Software Requirements Specification

for

SmartAgCloud

Version 1.0

Prepared by

Priya Yadav

Jessica Mathias

Ankita Chikodi

Priya Khadke

San Jose State University

19th Feb 2019

Table of Contents

Table of Contents ii

Revision History iii

1. Introduction 1

1.1 Purpose 1

1.2 Intended Audience 1

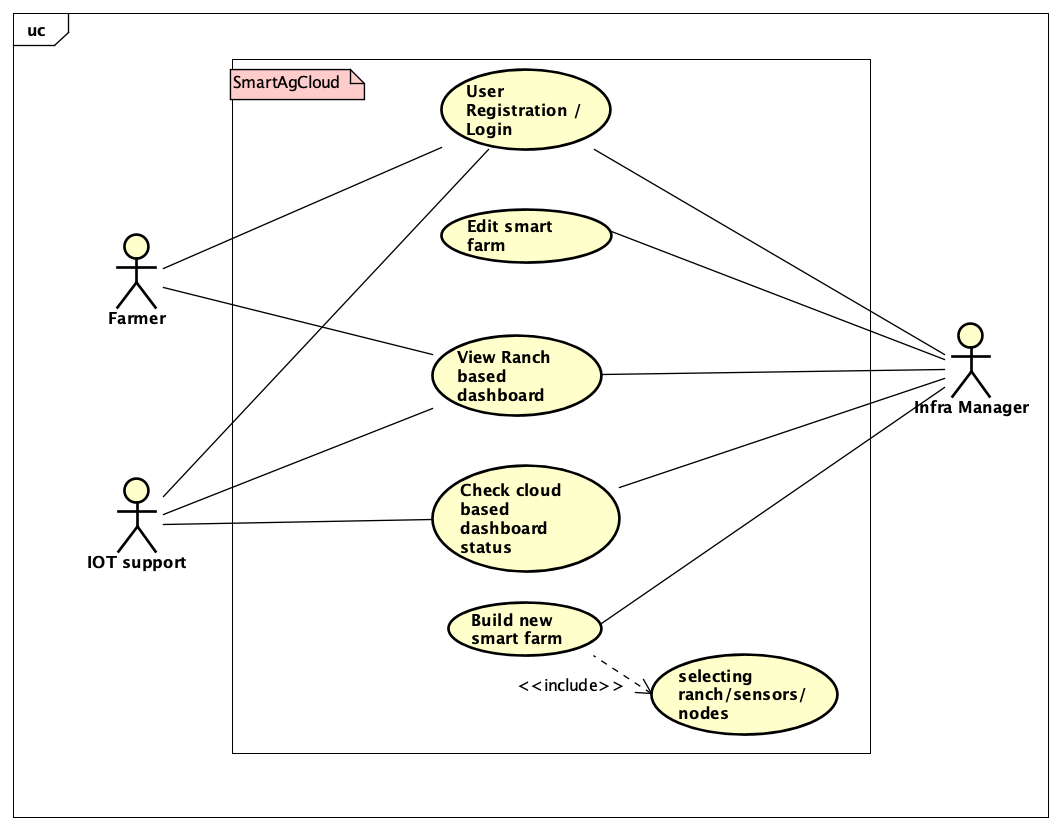
1.3 Application Scope 1

2. Overall Description 2

2.1 Product Perspective 2

2.2 Product Functions 2

2.3 User Classes and Characteristics 3

2.5 Operating Environment 4

2.4 Design and Implementation Constraints 4

2.5 User Documentation 4

2.6 Assumptions and Dependencies 5

3. External Interface Requirements 5

3.1 User Interfaces 5

3.2 Hardware Interfaces 5

3.3 Software Interfaces 5

3.4 Communications Interfaces 6

4. System Features 6

4.1 User Registration 6

4.2 User Login 6

4.3 Build Smart Farm 7

4.4 Sensor Data Management 8

4.5 Cloud Based Sensor Data Management 8

4.6 View Ranch-based dashboard 9

4.7 View Cloud-based dashboard 10

5. Other Nonfunctional Requirements 11

5.1 Performance Requirements 11

5.2 Safety Requirements – Reliability 11

5.3 Security Requirements 11

5.4 Software Quality Attributes 11

6. System infrastructure 12

7. System architecture and Component interaction design 12

8. Plan and schedule 12

PERT chart with ind component ownership and dates 12

9. Technology Selection 13

9.1 Cloud Provider 13

9.2 Back End Server Technology 13

9.3 MQTT 13

9.4 MySQL 13

9.5 Kafka- data bus option for scalable data transfer with increased number of sensors– (pending PoC) 13

9.6 Front-end JavaScript 13

9.7 Mobile App Android (Solution using mobile app builder) 13

9.8 Agriculture sensor simulation software 13

9.9 Abbreviation 13

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason for Changes** | **Version** |
|  |  |  |  |
|  |  |  |  |

