UNIVERSIDAD
DE GRANADA

TRABAJO FIN DE GRADO

GRADO EN INGENIERÍA INFORMÁTICA

Septiembre de 2018

CREATION OF A VOICE-DRIVEN CONTROLLER FOR HOME AUTOMATION

Autor
David Vargas Carrillo

Director
Juan Antonio Holgado Terriza

CONTENTS

SECTION 1 INTRODUCTION

“

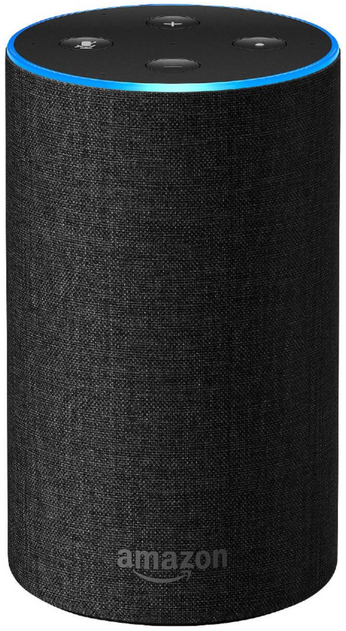
I am a HAL 9000 computer. I became operational at the H.A.L. plant in Urbana, Illinois on the 12th of January 1992. My instructor was Mr. Langley, and he taught me to sing a song.

-HAL 9000 on 2001: A Space Odyssey (1968)

SECTION 1 INTRODUCTION

Google Home

Apple HomePod



Amazon Echo

SECTION 1 INTRODUCTION

- **Problems in common:**
  - Very few **customization options**
  - **Limited compatibility** with domotic devices
  - They are generally **expensive**
  - There are **privacy concerns** regarding some of these companies
  - Generally, a **lack of a UI** that centralizes all the devices
  - **Availability**: at this moment, only Google Home is available in Spain, and it was not available when I began working on this project

SECTION 1 INTRODUCTION

- **General objectives:**
  - Design a system able to group home automation **devices from different makers**
  - Include **extra facilities**: automation and management from a mobile application
  - Make the system modular, extensible, safe and fully customizable
  - Make the system **accessible and adaptable**: usable with external screen, voice…
  - Make the common process **seamless and easy**

USABILITY

ADAPTABILITY

FUNCTIONALITY

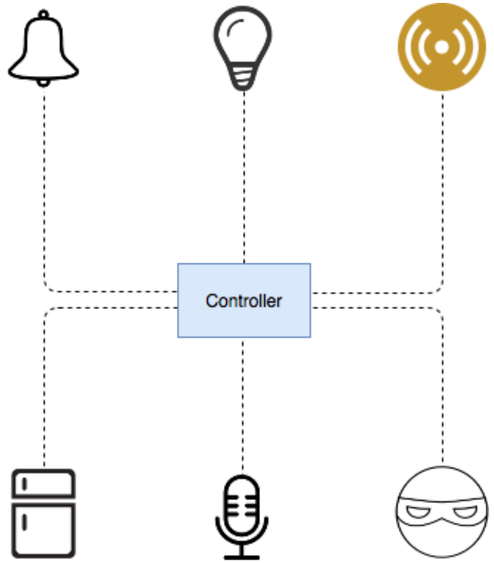
SECTION 2 MAIN CONCEPTS

SECTION 2 MAIN CONCEPTS

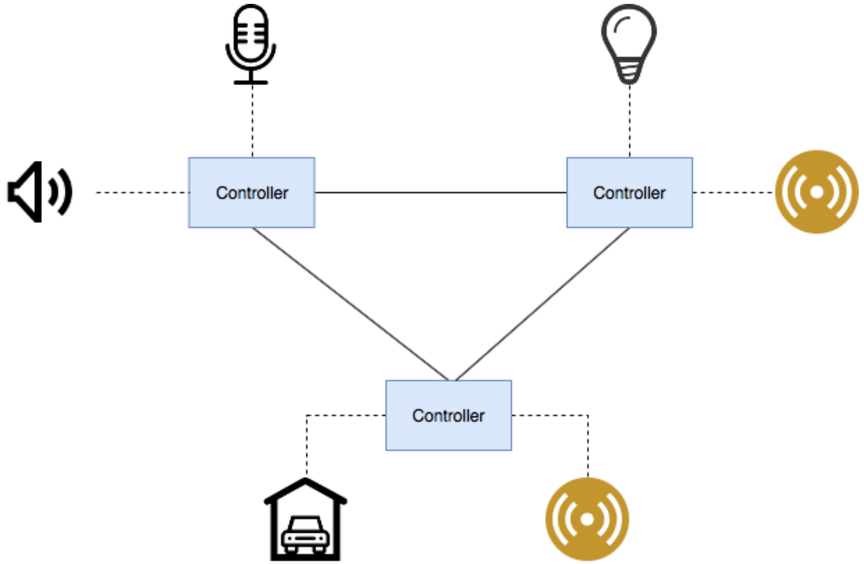
"**Home automation** is to employ sensors and control systems to monitor a dwelling, and accordingly adjust the various mechanisms that provide heat, ventilation, lighting, and other services."

