



About Mi Garden

Growing your own produce is a rewarding and simple pastime. Lots of people have indoor greenhouses and enjoy growing simple herbs or fruit in their own kitchen. With busy lives, and often unsuitable growing environment, the success of peoples' attempts is subject to much trial and error.

This IOT project is to create a sensor-driven system allowing users to monitor and control indoor growing conditions. By using sensors for light, heat and humidity the systems sends this information to a website interface allowing the user can access from anywhere. Alerts can be sent to the users' email or Twitter if the temperature gets too high or too low. Using their phone the user should be able to turn on a fan and adjust the lighting.

All of this information gets stored in a database to allow the user to review and refine their growing habitat leading to a more successful and rewarding growing experience. As much as the platform is aimed at kitchen gardeners its design should allow for scalability e.g., monitoring and controlling multiple greenhouses.

Tools, Technologies & Equipment

The tools and hardware required are:

Raspberry Pi + Sense Hat: Sense Hat comes with humidity and temperature sensors as well as LEDs which is perfect for synthesizing the environment for this prototype.

Mobile Phone: This is going to be the main point of interaction for the user.

Fan (5V): Connected the Raspberry Pi – user should be able to turn on and off remotely.

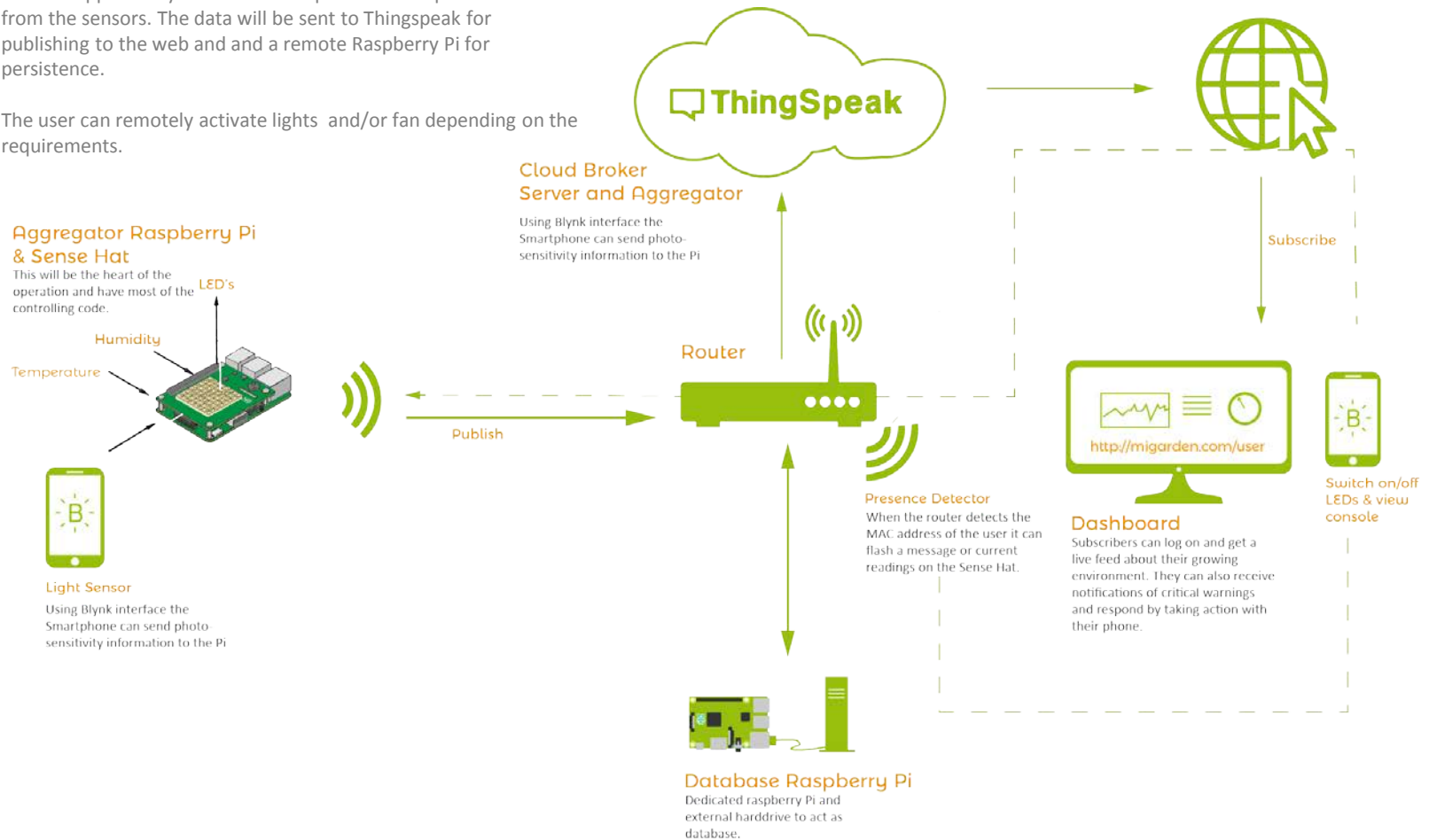
The technologies:

- Wi-Fi – used to transmit data from Pi
- Thingspeak – Data Aggregator and MQTT
- Blynk – Mobile interface for interacting with the platform
- JavaScript & Json – Handling data and programming events
- HTML & CSS – to style the online dashboard
- Cloud Storage – To store the database information

Pictorial Representation

This IOT app uses Python for the scripts used to capture the data from the sensors. The data will be sent to Thingspeak for publishing to the web and a remote Raspberry Pi for persistence.

