I. Cloud Robotics Communication Tool

A. Background

If a roboticist has to deploy a cloud based robot, the person was required to have certain knowledge about the different protocols and socket programming and has to make redundant code to deploy more robots. It would be more easier for the person if there was a tool which would take care of the networking part according to the user's needs. After careful research, it was found that there hasn't been a tool which had been developed to address this issue. This led to the development of a Cloud Robotics Communication Tool, which facilitates the easy deployment of a cloud robot. It would just take minutes to configure and start sending and receiving data to and from the cloud server. This tool has been under active development and new features are being added.

B. Design and Working

The tool is designed in such a manner that the user could deploy a cloud robot in a rapid manner. The tool takes a few user inputs and generates sender-receiver scripts in both the server and the robot client. The tool consists of two modules.

1) A Master Server Module and
2) A Client Configuration Module

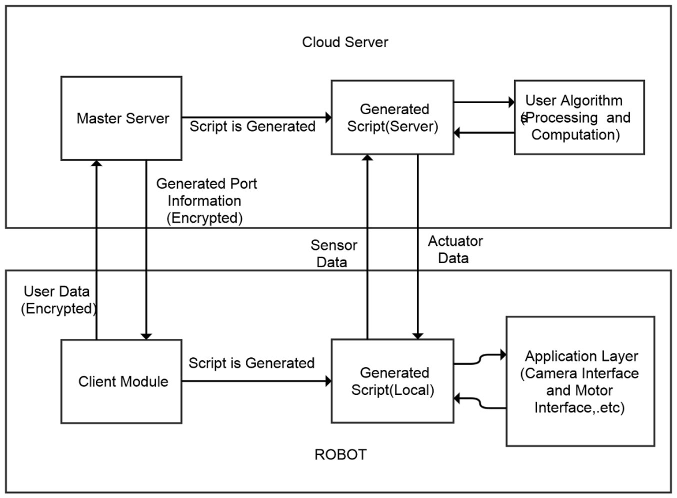

Fig. 1. Cloud Robotics Tool

1) Master Server Module: The Master Server Module is placed in the cloud server. The master server module, when executed, waits for commands from the client module. When it receives the appropriate command and user information, It calculates and assigns ports, Once the ports have been assigned, it starts to generate a sender-receiver script and also sends the port information back to the client module.The master server also starts the generated script automatically. The Master Server Module acts as the core which deploys the scripts,the master module is being developed so that it could also monitor the deployed robots and give back the information and status of the deployed robots.

2) Client Configuration Module: The client configuration module is placed in the client robot.The configuration module takes user inputs and sends it to the master server module. Once the master server generates the ports,it sends back the port details to the client module. The client module then generates the sender-receiver file for the robot and launches it automatically.However , the user has to program the application layer for the robot.If the user wants to set-up another cloud robot, He can run the client module on the second robot and enter the sensor and actuator information, A sender receiver pair similar to the first robot will be generated.

C. Encryption

The communication between the Master and the client module has been encrypted so that the sensitive port information cannot be intercepted and also the system cannot be hacked.There is a default Encryption Key set already, The user can change the encryption key according to his needs. The Encryption standard used here is AES Encryption in Counter Mode.The AES encryption is a block cypher based encryption method. The main design principle of AES is to achieve at least the security level of Triple DES while improve the performance in both software and hardware platforms. After three rounds of evaluation and selection for about four years, rijndael, a block cipher proposed by two Belgium cryptographers was selected as the official AES standard[12].The Counter Mode is a relatively new mode in which each cypher blocks are not dependent on the other cipher blocks.

D. Features

One of the main features of this tool is an easy way to plug-in algorithms into the code.By default The tool provides two algorithms,Face Detection(Haars Cascade) and HSV based object detection.The user can select either one of the algorithms while entering his sensor and actuator information.More features are being added to tool, as the development of the tool is still in progress. Some of the other tentative features are

i) Ability to add more than one algorithm.
ii) Ability to monitor the running processes in real-time.
iii) Feature to share sensor information among the processes generated in the cloud.

More features are likely to be added as the development of the tool progresses. New feature ideas are brought to our attention with active testing of the tool.

II. Documentation

The server module is pasted in the cloud server and open a new terminal and navigate to the folder, first the setup.sh file is executed by typing the command

```
./setup
```

This installs all the required dependencies.

Then the master server process is executed by typing the following command in the terminal

```
python masterserver.py
```

The Client module is pasted in the robot to be deployed.
Then the setup.sh file is first executed.
Once the setup is done, the client module is run by typing the following command.

```
python client.py
```

The client software asks for the ip address of the server. And then, to create a cloud connection, the user has to type the command

```
addrobot
```

The user will be prompted for number of sensors and then the number of actuators
The user will also be asked if he prefers to use any of the built-in algorithms.
Once the user enters all the data, A file named ”robot(robot number)” will be created
for example, when adding the first robot, the file created will be named as robot1.py and so on

This file will contain all the initializations and socket bindings necessary

For example, If the user gives 2 Sensors and 1 actuator,Two variables will be created called Sensordata0 and Sensordata1, to which the user can send sensor data. And for the actuator data will be received in a variable called actuatordata0