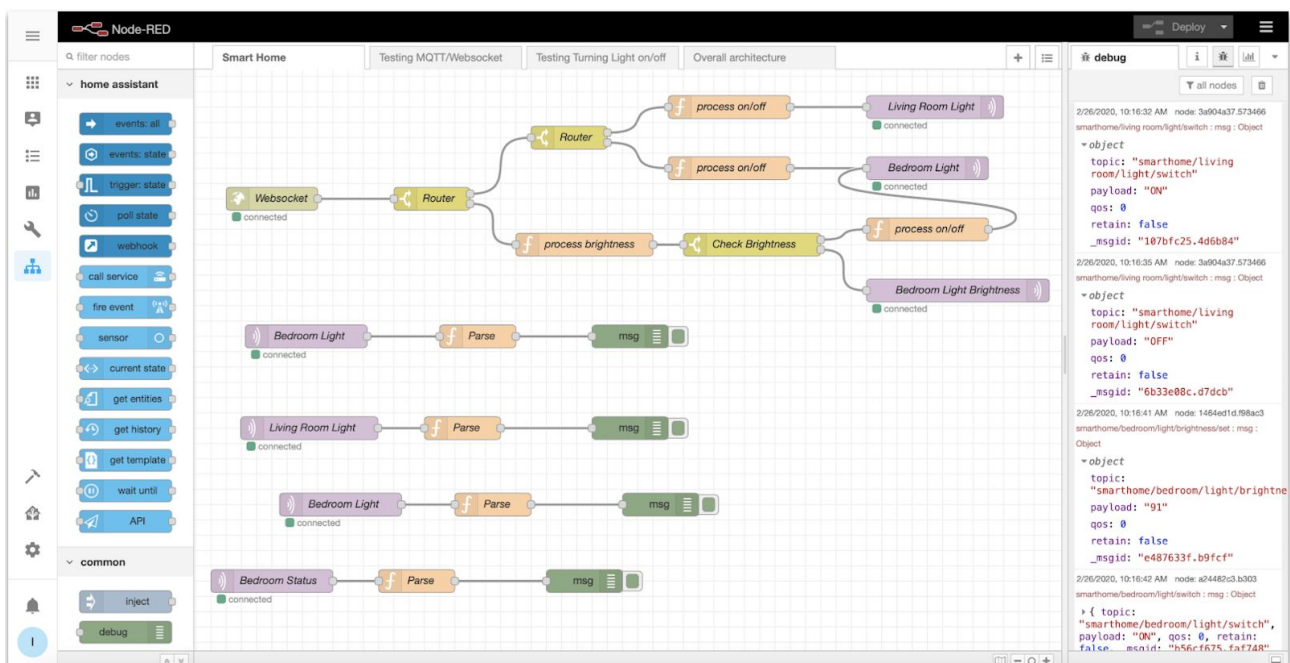


Figure 9.5: Node Red

• User Interface

The user interface that is provided is hosted on the supervision module. The user accesses this in order to control the devices via a web or mobile application. The Home Assistant software provides a great UI that is customisable, intuitive and integrates well with a variety of devices. As part of this project, the UI was designed keeping manageability in mind. As the number of devices that are a part of the smart home grow in number, it becomes difficult to control and monitor the devices.

To overcome this, the UI was designed in terms of pages and cards. The Figure 9.6 shows the UI that was developed for this purpose. There are pages for each room and the page contains the devices that are a part of their respective rooms. All the information is presented in terms of cards. In addition, there is also a Dashboard that gives an overall view of the entire smart home, its users along with some important information.

