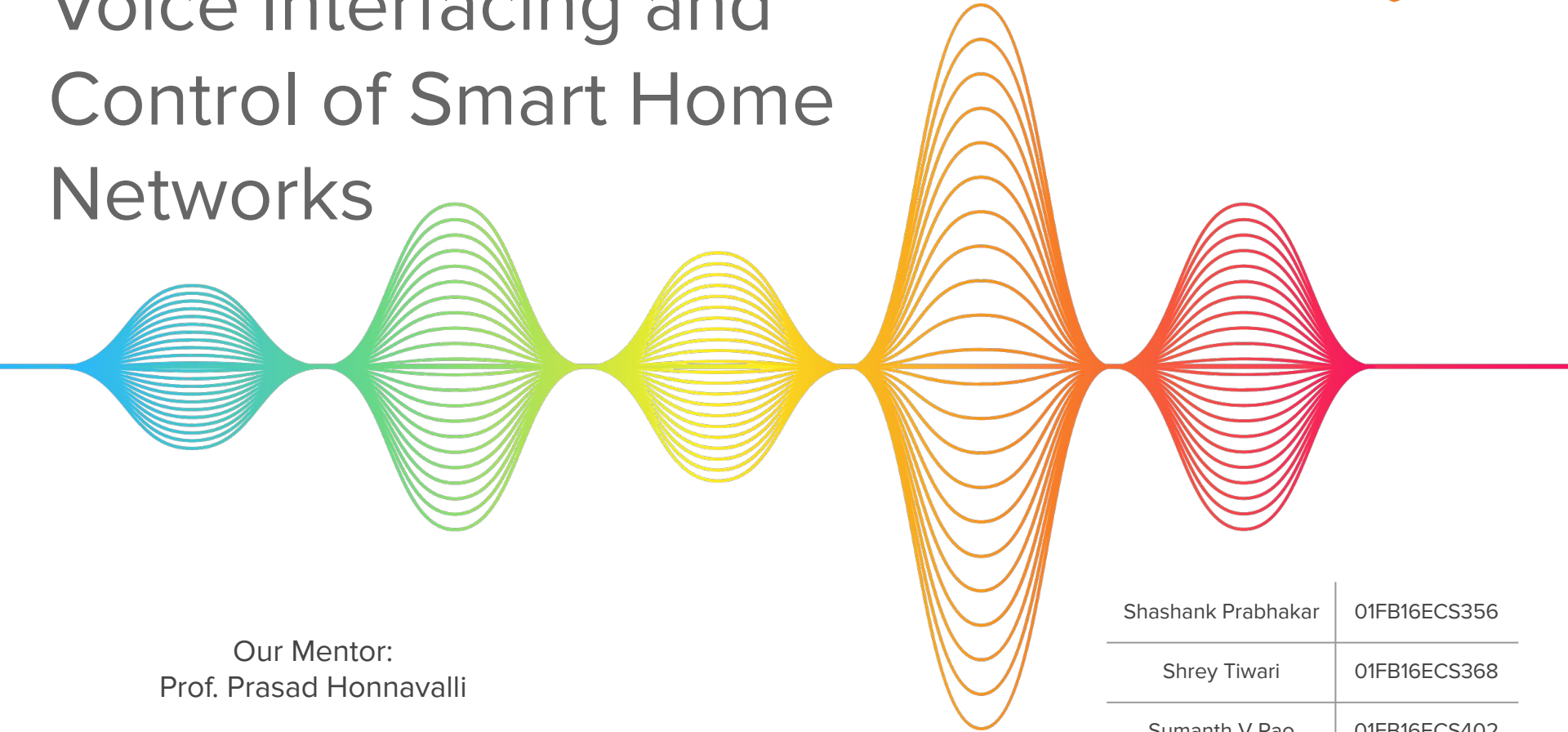


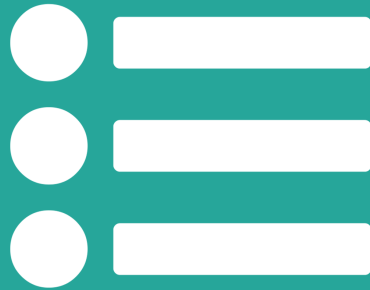
Voice Interfacing and Control of Smart Home Networks



Our Mentor:
Prof. Prasad Honnavalli

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

Table of Contents



- **Introduction**
- **Design**
- **Implementation**
- **Practicality**
- **Conclusion**

Introduction



Smart Home



Motivation



Aim of the Project

What is a Smart Home?