# Introduction

## Purpose

<make sure to include the cloud benefits like reduced cost due to reuse, elastic dynamic good coverage, reliability etc>benefits over normal sensor networks

This document describes the system requirements specification for an IOT-based smart agriculture infrastructure service management system as a service on a cloud namely SmartAgCloud. The document will discuss in detail the requirements and high-level design of the application. The document includes the details on defining scope along with assumptions and dependencies. The IOT based application will be designed and implemented for ranch field farms.

The application It supports large-scale on-demand IOT-based agriculture system infrastructure services for farmers. Each farmer could select install and deploy one or more IOT agriculture networks for their ranch fields.

## Intended Audience

This document is intended to be used by farmers, IoT managers, IoT supporters, business stakeholders, developers, architect and testers. This document can also be helpful for individual looking for smart farming solutions in agriculture domain. The document includes high level application understanding including product scope and functionality. It is recommended to follow the sections in order as presented by document to get high level understanding followed by low level details.

## Application Scope

SmartAgCloud is a cloud service solution for smart agriculture which helps farmers to monitor farms which are remotely located; thereby saving time and cost involved. The SmartAgCloud application is expected to achieve goals of implementing the solution on cloud and leverage the cloud benefits of OnDemand resource allocation, run time scaling, reduced cost, improved coverage. Below are the key features in scope for the SmartAgCloud.

1. Builds smart farm to monitor below parameters,

* Temperature,
* Humidity,
* Wind Speed
* Wind Direction
* Rain
* Measures Soil Moisture

The parameters will be monitored by simulated software. This software is expected to simulate below two real types of sensors,

* AcuRite
* Soil Moisture Meter Probe - 24 Inch
* Raspberry Pi

Out of Scope

Real sensors are out of scope of this application.

# Overall Description

## Product Perspective

Technologies and IoT have the potential to transform agriculture in many aspects. Namely, there are five ways IoT can improve agriculture:

Data, tons of data, collected by smart agriculture sensors, e.g. weather conditions, soil quality, crop’s growth progress or cattle health. This data can be used to track the state of your business in general, as well as staff performance, equipment efficiency, etc.

Better control over the internal processes and, as a result, lower production risks. The ability to foresee the output of your production allows you to plan for better product distribution. If you know exactly how much crops you are going to harvest, you can make sure your product won’t lie around unsold.

benefits-of-smart-farming

Cost management and waste reduction thanks to the increased control over production. Being able to see any anomalies in the crop growth or livestock health, you will be able to mitigate the risks of losing your yield.

Increased business efficiency through process automation. By using smart devices, you can automate multiple processes across your production cycle, e.g. irrigation, fertilizing, or pest control.

Enhanced product quality and volumes. Achieve better control over the production process and maintain higher standards of crop quality and growth capacity through automation.

As a result, all of these factors can eventually lead to higher revenue.

Now that we have outlined how IoT can be advantageously applied in the sphere of agriculture, let’s take a look at how the listed benefits can find their application in real life.

## Product Functions

Your ***SmartAgCloud*** is a IOT-based smart agriculture infrastructure service management system on a cloud to provide farmers with on-demand large-scale services in building their own IOT infrastructure networks by booking, configuration, installation, and deployment of desired IOT-based agriculture sensor networks in their ranches.

The key features of the application are stated below:

1. Monitors the agricultural conditions using data feed from smart sensors at real time
2. Allows farmers to configure smart ranch fields within farmlands.
3. Dashboard on web and mobile facilitate real time monitoring

* Allows farmers to select their own desired ranches within farmland
* Book the resources (different types of sensors) (sensor profile should include sensor type, name, value data type).

