-

## Project Research and Innovation

# Smart Pot

An intelligent way to manage gardens

**Submitted By**
Mansi Karode
Naveen Rajappa
Rahul Kumar Thaivalappi
Blessie Durairaj
Marcelo Ardiles

## Table of Contents

## Introduction

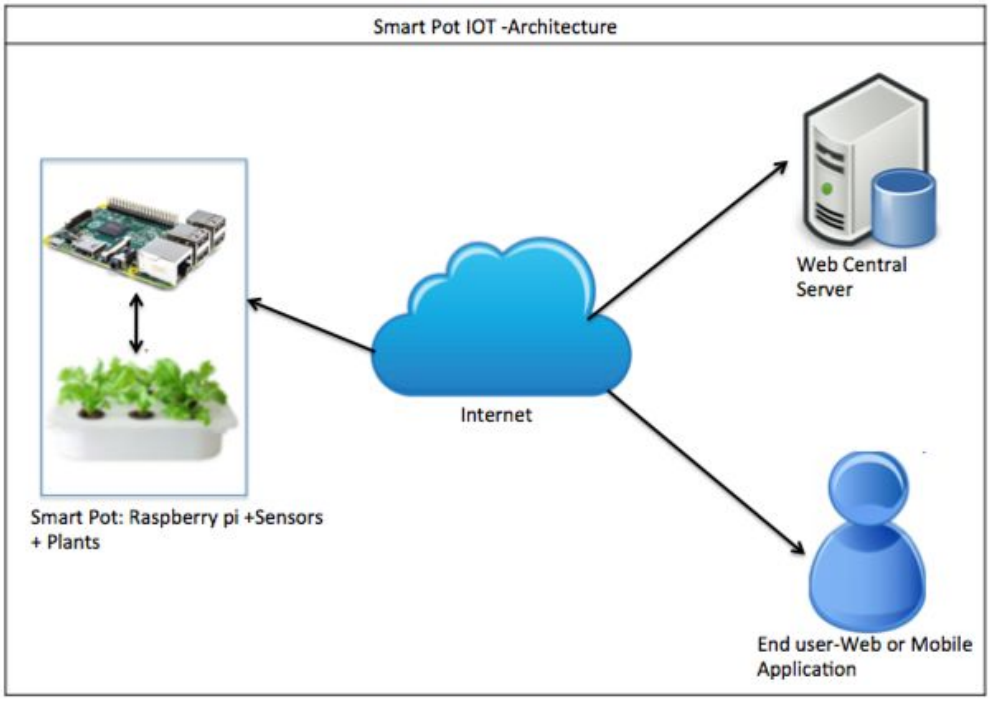
## Project Goal

An innovative solution which will provide people with an automated and economical way to manage their gardens.

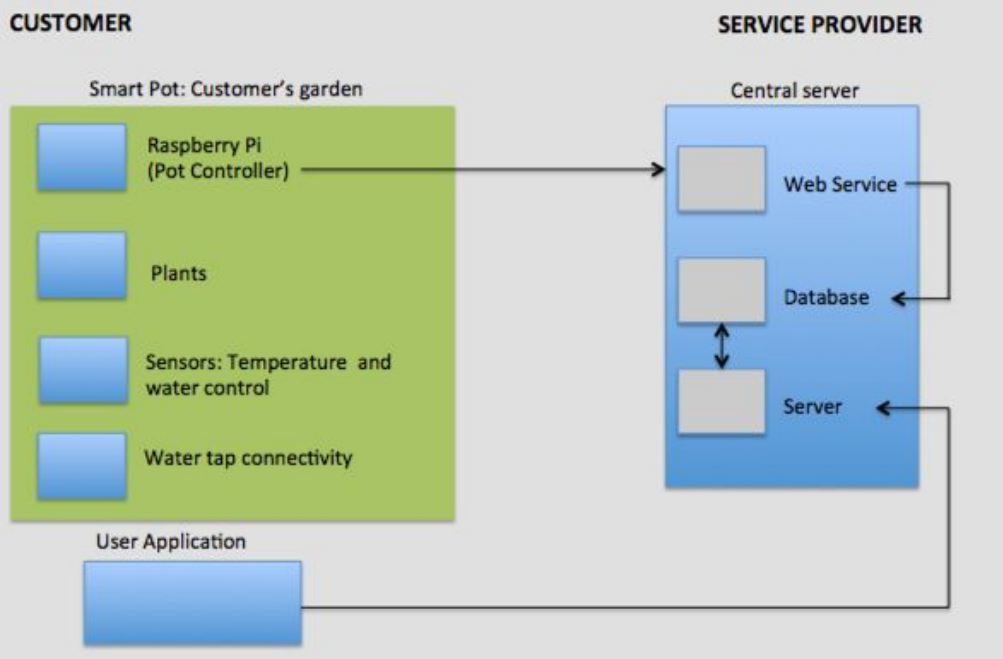
- Focusing on people of big cities with lack of time and space management for their plantations.
- A platform connecting servers,sensors and other tools taking care of water level and weather requirements by using internet services.
- Automatic watering to the plants.
- A web based interface is used to create weather-based irrigation scheduling programs,monitor the growth of the plant by photo capturing,own dashboard of the garden and knowledge sharing repository.

