**MASTER OF TECHNOLOGY**

**(INTELLIGENT SYSTEMS)**

**PROJECT REPORT**

**Mycroft Discovery and Application**

GROUP MEMBERS

LI XIAO XIA

YANG YANG

TAN WOEI MING

JIANG AI GUO

ANG JENN NING

MASTER OF TECHNOLOGY Institute of System Science, National University of Singapore

[**1. Project Overview** 4](#_heading=h.2et92p0)

[**2. Business Perspective** 4](#_heading=h.3dy6vkm)

[**3.** **Market Research** 5](#_heading=h.1t3h5sf)

[**4. Project Objectives** 8](#_heading=h.4d34og8)

[**5. System Specification** 9](#_heading=h.2s8eyo1)

[**5.1 Mycroft Architecture** 9](#_heading=h.17dp8vu)

[**5.2 Mycroft Data Privacy Handling** 11](#_heading=h.3rdcrjn)

[**5.3 Mycroft Home-UI Setting** 12](#_heading=h.26in1rg)

[**5.4 Mycroft Linux Installation and Application** 12](#_heading=h.lnxbz9)

[**5.5 Picroft Installation and Application** 13](#_heading=h.35nkun2)

[**5.6 Mycroft Personal Selene Server setup** 14](#_heading=h.1ksv4uv)

[**5.7 Mycroft Skill Management** 15](#_heading=h.z337ya)

[**5.7.1 Private Skill Create and Run** 15](#_heading=h.1y810tw)

[**5.7.2 Public Skill Create and Run** 15](#_heading=h.4i7ojhp)

[**5.8 Use Cases Specification** 16](#_heading=h.3as4poj)

[**5.8.1 Voice assistant for routes optimized parcel delivery backend and API Host Personal Server** 16](#_heading=h.1pxezwc)

[**5.8.2 Picroft Run Voice assistant for routes optimized parcel delivery API Call Skill** 17](#_heading=h.49x2ik5)

[**5.8.3 Load Project Skill** 18](#_heading=h.2p2csry)

[**5.8.4 DailyQuiz Skill and API call skill template** 18](#_heading=h.147n2zr)

[**5.8.5 Test ISS Projects** 19](#_heading=h.3o7alnk)

[**5.8.5.1** **ISS Projects** **Business Values** 19](#_heading=h.ihv636)

[**5.8.5.2 ISS Projects System Architecture** 20](#_heading=h.32hioqz)

[**5.8.5.3 ISS Projects API and Skill** 20](#_heading=h.ta0xjgwj68iy)

[**6. System Implementation** 21](#_heading=h.1hmsyys)

[**6.1 Mycroft Home Setup** 21](#_heading=h.41mghml)

[**6.1.1 Create Account** 21](#_heading=h.2grqrue)

[**6.1.2 Add Devices** 22](#_heading=h.vx1227)

[**6.2 Mycroft Linux Installation and Configuration** 23](#_heading=h.3fwokq0)

[**6.2.1 Mycroft Linux VM Installation and Configuration** 23](#_heading=h.1v1yuxt)

[**6.2.2 Mycroft download in Linux VM** 23](#_heading=h.4f1mdlm)

[**6.2.3 Mycroft Setup** 24](#_heading=h.2u6wntf)

[**6.2.4 Run Skill** 24](#_heading=h.19c6y18)

[**6.3 Mycroft Picroft Config** 25](#_heading=h.3tbugp1)

[**6.4 Private Skill Create and Run** 26](#_heading=h.1mrcu09)

[**6.5 Public Skill Create and Run** 26](#_heading=h.46r0co2)

[**6.6 Use Cases Implementation** 27](#_heading=h.2lwamvv)

[**6.6.1 Voice assistant for routes optimized parcel delivery - backend and API implementation** 27](#_heading=h.111kx3o)

[**6.6.1.1 Pick network (Nationwide Parcel Locker Network) address** 27](#_heading=h.3l18frh)

[**6.6.1.2 Address in Latitude and Longitude** 27](#_heading=h.206ipza)

[**6.6.1.3 Manhattan distance matrix** 28](#_heading=h.4k668n3)

[**6.6.1.4 Vehicle demand, capacity and depot** 28](#_heading=h.2zbgiuw)

[**6.6.1.5 Google OR-Tools to solve capacitated vehicle routing problem (CVRP)** 29](#_heading=h.3vac5uf)

[**6.6.1.6 Optimized vehicle routes** 29](#_heading=h.2fk6b3p)

[**6.6.1.7 Web API endpoint** 29](#_heading=h.upglbi)

[**6.6.1.8 Picroft Voice Assistant API Call** 29](#_heading=h.3ep43zb)

[**6.6.2 Load Project Skill** 29](#_heading=h.2szc72q)

[**6.6.3 DailyQuiz Skill** 29](#_heading=h.184mhaj)

[**6.6.4 Test ISS Project API and Skill** 30](#_heading=h.3s49zyc)

[**6.6.4.1 ISS Project API** 30](#_heading=h.6b0mt0ohnds)

[**6.6.4.2 mycroft-iss-project Skill** 34](#_heading=h.iyxgnjejqmyg)

[**7. Project Conclusions** 36](#_heading=h.279ka65)

[**Appendix: Project Proposal** 36](#_heading=h.meukdy)

[**Appendix: Mapped System Functionalities against MR, RS, CGS** 36](#_heading=h.36ei31r)

[**Appendix: Installation and User Guide** 38](#_heading=h.1ljsd9k)

[**Appendix: Individual reflection of project journey** 38](#_heading=h.45jfvxd)

[**Appendix: References** 39](#_heading=h.2koq656)

# 

# **1. Project Overview**

One day, your voice will control your home gadgets and support your daily life as an assistant which targets to improve quality in living. This day is coming along with cutting-edge technology Artificial Intelligence (AI). Some Top Tech Giants introduced their AI driven voice assistant products in the past years including Amazon Alexa, Google Assistant and Apple Siri to name but a few. These commercial products did bring consumers to the great platform to gain fascinating experiences but also resulted in unexpected data privacy exposure problems. To overcome such data privacy issues and open a new window for Geeks and Nerds to gain insights into voice assistants, Mycroft, the world’s first open-source voice assistant, comes to our views which is also as a name of a suite of software and hardware tools that use Natural Language Processing and Machine Learning.

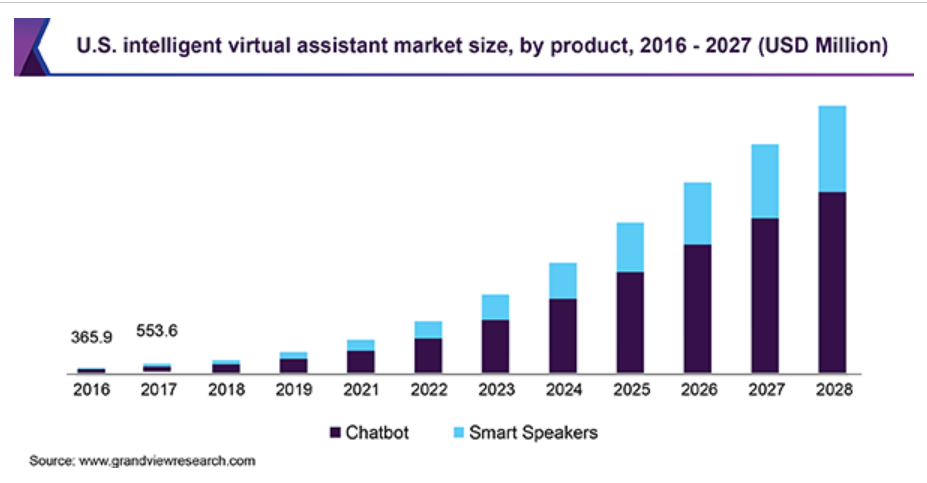
This project is to discover the data privacy handling, architecture, main components of Mycroft, interactions of components to make functionality and experiments on software and hardware tools for it. Main components are Wake Word Detection, Speech to Text (STT), Intent Parser, Text to Speech (TTS), Middleware (Mycroft core, home and API) and Skills. In addition, to set-up Mycroft core in devices like Mycroft Linux and Picroft (Raspberry PI based) followed by skill operation interfacing with Mycroft server and Personal server (setup based on Mycroft server), it brings all components together to fulfil the expected functionalities and experiments and can be treated as infrastructure for future exploration and usage.

Furthermore, our project builds customized skills as use cases which are running against devices (Mycroft Linux and Picroft) interfacing both Mycroft server and Personal server. And we also develop routes optimized parcel delivery system which is a REST API-enabled system to be called by skill to perform designed functions against Mycroft infrastructure.

The bright spot of the project is to document the standard setup along with detailed instructions which will speed up new user and developer onboarding Mycroft. And defined use cases enhanced the understanding about the Mycroft framework which can be used for future application. In addition, developed routes optimized parcel delivery system and applied skill running against Personal server is obviously extending the scope of Mycroft usage and bringing more value therefore potential attracting more focusing from users.

# **2. Business Perspective**

According to Intelligent Virtual Assistant Market Size, Share & Trends Analysis Report done by Grand View Research, the global intelligent virtual assistant market size was valued at USD 3.7 billion in 2019, growing at a CAGR (Compound Annual Growth Rate) of 34.0%, which expected to reach USD 45.1 billion by 2027. The growing use of smart speaker-based technologies for home automation and digitization in the retail sector has led to the implementation of conversational e-commerce is the major driving factor of the market. The smart speaker where voice assistants belonging to, its market share is expected to rise at a 34.7% CAGR and attain a market worth of USD 11.57 billion by the end of the global forecast in 2023.



source: <https://www.grandviewresearch.com/industry-analysis/intelligent-virtual-assistant-industry>

Chatbot and smart speakers listen, recognize, and respond to the individuals' requirements and assist consumers in various tasks. Thus, the devices are gaining popularity among the consumers for a variety of functions such as calling, shopping, reminders, setting the alarm, music streaming, and consulting. Amazon Alexa and Google Assistant accounted for a majority of the intelligent virtual assistant (IVA) market share of the smart speaker segment in 2019. Chatbot has enabled ease of accessibility in banking, retail, education, e-commerce, travel, and hospitality sectors.