<Summarize the major functions the product must perform or must let the user perform. Details will be provided in Section 3, so only a high-level summary (such as a bullet list) is needed here. Organize the functions to make them understandable to any reader of the SRS. A picture of the major groups of related requirements and how they relate, such as a top-level data flow diagram or object class diagram, is often effective.>

## User Classes and Characteristics

**System users:**

This system includes the following types of users:

* **Farmers** – They can access the system to check and configure their own IOT networks with

smart nodes (with diverse sensors) online by accessing SmartAgCloud to check their IOT sensor status

and statistic data,

* **IOT Supports** – They can access the system to find out the status of smart nodes and sensors.
* **Infrastructure Managers** – The persons who setup, configure, and manage smart nodes, cluster nodes,

and sensors, as well as their connectivity. In addition, they could monitor and track the status of their smart

nodes with diverse sensors.

## 2.5 Operating Environment

Linux

## Design and Implementation Constraints

<Describe any items or issues that will limit the options available to the developers. These might include: corporate or regulatory policies; hardware limitations (timing requirements, memory requirements); interfaces to other applications; specific technologies, tools, and databases to be used; parallel operations; language requirements; communications protocols; security considerations; design conventions or programming standards (for example, if the customer’s organization will be responsible for maintaining the delivered software).>

<NO real sensors used -only simulation software used for project implementation>

## User Documentation

<List the user documentation components (such as user manuals, on-line help, and tutorials) that will be delivered along with the software. Identify any known user documentation delivery formats or standards.>

## Assumptions and Dependencies

<List any assumed factors (as opposed to known facts) that could affect the requirements stated in the SRS. These could include third-party or commercial components that you plan to use, issues around the development or operating environment, or constraints. The project could be affected if these assumptions are incorrect, are not shared, or change. Also identify any dependencies the project has on external factors, such as software components that you intend to reuse from another project, unless they are already documented elsewhere (for example, in the vision and scope document or the project plan).>

<real sensors underlying in the field are connected with low network>

# External Interface Requirements

## User Interfaces

<Describe the logical characteristics of each interface between the software product and the users. This may include sample screen images, any GUI standards or product family style guides that are to be followed, screen layout constraints, standard buttons and functions (e.g., help) that will appear on every screen, keyboard shortcuts, error message display standards, and so on. Define the software components for which a user interface is needed. Details of the user interface design should be documented in a separate user interface specification.>

## Hardware Interfaces

<Describe the logical and physical characteristics of each interface between the software product and the hardware components of the system. This may include the supported device types, the nature of the data and control interactions between the software and the hardware, and communication protocols to be used.>

<list of sensors> -- tentative – with cost

## A close up of a map Description automatically generatedSoftware Interfaces

## Communications Interfaces

<Describe the requirements associated with any communications functions required by this product, including e-mail, web browser, network server communications protocols, electronic forms, and so on. Define any pertinent message formatting. Identify any communication standards that will be used, such as FTP or HTTP. Specify any communication security or encryption issues, data transfer rates, and synchronization mechanisms.>

# System Features

## User Registration

|  |  |
| --- | --- |
| **Use case:** User Registration | |
| **Actors:** Farmer/IoT Manager /Infrastructure Manager | |
| **Purpose: Register a user in SmartAgCloud Application** | |
| **Overview: Allows users to register in SmartAgCloud and assign respective roles to the users** | |
| **Type: Essential** | |
| **Preconditions**: NA | |
| **Postconditions:** The user must be able to register to the SmartAgCloud application via web or mobile  App and appropriate role must be assigned. | |
| **Special Requirements: NA** | |
| **Flow of Events** | |
| **Actor Action** | **System Response** |
| 1. This use case begins after the new user opens the SmartAgCloud site or mobile application and selects Register User option 2. Users fills in the registration form and registers to the system. | 1. System prompts the new user registration form.  2. The application will save the user details and assigns  appropriate roles (from Farmer/IoT Manager /  Infrastructure Manager) to the user. |
| 1. **Alternative Flow of Events** |  |
| Line 1: If the user is already registered, the user can login to the application by selecting login option. | |