## Product Overview

The smart pot will be based on the standards of IOT and will use a raspberry device as the main controller of the plantation space. The Smart Pot will use an internet connection to access to the central web server which will be responsible for centralizing configurations, statistics, user settings and will provide the main interface for end users. This architecture will give to users the ability to manage intelligently and efficiently all required aspect for growing a plant remotely with minimal interaction.




**Connectivity diagram**




**Following are the main challenges faced by potential customer, who are interested in gardening**

- Know-how about gardening
- Efficient and intelligent way to nourish the plants
- Lack of time to monitor plants growth
- Difficult to get expert advice and connectivity with other people.

## Our Objectives:

**Business Objectives:**

- Provide an innovative and economical solution for taking care of plants, fruits,tree and vegetables.
- Monitor the growth of plants.
- Recommendations for proper growth.
- Sharing of the knowledge and experience between users.

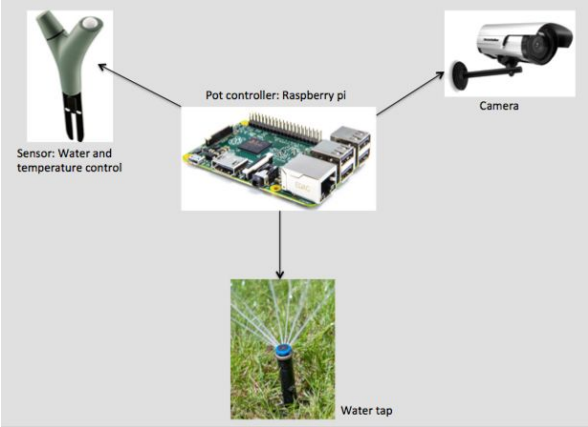
**Technical Objectives:**

- A sensor to measure and monitor the water supplied to the plant.
- Water level data to raspberry server from physical sensors.
- Continuous update of data from raspberry server to Web server.
- Update Weather information from Web server to raspberry and create an alert to the user.
- Web based interface for clients to monitor the growth of plants.

- Implement the safety measures of IoT i.e. Security of the smartpot infrastructure and the user's data.

## Detailed Product description

- *Dashboard represent parameter histories in Chart or graph representation.*
- *Pictures or animation of plantation area.*
- *XML representation of Garden Layout*
- *"Help" integrated in application*
- *Elements of the user interface such as Menu, Dialog box, button,tooltip, warning and error messages.*
- *Summary of alert to user*

<table>
<tr>
<td> 
</td>
<td>Functionalities of Pot controller:<br>
<ul>
<li>Images for the growth of plant</li>
<li>Temperature and water level values through the sensors</li>
<li>Controlling the water resources.</li>
<li>Exchange of data with web server</li>
</ul>
</td>
</tr>
</table>