Photo by Brian Babb on Unsplash

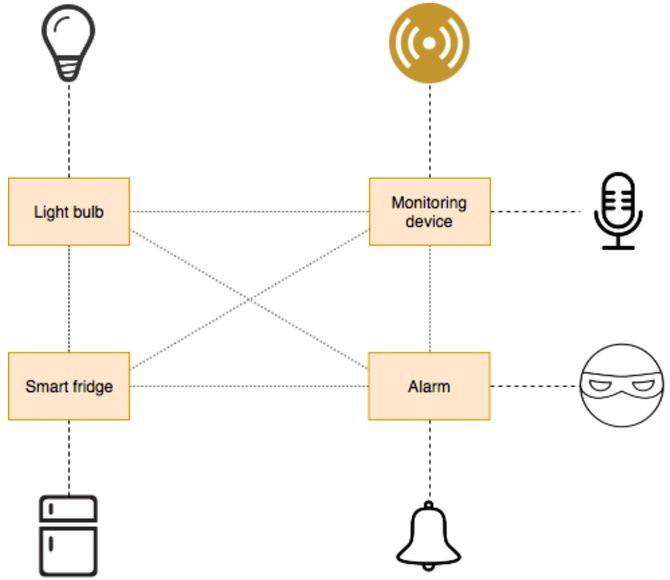
Home Automation System Design



Centralized architecture



Decentralized architecture



Distributed architecture

SECTION 2 MAIN CONCEPTS

"A **voice assistant** is an application program that understands natural language voice commands and can perform tasks or services for an individual"

