VOGO – Helmet-Harborer System

Team : Guns and Roses
Members : Muhammed Roshan, Rohit R Nath

1. Summary

Helmet theft become headache for most of the vehicle renting services. It create huge loss to most of the company and many users are not able to use the service securely due to unavailability of helmet during ride. Hence we are here with Smart and worlds cheapest solution for this problem. This document is intended to explain high level design of anti-helmet theft system.

2. Tools Used

To implement this POC, we must work on 3 regions such as, On-Bike unit, Server and UI application.

2.1 On-Bike Unit (OBU)

- Controller used – **ESP32** stand alone SoC with inbuild Wi-Fi and BT.
- Programmed using **Arduino IDE**
- Other components – **resistors**

Note : If the current IoT board in the VOGO bike having ADC, then the costless resistors are only needed for this implementation.

2.2 Server Unit

In Server Unit,

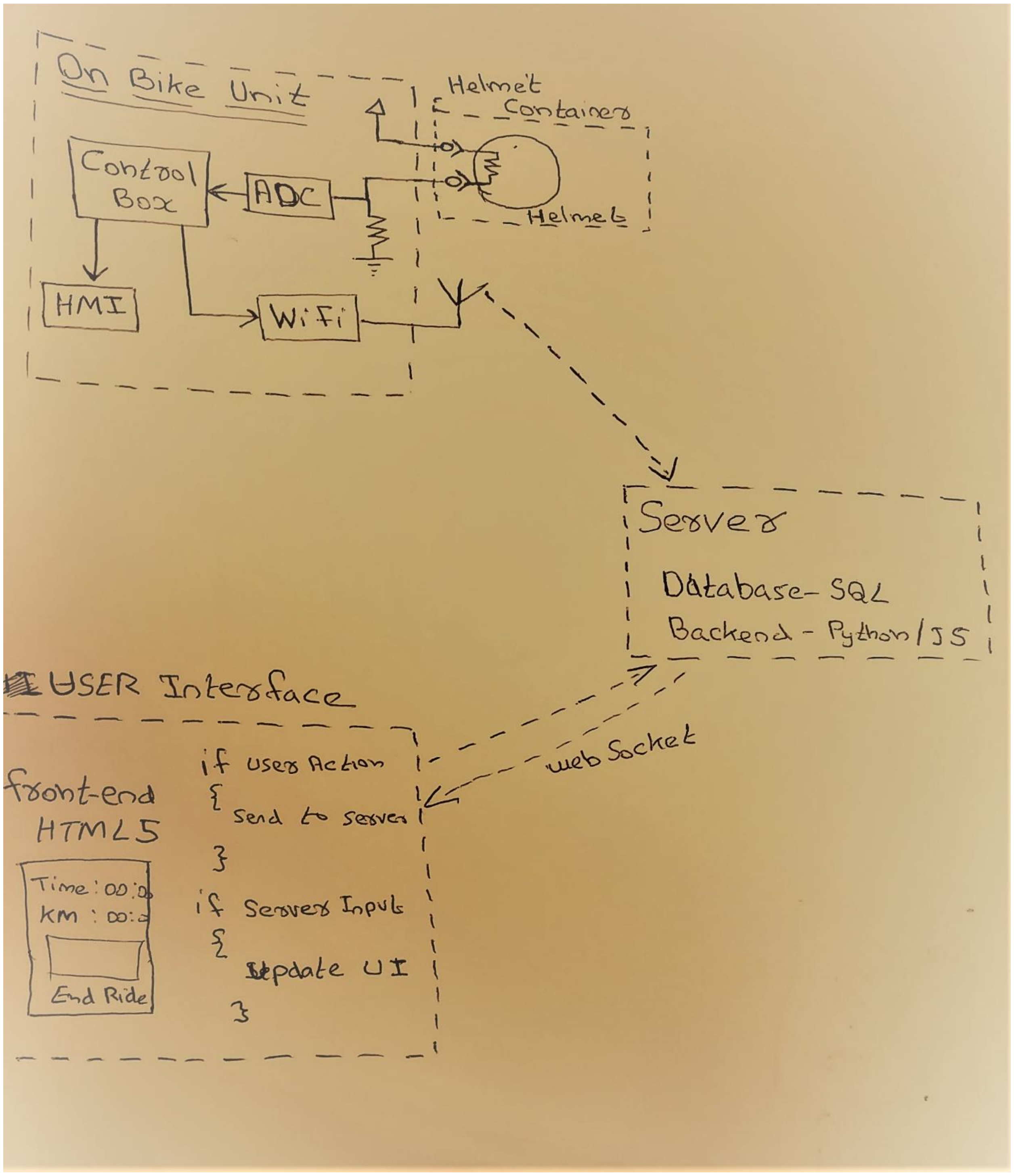
- Major backend process will take care by Python based scripts.
- SQL based database will used to store all user/ride details
- Javascript (NodeJs) based web sockets will be used to communicate with user's UI.