## User Login

|  |  |
| --- | --- |
| **Use case:** User Login | |
| **Actors:** Farmer/IoT Manager /Infrastructure Manager | |
| **Purpose: Logins a user in SmartAgCloud Application** | |
| **Overview: Allows users to login in SmartAgCloud and use application as per assigned role** | |
| **Type: Essential** | |
| **Preconditions**: The user must be registered in the application | |
| **Postconditions:** The user must be able to login to the SmartAgCloud application via web or mobile  App and appropriate role must be assigned. | |
| **Special Requirements: NA** | |
| **Flow of Events** | |
| **Actor Action** | **System Response** |
| 1. This use case begins after the new user opens the SmartAgCloud site or mobile application and selects Login option 2. Users enters the username and password and logins to the system. | 1. Application prompts the to enter username and password.  2. The application will authenticate the user details, and  Displays the home page. |
| 1. **Alternative Flow of Events** |  |
| Line 1: if username/password is invalid, return error message "Invalid username/password" to Output  and terminate this Use Case. | |

## Build Smart Farm

|  |  |
| --- | --- |
| **Use case:** Build smart farm | |
| **Actors:** Infrastructure Managers | |
| **Purpose:** Build smart farms to enable monitoring via sensors within selected farm areas | |
| **Overview: Allows *Infrastructure Managers* to build the one or more smart ranches within his**  **farm with farmers choice of sensors** | |
| **Type: Essential** | |
| **Preconditions**: The users are registered in SmartAgCloud application as an Infrastructure Managers | |
| **Postconditions:** The smart farm is setup by Infrastructure Managers by the choice of farmer is able to view  the selected farms in the dashboard | |
| **Special Requirements:**   1. ***Cluster node management*** – It allows Infrastructure Managers to add/update/delete/view cluster nodes for an agriculture IOT sensor network, and track cluster node status. 2. ***Smart node management –*** It allows Infrastructure Managers to add/update/delete/view smart nodes   controlled by a cluster, and track node status. | |
| **Flow of Events** | |
| **Actor Action** | **System Response** |
| 1. This use case begins after the Infrastructure Managers logins to the SmartAgCloud application with his username and password and selects Build Smart Farm option 2. Infrastructure Managers selects one ranch at a time from the provided farm map and then selects the smart sensors and required numbers of sensors 3. Infrastructure Managers may choose to build multiple unique ranches within this use case | 1. Verify the entered username and password   and allow user to login on authentication.   1. Triggers request to IoT managers to deploy and   validate the connections of the sensors. Display the selected smart ranch details and smart sensors details on farmers profile. |
| 1. **Alternative Flow of Events** |  |
| Line 1: if username/password is invalid, return error message "Invalid username/password" to Output  and terminate this Use Case. | |

## Sensor Data Management

|  |  |
| --- | --- |
| **Use case:** Sensor Data Management | |
| **Actors:** IOT Manager | |
| **Purpose:** To allow managers to manage the sensors data and connectivity. | |
| **Overview: Allows IoT manager to update sensors data reading, sensor’s profile and retrieve**  **sensors data for a specified time interval** | |
| **Type: Essential** | |
| **Preconditions**: The user is registered in SmartAgCloud application with IoT manager role | |
| **Postconditions:** The IOT agricultural data manager is able to update sensors data reading, add/delete/  update the sensor’s profile and retrieve the sensor’s data. The updated sensor profile is available for  farmers to build the smart farm. | |
| **Special Requirements:** none | |
| **Flow of Events** | |
| **Actor Action** | **System Response** |
| 1. This use case begins after the farmer logins to the SmartAgCloud application with his username and password and selects Build Smart Farm option 2. Farmer selects one ranch at a time from the provided farm map and then selects the smart sensors and required numbers of sensors 3. Farmer may choose to build multiple unique ranches within this use case | 1. Verify the entered username and password and  allow user to login on authentication.  2. Display the selected smart ranch details and smart c  Sensors details on farmers profile. |
| 1. **Alternative Flow of Events** |  |
| Line 1: if username/password is invalid, return error message "Invalid username/password" to Output  and terminate this Use Case. | |