## Tools and Technology

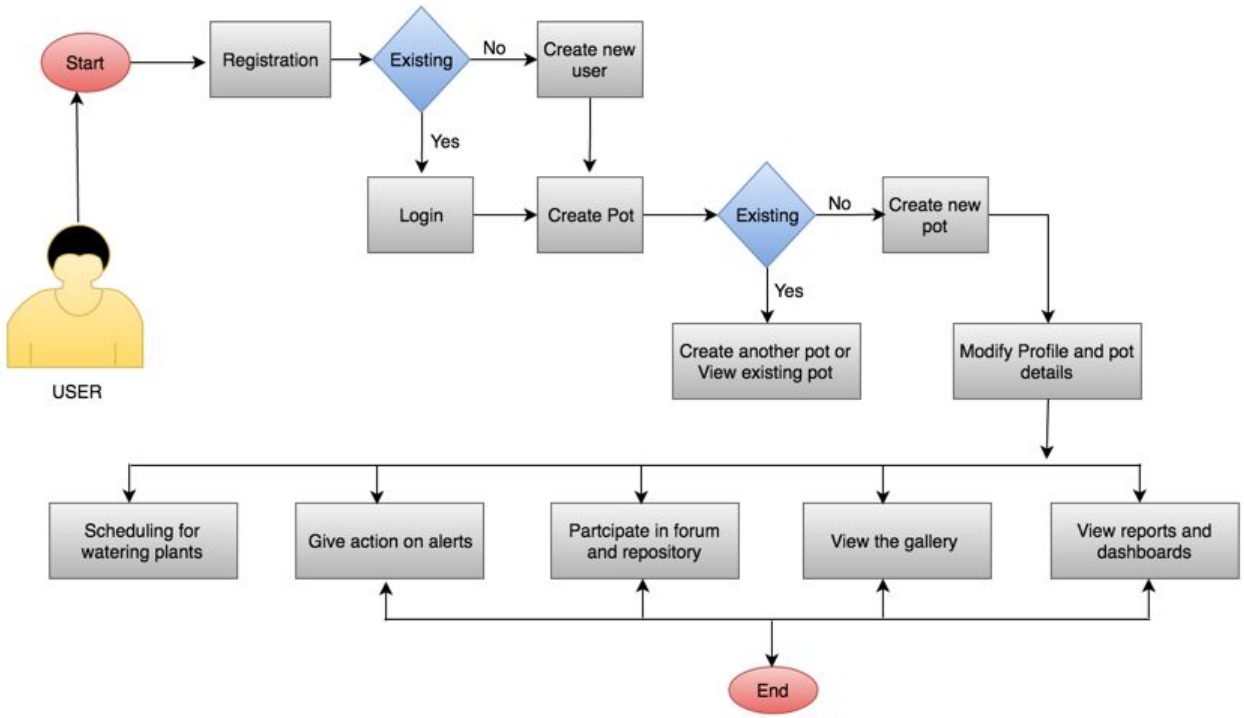
| Project Management | Embedded system | Application Development |
|---|---|---|
| • ProjectLibre<br>• Google Slides<br>• Google Documents<br>• Daily Meeting | • Raspberry Pie (Nano Pi M1)<br>• Sensors (NHT11) : Temperature and humidity<br>• Debian Operating System<br>• Server:Ubuntu<br>• Port knocking<br>• IP Tables<br>• Firewalls<br>• HTPC Server<br>• Certificates for authentication | • IDE : Eclipse Mars<br>• Framework : Maven, Hibernate<br>• Frontend : JavaScripts, CSS, Bootstrap<br>• Database : Mysql<br>• Server : Apache Tomcat 7<br>• DAO Methods<br>• Java programming language<br>• Testing :JUnits |