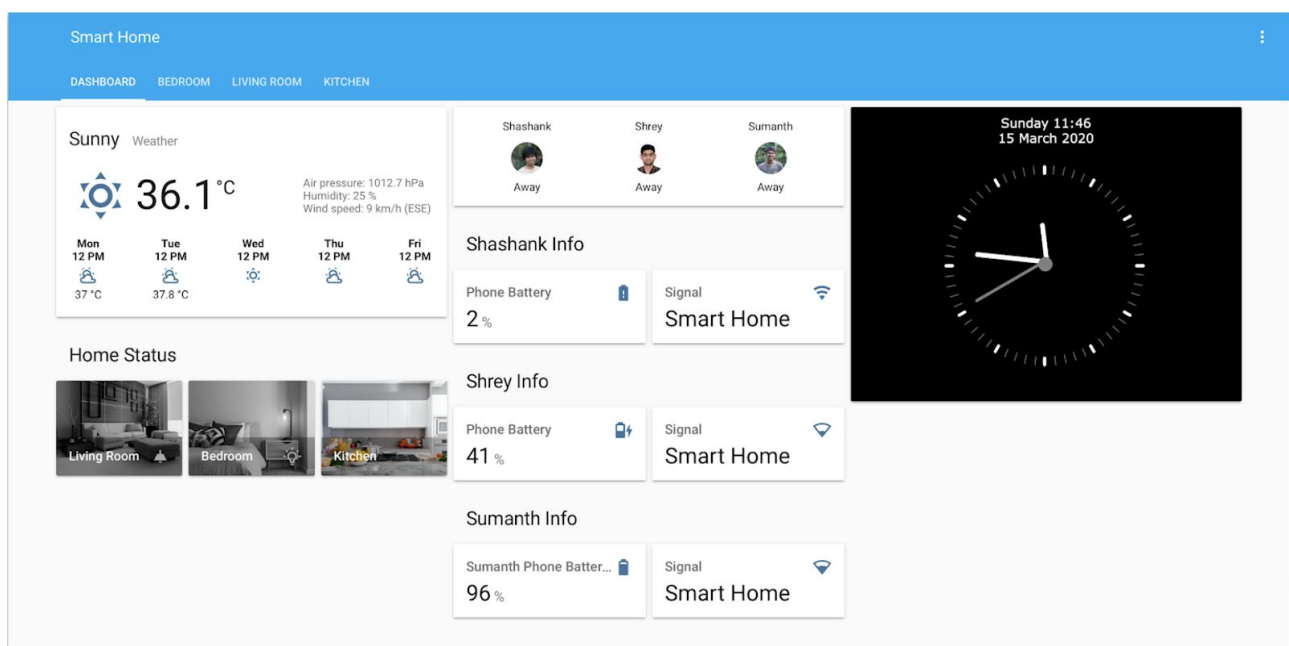


Figure 9.6: UI for the Home Assistant

It is also possible to access this locally hosted UI remotely over the internet. There is a landing website that is created on the abstraction server. This website gives some general information about the team members and the project itself. On this website is a link to the locally hosted UI. The figure Figure 9.7 depicts the landing home page.

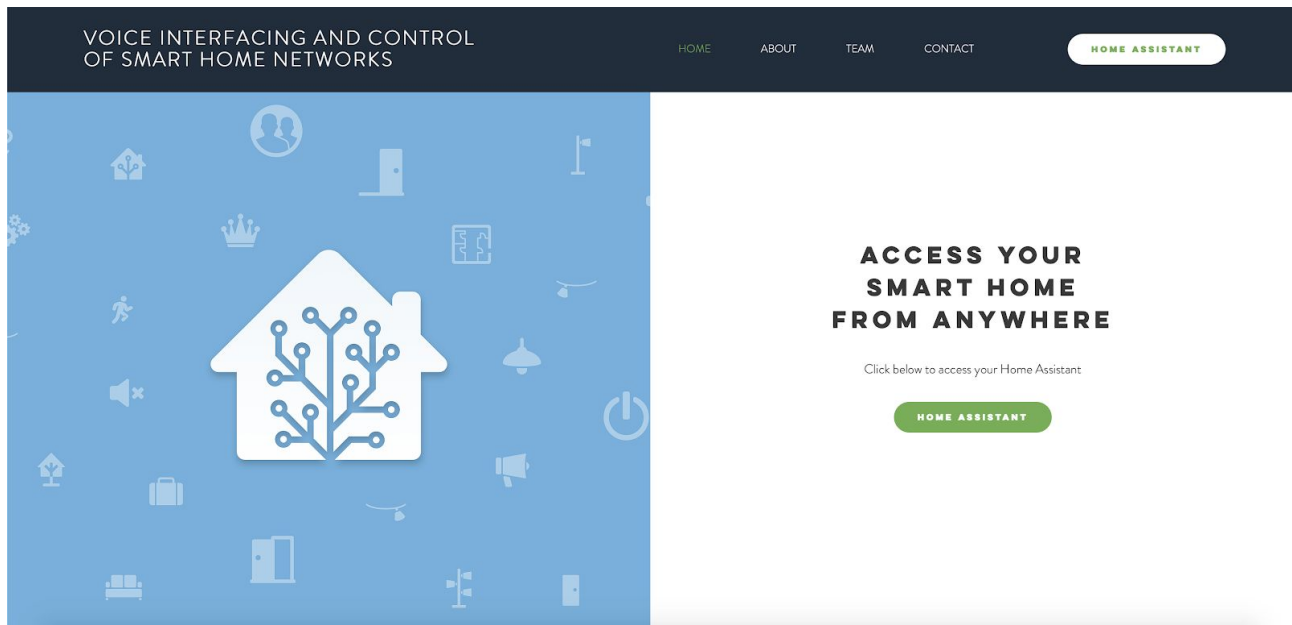


Figure 9.7: Website for the project

This section makes it clear that the project has multiple dimensions to it and some informed design choices as well as architecture decisions have been made. The different modules were implemented individually and integrated into the system to render a fully-functioning Home Automations system that is secure, scalable and platform-agnostic.