## Cloud Based Sensor Data Management

|  |  |
| --- | --- |
| **Use case:** Cloud-based sensor data management | |
| **Actors:** An IOT agriculture data manager | |
| **Purpose:** To allow managers to manage the sensors data and connectivity. | |
| **Overview: Allows IoT manager to update sensors data reading, sensor’s profile and retrieve**  **sensors data for a specified time interval** | |
| **Type: Essential** | |
| **Preconditions**: The user is registered in SmartAgCloud application with IoT manager role | |
| **Postconditions:** The IOT agricultural data manager is able to update sensors data reading, add/delete/  update the sensor’s profile and retrieve the sensor’s data. The updated sensor profile is available for  farmers to build the smart farm. | |
| **Special Requirements:** none | |
| **Flow of Events** | |
| **Actor Action** | **System Response** |
| 1. This use case begins after the farmer logins to the SmartAgCloud application with his username and password and selects Build Smart Farm option 2. Farmer selects one ranch at a time from the provided farm map and then selects the smart sensors and required numbers of sensors 3. Farmer may choose to build multiple unique ranches within this use case | 1. Verify the entered username and password and  allow user to login on authentication.  2. Display the selected smart ranch details and smart c  Sensors details on farmers profile. |
| 1. **Alternative Flow of Events** |  |
| Line 1: if username/password is invalid, return error message "Invalid username/password" to Output  and terminate this Use Case. | |

## View Ranch-based dashboard

|  |  |
| --- | --- |
| **Use case: View ranch-based dashboard of the monitored smart farm** | |
| **Actors:** Farmer/IoT Manager | |
| **Purpose:** To allow farmers to view real time status of monitored smart field via dashboard on Mobile app  Or web interface | |
| **Overview: Allows farmers to view ranch-based dashboard via mobile app or online interface** | |
| **Type: Essential** | |
| **Preconditions**: The user is registered in SmartAgCloud application with farmer role | |
| **Postconditions:** The farmer is able to view ranch-based dashboard for all built smart farms. | |
| **Special Requirements:** none | |
| **Flow of Events** | |
| **Actor Action** | **System Response** |
| 1. This use case begins after the farmer logins to the SmartAgCloud application with his username and password  2. Farmer selects view dashboard option.  3. Farmer selects a ranch to monitor | 1. Verify the entered username and password and   allow user to login on authentication.   1. Displays the dashboard for logged in farmer’s   profile   1. Displays the farm stats for the set period of interval the selected |
| **Alternative Flow of Events** | |
| Line 1: if username/password is invalid, return error message "Invalid username/password" to Output  and terminate this Use Case.  Line 2: Farmer may edit the default the time duration for the dashboard; the system will display the stats  As per edited duration. | |

## View Cloud-based dashboard

|  |  |
| --- | --- |
| **Use case: View Cloud-based dashboard of the monitored smart farm** | |
| **Actors:** Infrastructure /IoT Manager | |
| **Purpose:** To allow Infrastructure /IoT Manager to view real time status of all monitored smart field via  dashboard on Mobile app Or web interface | |
| **Overview: Allows** Infrastructure /IoT Manager **to view cloud-based dashboard via mobile app or**  **web interface** | |
| **Type: Essential** | |
| **Preconditions**: The user is registered in SmartAgCloud application with Infrastructure /IoT Manager role | |
| **Postconditions:** The Infrastructure /IoT Manager is able to view ranch-based dashboard for all  built smart farms and devices | |
| **Special Requirements:** none | |
| **Flow of Events** | |
| **Actor Action** | **System Response** |
| 1. This use case begins after the user logins to the SmartAgCloud application with his username and password  2. user selects view dashboard option.  3. user selects a ranch to monitor | 1. Verify the entered username and password and   allow user to login on authentication.   1. Displays the dashboard for logged in user   profile   1. Displays the farm stats for the set period of interval the selected |
| **Alternative Flow of Events** | |
| Line 1: if username/password is invalid, return error message "Invalid username/password" to Output  and terminate this Use Case.  Line 2: user may edit the default the time duration for the dashboard; the system will display the stats  As per edited duration. | |