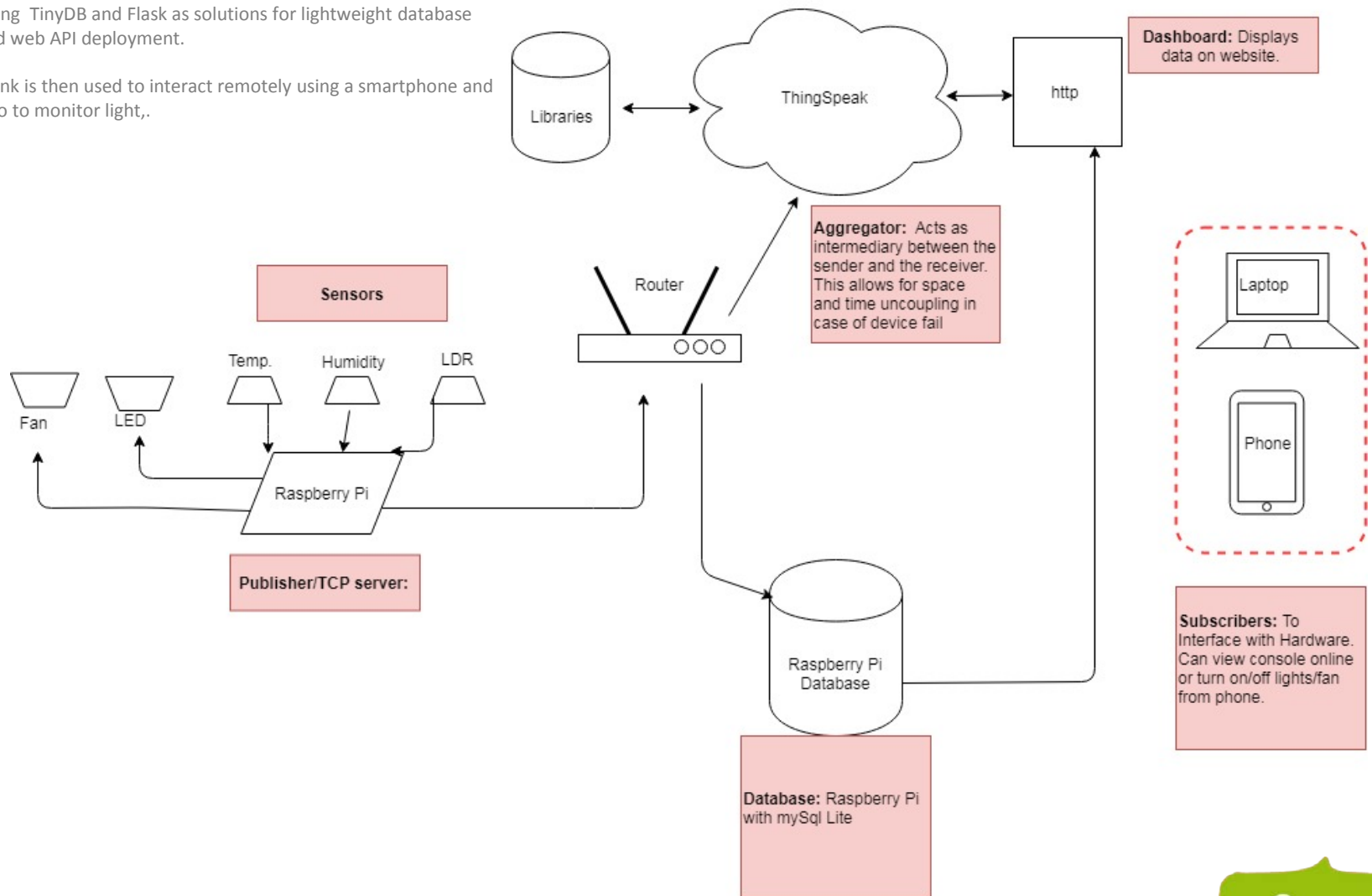
The user can remotely activate lights and/or fan depending on the requirements.



Informational Flow Diagram

Using TinyDB and Flask as solutions for lightweight database and web API deployment.

Blynk is then used to interact remotely using a smartphone and also to monitor light,.



Logic Flow Chart

