2020

Smart Garden System

ARCHITECTURE DOCUMENT

JEREMIAH SMITH

Introduction

The Smart Garden System is designed to modernize the household garden. The system will simplify maintaining a garden by automating maintenance tasks such as watering. The system will also be recording metrics including soil moisture, soil temperature, environmental temperature, water distributed, plant yield, direct sunlight, and direct shade; these metrics will allow the system to adapt and optimize the garden.

Requirements

Functional Requirements

Soil Gardens

The system will interface with sensors monitoring to soil-based plants to gather datapoints for soil moisture, soil temperature, humidity levels, barometric air pressure, environmental temperature and light levels.

The system will interface with actuators that will control the release of water, activating light sources and temperature control.

Hydroponic Gardens

The system will interface with sensors monitoring hydroponic plants to gather datapoints for water temperature, water pH levels, environmental temperature and light levels.

The system will interface with actuators that will control the flow of water, activating light sources and temperature control.

General

The system will have a control system that will display relevant garden statistics such as current sensor readings.

The system will have an interface that allows the user to configure garden systems.

Non-Functional Requirements

The system should be able to handle twenty concurrent messages at one time.

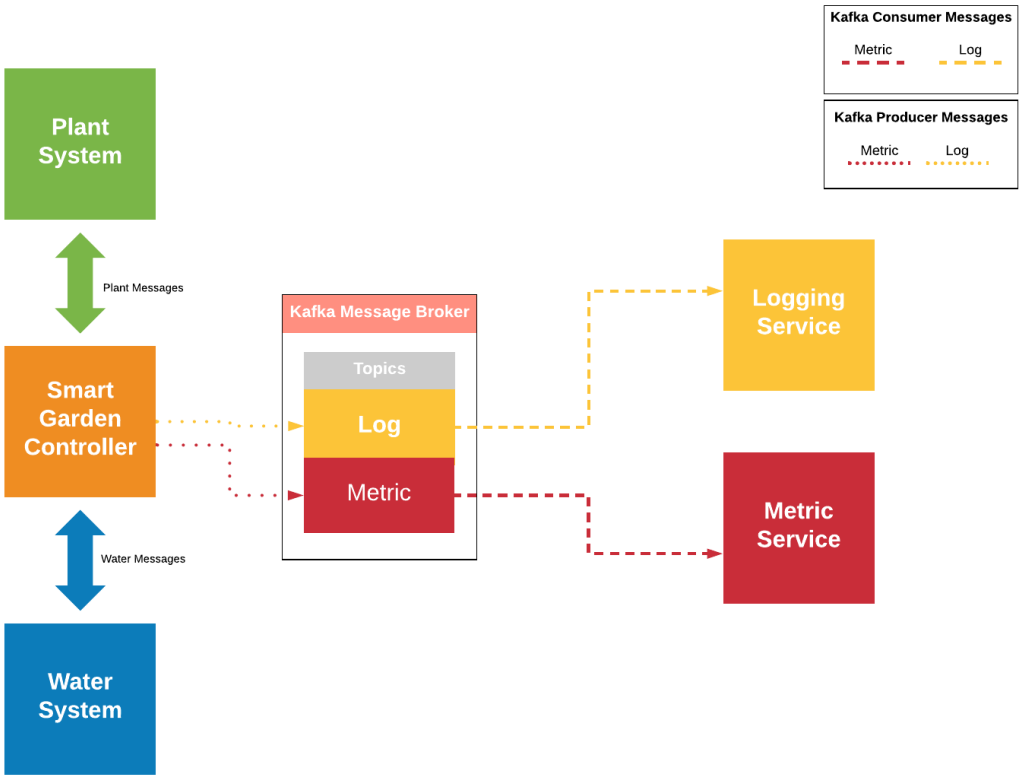
The system should have a throughput of forty requests per minute

The system should be able to support ten concurrent users.

The system will produce fifty gigabytes of data per year.

The system should not have a message loss over one percent.

Overall Architecture



Components

Logging Service

The logging service will be responsible for collecting, indexing and presenting log messages for all components in the system. Logging is a vital service in all applications it enables error tracking, usage statistics and transaction tracing.

Message Broker

The system will make use of a Message Broker to record logging and metric information messages; as well as pass sensor readings to the garden controller.

Metric Service

The metric service will be responsible for recording system statics such as CPU usage, Sensor Reads, Actuator activations and data layer statistics.

Smart Garden Controller

The smart garden controller is the information hub of the system. It will handle the displaying of information to the users of the smart garden system. It will be a will aggregate all information for each component in a unified display.

Plant Controller

The plant system will interface with all the sensors and actuators for the plants in the smart garden system. The system will have the ability to configure sensors, actuators and system properties.

Water Controller

The water system will interface with all the sensors and actuators for the watering system. The system will have the ability to configure sensors, actuators and system properties.

Component Architecture

Logging Service

Role

Logging is an important aspect of any software system. It enables transaction tracing, software failure identification and error correction. This service will be the aggregator of all logs with the system. It will store, index and present log messages.

Technology Stack

The logging service will use a tech stack commonly referred to as ELK (Elastic search, Kibana and Log Stash).