## Participants and Functionality

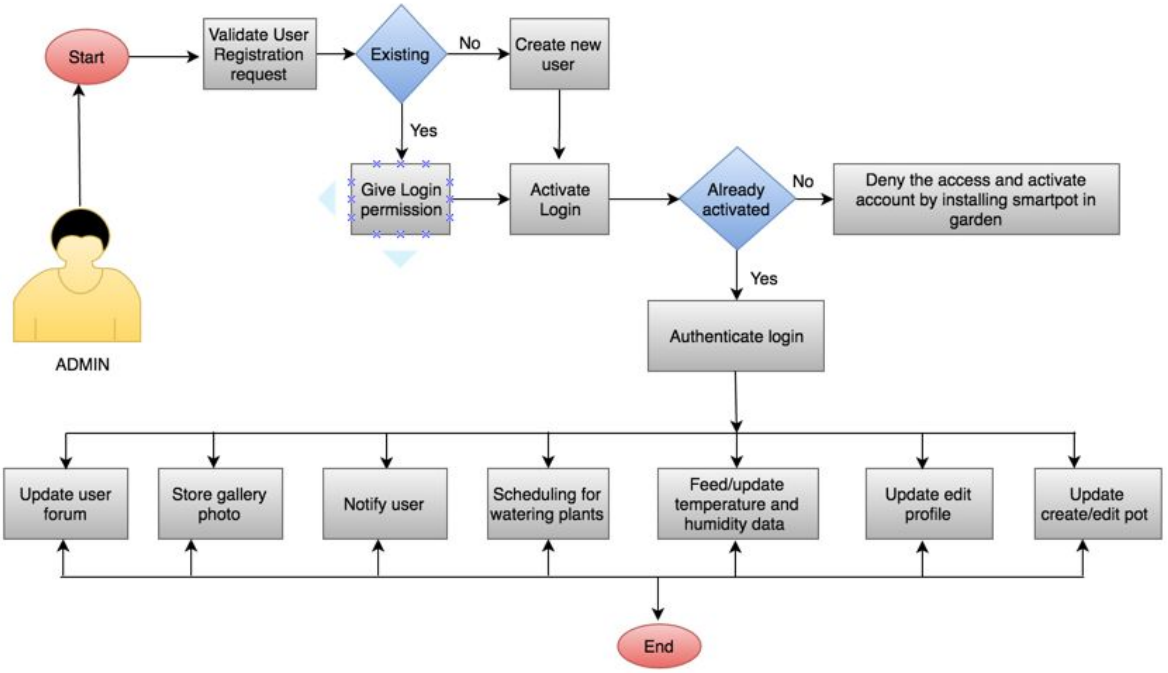
| SMART POT | CENTRAL SERVER | USER |
|---|---|---|
| • Initial setup assistance.<br>• Receive configuration from central server.<br>• Send statistics to central server.<br>• Schedule activities in the pot.<br>• Send alerts to central server.<br>• Receive news from central server.<br>• Controlling devices like camera,tap and sensors. | • Register users.<br>• Receive and Store statistics.<br>• Send alerts to users.<br>• Send news to Pot controller.<br>• Send Configurations to Pot controller.<br>• Continuous communication with smart pot.<br>• Maintain user forum. | • Register account.<br>• Activate garden.<br>• Create pot,select plant or vegetable to grow.<br>• Monitor the single and multiple gardens through camera<br>• Generate dashboards and reports<br>• Share the experience and solutions in forum. |