Given the boosted market perspective, there are big spaces for voice assistant products to play in the market varied in domains so does Mycroft. Considering the reported data privacy issue from Top Tech Giants, we believe that consumers shall consider choosing Mycroft which designs to handle data privacy much more transparently comparing those Giants products. Moreover, Mycroft as an open-source platform is attracting more and more Nerds and tech developers to onboard therefore speeding and building up a more friendly, open, extendable, scalable and power eco system. Eventually, these will bring much more convenience and profit to the end user, consumer and market.

# **3.** **Market Research**

Voice assistant which belongs to family member of Intelligent virtual assistant (IVA) or intelligent personal assistant (IPA) is a [software agent](https://en.wikipedia.org/wiki/Software_agent) that can perform tasks or services for an individual based on commands or questions via voice. It is able to interpret human speech and respond via synthesized voices. Users can use voice for interaction such as asking their assistants questions, controlling home automation devices and media playback via voice. As of 2017, the capabilities and usage of virtual assistants are expanding rapidly, with new products entering the market and a strong emphasis on both email and [voice user interfaces](https://en.wikipedia.org/wiki/Voice_user_interface). Amazon Alexa, Google Assistant and Apple Siri have been introduced to market to become the leading products in this area till today.

Voice assistants may be integrated into many types of platforms or across several of them:

* Into devices like [smart speakers](https://en.wikipedia.org/wiki/Smart_speakers) such as [Amazon Echo](https://en.wikipedia.org/wiki/Amazon_Echo), [Google Home](https://en.wikipedia.org/wiki/Google_Home), Mycroft Mark I(II)
* Built into a mobile operating system (OS), as are [Apple](https://en.wikipedia.org/wiki/Apple_Inc.)'s [Siri](https://en.wikipedia.org/wiki/Siri) on [iOS](https://en.wikipedia.org/wiki/IOS) devices
* Into a desktop OS such as [Cortana](https://en.wikipedia.org/wiki/Cortana_(software)) on [Microsoft Windows](https://en.wikipedia.org/wiki/Microsoft_Windows) OS and Mycroft on Linux OS
* In appliances, cars, as Mycroft in Jaguar vehicles and [wearable technology](https://en.wikipedia.org/wiki/Wearable_computer)

Voice assistants can provide a wide variety of services which including:

* Provide information such as weather; facts from [Wikipedia](https://en.wikipedia.org/wiki/Wikipedia); set an alarm; make to-do lists and shopping lists
* Play music from streaming services such as [Spotify](https://en.wikipedia.org/wiki/Spotify) and [Pandora](https://en.wikipedia.org/wiki/Pandora_Radio); play radio stations; read [audiobooks](https://en.wikipedia.org/wiki/Audiobooks)
* [Conversational commerce](https://en.wikipedia.org/wiki/Conversational_commerce)
* Assist public interactions with government
* Complement and/or replace customer service by humans
* Third party services like Amazon enables Alexa "Skills" and Google "Actions"; Mycroft enables Skills

According to market feedback on voice assistants, Google Home is the smartest, Amazon Alexa is most impactful and Apple Siri is seating on leading but cannot be comparable with Google and Amazon. Amazon, named as the biggest impact on the voice industry, is taking advantage of its selling goods to extend Alexa to lead in voice-based AI assistants. Given its long history in search and perhaps the most extensive knowledge graph on the planet, Google is [generally rated](https://www.forbes.com/sites/johnkoetsier/2018/04/24/ai-assistants-ranked-googles-smartest-alexas-catching-up-cortana-surprises-siri-falls-behind/) as the [smartest](https://www.zdnet.com/article/apple-siri-vs-amazon-alexa-vs-google-assistant-tests-reveal-which-is-smartest/) voice-based AI assistant.

Even though Google, Amazon and Apple are leading the voice assistant market, they are also raising concerns on data privacy issues like user’s voice data mid-handling and status devices from devices. And Google and Amazon profit from the data they gather with their devices and services therefore they will not give up these advantages easily as public expectation. Below are reported cases for data privacy issues from tech giants.

* [An Amazon Echo recorded a family’s conversation, then sent it to a random person in their contacts, report says](https://www.washingtonpost.com/news/the-switch/wp/2018/05/24/an-amazon-echo-recorded-a-familys-conversation-then-sent-it-to-a-random-person-in-their-contacts-report-says/)
* [Human workers are listening to recordings from Google Assistant](https://www.theverge.com/2019/7/11/20690020/google-assistant-home-human-contractors-listening-recordings-vrt-nws)
* [Apple 'sorry' that workers listened to Siri voice recordings](https://www.bbc.com/news/technology-49502292)

Mycroft is not designing as intrusive software and platform while providing similar functionality and service like Google and Amazon’s voice assistants. It allows users to decide on if they want to participate on an open-dataset which is used for training Mycroft to be more accurate on user’s speech.

As the first open-source voice assistant, Mycroft provides an extendable platform for developers to build their skills. And developers are able to set-up Personal server based on Mycroft server to fulfil their own needs. Even Speech To Text (STT) is still difficult to build locally, Mycroft’s open project openSTT has shared the lights on the path.

In our opinion, Mycroft will bring more attention to the market and embrace more services(skills) developed by developers from the Mycroft own developers and its community.

Below is comparison of notable voice assistants

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Intelligent personal assistant** | **Developer** | [**Free software**](https://en.wikipedia.org/wiki/Free_software) | [**Free and open-source hardware**](https://en.wikipedia.org/wiki/Free_and_open-source_hardware) | [Chromecast](https://en.wikipedia.org/wiki/Chromecast)  **integration** | **Smart phone app** | **Always on** | **Skill language** |
| [Alexa](https://en.wikipedia.org/wiki/Amazon_Alexa) | [Amazon.com](https://en.wikipedia.org/wiki/Amazon.com) | No | No | No | Yes | Yes | JavaScript |
| [Cortana](https://en.wikipedia.org/wiki/Cortana_(software)) | [Microsoft](https://en.wikipedia.org/wiki/Microsoft) | No | N/A | No | Yes | Yes |  |
| [Google Assistant](https://en.wikipedia.org/wiki/Google_Assistant) | [Google](https://en.wikipedia.org/wiki/Google) | No | N/A | Yes | Yes | Yes | C++ |
| [Mycroft](https://en.wikipedia.org/wiki/Mycroft_(software)) | Mycroft AI | Yes | Yes | Yes | Yes | Yes | Python |
| [Siri](https://en.wikipedia.org/wiki/Siri) | [Apple Inc.](https://en.wikipedia.org/wiki/Apple_Inc.) | No | No | No | Yes | Yes |  |

As highlighted in the project overview, our team has also built a minimum viable product (MVP) ‘Voice assistant for routes optimized parcel delivery’ which applied skill can run on Picroft device (based on Raspberry Pi). The delivery man tells his or her current address in postal code, Picroft will reply with the next address to go. Idea of this product is inspired by the changing landscape of the e-commerce delivery service with the deployment of 1,000 parcel locker stations across Singapore under the Nationwide Parcel Locker Network to be completed by end 2021 (Figure 1: Benefits of Pick Network: Nationwide Parcel Locker Network). We build this MVP for parcel delivery companies to quickly realize these business benefits by utilizing the Nationwide Parcel Locker Network (Pick Network) as soon as it is to be deployed.

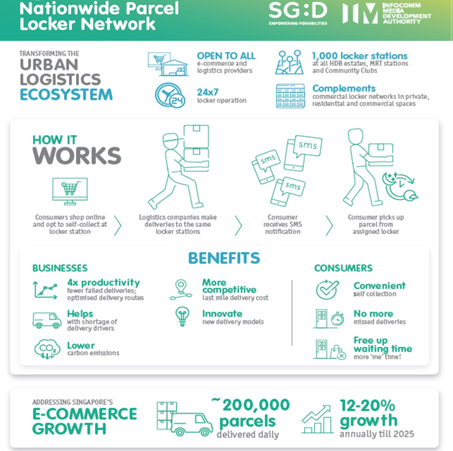


Figure 1: Benefits of Pick Network: Nationwide Parcel Locker Network

# **4. Project Objectives**

Our project aims to discover the data privacy handling, architecture, main components of Mycroft, interactions of components to make functionality and experiments on software and hardware tools for it.

And we built few skills to explore and test key knowledge gained from our discovery for Mycroft.

Below are main goals to achieve:

* Mycroft architecture
* Mycroft data privacy handing
* Mycroft install from Mycroft Linux connecting Mycroft server backend services
* Mycroft install from Picroft connecting Mycroft server backend services
* Private skill creates and runs against Mycroft Linux connecting Mycroft server backend services
* Public skill creates and runs against Mycroft Linux connecting Mycroft server backend services
* Mycroft Personal server setup
* Private and Public skills run from Mycroft Linux and Picroft connecting Personal server backend services

We define that Public skill is the skill which completes the Skill Acceptance Process and available publicly from Mycroft marketplace.

We define that Private skill is the skill which saves and runs in user’s device only but not deriving from skills in Mycroft marketplace.