Kibana

This is the presentation layer of the technology stack. It allows building dashboards, viewing log messages and filtering.

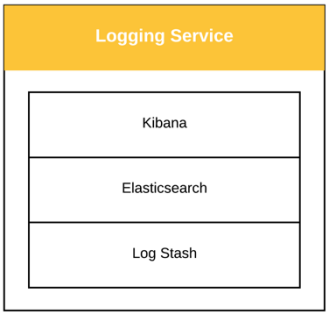
Elastic Search

This is indexing and searching layer of the technology stack. It will index log messages.

Log Stash

This is the storage layer of the technology stack. It is responsible for storing application log messages for indexing.

Architecture



Metric Service

Role

Collecting metrics is vital when identifying system load. The metric service will aggerate all system metrics such as CPU utilization, memory usage and network traffic. This service will help with identifying hardware usage and help identify places where the system can be improved. Future use of this data will also help in hardware failure predictions and maintenance scheduling.

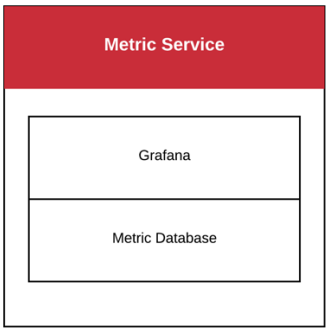
Technology Stack

The stack will be a commonly used opensource application called Grafana with a MySql datastore.

Grafana

This is the presentation layer of the service, it allows for building dashboards and viewing metric data in a friendly way.

Architecture



Message Broker

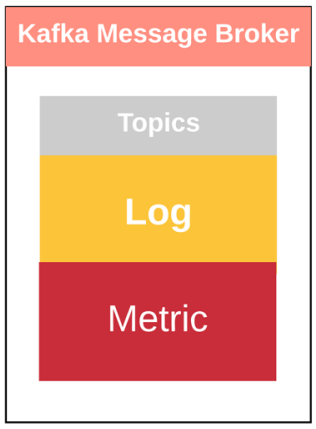
Role

The message broker is used to pass messages form the producers to consumers. In the case of this system the Smart Garden controller will produce messages that contain log messages and metrics. The Logging Service and Metric Service will consume the messages. The message broker allows for passing and process of data asynchronously.

Technology Stack

Apache Kafka will be used as the message broker in the system. Apache Kafka is a real-time, distributed data streaming platform. It allows for streams of data to be published and subscribed to similar to other message queues.

Architecture



Smart Garden Controller

Role

The Smart Garden Controller will act as the central hub of the of the system. It will provide user access to the system, be responsible for interfacing with the various plant, water and environmental control systems.

Technology Stack

User Interface

The user interface with be a web application created using the Angular Framework. There will also be mobile applications in Android and IOS.

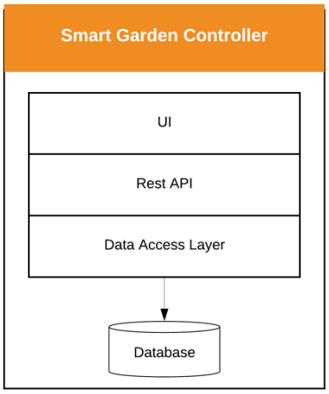
API

This UI will interface with the system through a Restful Web API. This API will be written with Java and the Spring Framework.

Data Store

The smart garden controller will store the information on a MySql database.

Architecture



Plant Controller

Role

The Plant controller will be responsible for communicating with the various sensors and actuators related to plant management. The system will allow for the controller to be configured via a user interface.

Technology Stack

User Interface

The UI will be a web application written with the Angular Framework and will operate as an Electron application.

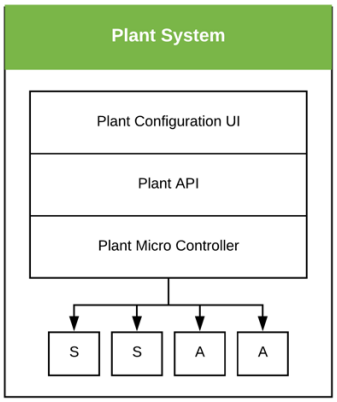
API

The UI will interact with the micro controller via a Restful API written in JavaScript using Node.js and Express Framework.

Microcontroller

The system will run on a Raspberry PI microcontroller. The Raspberry PI will serve the Application Interface and UI. There will be a series of sensors and actuators that will communication with Arduino Microcontrollers that will be connected to the Raspberry Pi GPIO pins.

Architecture



Water Controller

Role

The Water Controller will be responsible for communicating with the various sensors and actuators related to water system management. The system will allow for the controller to be configured via a user interface. The water controller will vary for Soil and Hydroponic gardens.

Technology Stack

User Interface

The UI will be a web application written with the Angular Framework and will operate as an Electron application.

API

The UI will interact with the micro controller via a Restful API written in JavaScript using Node.js and Express Framework.

Microcontroller

The system will run on a Raspberry PI microcontroller. The Raspberry PI will serve the Application Interface and UI. There will be a series of sensors and actuators that will communication with Arduino Microcontrollers that will be connected to the Raspberry Pi GPIO pins.

Architecture