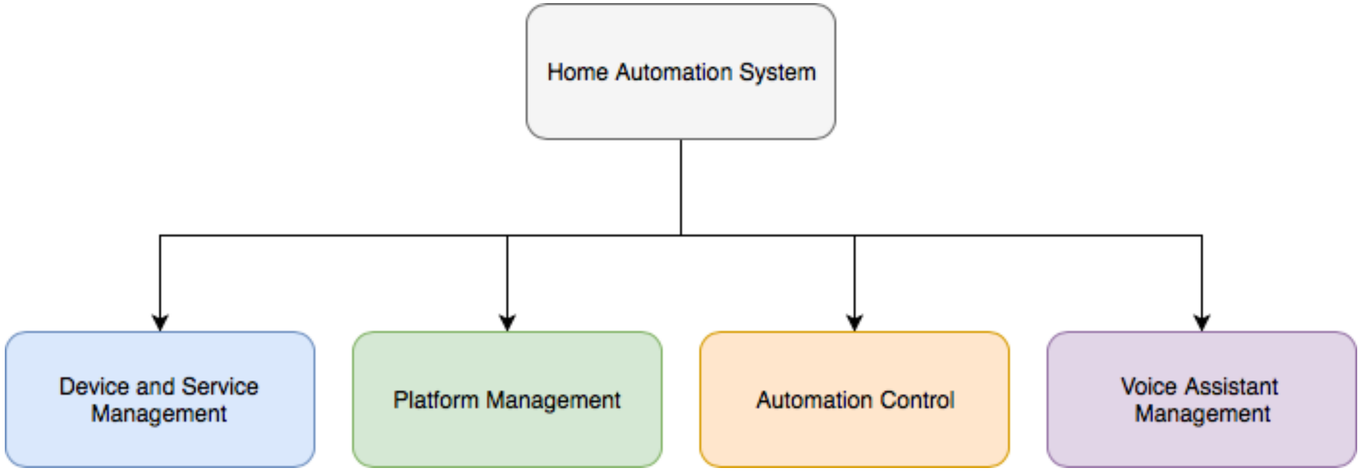
Photo by rawpixel on Unsplash

SECTION 3 VDHAC

VOICE-DRIVEN HOME AUTOMATION CONTROLLER

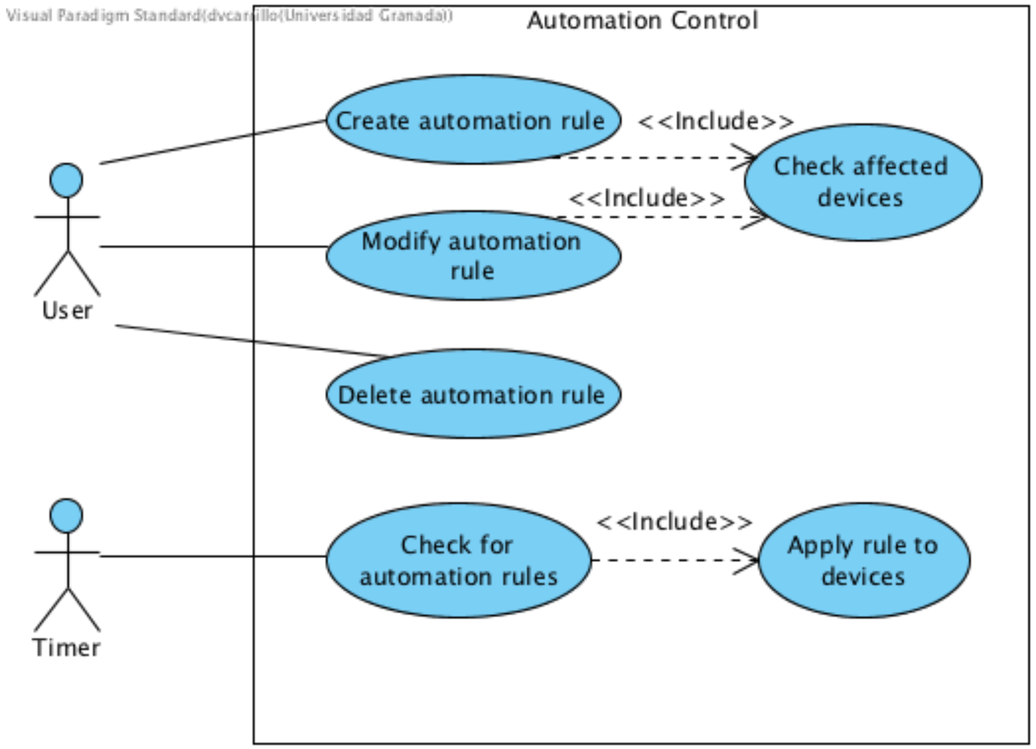
SECTION 3 VDHAC // SPECIFICATION

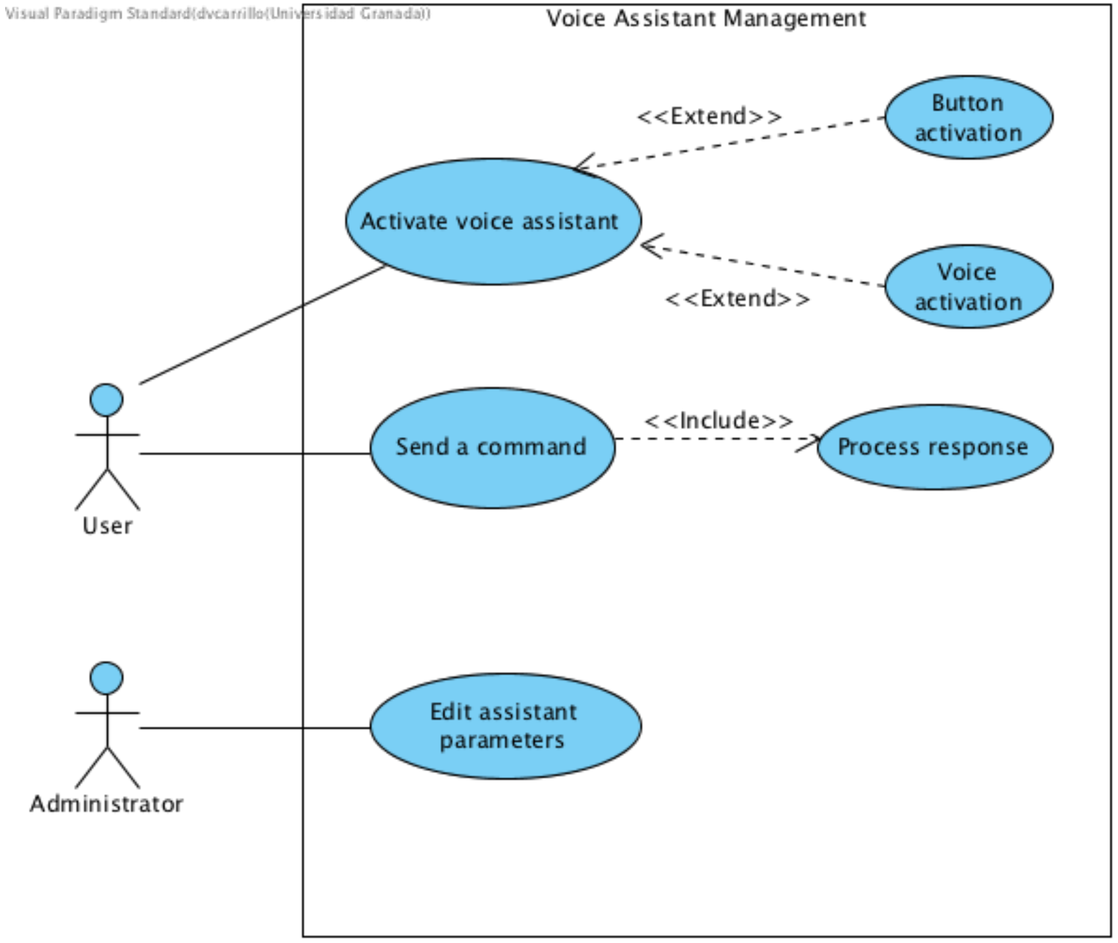
- **Use cases**
  - Description of actors:
    - User
    - Administrator
    - Timer
- Functional subsystems:
  - Device and service management
  - Platform management
  - Automation control
  - Voice Assistant Management