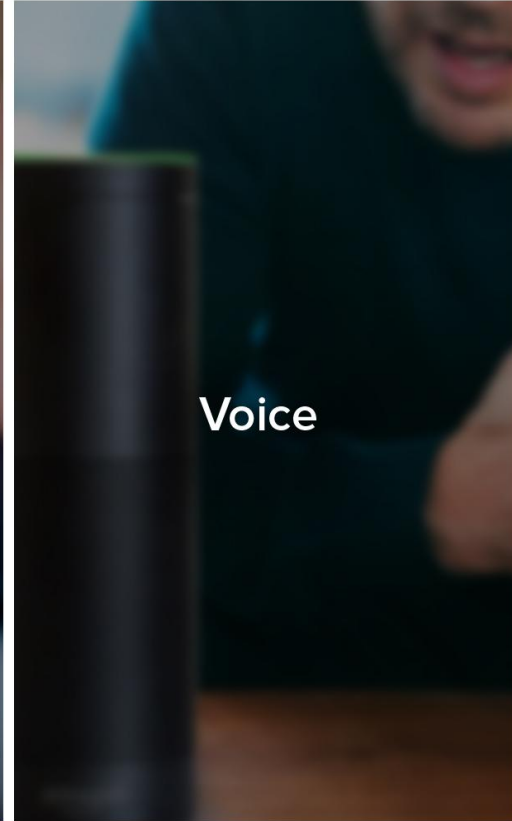
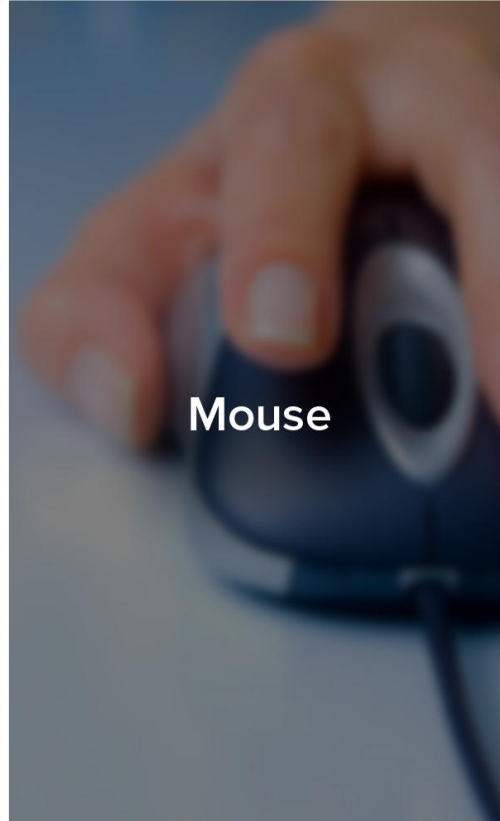
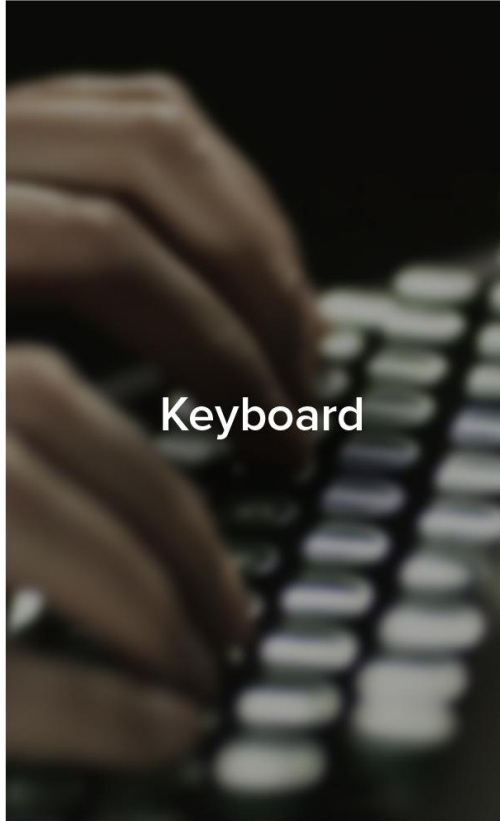


- **Must be 'Smart'**
Not just smart ways of controlling, but smart ways of living
- **Natural extension of human behaviour**
As customizable as a phone
- **Anticipate actions**
Must infer user intentions and actions
- **Connected**
The entities are interconnected and easily accessible

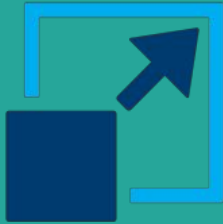


- **High Cost**
Expensive products and high installation/modification charges
- **Early stages of adoption in India**
Smart Home is still a new concept in our country
- **Low on security**
Many attack surfaces for IoT networks
- **Privacy Concerns**
Worry about one's personal data
- **Mutually exclusive ecosystems**
Market is dominated by a few companies and they have different sales channels.

Voice is the way ahead



Aim of the project



- **Cost-effectiveness**
The solution should be cheap. Must integrate with existing infrastructure.
- **Security**
The network should have security by design.
- **Scalability**
The design should not have any choke points. It must be easy to add more devices.
- **Manageability**
The interfacing with the network should be seamless and intuitive.
- **Configurability**
It should be possible to update and configure the network over the air.

Design



Security by Design

Misuse Case Study
and Threat Modelling



Architecture

System Components
and dependencies



Software Stack

Design choices
for Software

Use Cases and Misuse Cases

USE CASES

Control and interface with devices using voice commands

Control and interface with devices using the web application

Ability to perform compound actions

Adding, removing and discovery of devices

Adding and deleting users of the devices

Control third party devices

MISUSE CASES

Skill Squatting

Unauthorized access and control of an IoT device using voice commands

Information leakage from IoT home monitor to outsiders or guests

Use home IoT network as Botnet to launch DDoS attack

Privacy breach by exploiting vulnerabilities of voice assistant

Gain access IoT network via voice commands when not at home

COUNTERMEASURES

Certification systems and verification of skills

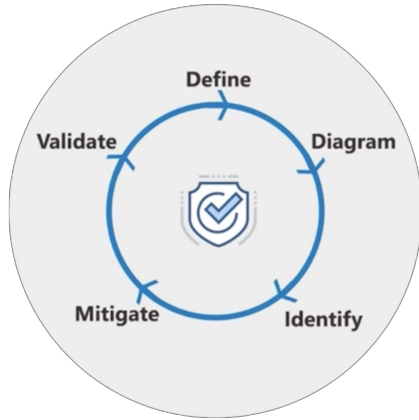
Voice recognition and unique passphrases

Access Control

Firewalls and Intrusion Prevention Systems

Kill Switch

Disabling voice interfacing subsystems in the absence of the owner



Threat modelling works to **identify**, **communicate**, and **understand** threats and **mitigations** within the context of protecting something of value