2.3 User Interface

- HTML5(html, CSS, JS) based progressive WebAPP with responsive window.
  - This webApp will suitable in any environment which is having internet browser feature.

3. Technical Architecture

3.1 HIGH-LEVEL DESIGN



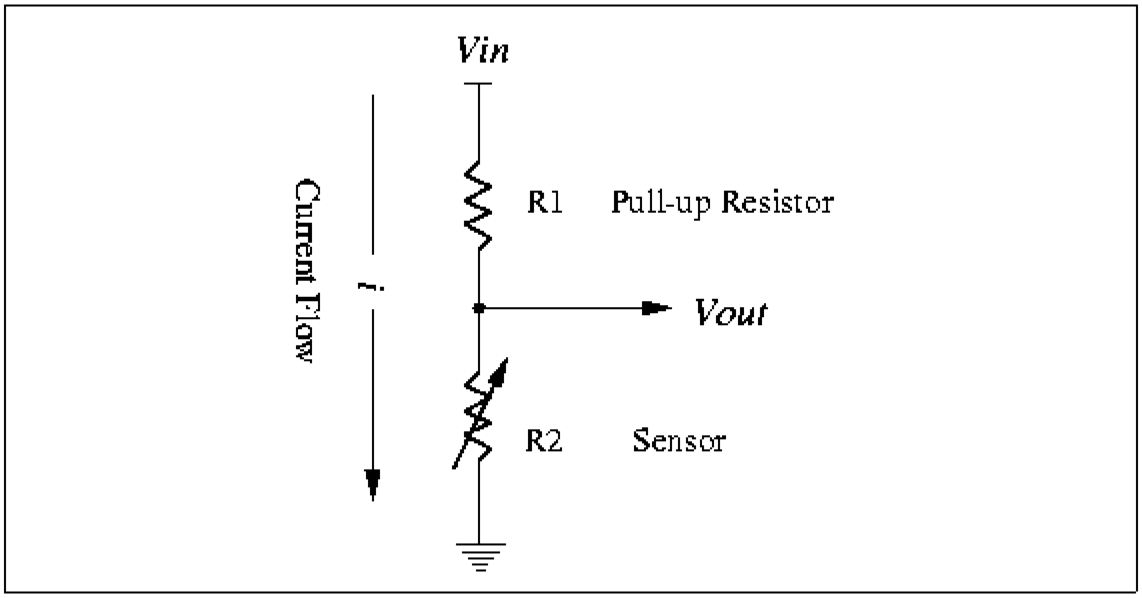
3.2 Blocks in detail

3.2.1 On-bike Unit(OBU)

In our POC the major part will be a tiny **costless add-on to the present IoT hardware**.

Resistors are the passive components which costs less than 10 paisa/ piece.

Here we are making the helmet containers intelligent enough to authenticate the helmet with simple but powerful circuit of electronics, **Voltage Divider**.



Here in the place of sensor, we will use a helmet which containing fixed resistance that complete the remaining circuit which is there in the container, the out put will read by out existing IoT hardware with the help of an ADC (Analog0 to Digital Converter).

The resistance of each helmet may different that will be the identity of each helmet.