## Receive Alerts

|  |  |
| --- | --- |
| **Use case: Receive alerts on monitored** | |
| **Actors:** Farmer /IoT Manager | |
| **Purpose:** Notify Farmer/Infrastructure Managers of any action required  on all monitored smart field via dashboard on Mobile app or web interface | |
| **Overview: Allows Infrastructure /IoT Manager to receive alerts on monitored smart farms in case**  **of emergency** | |
| **Type: Essential** | |
| **Preconditions**: The smart farm must be monitored, and users must be registered as farmers/IOT manager | |
| **Postconditions:** Farmers/IoT Managers are notified of any action required or monitored  parameter crossing minimum/maximum threshold | |
| **Special Requirements:** none | |
| **Flow of Events** | |
| **Actor Action** | **System Response** |
| 1. This use case begins after the user logins to the SmartAgCloud application with his username and password  2. user selects view dashboard option.  3. user selects a ranch to monitor | 1. Verify the entered username and password and    1. allow user to login on authentication. 2. Displays the dashboard for logged in user    1. profile 3. Displays the farm stats for the set period of interval the selected |
| **Alternative Flow of Events** | |
| Line 1: if username/password is invalid, return error message "Invalid username/password" to Output  and terminate this Use Case.  Line 2: user may edit the default the time duration for the dashboard; the system will display the stats  As per edited duration. | |

# Other Nonfunctional Requirements

## Performance Requirements

<If there are performance requirements for the product under various circumstances, state them here and explain their rationale, to help the developers understand the intent and make suitable design choices. Specify the timing relationships for real time systems. Make such requirements as specific as possible. You may need to state performance requirements for individual functional requirements or features.>

## Safety Requirements – Reliability

<Specify those requirements that are concerned with possible loss, damage, or harm that could result from the use of the product. Define any safeguards or actions that must be taken, as well as actions that must be prevented. Refer to any external policies or regulations that state safety issues that affect the product’s design or use. Define any safety certifications that must be satisfied.>

## Security Requirements

<Specify any requirements regarding security or privacy issues surrounding use of the product or protection of the data used or created by the product. Define any user identity authentication requirements. Refer to any external policies or regulations containing security issues that affect the product. Define any security or privacy certifications that must be satisfied.>