S

SPOOFING

T

TAMPERING

R

REPUDIATION

I

INFORMATION DISCLOSURE

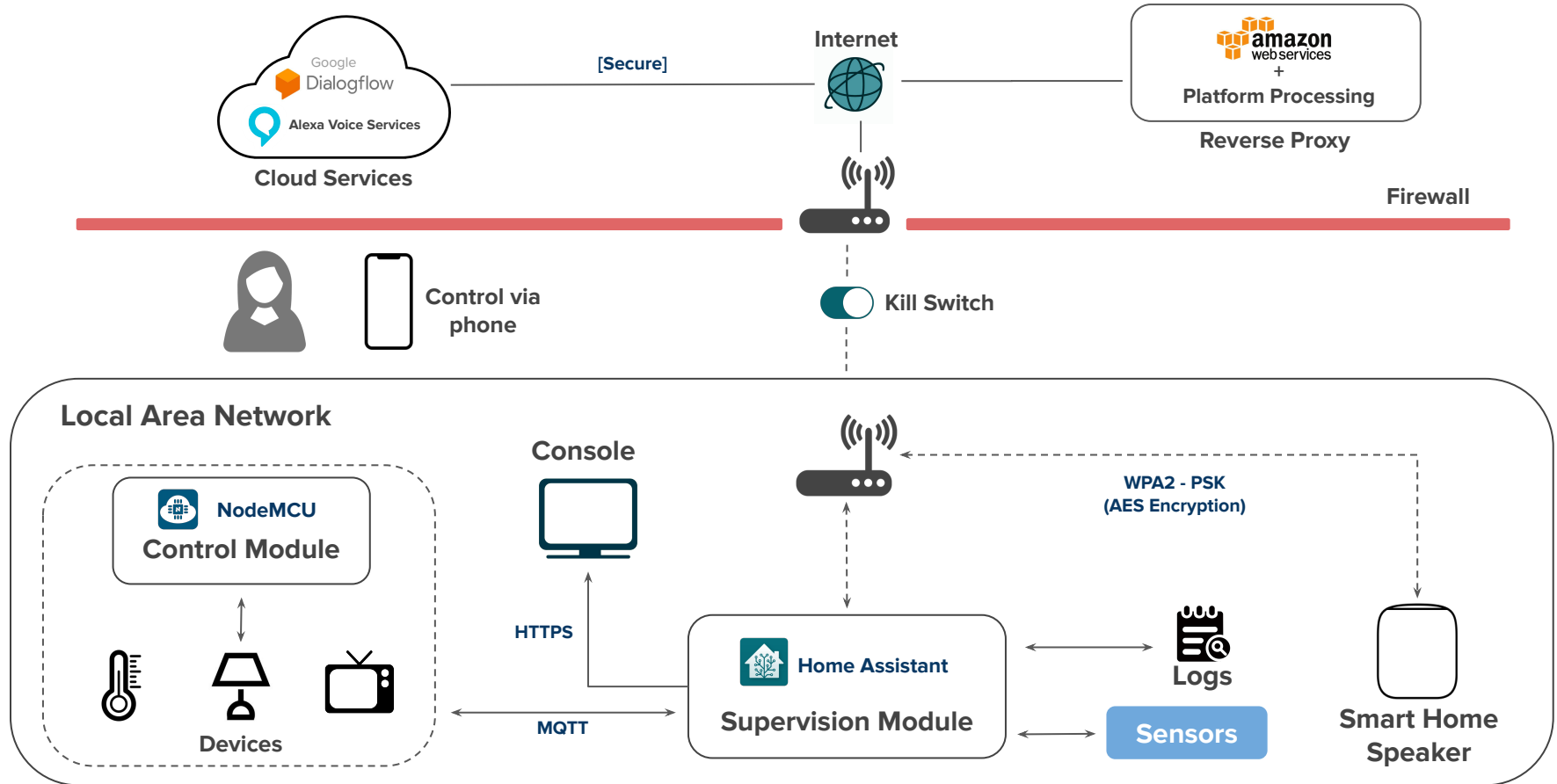
D

DENIAL OF SERVICE

E

ELEVATION OF PRIVILEGE

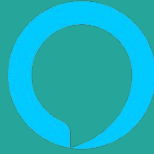
Architecture incorporating design choices



Software Stack



Actions on
Google



Alexa Skills



Home Assistant



Node.js

- **Actions on Google**
To develop an interactive voice assistant capable of interfacing with our Smart Home
- **Alexa Skills**
To interact in a similar manner with the popular alternative - Amazon Alexa
- **Amazon Web Services**
To host the voice assistant abstraction server
- **Home Assistant**
Event driven Operating System - serves as the brain of the Smart Home
- **Node.js**
Framework to develop the server endpoints

Implementation



Security

Two-router setup and authentication



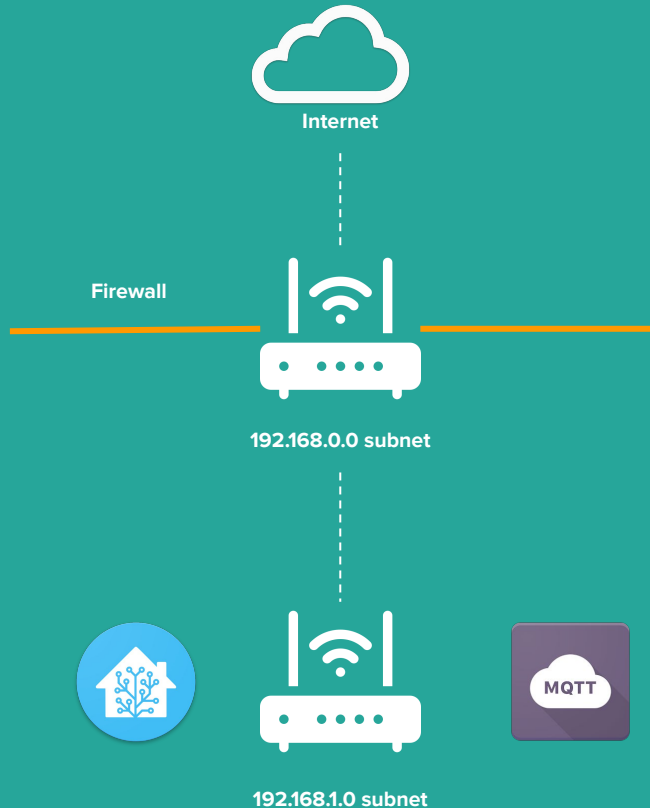
Platform Agnostic

Ability to control Smart Home with both Google Assistant and Amazon Alexa



Intuitive UI

More information, Better entity cards.