## Data Flow diagram

## 1. User Side

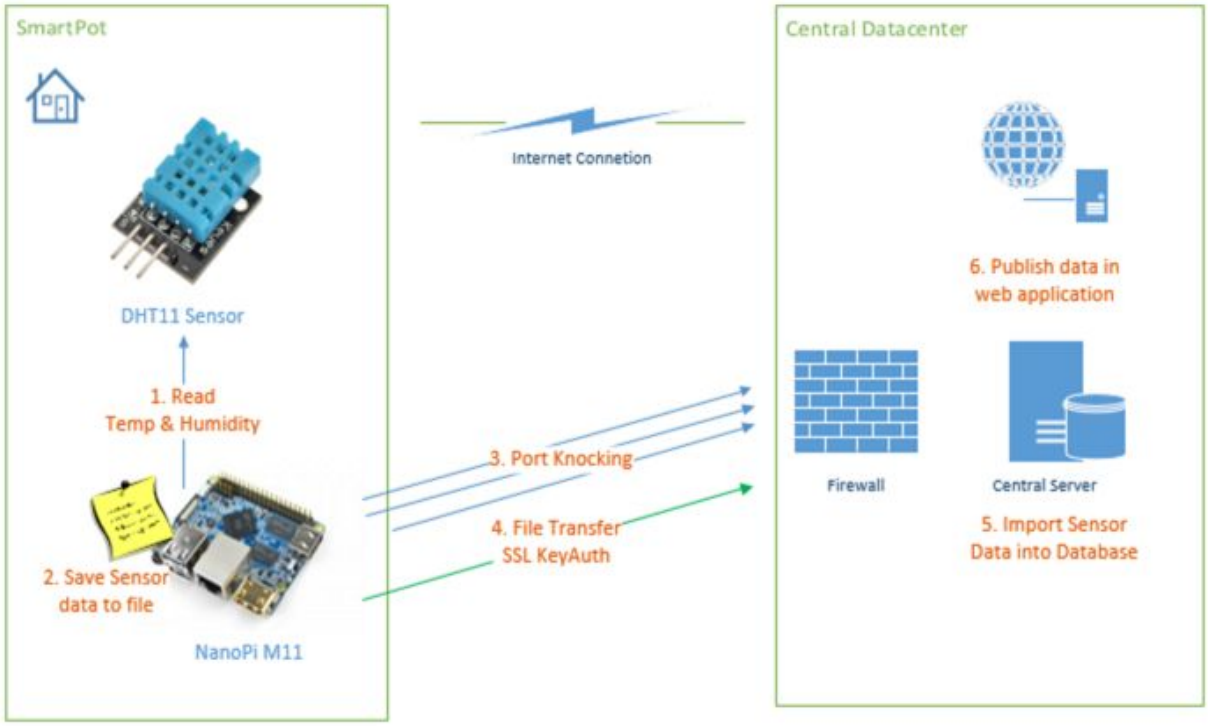



## 2. Admin Side




## Application Development

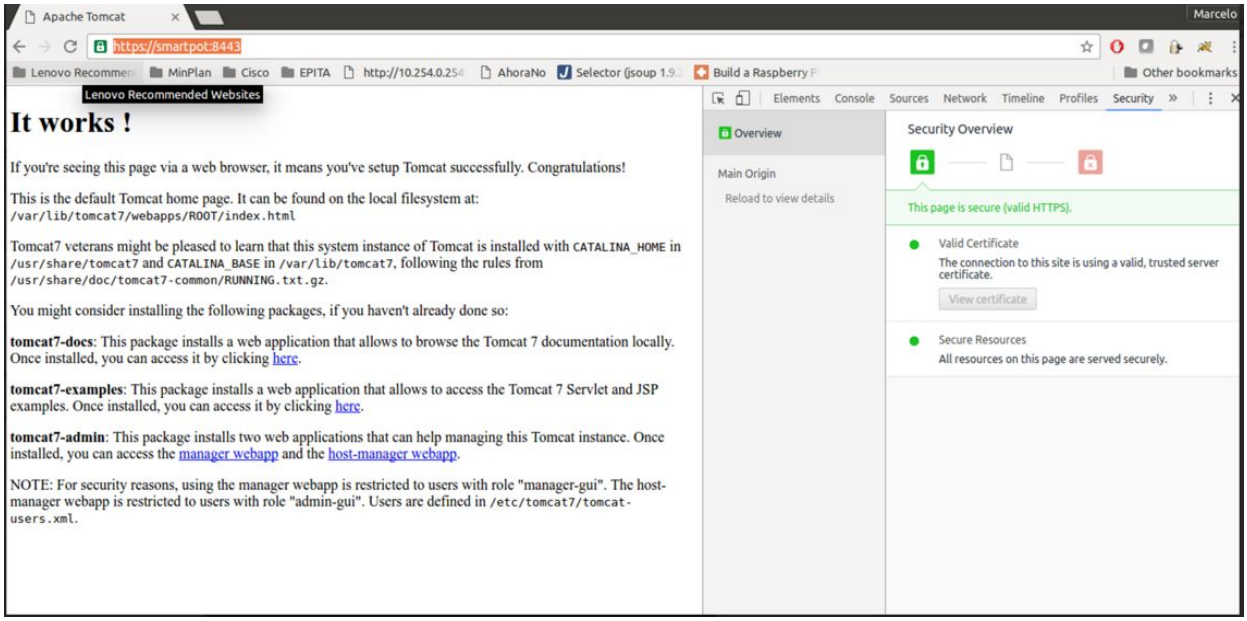
## Smart Pot Communication Flow:Exchange of data