<include user auth details>

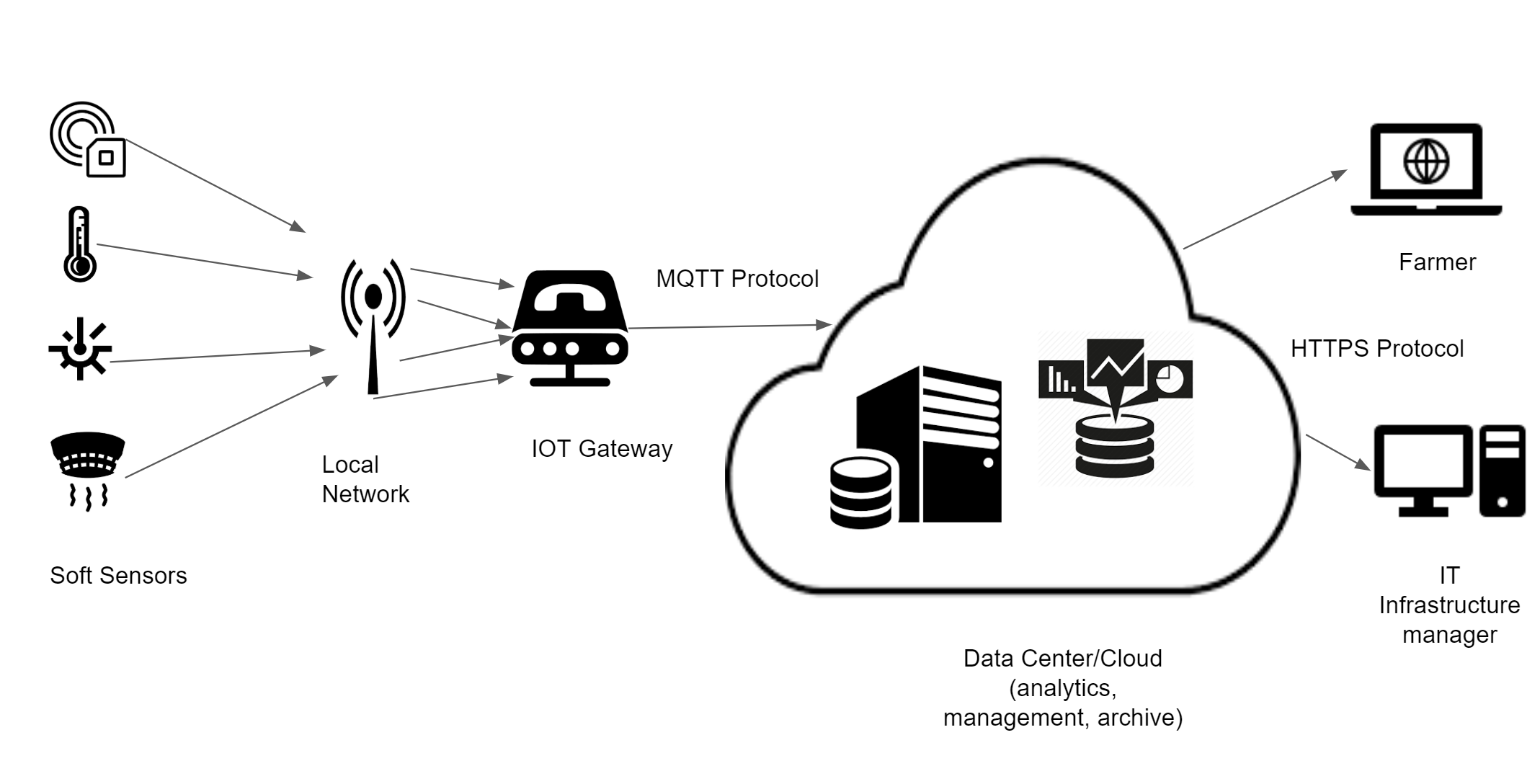
## Software Quality Attributes

<Specify any additional quality characteristics for the product that will be important to either the customers or the developers. Some to consider are: adaptability, availability, correctness, flexibility, interoperability, maintainability, portability, reliability, reusability, robustness, testability, and usability. Write these to be specific, quantitative, and verifiable when possible. At the least, clarify the relative preferences for various attributes, such as ease of use over ease of learning.>