- **Local Area Network**
Second router connects to the first router as just another device
- **Network Segmentation**
Only users connected to the second router have access to Home Assistant
- **Kill Switch**
Increased security and privacy
- **Websockets**
Does not open up ports in the Firewall. Efficient means of communication

Google Dialogflow Interface

Training phrases ?

Search training phrases

” Add user expression

” Switch off the kitchen lights

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

” 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

” Turn the living room lights on

Try it now

See how it works in [Google Assistant](#).

Agent

USER SAYS [COPY CURL](#)

Set the kitchen light brightness to 100%

DEFAULT RESPONSE

Sure! The light in your Kitchen has been set to 100% brightness.

INTENT


[Light_brightness](#)

ACTION

Not available

PARAMETER	VALUE
House_rooms	Kitchen
percentage	100%

Conversation with
Google Assistant

 alexa developer console

[Your Skills](#) [Home-Buddy](#) [Build](#) [Code](#) [Test](#) [Distribution](#) [Certification](#) [Analytics](#)

English (US)

Save Model

View Model Versions

Build Model

Update live skill

Evaluate Model

CUSTOM

Interaction Model

Utterance Conflicts (0)

Invocation

Intents (9)

+ Add

Appliance_toggle

appliance_mode

house_room

target_appliance

Light_brightness

house_rooms

percentage

Fan_speed

Configure_device

Dimmable_confirmation

Updates to sample utterances qualify for instant live updates. [Learn more](#) about live updates to your skill.

Intents / Appliance_toggle

Sample Utterances (5) ?

Bulk Edit

Export

What might a user say to invoke this intent?

turn the {target_appliance} in the {house_room} {appliance_mode}

switch {appliance_mode} the {house_room} {target_appliance}

turn my {house_room} {target_appliance} {appliance_mode}

turn the {house_room} {target_appliance} {appliance_mode}

turn {appliance_mode} {house_room} {target_appliance}

< 1 - 5 of 5 >

Show All

Platform Agnostic Design

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

Received a POST request, forwarding it to the router
Reached the router.
The request was made using Alexa.
The intent extracted is: SetLightBrightness
Routing for the intent: SetLightBrightness
In the SetLightBrightness processRequest handler.
The important parameters are:
Target Room: living room bedroom
Brightness Percentage: 30
-----x-----

Received a POST request, forwarding it to the router
Reached the router.
The request was made using Alexa.
The intent extracted is: SetLightBrightness
Routing for the intent: SetLightBrightness
In the SetLightBrightness processRequest handler.
The important parameters are:
Target Room: bedroom
Brightness Percentage: 30
-----x-----

Received a POST request, forwarding it to the router
Reached the router.
The request was made using Google Assistant.
The intent extracted is: SetLightBrightness
Routing for the intent: SetLightBrightness
In the SetLightBrightness processRequest handler.
The important parameters are:
Target Room: bedroom
Brightness Percentage: 30
-----x-----
```

```
1: node, ngrok
ngrok by @inconshreveable (Ctrl+C to quit)

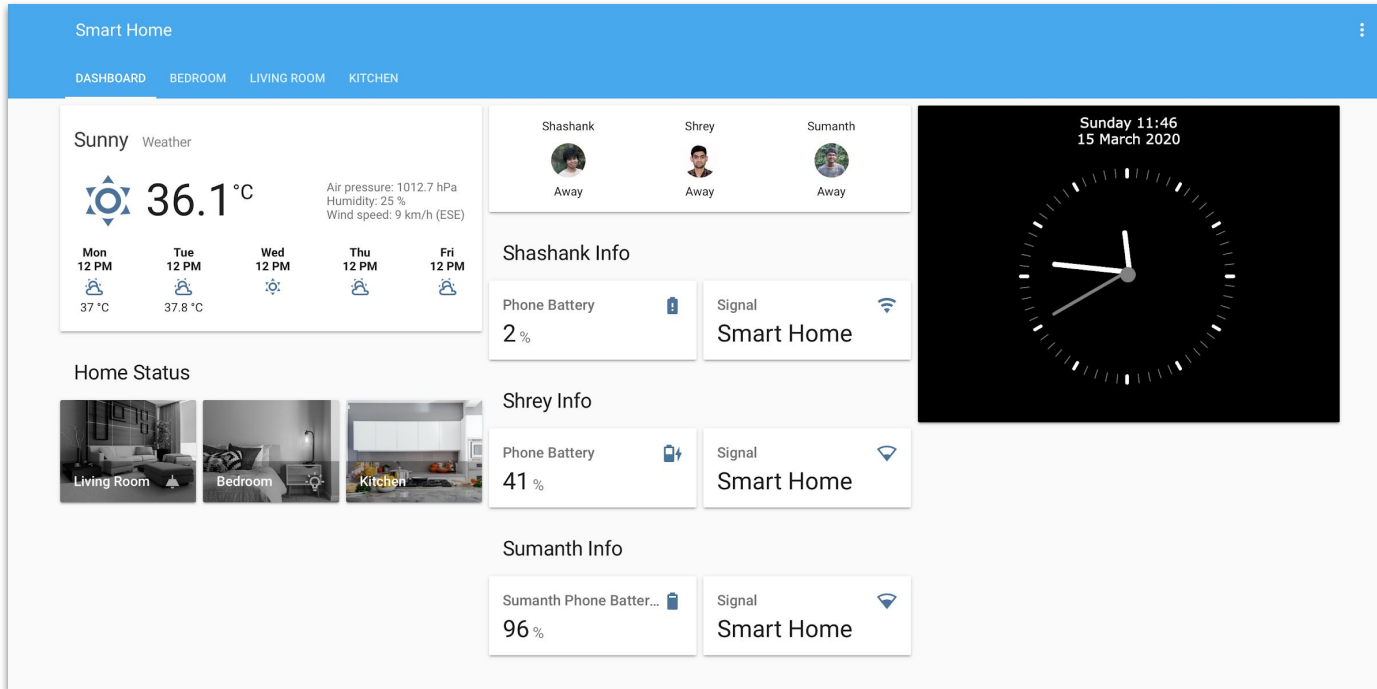
Session Status      online
Account             Sumanth V Rao (Plan: Free)
Version             2.3.35
Region              United States (us)
Web Interface        http://127.0.0.1:4040
Forwarding           http://81a45cb0.ngrok.io -> http://localhost:8080
                    https://81a45cb0.ngrok.io -> http://localhost:8080