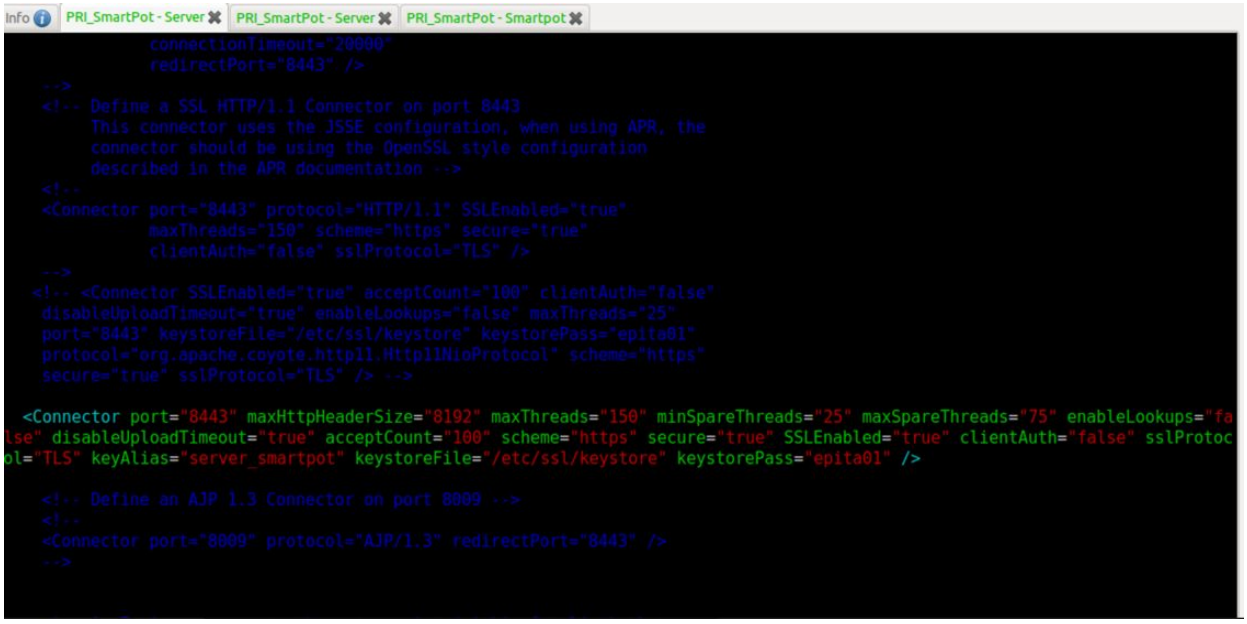
## Smart Pot : Security Issues and Steps

- **Secure Architecture**
- **Secure networking** using IP tables and firewall
- **Port knocking** for managing open connection
- Key authentication for **secure transfer** SSL
- HTTPS Tomcat server with **certificates**
- Operating system hardening
- **Security evaluation**
- Owasp Lapse Project for **secure scanning** of java **applications**