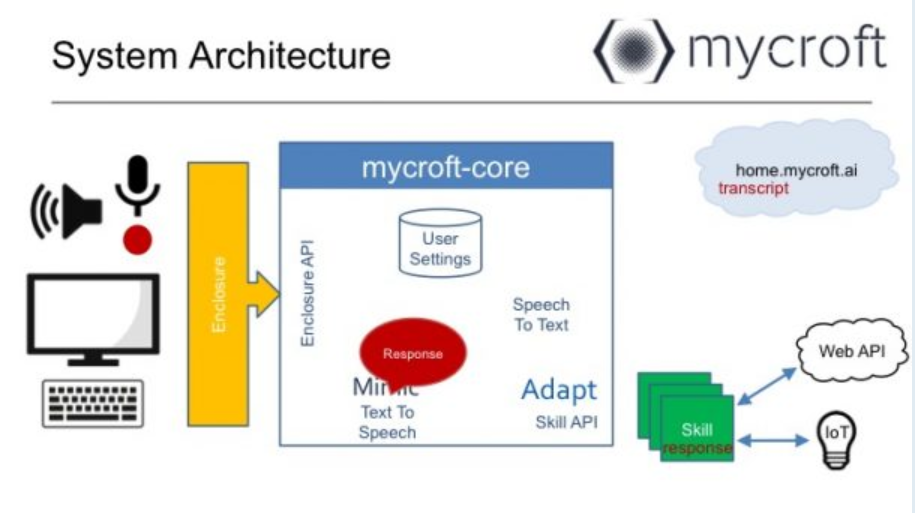
In addition to set-up the Mycroft infrastructure and create skills to run it, we also build the REST API-enabled backend service to run the ‘Voice assistant for routes optimized parcel delivery’ MVP, with below objectives

* Obtain actual addresses of the Nationwide Parcel Locker Network
* Get Latitude and Longitude of these addresses through OneMap API
* Python code that utilizes the Google OR-Tools for vehicle routing optimization on these addresses
* Web API endpoint for the Mycroft devices like Picroft to post request and get data

# **5. System Specification**

## **5.1 Mycroft Architecture**

Mycroft as the name of a suite of software and hardware tools which uses [natural language processing](https://en.wikipedia.org/wiki/Natural_language_processing) and [machine learning](https://en.wikipedia.org/wiki/Machine_learning) to provide an open source voice assistant.



Source: <http://imgur.com/AIYaD3r.png>

Mycroft is a modular or components-based platform. Its main components are as below:

* Wake Word Detection
* Speech to Text (STT)
* Intent Parser
* Text to Speech (TTS)
* Middleware (Mycroft core, home and API)
* Skills
* Devices and Enclosures

**Wake Word detection**

A Wake Word is a phrase you use to tell Mycroft you're about to issue a command. By default, this is “Hey Mycroft”, but you can configure your own Wake Word in your [Mycroft Home](https://home.mycroft.ai/) account.

There are two technologies that Mycroft.AI currently uses for Wake Word detection:

* [PocketSphinx](https://github.com/cmusphinx/pocketsphinx):

PocketSphinx is part of the broader [CMUSphinx package](https://cmusphinx.github.io/), developed by [Carnegie Mellon University](https://www.cmu.edu/). PocketSphinx is a lightweight speech recognition engine, specifically tuned for handheld and mobile devices.

Because PocketSphinx is trained on English speech, your Wake Word currently needs to be an English word. Wake Words in other languages, like Spanish, French or German, won't work as well.

* [Precise](https://mycroft.ai/documentation/precise):

Unlike PocketSphinx, which is based on Speech to Text technology, Precise is a neural network that is trained on audio data. It doesn't matter what *words* you want to use for your Wake Word. Instead, you train it on *sounds*. The downside is that Precise needs to be trained on your chosen Wake Word. Precise is the default Wake Word Listener for the "Hey Mycroft" wake word, PocketSphinx provides a fallback to this if Precise is unavailable.

**Speech to Text (STT)**

Speech to Text (STT) software is used to take spoken words, and turn them into text phrases that can then be acted on.

Mycroft is working with Mozilla to build [DeepSpeech](https://github.com/mozilla/DeepSpeech). A fully open source STT engine, based on Baidu’s Deep Speech architecture and implemented with Google’s [TensorFlow](https://www.tensorflow.org/) framework.

DeepSpeech is not yet ready for production use and Mycroft currently uses [Google STT](https://cloud.google.com/speech/) as the default STT engine.

Mycroft also supports other STT engines that can be configured using the [Configuration Manager](about:blank):

* [IBM Watson Speech to Text](https://www.ibm.com/watson/services/speech-to-text/) (IBM API key required)
* [wit.ai Speech to Text](https://wit.ai/blog/2014/02/12/speech-api) (wit.ai API key required)

**Intent parser**

An intent parser is software which identifies what the user's *intent* is based on their speech. An intent parser usually takes the output of a Speech to Text (STT) engine as an input.

An intent parser can then match the *intent* with a suitable Skill to handle the *intent*.

* [Adapt intent parser](https://github.com/MycroftAI/adapt):

Adapt is the default intent parser for all Mycroft platforms. Adapt was developed by Mycroft and is available under an open-source license.

* [Padatious](https://github.com/MycroftAI/padatious):

Padatious is a neural network based intent parser. Padatious is currently under active development by Mycroft and is available under an open-source license. It is likely that some Mycroft platforms will switch to using Padatious in the future instead of Adapt.

**Text to Speech**

Text to Speech (TTS) software takes written text, such as text files on a computer, and uses a *voice* to *speak* the text. Text to Speech can have different voices, depending on the TTS engine used.

* [Mimic](https://github.com/MycroftAI/mimic): Mycroft's default local text to speech (TTS) engine, based on CMU's Flite (Festival Lite)
* [Mimic2](https://github.com/MycroftAI/mimic2): Mycroft's own cloud-based text to speech (TTS) engine, based on Tacotron providing a much better voice quality. In your home.mycroft.ai account, you can select voices from these as well as
* [Google TTS](https://play.google.com/store/apps/details?id=com.google.android.tts): you need to choose which voice to use even more TTS engines are available but require manual configuration.

**Middleware**

The Mycroft middleware has two components:

* [Mycroft Core](https://github.com/MycroftAI/mycroft-core): this code, written in Python, is the core software that provides the 'glue' between other modules. Mycroft Core is available under an Apache 2.0 open-source license.
* [Mycroft Home and Mycroft API](https://home.mycroft.ai/): this is the platform where data on Users and Devices is held. This platform provides abstraction services, such as storing API keys that are used to access third-party services to provide Skill functionality. The code for this platform is available under an AGPL 3.0 open-source license.

**Mycroft Skills**

[Mycroft Skills](https://github.com/MycroftAI/mycroft-skills) are like 'add-ons' or 'plugins' that provide additional functionality. Skills can be developed by Mycroft Developers, or by Community Developers, and vary in their functionality and maturity.

[Mycroft Skills Kit (MSK)](https://github.com/mycroftai/mycroft-skills-kit) is a Python-based utility that has been created to make it easier for Skill Authors to create, test and submit Skills to the [Skills Marketplace](https://market.mycroft.ai/).

[Mycroft Skills Manager (MSM)](https://github.com/mycroftai/mycroft-skills-manager) is a command line tool used to add, manage and remove Skills on any Mycroft installation.

**Devices and Enclosures**

Mycroft is designed to run on many different platforms. Each dedicated platform is called a device, these include:

* **Mark 1** - our first reference hardware device using a dedicated software image.
* **Mark 2** - our latest reference hardware device using a dedicated software image.
* **Picroft** - any Raspberry Pi 3 or 4 that is running the Picroft software image.

The enclosure refers to the specific code that is required for that device. It might define unique functionality such as the eyes on the Mark 1, or a specific way of interacting with the hardware, such as controlling the volume levels at a hardware level via i2c.

## **5.2 Mycroft Data Privacy Handling**

It’s known that adequate data is very important for Machine Learning (ML) therefore allowing ML’s algorithms to be more accurate on predicting the outcomes so does for Mycroft. But Mycroft never force users to share their data to approach the accuracy of its product by instead leaving it to the users themselves to decide.

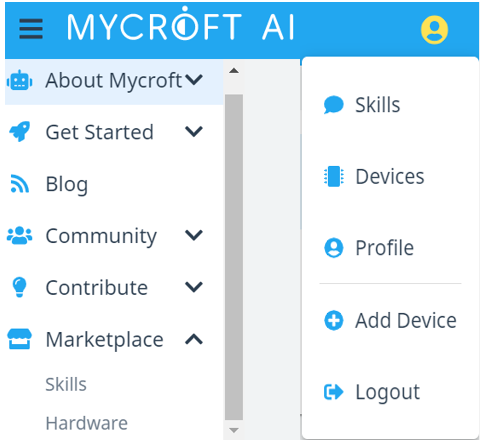
Users can choose to select Opted-out or Opt-in against Mycroft Open dataset which is used for Mycroft Machine Learning. This function is under the Profile menu in Home Mycroft AI.



## **5.3 Mycroft Home-UI Setting**

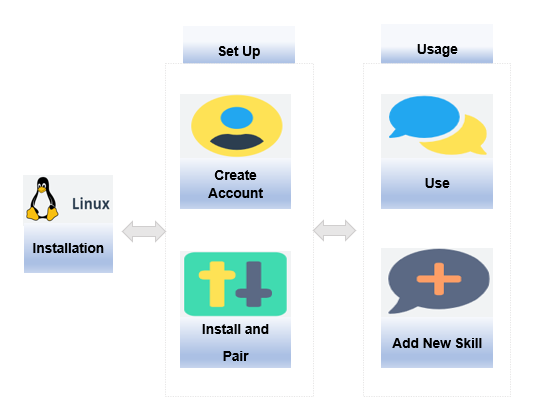
Mycroft Home is the UI where you can search all existing Skills in Mycroft Marketplace for your coming usage from Mycroft without sign up required and devices you can purpose. With account creation followed by sign in on the home page, you can maintain below menus to support the provided functions.