Connections          ttl    opn    rt1    rt5    p50    p90
6                  0      0.01   0.01   5.20   5.28

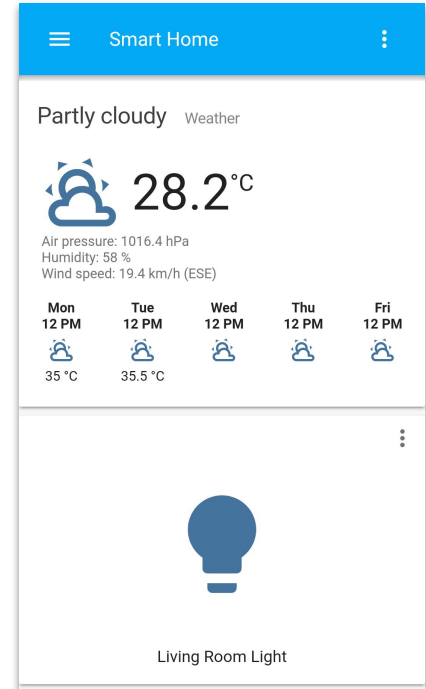
HTTP Requests
-----
POST /              200 OK
POST /              200 OK
POST /              200 OK
POST /              200 OK
POST /              200 OK
POST /              200 OK
```

Server for voice assistant abstraction

User Interface



Web UI



Mobile App UI

Practicality



Configurability

Code restructuring for
Hands-free addition of devices



Error Handling

Deal with error conditions
and invalid actions




Manageability

Uniform agent on all
NodeMCU modules





- Added functionality to server to **handle error conditions** and let the user know about **invalid actions**
- Added functionality to the server to support **hands-free configuration** of new smart home devices

Error Handling

Try it now 

Yes

 DEFAULT RESPONSE 

Another device with the same name exists in the mentioned room. Please rename your device

CONTEXTS RESET CONTEXTS

confirm-name light-dimmability

INTENT