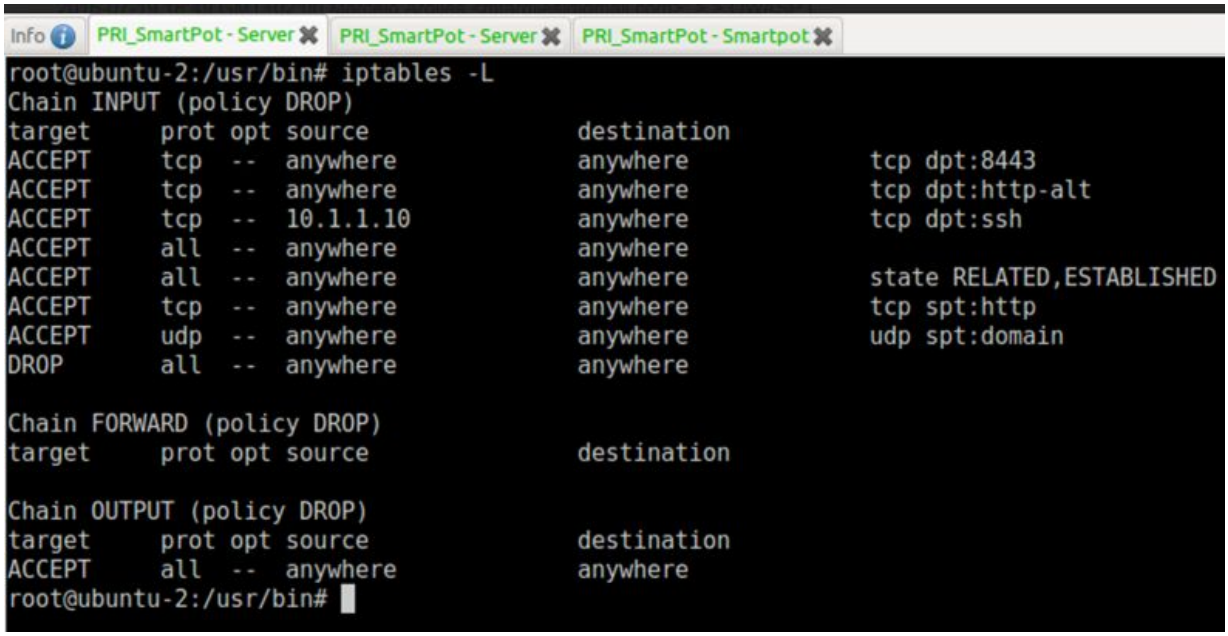
## Smart Pot : HTTPs connection and certificates



## Smart Pot : Configuration with Tomcat server




## Smart Pot : Low Level Programming (Embedding system)



Info | PRI_SmartPot - Server | PRI_SmartPot - Server | PRI_SmartPot - Smartpot

```
#include <stdio.h>
#include <stdlib.h>
#include "libfahw.h"

#define BUF_SIZE            (64)
#define DRIVER_MODULE       "dht11"

int main(int argc, char ** argv)
{
    int ret = -1;
    int dhtTemp=0, dhtHdty=0, board;
    char modStr[BUF_SIZE];
    int pin = GPIO_PIN(7);

    if ((board = boardInit()) < 0) {
        printf("Fail to init board\n");
        return -1;
    }
    if (board == BOARD_NANOPI_T2)
        pin = GPIO_PIN(15);

    sprintf(modStr, "modprobe %s gpio=%d", DRIVER_MODULE, pintoGPIO(pin));
    system(modStr);
    if ((ret = dht11Read(DHT_HUMIDITY, &dhtHdty)) != -1) {
        printf("[humidity:%d,", dhtHdty);
    } else {
        printf("Faided to get humidity\n");
    }
    if ((ret = dht11Read(DHT_TEMP, &dhtTemp)) != -1) {
        printf("temperature:%d", dhtTemp);
    } else {
        printf("Faided to get temperature\n");
    }
    system("rmmod "DRIVER_MODULE);
    return ret;
}
```

```
root@smartpot:/usr/bin# sendfiles.sh
hitting tcp 10.1.1.2:7821
hitting tcp 10.1.1.2:2320
hitting tcp 10.1.1.2:4321
hitting tcp 10.1.1.2:7831
/root/files/073791a1672940d01d07b9e39277f234_01_1970-01-01 08:14:28.txt
073791a1672940d01d07b9e39277f234_01_1970-01-01 08:14:28.txt
hitting tcp 10.1.1.2:7831
hitting tcp 10.1.1.2:4321
hitting tcp 10.1.1.2:2320
hitting tcp 10.1.1.2:7821
root@smartpot:/usr/bin# cat "/root/files/processed/073791a1672940d01d07b
[humidity:36000,temperature:26000,date:1970-01-01 08:14:28]
```

## Risk and Threats

| Risk Assessment Matrix | | | | |
|---|---|---|---|---|
| | Impact | | | |
| Consequence | | Likely | Unlikely | Mitigations |
| | Disconnection between pot controller and central server | | HIGH | Signal loss handled by raspberry pi and notify automatically. |
| | Fault data analysis and exchange | | HIGH | Strong data validation and data integrity. |
| | Hardware failure | HIGH | | Regular check up, auto detection and notification to user and service provider at the time of failure. |
| | User understanding of the system | LOW | | Providing knowledge sharing repository |