* Skills loaded in your devices
* Add/Edit/Remove device
* Maintain your profile



## **5.4 Mycroft Linux Installation and Application**

Mycroft is an open-source software that runs where you want it to run, including on your own hardware such as Linux. If you are a technical user, run Mycroft on your favorite Linux distro, including Ubuntu/Debian, Arch, and Fedora. Clone the repository and run the included setup script. Mycroft also provides the Alpha release of the Mycroft Snap package for installation. But this Alpha release contains some known usability bugs which may affect experiments for the project. To consider this situation, we choose to install Linux distro but not explore further on Alpha release.



As our team does not have a Linux OS at hand, we then decide to install Linux VM in Oracle VirtualBox to finish the installation.

After finishing the Linux VM installation, Setup is required to be done before the using which includes below 2 functions.

* Create Account
* Mycroft-core installation and Device Pair

Usage part contains 2 operations on:

* Talk to/Use Mycroft
* Add New Skills

## **5.5 Picroft Installation and Application**

Picroft is the Raspberry PI installed with Mycroft voice assistant. It could communicate with Selene Server to pair with the Picroft, register account, install and uninstall the skills. Picroft is a light lifting client that interacts with users and links to the server through API, where the heavy lifting function, algorithm or model runs.

There are many applications of Picroft clients. It is a home voice assistant for remote smart home to activate home appliances through voice, for example lighting, fan and air-conditioning. It is also a mobile client to assist users when driving, for example turn on/off radio, switching channel, search for map location and best route.

Picroft is installed with Mycroft core skill and load project skill. The core skill is basic voice activation skill developed by Mycroft. It translates the Speech to Text (STT) through Google STT. Load project skills allow you to install additional skills from GitHub. This is a user customized skill like smart home control, voice assisted route in the car.



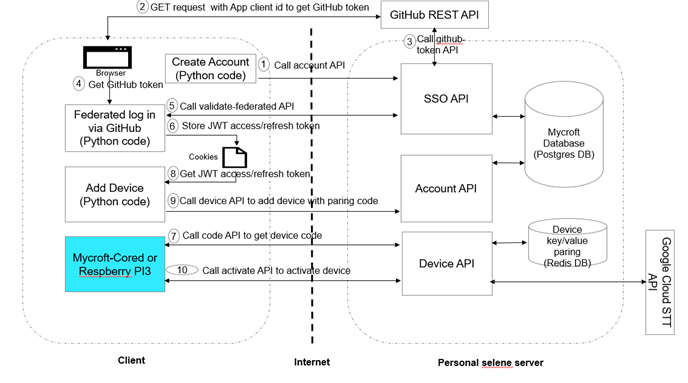
## **5.6 Mycroft Personal Selene Server setup**

Selene server provides the services used by Mycroft Linux or Picroft (based on Raspberry PI) to manage devices, skills and settings. It also can be used to build the customized services (such as hosting APIs) which could be used by the customized skill. We manage to setup the personal Selene server and make Mycroft Linux or Picroft to connect this personal server to perform normal/customized functions. The general steps to setup the personal Selene server are as below. For detail, please refer to “**3.3 Mycroft Personal Server Installation and Setup**” section under **User Guide**.

* Download source code of selene-backend from GitHub
* Install Postgres DB and setup it
* Install Redis DB and setup it
* Build database structure and insert the required data
* Setup GitHub REST API to do basic of authentication
* Setup Single Sign On API, Account API and Device API
* Setup Google Cloud Speech-To-Text API.
* Setup the firewall rules on Ubuntu

After completing the above steps and changing IP address of remote server in client device to point to this personal server, the client devices can communicate with the personal server to perform pairing code, activating device, Speech to Text, Text to Speech and triggering the skill activities.

The overall process as below:



## **5.7 Mycroft Skill Management**

### **5.7.1 Private Skill Create and Run**

Private skill means that created skill saves and runs in the user's device only but not deriving from Mycroft marketplace.

By directly saving the source code directory of skill under the skill folder in Mycroft core and waiting for skill configuration ready, you can run this private skill from CLI window.

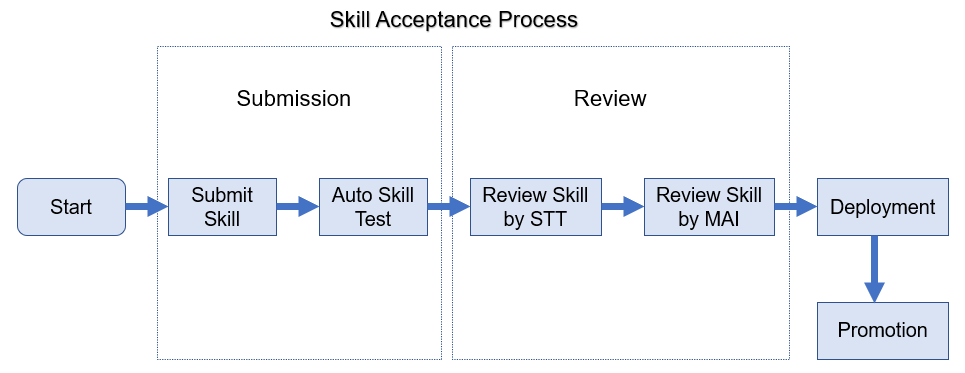
To modify the skill, you can edit the skill files directly in the skill folder to reflect the expected change.

To delete the skill, you can just remove the whole folder of this skill out of the skill folder.

### **5.7.2 Public Skill Create and Run**

Public skill is the one which completes the Mycroft Skill Acceptance Process and available publicly from Skills of Mycroft marketplace for user’s downloading and use.

Skills Acceptance Process outlines how Skills are tested, reviewed, deployed and promoted through the Mycroft Marketplace which consists below 6 stages.



* [Submitting your Skill](about:blank)
* [Automatic testing](about:blank)
* [Review by Skill Testing Team](about:blank)
* Code Review
* Information Review
* Functional Review
* [Review by Mycroft AI](about:blank)
* [Deployment](about:blank)
* [Promotion](about:blank)

Mycroft provides Mycroft Skills Kit(msk) which is a Python-based utility that has been created to make it easier for **Skill** Authors to create, test and submit **Skills** to the [Skills Marketplace](https://market.mycroft.ai/).

msk currently supports the following features:

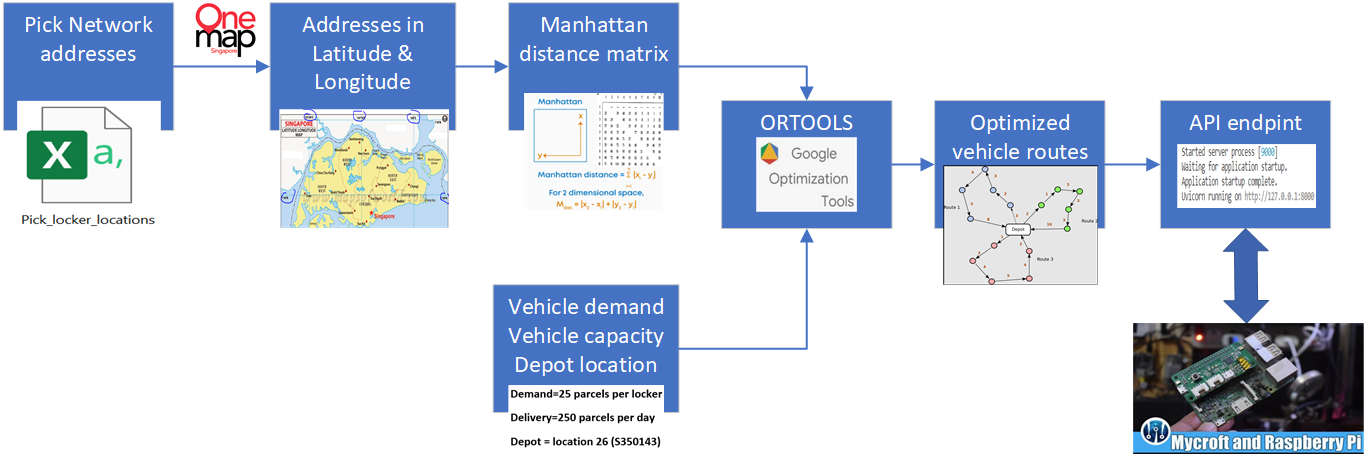
* Create a new Skill
* Create an Intent test for a Skill
* Upload a Skill
* Upgrade an existing Skill

We use msk to create new skills followed by intent test build-up to prepare for skill submission.

## **5.8 Use Cases Specification**

### **5.8.1 Voice assistant for routes optimized parcel delivery backend and API Host Personal Server**

**System architecture**



The backend of the routes optimized parcel delivery system is hosted in a Mycroft Personal Server that is built by our team. It uses the Google OR-Tools to optimize the vehicle delivery routes for a given list of parcel locker addresses which are obtained from the Pick Network (Nationwide Parcel Locker Network) website. An API endpoint is built to enable API calls from various Mycroft devices including the Picroft (based on Raspberry Pi) in our use case. It takes the input of the current delivery address and returns the next address to go and also the number of remaining delivery locations.

### **5.8.2 Picroft Run Voice assistant for routes optimized parcel delivery API Call Skill**

Picroft interacts with delivery API call which works with backend personal server, and return the delivery address through voice control. The API call skill can be downloaded from GitHub, then activate the delivery process, check the current and next delivery location.