10. RESULTS AND CONCLUSION

A systematic approach to creating a home automation system was followed throughout the project. For defining requirements, designing an architecture, considering security features and for every other step during the execution of this project, a methodical way of clearly defining what is to be done was followed. This helps validate our results along well-defined verticals.

- **Security by design**

After following the use case and misuse analysis, followed by the STRIDE Threat Modelling Technique, security features were incorporated into the architecture of this system. The two-router setup described earlier facilitates secure access to the Home Assistant, while the user is still connected to the outside internet. Similarly, the choice to use WebSockets also ensures secure, full-duplex communication. Thus, the promised security was offered in the solution.

- **Platform-agnostic nature**

Since the home automation system supports both Google Assistant and Amazon Alexa voice assistants, the choice of voice assistant does not limit the functionality of the system. The user is free to choose any of the two smart home speakers, and the solution is compatible with both.

- **Easy-to-use Interface**

The fact that voice can be used to interact with the system ensures least effort from the end user. The voice commands that the user can give to the system are intuitive in nature. The other means of interaction, that is, through the web interface or on the smartphone application, is also user-friendly. Easy-to-understand icons and interaction elements help the user access the smart home easily.

- **Configurability**

The code was restructured to ensure easy addition of new devices.

- **Hands-free addition of new devices**

The code structure that has been used ensures that different devices in different rooms can all be controlled using a single uniform code running on the control modules. This enabled hands-free addition of new devices, a very convenient feature in the smart home. This has been tried with a number of devices, both on the Google and Amazon platforms.

- **Error handling**

Invalid and incomplete commands have been dealt with in an elegant manner. The voice assistant prompts the user to provide enough information, in case the user provides incomplete information. Conflicts in device names and other inconsistencies on the user's end have also been handled so that the user's intent can be fulfilled to as large an extent as possible.

- **Easy Manageability**

Uniform code running on all control modules means easier manageability. Any change that has to be done should be done in only one place and this makes it convenient for the developer.

All in all, a holistic study and an in-depth understanding of various technologies, architectures, the market space, and customer requirements helped arrive at the home automation system described in this project. All the requirements that were promised were met in this project.

11. DISCUSSION

The project has been executed successfully. Though enough thought has been put into the implementation of the 'Proof Of Concept', there are some aspects that can be added to the solution to make it a product that a consumer can buy. These, when added in the future will make it a well-rounded consumer product. Some of the domains that demand focus in the future are:

The following features are possible ideas which can be incorporated in the future

- **Voice Recognition**

By adding the ability to recognize the users' voices uniquely, the voice itself can be used as a means of authentication to access the smart home, where unauthorized voices would not be allowed to issue commands. Though Google Assistant already offers this solution, integrating it with the smart home would add more value to it.

- **Compound Actions**

This is the ability to perform a bunch of actions through the invocation of a single command by the user. If the home automation system is able to understand the intents of the user and perform the required actions without the need for explicitly giving the command, this would render the entire solution more capable and desirable.

- **Machine Learning and Artificial Intelligence**

The ability to engineer a truly intelligent smart home which requires little to no management on the users' part and can make decisions based on the usage patterns and historical data. The possibilities of enhancing the solution using machine learning are endless. This would help decrease the human effort involved and provide a better experience.

12. REFERENCES

- [1] Y. Igarashi, M. Hiltunen, K. Joshi, and R. Schlichting, “An Extensible Home Automation Architecture based on Cloud Offloading”, In 18th International Conference on Network-Based Information Systems, 2015.
- [2] D. Greenstreet and J. Smrstik, Texas Instruments, “Voice as the user interface – a new era in speech processing”, May 2017.
- [3] V. Stangaciu, V. Opa[^]rlescu, P. Csereoka, R. D. Cioarga [~], and M. V. Micea. “Scalable Interconnected Home Automation System”, In 21st International Conference on System Theory, Control and Computing (ICSTCC), 2017.
- [4] D. Kumar, R. Paccagnella, P. Murley, E. Hennenfent, J. Mason, A. Bates, and M. Bailey, “Skill Squatting Attacks on Amazon Alexa”, In 27th USENIX Security Symposium, ISBN 978-1-931971-46-1, Aug., 2018.
- [5] Booz Allen Hamilton, “2019 Cyber Threat Outlook”, 2019.
- [6] K. Bhardwaj, J. C. Miranda, and A. Gavrilovska, “Towards IoT-DDoS Prevention Using Edge Computing”, Georgia Institute of Technology.
- [7] J. Cichonski, J. Marron, and N. Hastings, “Security for IoT Sensor Networks”, National Institute of Standards and Technology, February 2019.
- [8] L. Røstad, “An extended misuse case notation: Including vulnerabilities and the insider threat”, Norwegian University of Science and Technology, Trondheim, Norway, 2006.
- [9] M. Fagan, M. Yang, A. Tan, L. Randolph, and K. Scarfone, “Security Review of Consumer Home Internet of Things (IoT) Products”, National Institute of Standards and Technology, U. S Department of Commerce Tech. Report, October 2019.
- [10] C. Withanage, R. Ashok, C. Yuen, and K. Otto, “A Comparison of the Popular Home Automation Technologies”, In IEEE Innovative Smart Grid Technologies, 2014.

