Stay Connected

Distributed Active Climate Control System User Console Design

Team 10

Faisal Rahman | Pratik Baniya | Vera Wang | Tejal Padharia

Stay Connected - A Distributed Active Climate Control System as a Service

Professor: Rakesh Ranjan

Students: Vera Wang, Tejal Padharia, Pratik Banya and Faisal Rahma

***Abstract*— According to the World Population Prospect of the United Nations in 2017, the current world population of 7.6 billion is expected to reach 8.6 billion in 2030, 9.8 billion in 2050 and 11.2 billion in 2100 [1]. Additionally, the vast majority of this growth from 2017 to 2050, it is expected that half of the world's population growth will be concentrated in just nine countries mainly in africa and asia [1]. With this increase in population and more percentage of existing population moving into middle class across the developing world results in increase in food demand. However, the ever increasing global temperature and the rise in sea levels are resulting in unpredictable and increasingly severe weather events across the world. This pattern of intense rain and snow storms and periods of drought is becoming the *new normal* in our everyday weather as levels of heat-trapping gases in the atmosphere continue to rise [3]. These adversities severely effects agriculture leading to Crop yields, aquatic populations and forest productivity to decline, invasive insect and plant species to proliferate and desertification, soil salinization and water stress to increase [4]. In the face of these adversities, innovation in how we grow food becomes pertinent and increasingly Indoor farming in the form of Vertical Farming has been garnering a lot of praise in its feasibility to localize and distribute not the food but the production itself. Farming in an indoor environment requires artificial lighting, irrigation, temperature control and nutrients which can be controlled automatically by a WiFi enabled microcontroller with an array of sensors delivering precisely the exact amount of resources each plant needs without any wastage. But, when this system is scaled up to feed the world population it becomes increasingly necessary to find a way to aggregate, control and analyse the information from the large number of interconnected IoT devices. Stay Connected offers a System as a Service for sensor control, data collection and analytics.**

Keywords —Smart devices, Wireless sensor networks, Internet Of Things, System as a Service, Vertical Farming, Pattern Analysis

I. INTRODUCTION

Stay Connected is a Distributed Active Climate Control System as a Service which can be used in a variety of applications including vertical farming, data centers. Users are offered a single page web application which allows them to securely authenticate and access the sensors. The access control offers network and device isolation by role based permission in mainly five categories including create, read, update, delete and operate.

II. DESIGN

Our design is a Distributed Active Climate Control System. In the data centers the server racks are laid out in grids with cooling units and lighting units (for maintenance and operation). A climate control device like fan can be turned on and off based on temperature. User Interface perspective our task is to design an intuitive and productive user interface for two main user roles through persona development, empathy mapping and requirement analysis. DACCS deployment topology consists of multiple **Sites** (Physical Sites) each consisting of multiple **Locations**(Buildings), each consisting of multiple **Zones** (NxM grid of sensors) which in turn is made of multiple **Nodes** (Climate Controller Sensor Node, one per control element).

Furthermore, the service offers a framework which provides integration and analytics of 3rd party sensors. As an sample, we have integrated 3rd party sensors from NOAA using their RESTFul Data APIs. The system allows for the capture of useful data including temperature, humidity, latitude, longitude and are used in analytics and generate pattern graphs showing high and low temperature, change in temperature, etc.

A. *Design Methodology*

For this paper we have used Human Centric Design which starts with identification of personas of the systems users and their roles definition. Then an empathy map is created based on what they might think, feel, see, hear, say, and want. This is followed by an interview with a sample user, which in our case for us was a Site Operations Engineer. Finally, a sketch of the User Interface was developed and exported to Invision App for usability testing. This methodology helped us identify user expectations before development of user models, permission models, sensor network models were ever taken on.