Below are the examples of voice control:

1. Let’s begin deliver
2. Please let me know the current deliver
3. What is the next deliver?

Starting with the “begin deliver”, the API call returns the first delivery address, checking the “current deliver” address and the “next deliver” address.



### **5.8.3 Load Project Skill**

Picroft with load project skill, is a specially designed skill to load projects from GitHub. Users create skills as projects and store them in GitHub repository. By updating the GitHub path in the projectlist.txt, you could use voice command to show, install and uninstall the projects.

Below are the examples of voice control:

1. Show project
2. Select project <One|Two|Three>
3. Load project
4. Remove project

Through the voice control, users could avoid the hassle to debug and installation, but one-voice to kick start the process. Each project may integrate the API call to return project output.



### **5.8.4 DailyQuiz Skill and API call skill template**

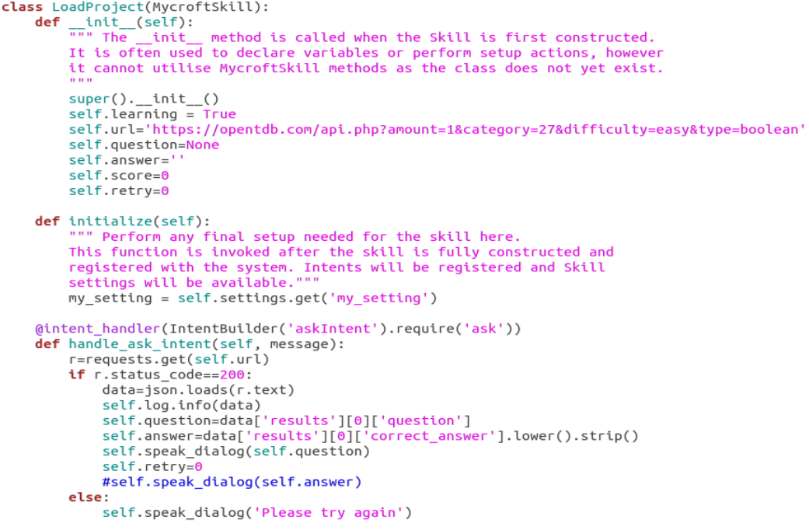
DailyQuiz skill can be downloaded from GitHub. This is an interactive skill implemented as API call to get a quiz question from opentdb database. Picroft will ask a quiz and then the user replies with an answer (true/false). Score will be recorded and accumulated by Picroft.

Below are the examples of voice control:

1. Test me
2. True / False
3. Score
4. Reset



API call skill template can be specified for each project to be implemented as below.



### **5.8.5 Test ISS Projects**

We utilize the personal selene server which is built from **5.6** **Mycroft Personal Selene Server setup** to create the customized service to host ISS group projects. On the client side, we build the customized skill which is enabling ISS lecturer to test these ISS group projects via voice assistance.

#### **5.8.5.1** **ISS Projects** **Business Values**

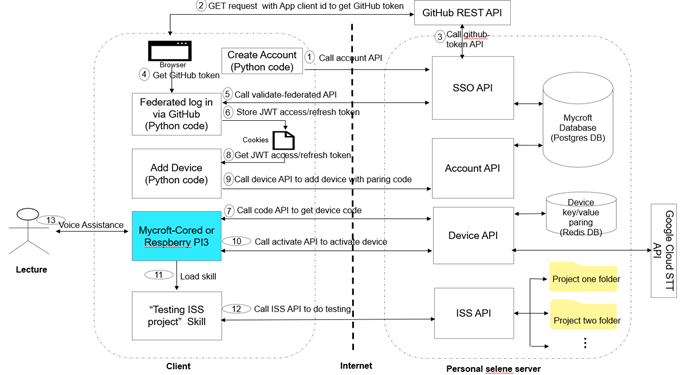
Personal selene server is easy to build the customized APIs using python flask based on the selene-backend framework. It’s able to host multiple APIs for one project for the different functions. It’s also able to add new APIs for a new project easily and fast. The environment for group projects only requires to be setup one-time on the personal selene server, then the APIs of projects can be accessed/utilized by the multiple clients via the internet. It can reduce the efforts for the environment setup of projects. On top of these, we build the corresponding Skill used to test APIs in the client by lecturer’s voice input. The structure of Skill is built scalable and flexibly. It’s easy to build the intended tasks in our Skill to test hosted APIs based on the interface specification and test cases students provided.

Based on this design, multiple ISS lectures are able to test the group projects concurrently via the different clients. It also can help ISS lectures to reduce workload and improve work efficiency by voice assistance during testing and verifying the group projects.

ISS group projects that integrate with personal selene server and our customized Skill are required to meet the following two criteria:

* The interface of project (first entry function) is required to build using python code
* Input/Output for the interface are required to be text or JSON format

#### **5.8.5.2 ISS Projects System Architecture**

****

#### **5.8.5.3 ISS Projects API and Skill**

For this use case, the APIs of all ISS group projects could be configured in the personal selene server and tested by ISS lectures.

A skill **mycroft-iss-project** is built to trigger the specific ISS project.

Mainly, following three intent handlers in this skill do perform designed functions based on the lecturer's specific voice input.

* Show ISS Projects Info
* Test Iris Demo
* Test Delivery Project

**Process Flow in Skill**

Diagram, schematic

Description automatically generated

# **6. System Implementation**

## **6.1 Mycroft Home Setup**

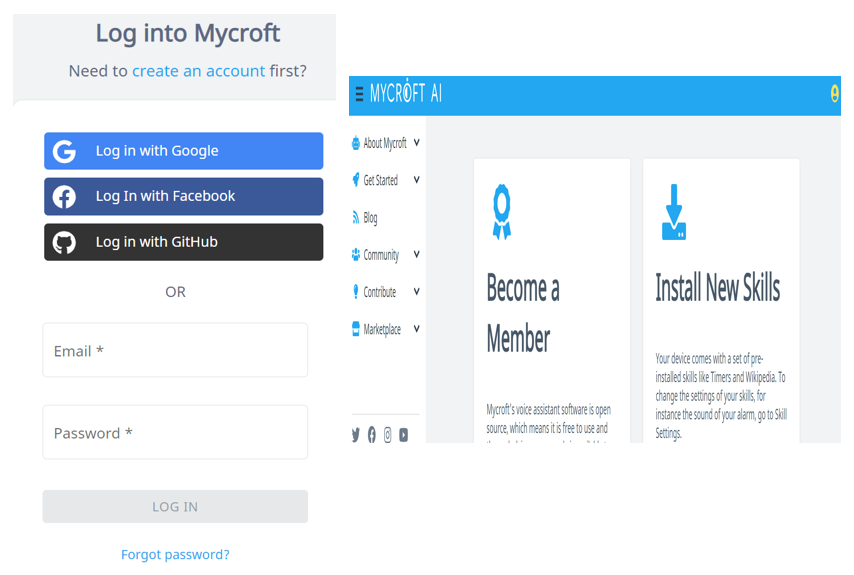
### **6.1.1 Create Account**

Click the **Create Account** button in the Download segment of [Get Started Page](https://mycroft.ai/get-started/#download) to create your own Mycroft account. Or you may login with any of your Google Account, Facebook Account and GitHub Account.

To login with Mycroft account, you need to input your registered email and password following by clicking the LOGIN button.

You may login with Google/Facebook/GitHub Account, and you can select one type of account to click the specific button and then choose an account to continue on Mycroft AI.

After login, you are directed to your Mycroft Home dashboard thereby able to maintain your profile, skills and devices.

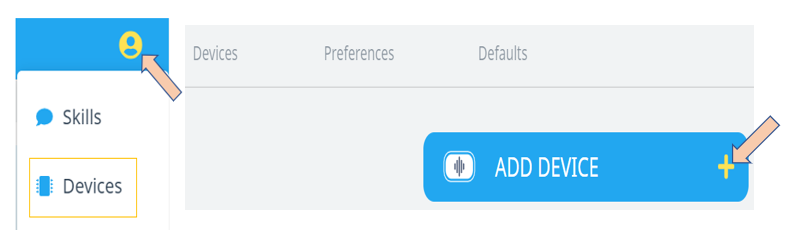


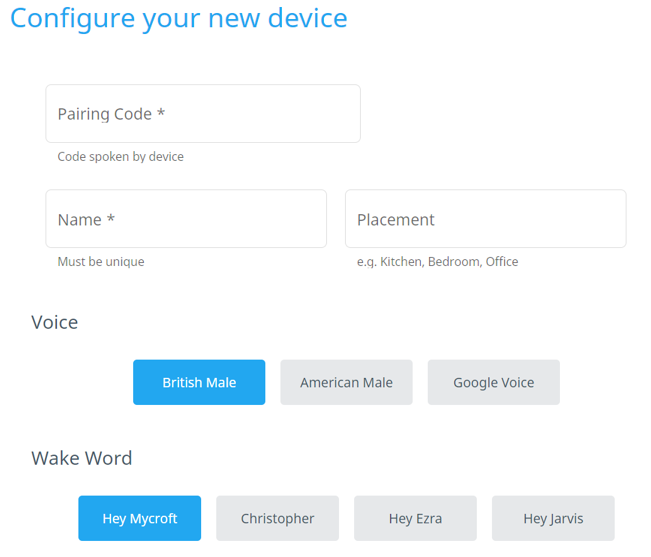
### **6.1.2 Add Devices**

By clicking the Devices menu in upper right of dashboard followed by clicking ADD DEVICE, you can add your device with all mandatory fields (with \*) to configure your new device. Within the fields, the Pairing code is the 6-character Registration Code which you can get from 6.2.3.1 Pair the Device. You can choose your favorite Voice (speaking accent, by default, British Male) and Wake Word (by default, Hey Mycroft).