ConfirmDimmability

ACTION



Not available

alexa developer console

< Your Skills Home-Buddy Build Code Test

Skill testing is enabled in: Development

Alexa Simulator Manual JSON Voice & Tone

English (US)  Type or click and hold the mic 

Which pin on the NodeMCU is the device connected to?

3

Is your light dimmable?

yes

Another device with the same name exists in the mentioned room. Please rename your device

Code Structure Redesign

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

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

Smart Home

Room Name

NodeMCU
ID

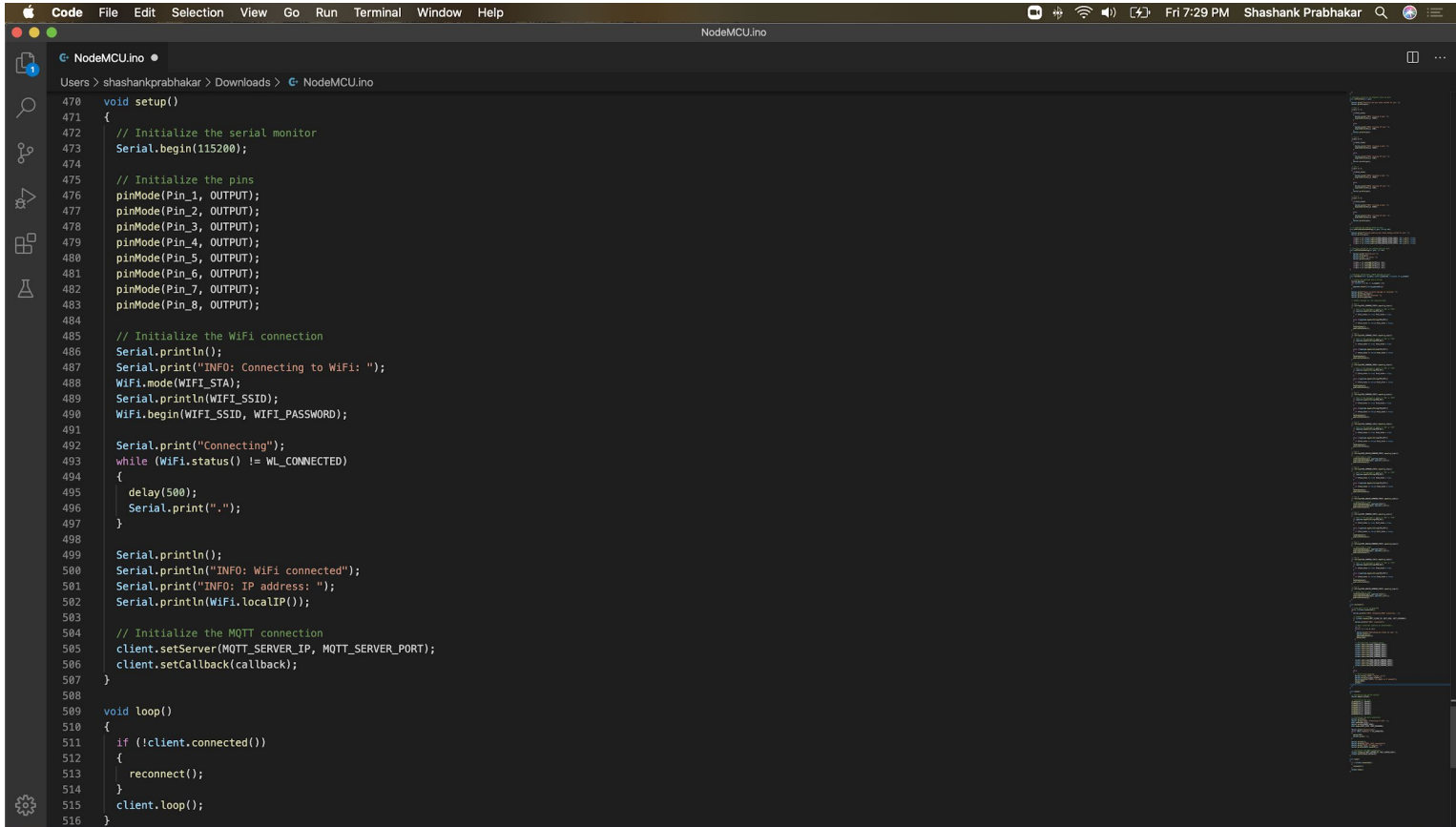
Pin Number

Adjustable



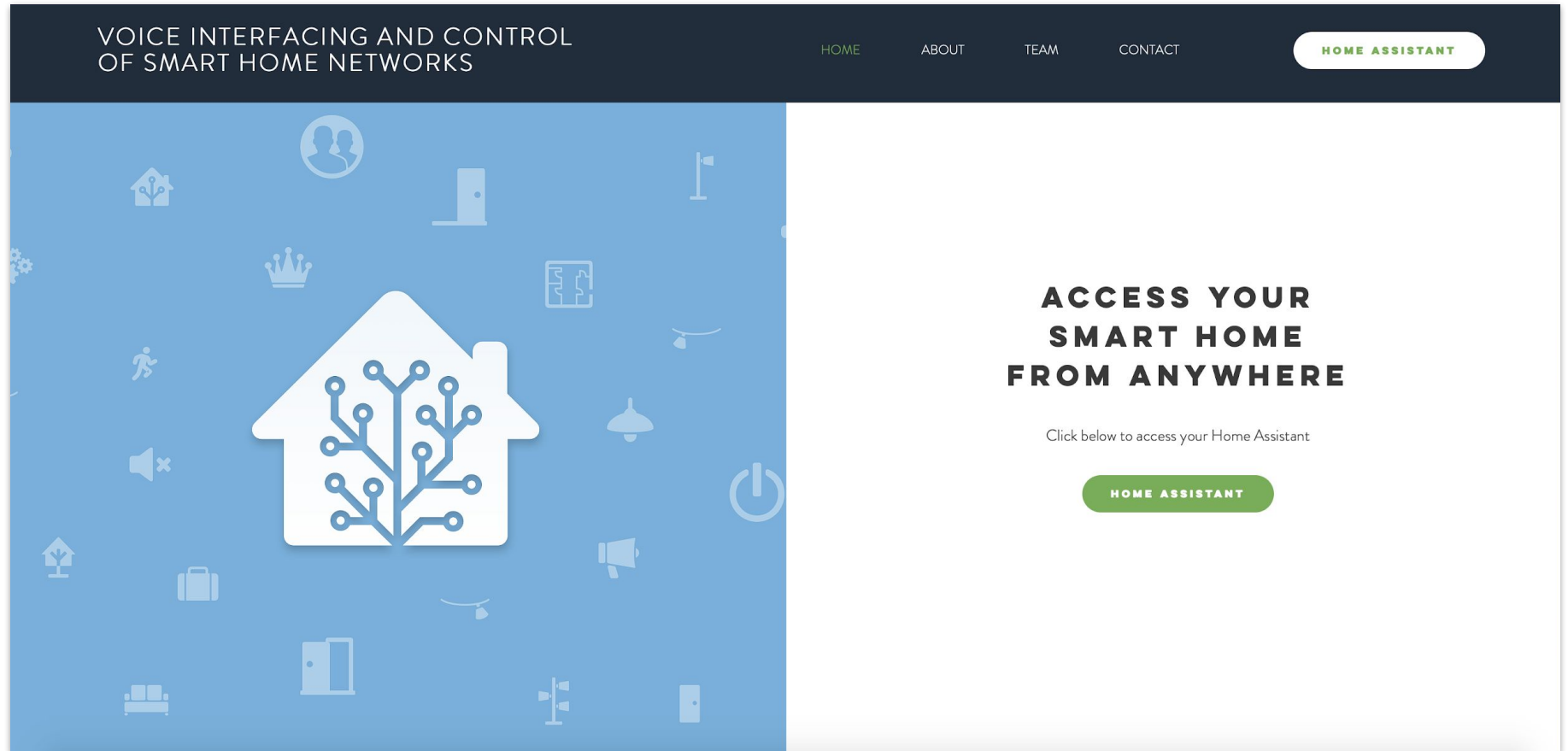
- **Main challenge in a Smart Home: Manageability**
 - Easily add new appliances
 - Modifying existing devices
 - Remove devices from the network
- To achieve manageability, the need to have uniform agent running on all NodeMCU modules
- Access Home Assistant from outside local network
- Enables scalability - Reduce dependency on NodeRED

Single code for all modules