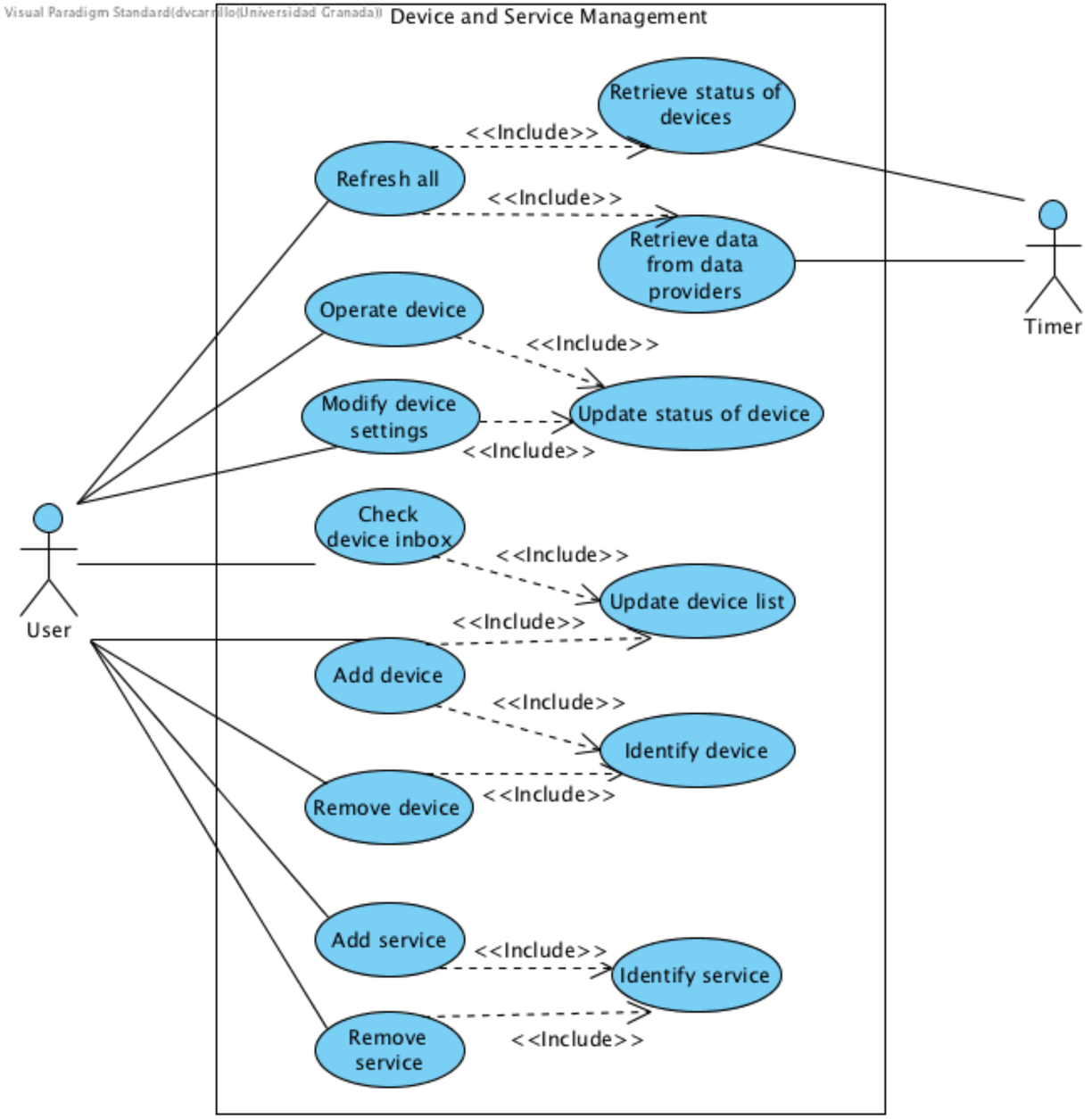


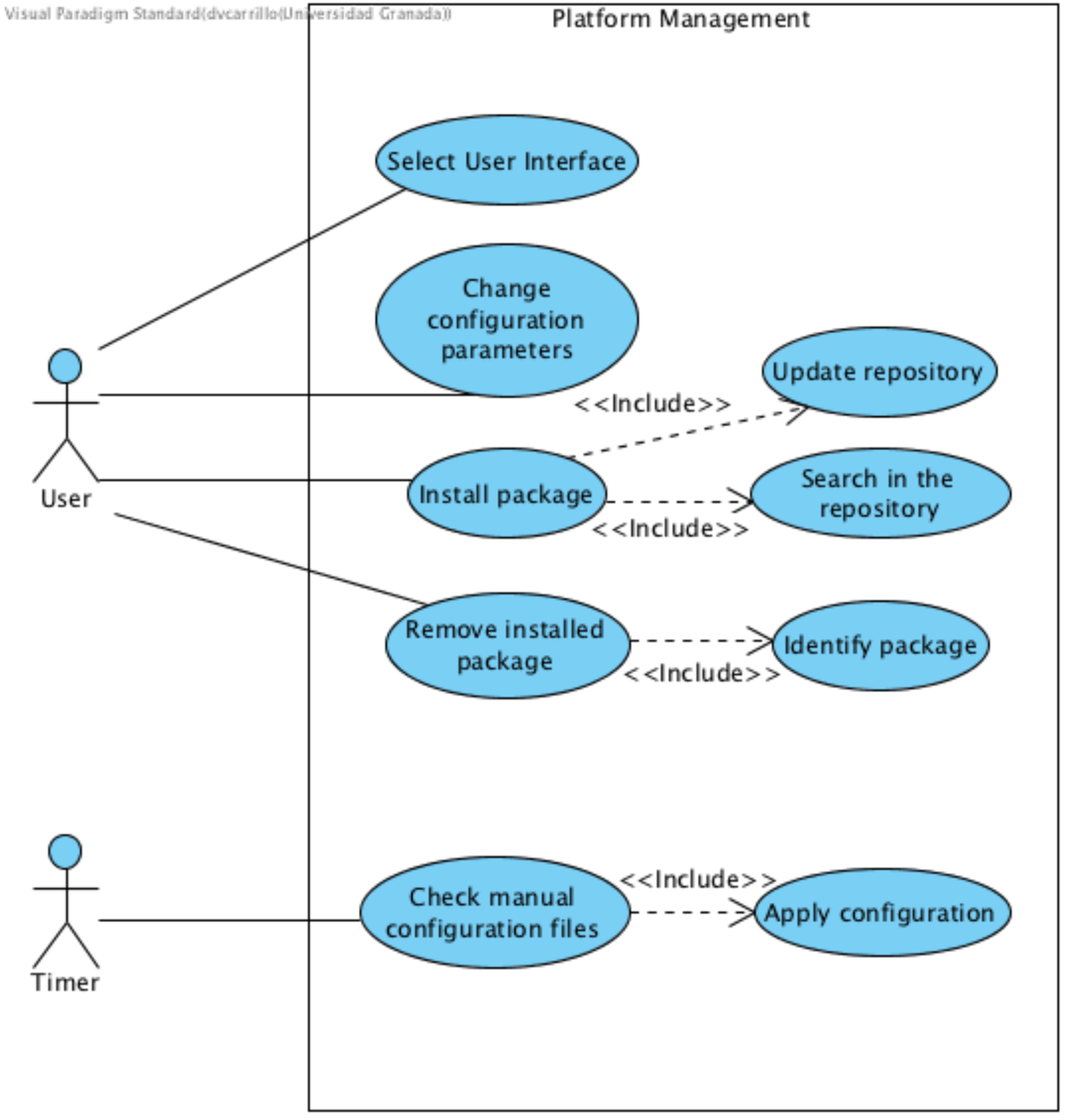
SECTION 3 VDHAC // SPECIFICATION

- Use cases diagrams



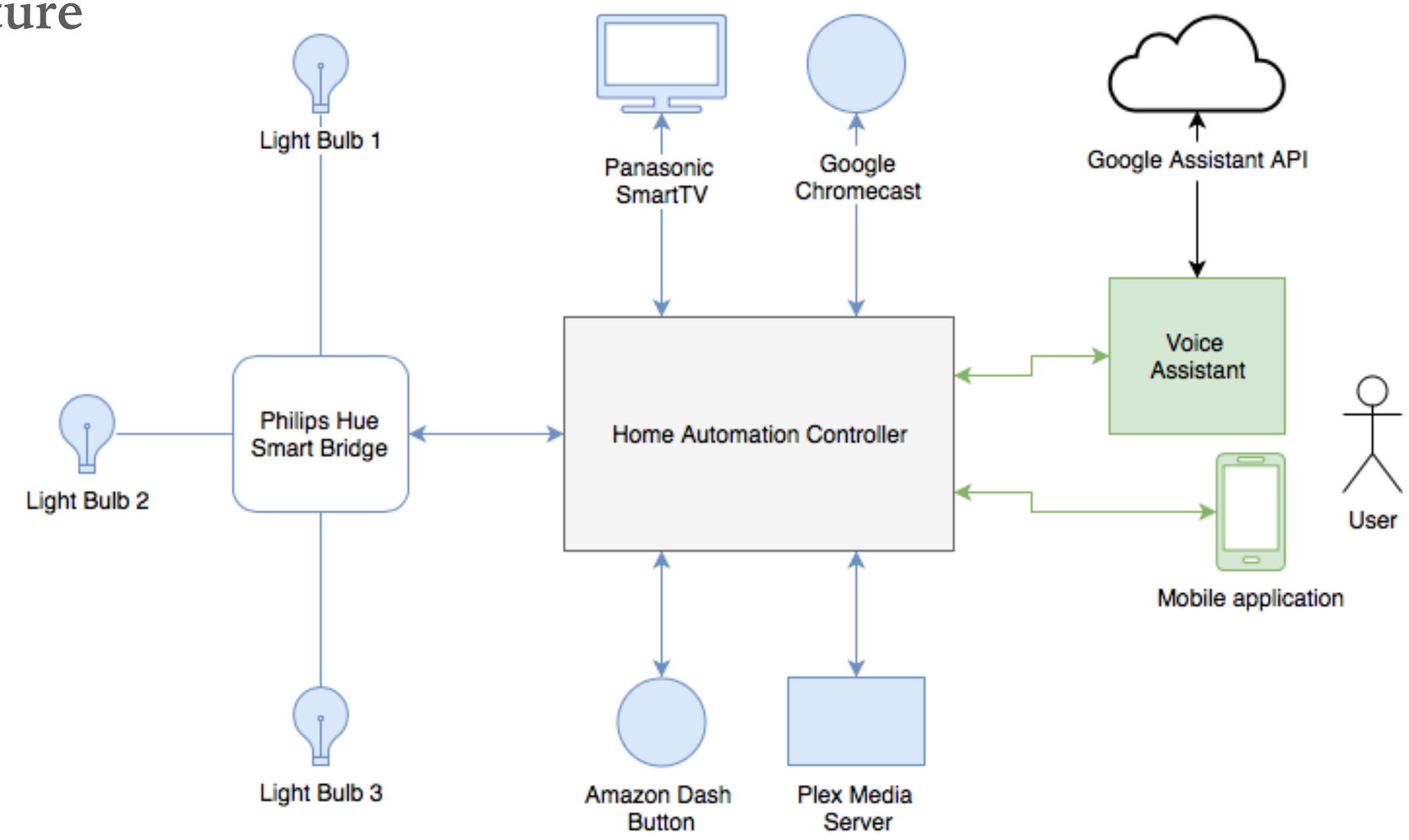



Visual Paradigm Standard(dvcarrillo(Universidad Granada))
Device and Service Management
Retrieve status of devices
Refresh all
Retrieve data from data providers
Timer
Operate device
Modify device settings
Update status of device
User
Check device inbox
Update device list
Add device
Identify device
Remove device
Add service
Identify service
Remove service
<<Include>>


Visual Paradigm Standard(dvcarrillo(Universidad Granada))
Platform Management
Select User Interface
Change configuration parameters
Update repository
Install package
Search in the repository
User
Remove installed package
Identify package
Timer
Check manual configuration files
Apply configuration
<<Include>>