For someone, “Christopher” as a wake word may be easier to be recognized by Mycroft compared with “Hey Mycroft”. You can choose the most suitable for your voice.

After adding the device and all skills configuration updates done, you can interact with Mycroft via voice command or command line which is running skill in device.





## **6.2 Mycroft Linux Installation and Configuration**

### **6.2.1 Mycroft Linux VM Installation and Configuration**

**VM VirtualBox:** Oracle\_VM\_VirtualBox\_6

**Linux Image:** Ubuntu 20.04.1

**Project using VM Machine**: ai-vm-u20.04-v03 following the instruction from [telescopeuser/iss-vm](https://github.com/telescopeuser/iss-vm)

When Linux VM installing followed by login, you can go ahead to download Mycroft Linux into the above VM machine.

### **6.2.2 Mycroft download in Linux VM**

Click the **Download for Linux** button in the Download segment of [Get Started Page](https://mycroft.ai/get-started/#download) to get instruction on how to download Mycroft Linux and run it in Project VM Machine.

The Instruction link: [MycroftAI/mycroft-core](https://github.com/MycroftAI/mycroft-core)

Follow below steps to get code on your device:

* cd ~/
* git clone <https://github.com/MycroftAI/mycroft-core.git>
* cd mycroft-core
* bash dev\_setup.sh

The script dev\_setup.sh sets up dependencies and a [virtualenv](https://virtualenv.pypa.io/en/stable/) to make sure Mycroft Linux running with needed ENV configuration.

### **6.2.3 Mycroft Setup**

Before continuing on below Mycroft Setup, you need to create an account at first which follows 6.1.1 Create Account.

**Pair the Device**

After runningscript dev\_setup.sh in terminal, you can run start-mycroft.sh debug to start your Mycroft device.



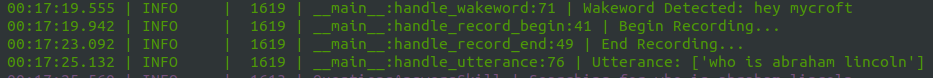
For the first starting your device, there are 6-character Registration Code displaying in CLI which you need to use to finish the paring against your Mycroft Home page.

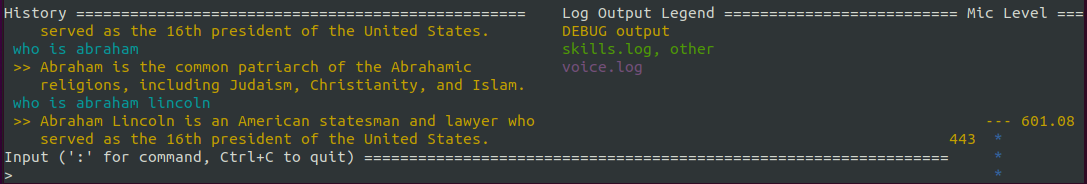
Following up 6.1.2 to add the device, your device should pair with your Mycroft. When the wording “Remote configuration updated” appears in the Log Output window, you can start to talk with Mycroft.



### **6.2.4 Run Skill**

When you speak a wake word like “Hey Mycroft”, Mycroft recognizes it and shows the message “Begin Recording…” in the Log Output window. You voice the intent(utterance) based on a specific skill which you are running like “who is Abraham Lincoln”. In general, you are given 3-8 seconds to finish the voice recording until the message “End Recording…” appears. Mycroft parses the intent, then calls STT service to recognize and translate your speech to text and then display in the Output window, finally you hear a voice speaking out the response of the intent.





Functions vary based on skill design and implementation therefore device behavior different functions serve by this skill.

## **6.3 Mycroft Picroft Config**

Mycroft Discovery and Application project uses Raspberry PI running Mycroft voice assistance to perform project loading. Raspberry PI loaded with Mycroft-LoadProject skill allows users to load, remove projects through voice command. The voice command is translated from voice to text in Raspberry and interacts with Mycroft AI service (Personal Selene Server) to download the project (skills) from GitHub.

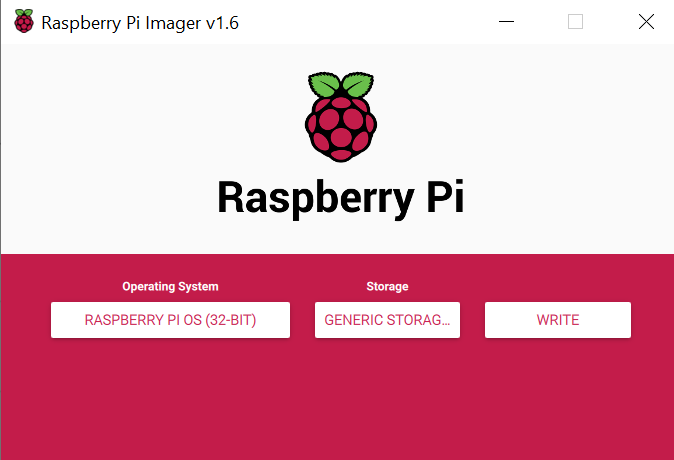
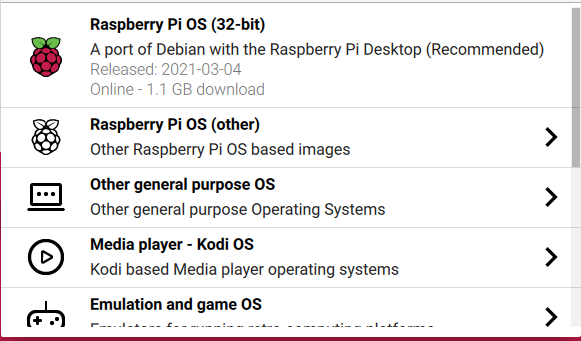
In order to use the voice assisted skill, Mycroft-LoadProject skill needs to be installed. Below are the steps:

1. Raspberry OS setup
2. Mycroft Core install from Mycroft AI GitHub
3. Mycroft-LoadProject skill installation

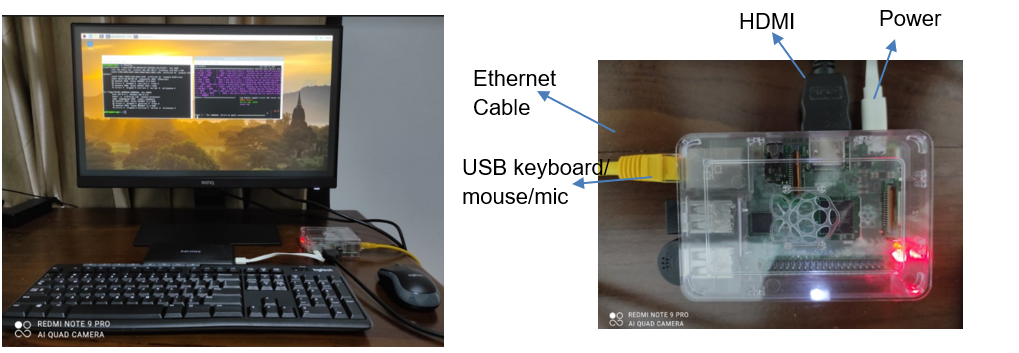
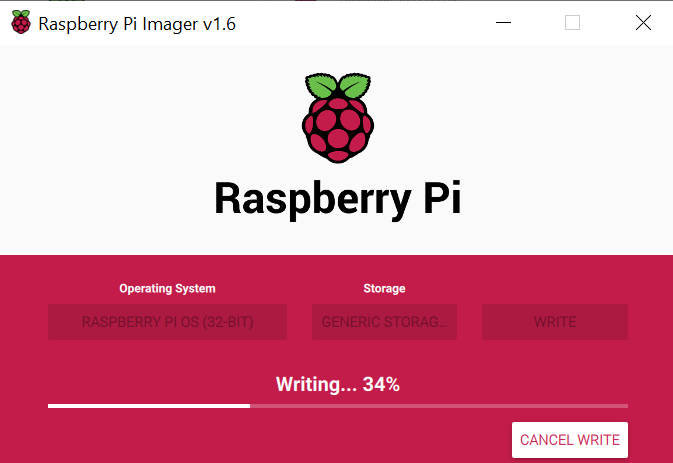
**Raspberry OS setup**

The steps to install Raspberry PI OS are preparation of hardware, download the Raspberry PI Image and install the OS.

Before the installation, a Raspberry PI 3 Model B (or above) and 16GB micro-SD card are needed. The Raspberry needs to have a HDMI display, USB port to support mouse and keyboard and Ethernet port. Then, download the image from the Internet and set it up on the desktop with micro-SD slot support. After the image setup, slot in the micro-SD card, select and run the “Raspberry PI OS” (recommend option).

Once the selection is done, click “Write”. The image will download the OS version from the internet and write to the SD card.



After the Raspberry PI OS is set up, slot the SD card into the Raspberry PI and connect the accessories (mouse, keyboard, HDMI to monitor, ethernet cable). Turn on the Raspberry PI, the OS will continue the setup, this will take a couple of minutes.

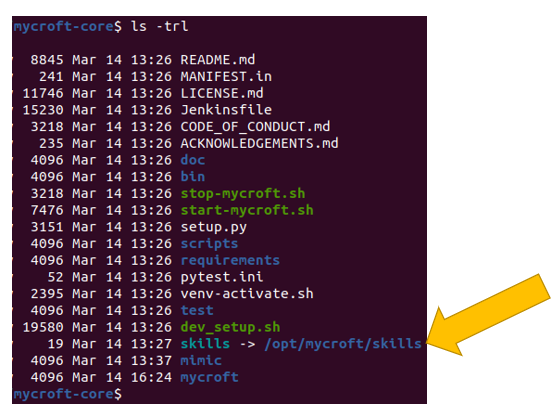
**Mycroft Core Setup**

Login to Raspberry with the pre-install OS, the default user is “pi” and password is “raspberry”. The OS should already pre-install Python and Git, proceed to clone the Mycroft-core repository following the steps in 6.2.2 to get code on Raspberry.