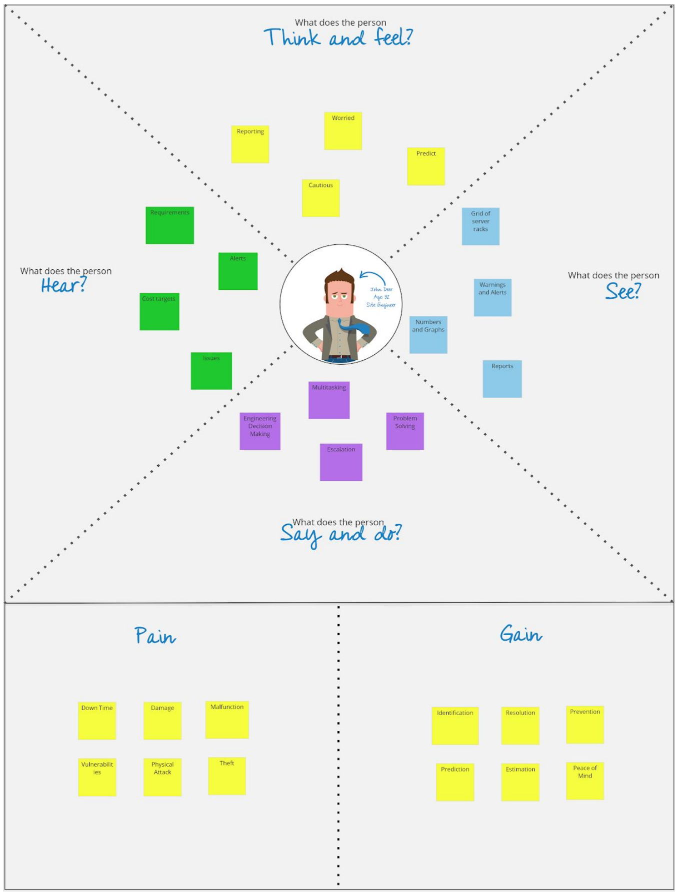


Figure 1: Empathy Map

B. Sensor

For the individual node, a DHT-11 temperature and humidity module was used along with the Arduino UNO controller to detect the temperature and humidity. We have also added 3V DC with fan blade to implement the automatic cooling feature. When the temperature or humidity reaches a threshold, the fan will be turned on automatically. In addition, the user is able to control the fan of each node in the dashboard. There are three additional conditions: voltage, current and apparent power sending from each node for monitoring. We use Emon library to extract the power usage.

Each Arduino sends data every 30 seconds to the master node through wifi ESP8266 module. Each node_id is identified using the MAC address of the wifi board. We use socket.io for data transmission between the Arduino and the server. The data is sent as a JSON object and stored in the database on the server side. An user could onboard each node on the dashboard to monitor the data. Since all the hardware pieces are low-cost, our distributed system can apply for a wide variety of use in the industry.

We have included virtual sensor onboarding scheme in the system due to the possible hardware failure and data inaccuracy. We have added third-party API such as NOAA Web Service API where virtual sensors could be subscribed by the user. The user is able to add a node in the dashboard, where a list of sensors are presented for the user to subscribe. Once the user subscribes the sensor, the node will be added to the dashboard for future monitoring. This feature provides flexibility and accuracy of data for many uses.

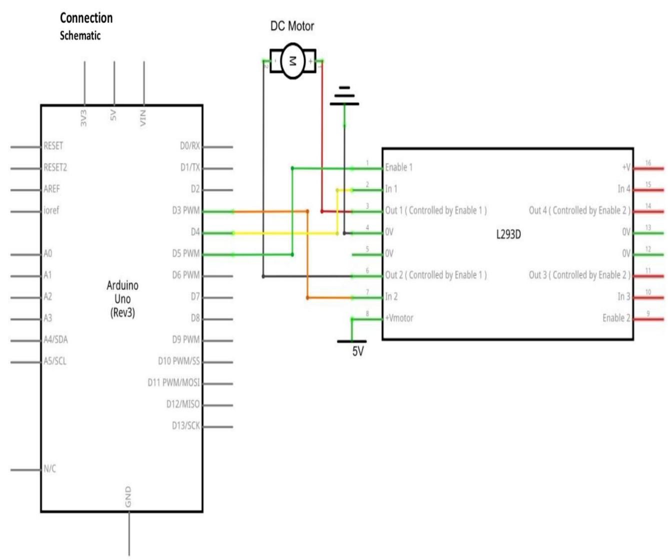


Figure 2. Arduino connection with DC motor

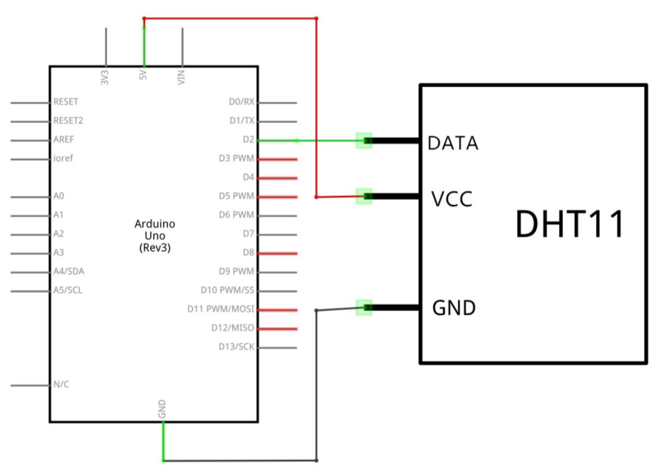


Figure 3. Arduino connection with DHT-11 sensor

C. Database

Containerized cloud hosted PostgresDB instance with TimescaleDB plugin for Timeseries dataset in EC2 AutoScaling Group within ECS Cluster.