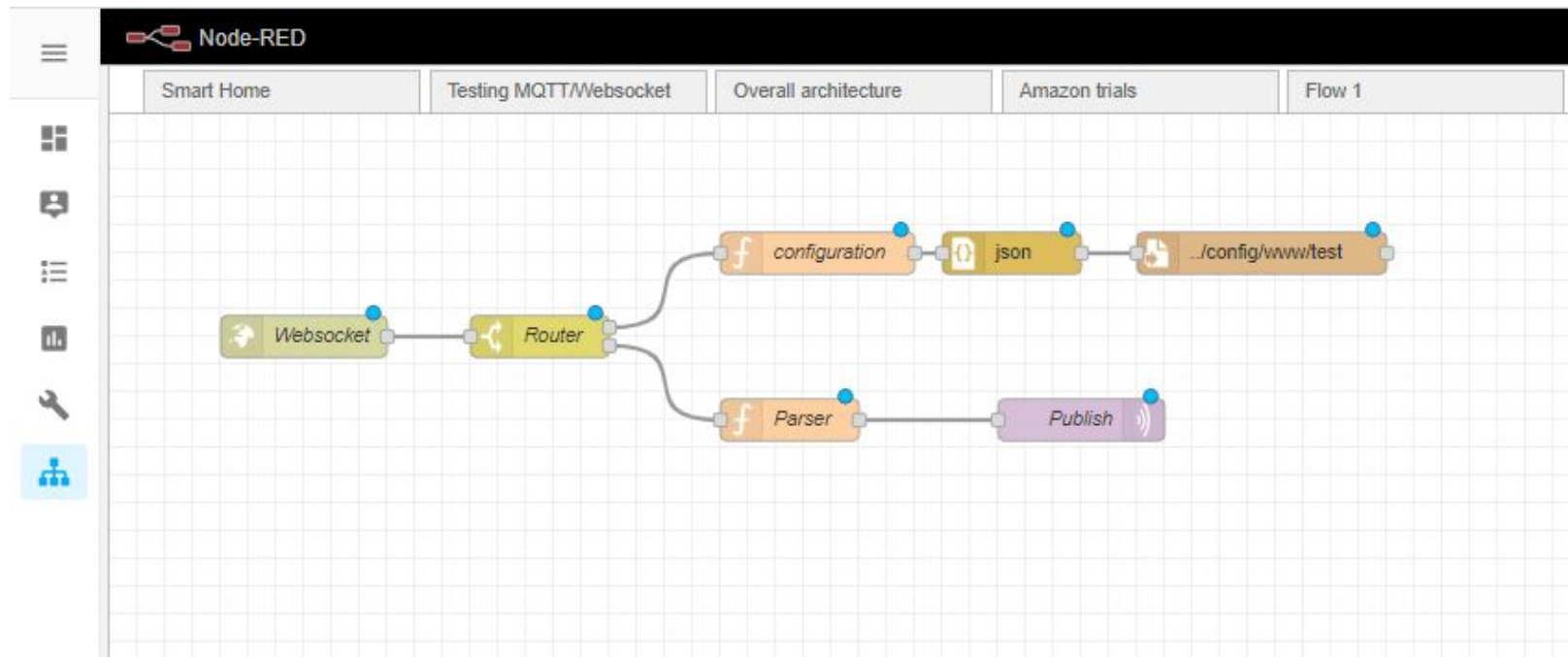
A screenshot of a macOS-style IDE window titled 'NodeMCU.ino'. The window has a menu bar with 'Code', 'File', 'Edit', 'Selection', 'View', 'Go', 'Run', 'Terminal', 'Window', and 'Help'. The status bar at the top right shows system icons (Bluetooth, Wi-Fi, battery), the time 'Fri 7:29 PM', and the user name 'Shashank Prabhakar'. The main editor area shows C++ code for a NodeMCU project. The code includes comments for initializing the serial monitor, pins, Wi-Fi connection, and MQTT connection. The code is organized into two main functions: 'void setup()' and 'void loop()'. The 'void setup()' function initializes the serial monitor, pins, Wi-Fi, and MQTT. The 'void loop()' function checks if the MQTT client is connected and reconnects if necessary. The code is saved as 'NodeMCU.ino' in the 'Downloads' folder. The IDE interface includes a sidebar with icons for Explorer, Search, Source Control, Run and Debug, and Extensions. The Explorer sidebar on the right shows a file tree with various files and folders.