3.2.1.1 Control Flow

1. When user going to start the ride, our IoT hardware will read the analog voltage at the Vout region with the help of ADC and will store that to non-volatile memory as well as send this to cloud server.
2. When the user going to end the ride, the system will check for the same analog voltage at Vout end with some tear threshold(Running Median will use).
3. If the same voltage reading is available at the Vout end then we can ensure the helmet is there in the container, if helmet is not there, the information will send to server as will as HMI will alert the rider to put it back to helmet container.
4. If server getting any notification that, the rider tried to end ride without putting the helmet back, the server will consider this as a suspicious activity and it will report this to user interface.
5. Whenever the suspicious notification get by UI from server, they will block the end-ride, this will leads to continue the run time.

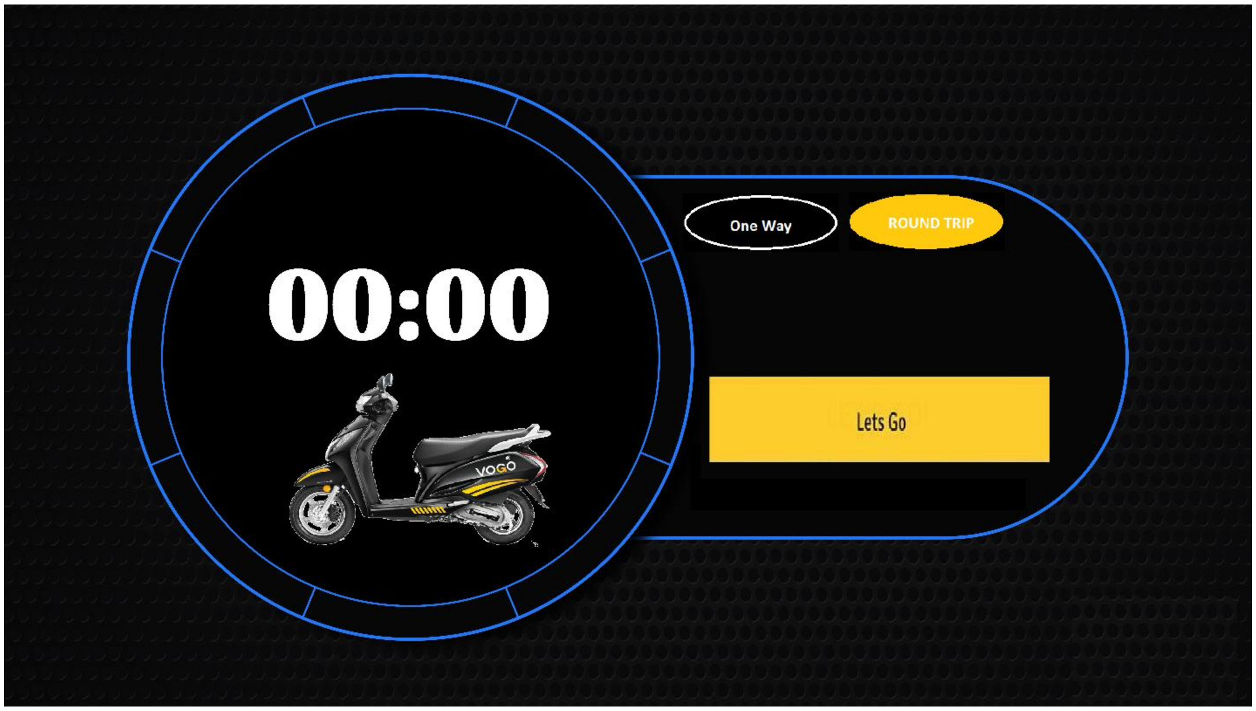
3.2.2 Server Unit

The core server part will do by python scripts, which will communicate to the bike and javascript (nodeJS) based web sockets will communicate with user interface.

3.2.3 User Interface

User interface will be a progressive webApp which made using HTML5.

Sample:



4. Feasibility

The solution to this problem can be implemented by using RFID and other technologies. But implementing those technologies in existing system is more expenssive as well as difficult. Hence our focus was to create cheap but smart solution. And we are here with a solution that can implement on any existing system with minimal cost and effort with zero compremize on the efficiency. We can implement this system at a cost atleast **50 times lesser** than any other solutions.