DDL:

```
create table site (
	site_id integer not null
		constraint site_pkey
			primary key,
	site_name varchar(45) not null,
	site_address varchar(256)
);

create table location (
	location_id integer not null
		constraint location_pkey
			primary key,
	location_name varchar(45),
	location_address varchar(256),
	site_id integer not null
		constraint fk_location_site
			references site);

create unique index
location_location_id_key
	on location (location_id);

create table zone (
	zone_id integer not null
		constraint pk_zone_id
			primary key,
	zone_name varchar(45) not null,
	zone_address varchar(256) not null,
	location_id integer not null
		constraint fk_location_id
			references location);

create unique index zone_zone_id_key
	on zone (zone_id);

create table node (
	node_id integer not null
		constraint node_pkey
			primary key,
	node_address varchar(256),
	zone_id integer not null
		constraint fk_zone_id
			references zone,
	node_name varchar(256) not null,
	status integer default 0
);

create unique index node_node_id_key
	on node (node_id);

create table condition (
	read_time timestamp with time zone
not null
		constraint condition_pkey
			primary key,
	temp double precision,
	humidity double precision,
	voltage double precision,
	current double precision,
	var double precision,
	node_id integer not null
		constraint fk_node_id
			references node );

create index condition_read_time_idx
	on condition (read_time);

create table permissions (
role_id integer not null
		constraint permissions_pkey
			primary key,
	permissions jsonb,
	role_name varchar(256) not null
);

create table "user"(
user_id varchar(128) not null
		constraint user_pkey
			primary key,
	first_name varchar(256) not null,
	last_name varchar(256) not null,
	title varchar(256) not null,
	role_id integer not null
		constraint fk_role_id
			references permissions
);

create unique index
permissions_role_id_uindex
	on permissions (role_id);
SELECT create_hypertable('conditions',
'time', 'location', 4);
```

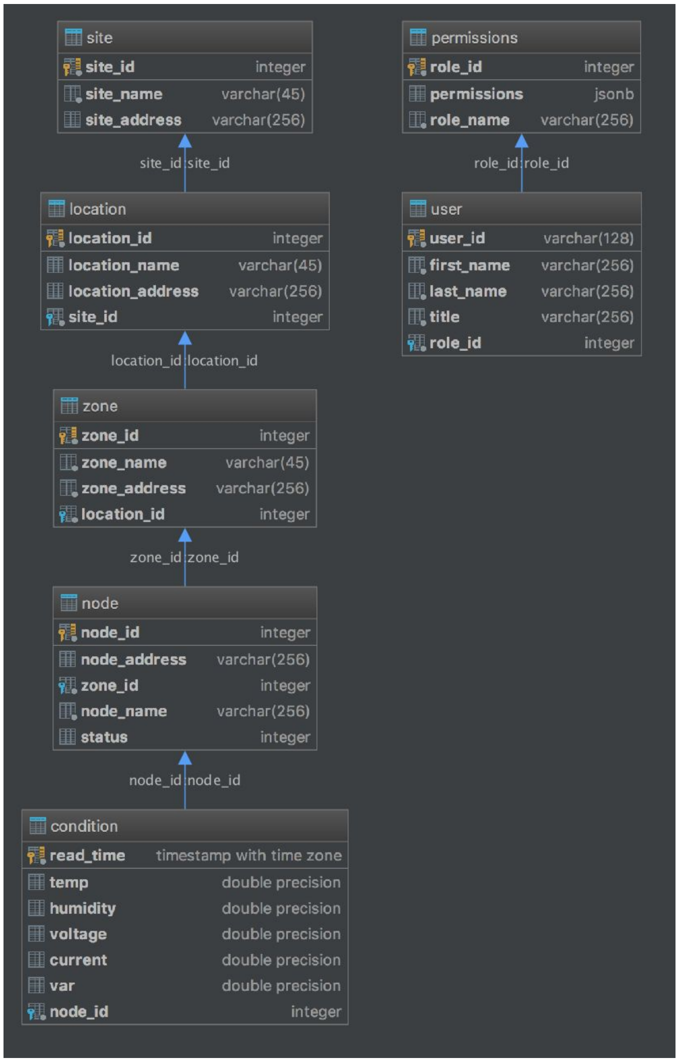


Figure 4: Database Diagram

D. Authentication and Access Control

To make the application easily accessible we have implemented Single sign on using google account. This feature has been implemented using google firebase Authentication.

The reason behind using this is that firebase provides backend services, easy to use SDK's and ready to use User Interface libraries that can be used to authenticate a user. Access Tokens received after the user logs in are send to the server via HTTPS headers. The Access Id is authenticated in our custom made server using **Firebase Admin Auth SDK.**

Permissions from any of five categories Create Read Update Delete Operate and are added to each object accessible to user.