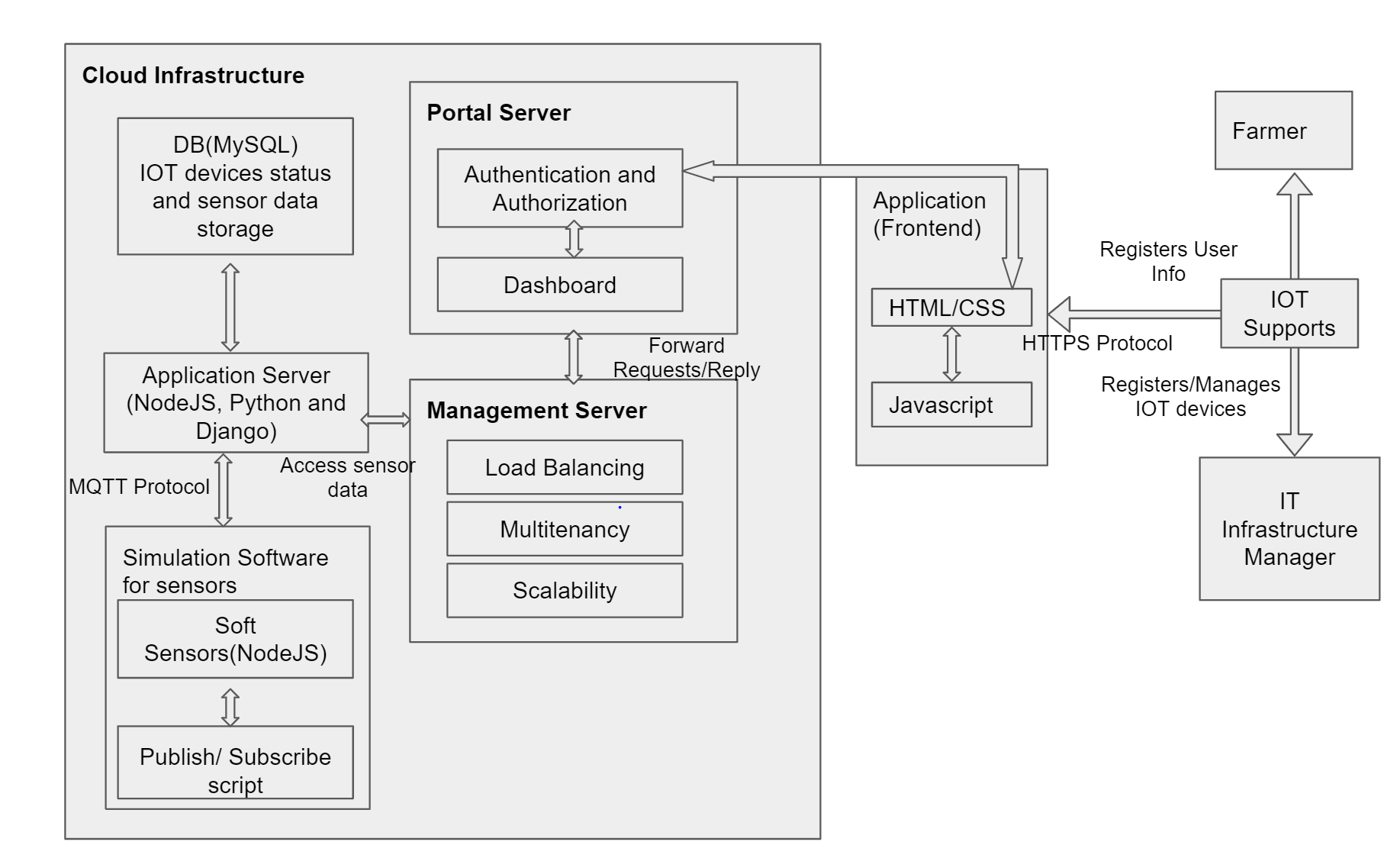
# System infrastructure

# 

# 



# System Architecture and Component interaction design



The overview of our SmartAg architecture consists of 2 components namely cloud infrastructure and entities.

The three entities are farmer, IOT supports and IT Infrastructure manager

1. Farmer: They access the system to get updates and notifications about SmartAg through the dashboard
2. IOT Supports:They monitor the status of smart nodes and sensors through the customized dashboard
3. Infrastructure Managers – They register/manage smart sensor nodes for the system through the dashboard

The cloud infrastructure consists of components such as simulation software, application server, management server, portal server and a database.

1. Simulation software for Sensors: This software simulates the data generated by various sensors in the network at real time and passes it on to the cloud database
2. Databases: The database used here is the Dynamo DB which is a part of the AWS IOT infrastructure. The sensor data sent from the simulation software is handled by the application server which then stores it in the Database. Whenever there is a request from the entities or notification to be sent to the entities, the application server hits the database, retrieves the information and sends it across to the management server and portal server
3. Management server: The management server handles load balancing when there is increased demand for sensor data information, resource pooling and scalability(addition and deletion of sensors based on how the farmers need)
4. Portal Server: The portal server consists of the dashboard and user authentication. The farmer, IOT support or IOT Manager can access the application through the dashboard and customize their dashboard as per their needs

# Plan and schedule

# PERT chart with ind component ownership and dates

# Technology Selection

|  |  |  |  |
| --- | --- | --- | --- |
| **Technology** | **Name** | **Version** | **Remarks** |
| Language | NodeJS, Java, SQL, Python |  | Programming language |
| Communication Protocol | MQTT |  | To get data from devices |
| UI | HTML, CSS, JS |  |  |
| DB | MySQL |  |  |
| MQTT Broker | Mosquitto |  |  |
| Data bus | Kafka |  | Real Time Data Transfer |
| Cloud | AWS | NA | To build cloud infrastructure |
| Sensor Simulation | NodeJS |  |  |
| Mobile App Development | Android Dev Studio |  |  |

## Cloud Provider

SmartAgCloud application is a solution on cloud, for which AWS (Amazon Web Service) compute service will be leveraged. AWS offers good range of instances. AWS will also provide the real time scaling of configured instances and backend with dynamic change in traffic using application load balancer.

## Back End Server Technology

NodeJS

## MQTT

MQTT offers below benefits,

1. MQTT has good integration with NodeJS (the backend logic)
2. MQTT is performance efficient the monitored data is received single connection

## MySQL

## Front-end JavaScript

## Mobile App Android

## Abbreviation

|  |  |  |
| --- | --- | --- |
|  |  |  |
| 1 | MQTT | Message Queuing Telemetry Transport |
| 2 | AWS | Amazon Web Services |
| 3 | APK |  |
|  |  |  |