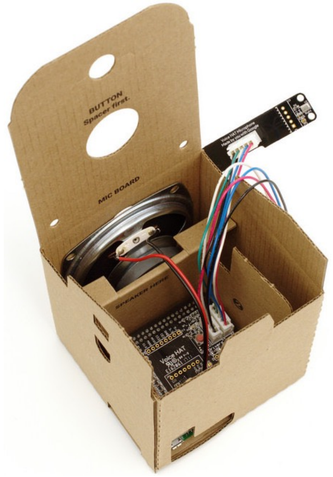
SECTION 3 VDHAC // SPECIFICATION

- Architecture



SECTION 3 VDHAC // SPECIFICATION

HARDWARE



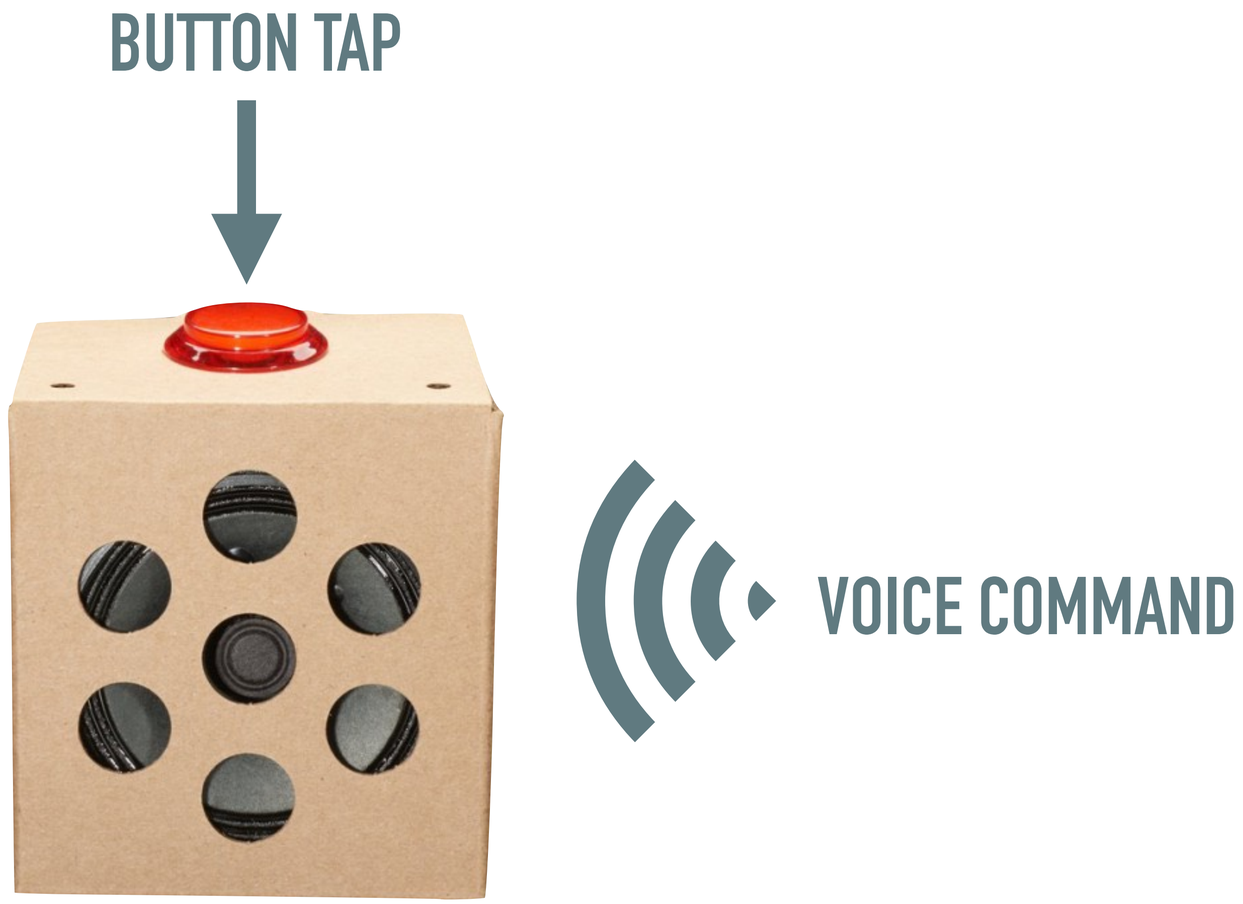
Home Automation Controller

SOFTWARE

SECTION 3 VDHAC // OPENHAB

- OpenHAB (open Home Automation Bus) is a **completely free, technology agnostic and open source platform for home automation**
- **Benefits**:
  - Capable of **integrating different domotic systems, devices and technologies** into a single solution
  - Provides **uniform user interfaces**, and a **common approach to automation rules** across the entire system
  - **Modular**, composed by *bindings*, that usually represent a device, a set of devices or a service that can be integrated in openHAB
  - Backed by an **active community**

SECTION 3 VDHAC // VOICE ASSISTANT



SECTION 3 VDHAC // VOICE ASSISTANT

- **Development of the voice assistant**
  - Based on a script that communicates with Google Assistant

- The STT part is done by the Google Assistant Library
- The command processing and TTS parts are done locally

➤ Speech-To-Text (STT)

Name	Assistant Library	Cloud TTS[17]	Pocketsphinx[11]	Watson STT[23][22]
Maker	Google	Google	CMUSphinx	IBM
Location	Cloud	Cloud	Local	Cloud
Voice HAT support	Yes	Yes	No	No
Required space	Insignificant	Insignificant	40 MB + language models	Insignificant
Recognition quality	Excellent	Excellent	Good	Excellent
Languages	English	120 supported	13 supported[1]	13 supported
Prince	Free (comes with the AIY Kit)	$0.006 USD/15 seconds after 60 minutes/mo.	Free	$0.02 USD/min. after 1,000 minutes/mo.[2]
Open source	No	No	Yes	No

[1]There are at the moment 13 language models available to download in their website. Although new languages can be added by creating new language models