E. Stay Connected Web Service

Stay Connected web service offers RESTful APIs for data access and Sensor control in compliance with RESTful design principles. The containerized Expressjs service is hosted in AWS in EC2 Auto Scaling group as an ECS cluster.

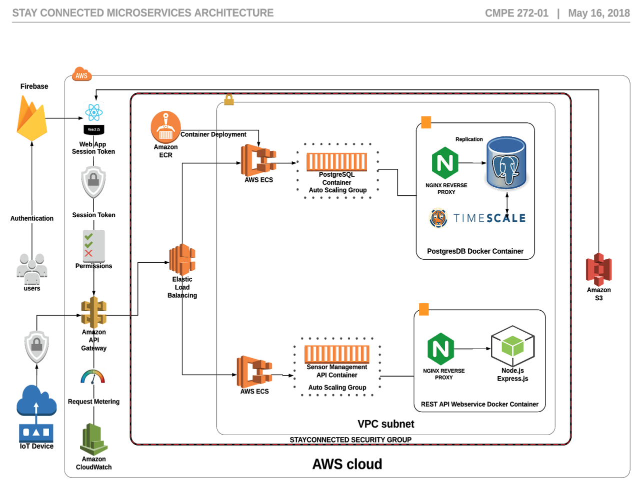


Figure 5. System Architecture

F. User Interface

The application provides user-friendly interface, where an user is able to monitor nodes in such an order: sites->locations->zones->nodes. In the navigation bar, the user is able to select each field individually and monitor real-time data along with graphical analysis. Each user has different privilege of access control and after logging in, the dashboard will only present subscribed nodes according user's privilege. The user is able to navigate all the information in a smooth order.

III. Math

A. Equations

```
while(DHT.humidity >= thresholdHumidity
|| DHT.temperature >= thresholdTemperature)
{
turnOnFan();
delay(10000);
chk = DHT.read11(DHT11_PIN);
sendToWifi();
}
turnOffFan(); // disable fan
```

```
void turnOnFan(){
    digitalWrite(ENABLE, HIGH);
    digitalWrite(DIRA,LOW);
    digitalWrite(DIRB,HIGH);
}

void turnOffFan(){
    digitalWrite(ENABLE, LOW);
}
```

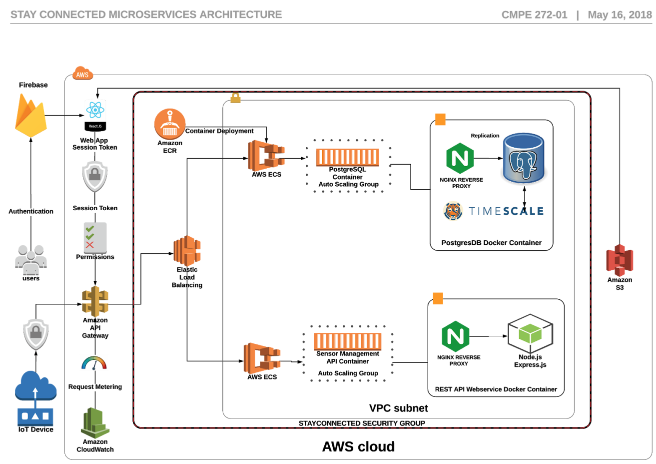
IV. Units

Temperature - Celsius
Humidity - Percentage water per cubic meter of air
AC Current - Ampere
AC Voltage - Volt
AC Active Power - (voltage-ampere) Watt
AC Reactive Power - (voltage-ampere-reactive) var

V. Conclusion

Due to the emerging technology of IoT and increasing needs for scalable and low-budget application, we have delivered a Distributed Active Climate Control System as a Service called Stay Connected. We have combined low-budget hardware pieces with third-party virtual sensors to provide a framework with data management, integration and analysis. An user is able to onboard either type of sensors for specific needs, monitor and control the entire system with user-friendly web application. Each node updates data to the server periodically, in addition, it contains a cooling device which can be controlled automatically or manually by the user. Furthermore, the service provides analysis based on historical data for user observation. The system is highly scalable and consistent to help users to achieve business success with reasonable cost.

VI. Appendix



VII. References

[1]New Plant Growth Technology Could Alleviate Climate Change And Food Shortage
[2]Thompson, B., Cohen, B., Ebooks Corporation, Cohen, Marc J, SpringerLink, & LINK. (2012). The impact of climate change and bioenergy on nutrition (Higher Education: Handbook of Theory and Research). Dordrecht ; New York : Rome: Springer ; FAO.
[3]https://www.ucsusa.org/global-warming/science-and-impacts/impacts/global-warming-rain-snow-tornadoes.html
[4] https://www.ncdc.noaa.gov/cdo-web/webservices/v2
[5] https://www.elegoo.com/
[6] https://github.com/openenergymonitor/EmonLib
[7] https://firebase.google.com/