When Mycroft core setup is done, you can run a loaded skill following the instructions from 6.2.3.

## **6.4 Private Skill Create and Run**

To create private skill, you can get or clone (via git) the skill directly in the skill folder under ~/mycroft-core. Waiting for the message “Remote configuration updated” appearing, you may start to run this skill following the method described in 6.1.4.



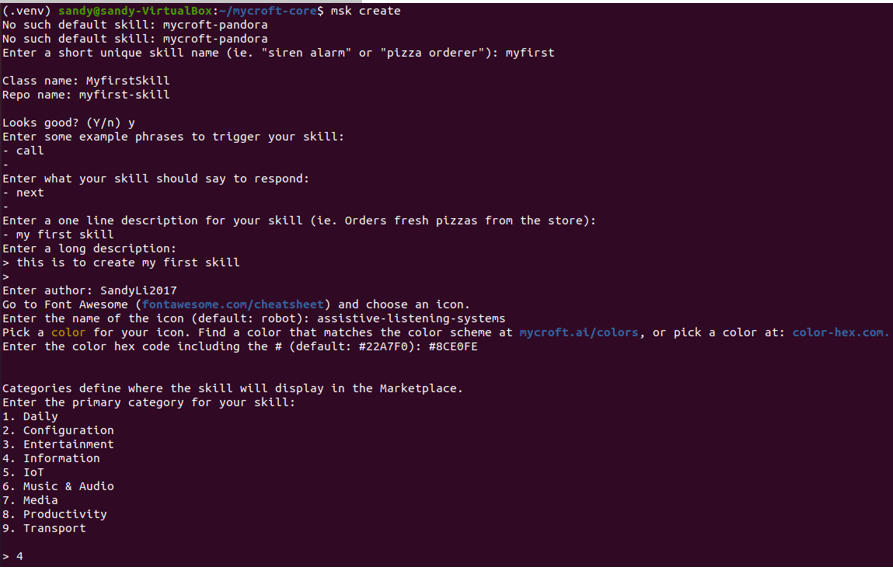
Our project can also get private skill via running Load Project skill with configuration to download to device from GitHub and run it.

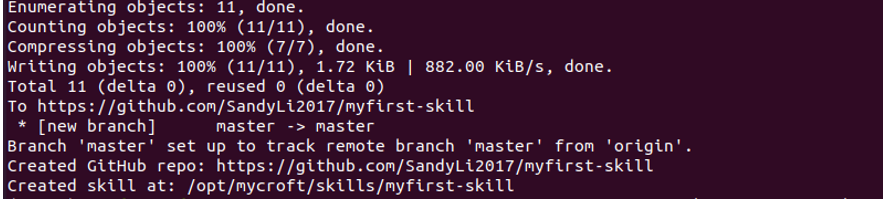
## **6.5 Public Skill Create and Run**

To design and develop your new skill targeting for submitting to Mycroft to finish Skill Acceptance Process, you can use Mycroft Skills Kit(msk) to create a skill and its intent test.

* cd ~/mycroft-core # or the path to your mycroft-core installation
* source venv-activate.sh; pip install msk
* msk create

After filling up all mandatory fields, the msk create command is finished and all skill folders are saved in the skill folder of mycroft-core and your GitHub as a repository. You may go ahead for functions buildup of your skill followed by a test.





Given all stages (except promotion) of process finishing, you can download this skill from the marketplace of Mycroft home page followed by running its designed functions using method described in 6.1.4.

## **6.6 Use Cases Implementation**

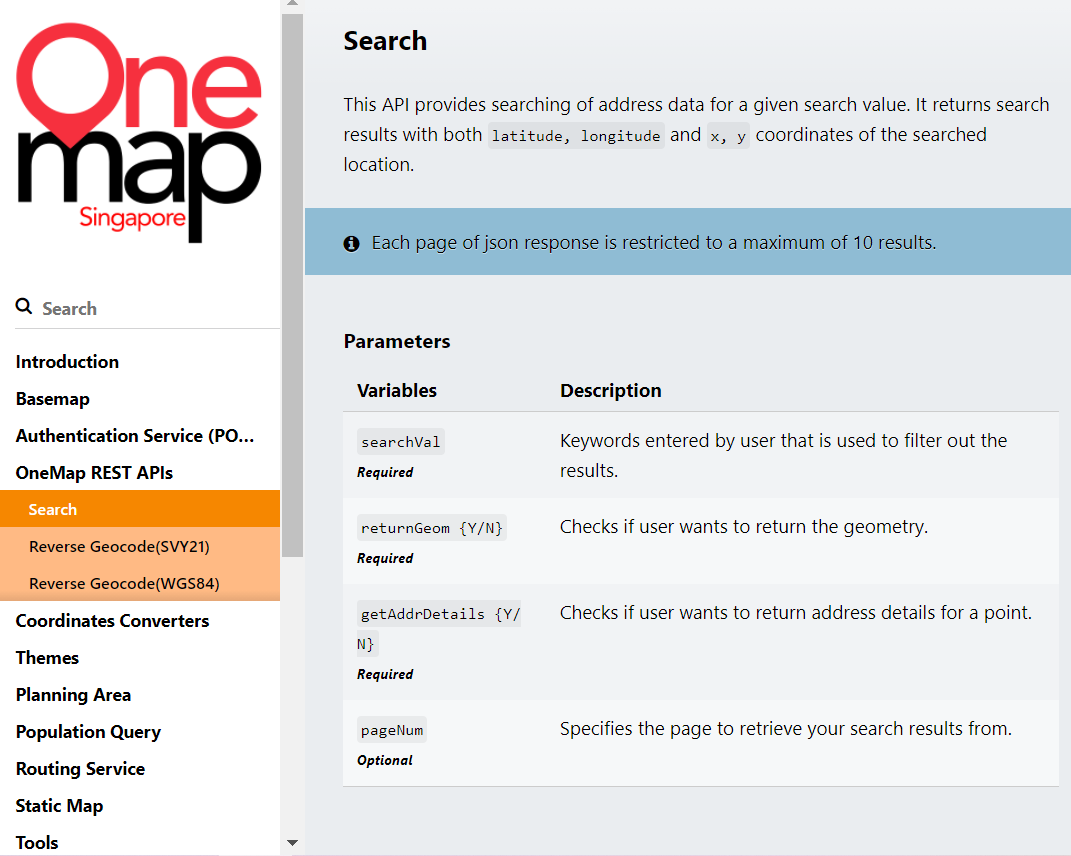
### **6.6.1 Voice assistant for routes optimized parcel delivery - backend and API implementation**

#### **6.6.1.1 Pick network (Nationwide Parcel Locker Network) address**

As of 29 Mar 2021, there are already 88 Pick Network lockers deployed. We get the addresses of these lockers from Pick Network website.

#### **6.6.1.2 Address in Latitude and Longitude**

Through OneMap API (call document @ https://www.onemap.gov.sg/docs/#search), we are able to get the Latitude and Longitude of these addresses



#### **6.6.1.3 Manhattan distance matrix**

**Manhattan distance**

Definition: The distance between two points measured along axes at right angles. In a plane with p1 at (x1, y1) and p2 at (x2, y2), it is |x1 - x2| + |y1 - y2|.

The Manhattan distance is used as it is closer to the actual distance in most cases in the Singapore city setting, as illustrated below: the blue color route is the actual vehicle travel distance. The green route is the Manhattan distance while the red route is another popular distance called Euclidean distance.

**Manhattan distance matrix**

The Manhattan distance matrix is obtained by computing the distance of all possible pairs of addresses.



#### **6.6.1.4 Vehicle demand, capacity and depot**

In order to use Google OR-Tools to solve capacity constrained vehicle routing problems, we have to input 3 more parameters, other than the distance matrix.

**Demand** = 25 (parcels): each location/locker has a *demand* corresponding to the quantity of the parcels to be delivered. In our case we know that each locker has a maximum capacity of 50 compartments, and a reasonable estimation is that 50% of these compartments will be utilized during the initial phase of deployment and adoption. The demand will go up when the Pick network gains traction over time.

**Capacity** = 250 (parcels): each vehicle has a capacity, the maximum number of parcels that the vehicle can hold and deliver on each day. 250 is the figure achieved by the delivery companies during the pilot rollout of the Pick network.

**Depot** = 26 (postal code = 350143): this is simply the depot, starting and ending point of the vehicle routing. In this case we use the address that is close to the central warehouses of a leading delivery company.

#### **6.6.1.5 Google OR-Tools to solve capacitated vehicle routing problem (CVRP)**

In the Vehicle Routing Problem (VRP), the goal is to find optimal routes for multiple vehicles visiting a set of locations. In our use case, there is capacity constraint on the vehicles: each vehicle can only deliver a maximum of 250 parcels on each day. With this it becomes a capacitated vehicle routing problem (CVRP).

We used some reusable functions and methods from Google OR-Tools to solve our CVRP, with the input of distance matrix, demand, capacity and depot. The solution, which contains the optimized vehicle routes, is stored in the JSON format {vehicle number 0: [sequence of postal code for vehicle 0], vehicle number 1: [sequence of postal code for vehicle 1], ……}.

#### **6.6.1.6 Optimized vehicle routes**

A Python function is created. It takes in the input of the vehicle number and current location of the vehicle, searches the stored optimized vehicle routes, then returns the next to go delivery location and also the remaining number of delivery locations.