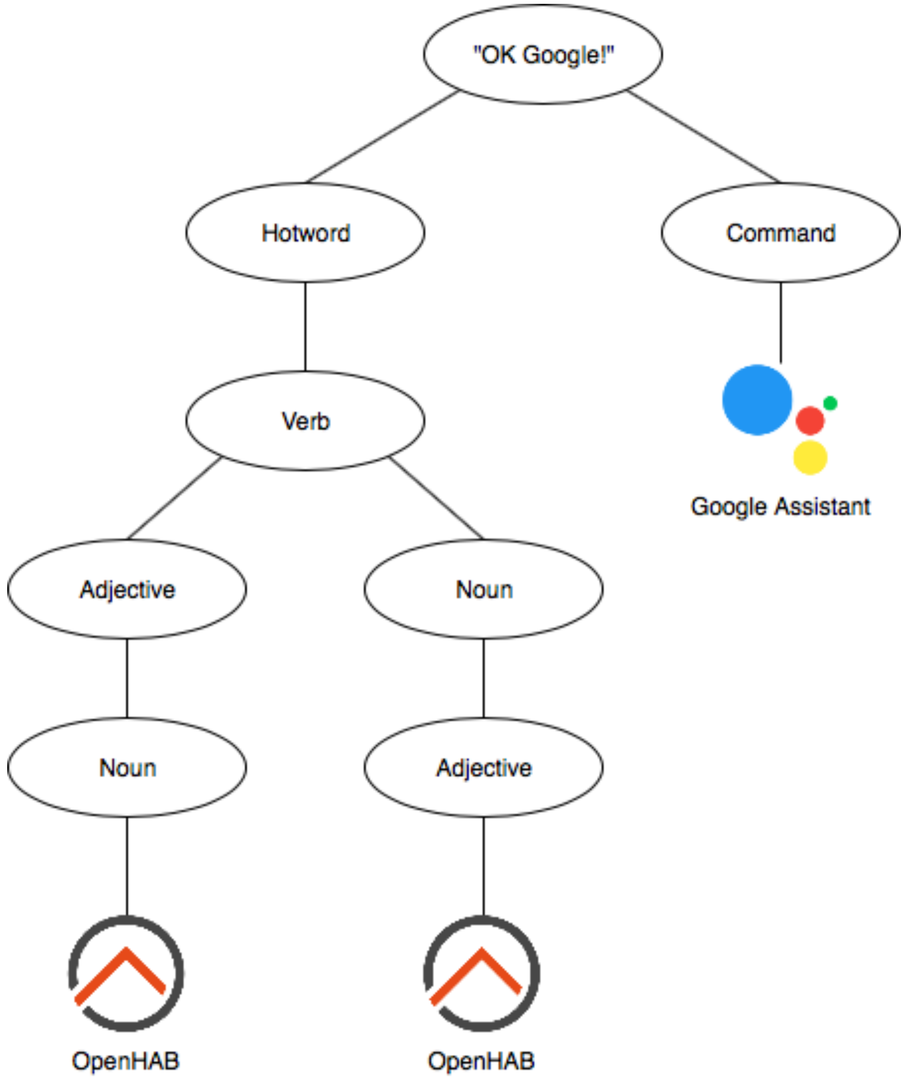
[2]This price is maintained until 250,000 minutes are used. The complete set of prices is: $0.02 USD for minutes 1,001 - 250,000, $0.015 USD for minutes 250,001 - 500,000, $0.0125 USD for minutes 500,001 - 1,000,000, $0.01 USD for minutes 1,000,001 and up

SECTION 3 VDHAC // VOICE ASSISTANT

- **Text-To-Speech (TTS)**
  - We used *pico2wave*, a Linux Text-To-Speech engine
  - It supports several languages and voices
  - It is possible to configure different languages for each response

SECTION 3 VDHAC // VOICE ASSISTANT

- Command processing



Extra functionalities

- **Automation rules**
  - OpenHAB Cloud Connector Binding + IFTTT
- **Mobile application**
  - OpenHAB mobile app
- **Everywhere access**
  - Run openHAB behind a reverse proxy
  - Make the runtime accessible from the public Internet
  - myopenHAB service