-
- [11] R Style Labs, “Building Intelligent Connected Home Solutions: Challenges and Ways to Overcome Them”, 2019.
- [12] D. Dodson, T. Polk, M. Souppaya, W. C. Barker, P. Grayeli, and S. Symington, “Securing Small-Business and Home Internet of Things (IoT) Devices: Mitigating Network-Based Attacks Using Manufacturer Usage Description (MUD)”, National Institute of Standards and Technology, U.S Department of Commerce Tech. Report Draft, November 2019.
- [13] J. Coumau, H. Furuhashi, and H. Sarrazin, “Smart Home is Where the Bot is”, *usenix.org*, [Online]. Available: <https://www.usenix.org/system/files/conference/usenixsecurity18/sec18-kumar.pdf>. [Accessed Sept. 18, 2019].
- [14] C. Pradyumna, “Google Actions VS Alexa Skill - A Comparative Study”, *medium.com*, [Online]. Available: <https://medium.com/archieai/google-actions-vs-alexa-skill-a-comparative-study-4e7bdd3c3c62>. [Accessed Sept. 2019].

13. BIBLIOGRAPHY

Apart from the papers and articles that were read to bring different parts of the project together, many other information sources were visited to help in the execution of this project.

- <https://www2.meethue.com/en-us/works-with/amazon-alexa/echo-plus>
- <https://dev.to/webhookrelay/controlling-gadgets-with-google-home-ifttt-and-node-red-3ea2>
- Amazon Alexa Skills Documentation: <https://developer.amazon.com/en-US/docs/alexa/ask-overviews/build-skills-with-the-alexa-skills-kit.html>
- Google Dialogflow Documentation: <https://cloud.google.com/dialogflow/docs>
- Trends shaping the Internet of Things Business Landscape: https://www.mckinsey.com/business-functions/digital-mckinsey/our-insights/ten-trends-shaping-the-internet-of-things-business-landscape?cid=other-eml-alt-mip-mck&hlk_id=67f67df07d474f8c9287d992adfd11f7&hctky=10221563&hdpid=0cacce55-f7de-4830-8089-2575a5a9d632
- NodeRED Documentation: <https://nodered.org/docs/tutorials/>
- The IoT Business Model: [https://hbswk.hbs.edu/item/the-internet-of-things-needs-a-business-model-here-it-is?cid=spsmailing-28183089-WK%20Newsletter%2007-24-2019%20\(1\)-July%2024,%202019](https://hbswk.hbs.edu/item/the-internet-of-things-needs-a-business-model-here-it-is?cid=spsmailing-28183089-WK%20Newsletter%2007-24-2019%20(1)-July%2024,%202019)
- Features of Google Home Hub: <https://www.smartneighbor.com/blogs/neighbor-notes/5-things-you-can-do-with-the-google-home-hub>
- Comparison between Google Assistant and Amazon Alexa: <https://medium.com/archieai/google-actions-vs-alexa-skill-a-comparative-study-4e7bdd3c3c62>

14. APPENDIX

This section contains the full forms of certain abbreviations and the meanings of some words used in the document.

1. IFTTT - If This Then That

A web service that helps create conditional actions based on the occurrence of events. For instance, this can be to send a notification on a smartphone when a certain event occurs.

2. Blynk

An internet-based service that allows one to control microcontrollers over the internet.

3. STRIDE - Spoofing Tampering Repudiation Information-Disclosure Denial-of-Service Elevation-of-Privilege

A security threat model to help devise better countermeasures against cyber threat activities

4. Node-RED

Flow-based visual programming tool

5. MQTT - Message Queuing Telemetry Transport

A lightweight, publisher-subscriber based message transfer network protocol

6. NodeMCU - Node MicroController Unit

A low-cost Wi-Fi development System-on-a-chip (SoC)