#### **6.6.1.7 Web API endpoint**

An API endpoint is built using the FastAPI Web framework. It accepts HTTP requests with 2 input parameters: vehicle number and location in postal code format. It then calls the backend Python function and returns the desired output as described in 6.7.1.6.

#### **6.6.1.8 Picroft Voice Assistant API Call**

Picroft installed with voice command (“begin deliver”, “current deliver”, “next deliver”) to assist drivers on the deliver address.

### **6.6.2 Load Project Skill**

Load Project skill is an automated installation skill through voice command. Implementing this skill could save time, reduce the debug hassle when creating a project. Instead of long documented instruction, it can be replaced by a voice command.

### **6.6.3 DailyQuiz Skill**

This skill demonstrated API call skill. Today, there are many applications hosted with API, from weather, news, traffic status, flight schedule to interactive API like quiz, chatbot. The advantage of API skill is keeping the heavy lifting function and algorithm running at the server. This could make the voice application scalable.

### **6.6.4 Test ISS Project API and Skill**

#### **6.6.4.1 ISS Project API**

For this use case, we assume there are two ISS group projects to be tested by ISS lecturers. Project one is Iris Demo, and project two is Delivery Project. We provided the information about the API of Iris Demo here. For the full information of APIs of two projects, please refer to “**Use Case #5 – Test ISS Project**” section in file MDAA\_Installation\_User\_Guide.

|  |  |
| --- | --- |
| **Project One** | **Iris Demo** |
| **URL** | http://<Public IP Address of selene server:5000>/iss/iris/predict |
| **HTTP Method** | Post |
| **Content-Type** | application/json |
| **Input Parameters** | |  |  |  | | --- | --- | --- | | **S/N** | **Param Name** | **Data Type** | | 1 | sepal\_length | decimal | | 2 | sepal\_width | decimal | | 3 | petal\_length | decimal | | 4 | petal\_width | decimal |   JSON input sample:  {“sepal\_length”:5.5,  “sepal\_width”:3.1,  “petal\_length”:5.8  “petal\_width”:2.0  } |
| **Output Data** | |  |  |  | | --- | --- | --- | | **S/N** | **Output Data** | **Data Type** | | 1 | prediction | string | | 2 | probability | decimal |   JSON output sample:  { “prediction”:”virginica”,  “probability”:0.87  } |
| **Project Path** | The actual project of Iris demo was placed at “ **/opt/selene/selene-backend/shared/iss\_project/project1**” path in personal selene server |
| **Steps to build API** | 1. Define an endpoint and URL for Iris Demo in ‘api.py’ file at “**/opt/selene/selene-backend/api/iss/iss\_api**” path      1. Create an endpoint named “iss\_project1.py” for Iris Demo at “**/opt/selene/selene-backend/api/iss/iss\_api/endpoints**” path      1. Defile ‘post’ function and ‘\_call\_project\_api’ function in ‘iss\_project1.py’ file. The first entry function of Iris Demo will be called under ‘\_call\_project\_api’ function. |

#### **6.6.4.2 mycroft-iss-project Skill**

When ISS API is ready, we build the following three intent handlers in this skill to perform designed functions based on the lecturer's specific voice input.

* Show ISS Projects Info
* Test Iris Demo
* Test Delivery Project

Lecturers can accomplish the same task by expressing their intents in multiple ways given that the intent parser can be used to extract from the lecturer's speech key data elements. We list the details of the “Test Iris Demo” intended task here. For the full information of these three intended tasks, please refer to “**Use Case #5 – Test ISS Projects** ” section in file MDAA\_Installation\_User\_Guide.

**“Test Iris Demo” intended tasks**

|  |  |
| --- | --- |
| **Intended Task Two** | **Test Iris Demo** |
| **Statements** | The following statements can be used to trigger this intended task   * The statement includes “test Iris” key word * The statement includes “test project one” key word * The statement includes “test demo one” key word |
| **API Involved And Input Parameters** | 1. The URL for the API of Iris Demo is as below:  http://<Public IP Address of selene server:5000>/iss/iris/predict  2. The input parameters for this API are stored in a JSON file named ‘iris\_input.json’ under “input” folder of this Skill folder. The values for these input parameters can be changed so that this intended task is able to get the different results. |
| **Expected Response** | Mycroft-Core or Raspberry PI 3’s text-to-speech engine will convert the statement below to speech:  “The API was called successfully with the returned result is: prediction is {prediction}, probability is {probability}.”  *Note: Variables enclosed by curly brackets are returned data from API* |
| **Test Result** |  |

## **7. Project Conclusions**

During the beginning phase (around one month) of project processing, we decided to do a route optimized box delivery system. We then changed the idea to take Mycroft as our project after discussing the system prototype with the lecturer as Mycroft was more attractive for the team in this semester. In order to understand and make Mycroft more extendable, we defined two streams. One stream was applied device interfacing Mycroft server, and the other stream was personal server setup and applied device interfacing with it. On top of these, our team was continuing the initial delivery system (named as routes optimized parcel delivery now) implementation but using it like a backend function for this system to behave as voice assistant.

Currently, we have gained below achievements.

* Mycroft Linux and Picroft interfacing Mycroft servers are able to run both private and public skill
* Personal server has set-up and is able to provide services supporting skills running against Mycroft Linux and Picroft devices
* Routes optimized parcel delivery system developed and deployed on personal server and Mycroft Linux and Picroft interfacing this personal server can run the functions of this system.

Even though the time was very tight, our team did work very hard and cooperate effectively to fulfil our project goals. When the course was held on campus, we met after each course ending to share the knowledge and the progress of the assigned task. If there was an online course, a zoom session was scheduled for sharing and discussion for that week. It’s a good chance for us to apply and map learned knowledge to the project, meanwhile, to grow up together to adopt new things.

Finally, there are some improvements which we can consider for future processing.

* To setup better STT engine by instead of currently google STT with not desired accuracy
* Develop own UI interfacing personal server to make full picture of personal usage
* Deep further on Mycroft components for future needs
* Create a skill matured enough to explore and complete Skill Acceptance Process

# **Appendix: Project Proposal**

# **Appendix: Mapped System Functionalities against MR, RS, CGS**

|  |  |
| --- | --- |
| **Knowledge of Courses** | **System Functionality** |
| Machine Reasoning   * Text Processing/NLP | * Typed utterance in input of CLI after starting Mycroft which is treated as text is processed and recognized to respond. |
| Reasoning System   * Search Based Reasoning | * Speech to Text (STT) engine is transcribing utterance to intent as input to Adapt Intent Parse which matches the intent to skill to handle it. * Solve capacitated vehicle routing problem (CVRP) |
| Cognitive System   * Speech recognition systems * Speech recognition engine | * User’s speech is transcribed by Speech to Text (STT) engine (Google) feeding to intent parse to match the intent to skill to return text, and then returned text is converted to speech (Padatious) and stream to device. * PocketSphinx is a lightweight speech recognition engine using in Mycroft to detect Wake Word based on Speech to Text (STT) technology. |
| Machine Learning   * Neural Network * Neural TTS | * Precise is a neural network using in Mycroft that is trained on audio data to detect user’s chosen Wake Word. * Mycroft Text to Speech (TTS) uses [Mimic2](https://github.com/MycroftAI/mimic2) TTS engine based on Tacotron which is neural TTS to take test as input to generate corresponding spectrogram as output. |

# **Appendix: Installation and User Guide**

Please refer to the file “MDDS\_Installation\_User\_Guide”.

# **Appendix: Individual reflection of project journey**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Team Member** | **Role** | **Personal Contribution** | **Most useful learnt** | **Knowledge and skill application** |
| LI XIAOXIA | Team Leader | Overall system architect and design, Mycroft Linux application, Project management and documentation. | Understand Mycroft Architecture and components interfacing and interacting; Mapping knowledge and technology points gained from implementing the Mycroft Linux and application with knowledge learnt from courses;  Market value of voice assistants | Voice assistants’ product and framework could be used in bank, government and other servicing domains to relieve the human resources depending and save cost for organization.  It can also be applied in smart home schemes to improve the consumer’s living quality and involve more people to a high-tech platform for catching up the trend. |
| YANG YANG | Team Member | Overall system design and implementation, Personal server setup, application and documentation. | Learnt how to setup a personal selene server and make Mycroft-core connect to this server, and utilized them to build the customized APIs and Skill to apply to a use case. | Network knowledge,  REST API,  Python Flask,  Knowledge in Intelligent Voice Assistant |
| TAN WOEI MING | Team Member | Overall system architect, design and implementation, Picroft application and documentation. | Raspberry Pi application with voice command helps to open up IoT interaction (human) implementations. | Python application on Mycroft Intent |
| JIANG AIGUO | Team Member | Voice assistant for routes optimized parcel delivery design, implementation and documentation.  Iris [species](https://www.google.com/search?rlz=1C1CHBF_enSG836SG836&q=iris+species&spell=1&sa=X&ved=2ahUKEwjrnqOowKPwAhXSdCsKHfMwARgQkeECKAB6BQgBENUB) prediction application as a demo application to build into ISS Project API and Mycroft Skill | How to find a good business problem and use AI/ML to solve it. | OR-Tools to solve capacitated vehicle routing problems. Machine learning to solve classification problems. |
| ANG JENN NING | Team Member | Overall system implementation, promotional video, system presentational video | Understanding of fundamental of voice assistant and Mycroft Implementation with skills | Project idea pitching and marketing intelligent system with visual presentation |

# **Appendix: References**

1. <https://www.grandviewresearch.com/industry-analysis/intelligent-virtual-assistant-industry>

2. <https://mycroft-ai.gitbook.io/docs/>

3. <https://en.wikipedia.org/wiki/Virtual_assistant>

4. <https://www.raspberrypi.org/software/>