Our Use Case

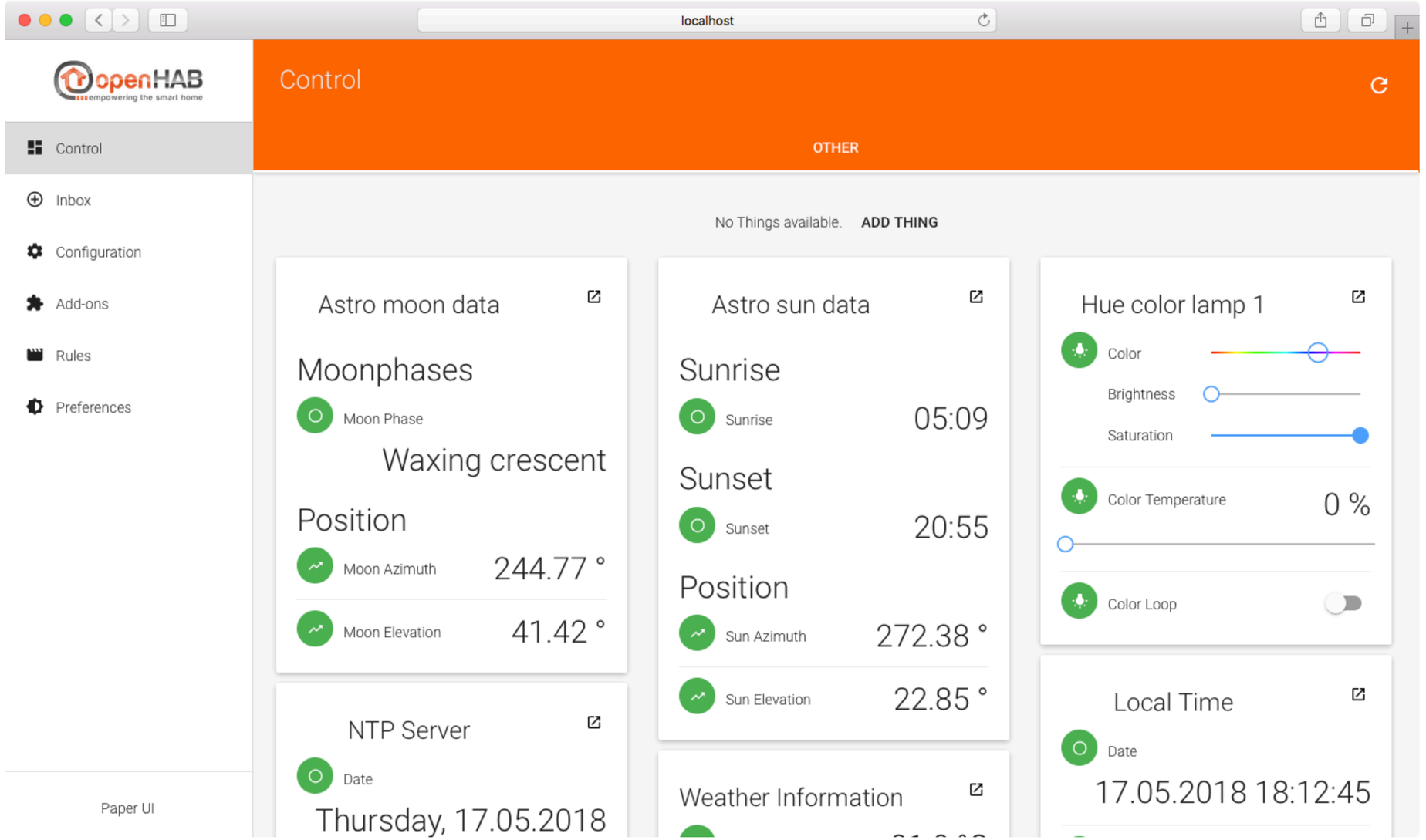
- Integrating the **Philips Hue Lightning System** to create a functional and usable home automation environment

SECTION 3 VDHAC // USE CASE

- First setup of Philips Hue:
  - We need the Philips Hue mobile app

- Adding the Philips Hue lighting system to openHAB:
  - **Automatically via PaperUI**
  - **Manually via configuration files:**
    1. Specify the Items in a openHAB *.item* file
    2. Create a Sitemap for displaying these items in Basic UI

SECTION 3 VDHAC // USE CASE



Voice Assistant Capabilities

- **For all the lights (through a Group Item):**
  - Turn on and off
- **For each light:**
  - Turn on and off
  - Change the light color (red, yellow, blue, pink, green), maintaining its brightness
  - Change the light color temperature (cold, warm, natural)
  - Increase and decrease its brightness
- **For the system:**
  - Turn off
  - Reboot

SECTION 4 CONCLUSIONS

SECTION 4 CONCLUSIONS

- Our objectives have been **mostly reached**

- **Found issues:**
  - Its setup requires a technical knowledge
  - Relies on third party solutions

- **Future work:**
  - Automate processes
  - Improve command processing

Objective	Fulfillment	Observations
1. Integrate a home controller in a embedded system	100%	We have successfully installed and integrated openHAB in the Raspberry Pi
2. Integrate a voice assistant in the same embedded system as the home automation controller	100%	The custom assistant integrates Google Assistant and a voice assistant for openHAB
3. Explore current home automation systems and voice assistants, focusing on open-source solutions	80%	We have explored some of these systems on Chapter 3, but the inclusion of open source solutions has been reduced
4. Explore how we can express automation rules to manage home devices	70%	We have successfully implemented an automation service using IFTTT and the REST API of openHAB. Some other services might have been explored
5. Explore options for managing the system from a mobile application	90%	The system is currently manageable thanks to the Cloud Connector and the mobile application of openHAB
6. Explore options for providing everywhere access to the system	100%	We have explored these options on Chapter 6 and we have tested the service myopenHAB
7. Explore safety and privacy concerns related to the home automation system	30%	Some safety concerns have been covered through this work, but we have not covered the main aspects related to privacy
8. Provide an adaptive and responsive user interface, usable on touch and non-touch screens	100%	We explained how to configure Basic UI and PaperUI. Both user interfaces comply with these points
9. Connect the virtual assistant to the domotic system, to manage directly the configured devices	100%	The custom virtual assistant communicates with openHAB thanks to the REST API of openHAB and the Requests Python library
10. Develop the home automation system applying usability techniques in order to obtain an intuitive and functional solution.	70%	The current solution is completely functional and usable. However, more devices should have been tested

RELEVANT BIBLIOGRAPHY

- Costin Bādicā, Marius Brezovan, and Amelia Bādicā. An overview of Smart Home Environments: Architectures, technologies and applications. 2013.
- Mark D. Gross. Smart House and Home Automation technologies. 1998.
- Muhammad Asadullah and Ahsan Raza. An overview of Home Automation systems. 2016.
- Bruce T. Lowerre. The HARPY speech recognition system. 1976.
- W3C - speech recognition grammar specification version 1.0 (https://www.w3.org/TR/speech-grammar/)

THANK YOU