```
470 void setup()
471 {
472     // Initialize the serial monitor
473     Serial.begin(115200);
474
475     // Initialize the pins
476     pinMode(Pin_1, OUTPUT);
477     pinMode(Pin_2, OUTPUT);
478     pinMode(Pin_3, OUTPUT);
479     pinMode(Pin_4, OUTPUT);
480     pinMode(Pin_5, OUTPUT);
481     pinMode(Pin_6, OUTPUT);
482     pinMode(Pin_7, OUTPUT);
483     pinMode(Pin_8, OUTPUT);
484
485     // Initialize the WiFi connection
486     Serial.println();
487     Serial.print("INFO: Connecting to WiFi: ");
488     WiFi.mode(WIFI_STA);
489     Serial.println(WIFI_SSID);
490     WiFi.begin(WIFI_SSID, WIFI_PASSWORD);
491
492     Serial.print("Connecting");
493     while (WiFi.status() != WL_CONNECTED)
494     {
495         delay(500);
496         Serial.print(".");
497     }
498
499     Serial.println();
500     Serial.println("INFO: WiFi connected");
501     Serial.print("INFO: IP address: ");
502     Serial.println(WiFi.localIP());
503
504     // Initialize the MQTT connection
505     client.setServer(MQTT_SERVER_IP, MQTT_SERVER_PORT);
506     client.setCallback(callback);
507 }
508
509 void loop()
510 {
511     if (!client.connected())
512     {
513         reconnect();
514     }
515     client.loop();
516 }
```

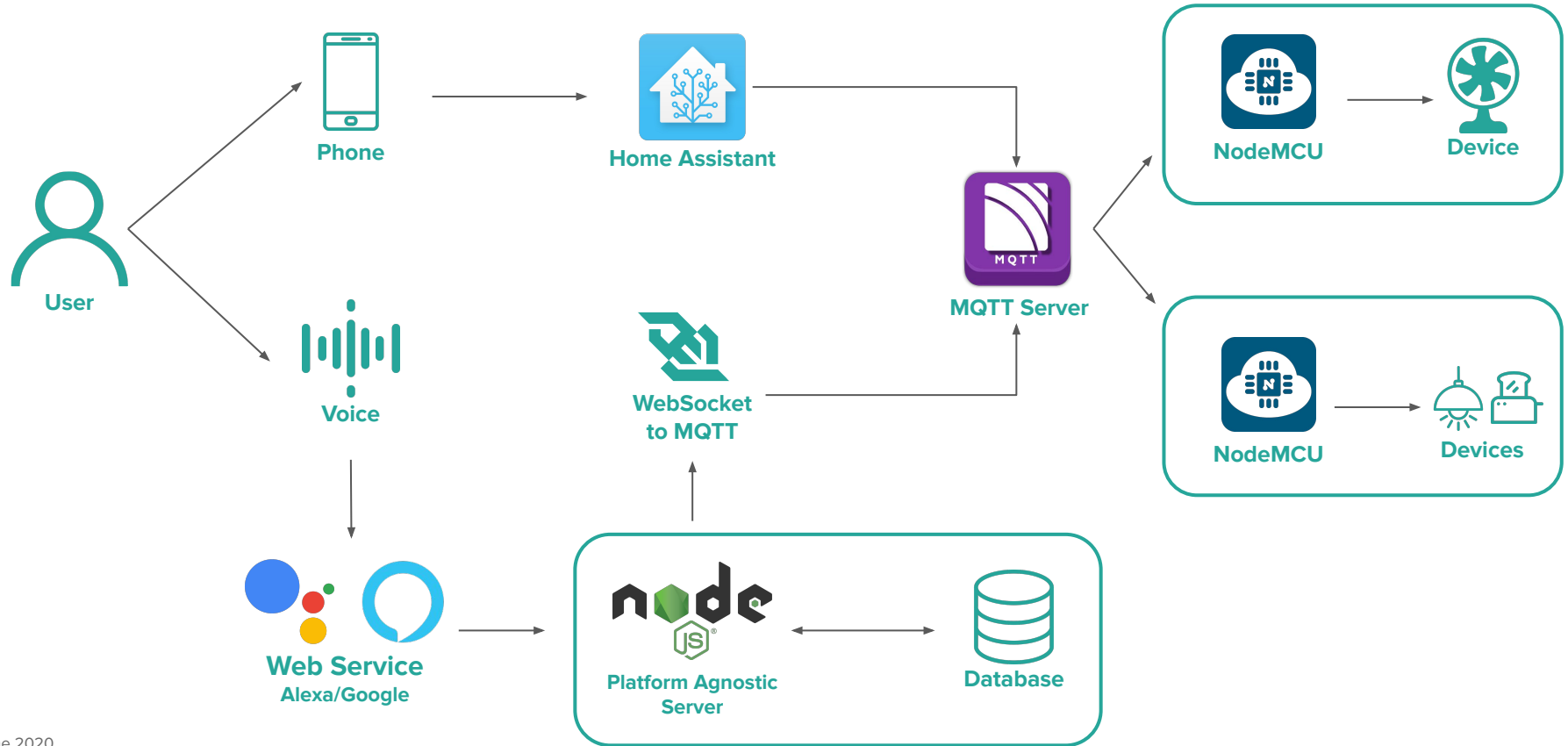

Remote Access - Public Web Page UI



NodeRED



Workflow



Conclusion



Outcome



Next Steps



Learnings



+



+



- **Security by Design**
To incorporate security choices from the beginning
- **Platform agnostic solution**
Should be able to use smart home system with both Google Home and Amazon Alexa
- **Easy to use and Robust design**
Solution that is intuitive by nature and resilient by design

Next steps



- Research Journal
- Apply for patent

010101010101010101010001010101101101101101010
101010101010100010100101010101010010110101011
01010101010101010101010001010101101101101010
101010101010100010100101010101010010100000101
01010101010101010101010001010101101101101010
101010101010100010100101010101010010100000101
011101010101010000001010101010101010101010101
111101010010010010010101101001001001001010111101
010101001010101010001001001001001001001011010
01010101010101010101010001010101101101101010
101010101010100010100101010101010010101010101
01010101010101010101010001010101101101101010
10101010101010001010010101010101001010101010
01010101010101010101010001010101101101101010
101010101010100010100101010101010010110110101
01110101010101010000001010101010101010101111
111101010010010010010101101001001001001010111001
010101001010101010100010010010010010010010100
01010101010101010101010001010101101101101010
101010101010100010100101010101010010101010100
01010101010101010101010001010101101101101010
101010101010100010100101010101010010101010101
01010101010101010101010001010101101101101010
101010101010100010100101010101010010101010100
011101010101010100000010101010101010101010000

- **Data Structures and Algorithms**
- **Microcontrollers and IoT**
- **Computer Networking**
- **Cyber Security and Information Security**
- **Web Technologies**
- **Cloud Computing**
- **Software Engineering**

Thank you