**SERIS**

Solar Energy Research Institute Singapore



Cloud Based Real-time Analytical Monitoring of Photovoltaic Systems and Weather Parameters Project

High Level Design (HLD)

|  |  |
| --- | --- |
| Filing Reference | SE25PT7SERIS/TECH/DESIGN/HLD/HLD.doc |
| Document Title | High Level Design (HLD) |
| Version | 1.3 |
| Author | Treza Bawm Win |
| Date Created | 14/03/2018 |

|  |  |  |
| --- | --- | --- |
| **Approved by:** | | |
| Name | Designation | Date |
|  |  |  |
| **Authorized by:** | | |
| Name | Designation | Date |
|  |  |  |

**Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Revision** | **Date** | **Author** | **Description** |
| 1.0 | 14/03/2018 | Treza Bawm Win | Initial version |
| 1.1 | 25/03/2018 | Treza Bawm Win | * Product vision * Multitier arch * Technologies stack * Performance |
| 1.2 | 22/04/2018 | Treza Bawm Win | Update the design diagram   * Add AWS Kinesis and S3 * Remove SNS * Add overview use-case diagram |
| 1.3 | 22/04/2018 | Treza Bawm Win | Update as per review comments. |
| 1.4 | 05/05/2018 | Treza Bawm Win | Add AWS Container session for Docker. |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **For Internal use** | | **Date** | **Department** |
| Authorized By |  |  |  |
| Released By |  |  |  |

Table of Contents

[1 Introduction 4](#_Toc513298945)

[1.1 Product Perspective 4](#_Toc513298946)

[2 System Architecture 5](#_Toc513298947)

[2.1 End to End architecture in diagram 5](#_Toc513298948)

[2.2 Multitier Architecture 6](#_Toc513298949)

[2.3 Technologies 6](#_Toc513298950)

[2.3.1 AWS IoT 6](#_Toc513298951)

[2.3.2 AWS Lambda 6](#_Toc513298952)

[2.3.3 DynamoDB 6](#_Toc513298953)

[2.3.4 AWS S3 6](#_Toc513298954)

[2.3.5 AWS Kinesis 7](#_Toc513298955)

[2.3.6 AWS EC2 7](#_Toc513298956)

[2.3.7 Backend Web-Service API 7](#_Toc513298957)

[2.3.8 Front-end web application 7](#_Toc513298958)

[2.3.9 Amazon Container 7](#_Toc513298959)

[2.3.10 Chart Libraries 7](#_Toc513298960)

[2.3 Application architecture 8](#_Toc513298961)

[2.4 System Interactions (Sequence Diagrams) 9](#_Toc513298962)

[2.5 Database design 9](#_Toc513298963)

[2.6 Coding standards and packaging 9](#_Toc513298964)

[2.7 Performance 9](#_Toc513298965)

1. Introduction

This document describes how the modules interact with each other in a high-level perspective. It also clearly states the detail necessary design model and architecture design prior to the actual implementation or coding. The architecture proposed in this document is designed to address the demand for the performance as well as the scalability and high availability.

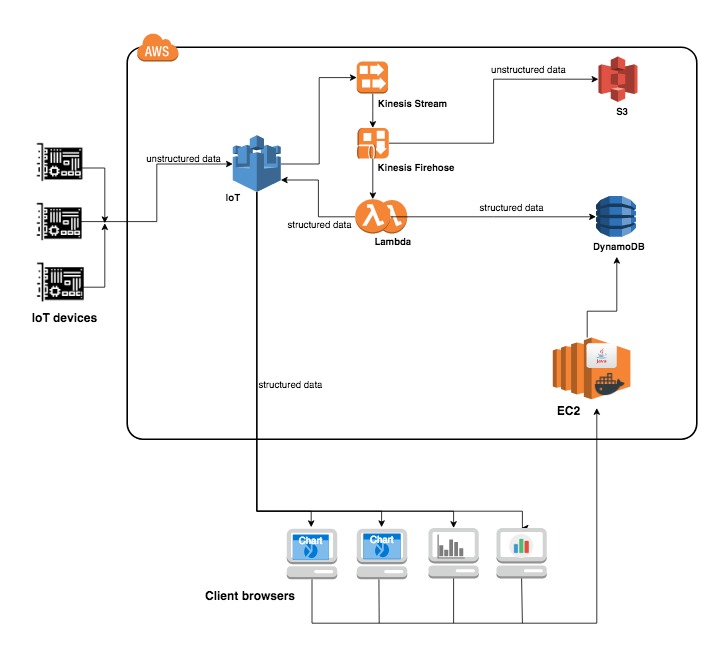
## Product Perspective

As the SERIS is a real-time cloud based analytical monitoring system, the system is required to be designed with the vision of handling high volume real-time data transaction per second. The AWS cloud services have been chosen as the pivotal integrator of the project components as it comes with options to support systems that are highly scalable and available in nature.

On the other hands, the web-based system will be required to develop which will allow multiple concurrent different types of users to access the system in order to see live analytical data in a very good graphical user interface (GUI).

1. System Architecture

## End to End architecture in diagram



## Multitier Architecture

1. **Presentation Layer**: the web application will be implemented in React platform.
2. **Business Layer**: Java back-end application which has business process handlings and expose as REST web services.
3. **Data access Layer**: will be implemented using JDBC with DAO pattern.

## Technologies

### AWS IoT

Amazon Web Services (AWS) IoT service will be deployed in order to get data from the client’s IoT devices to AWS cloud for data storage in Amazon S3 and further data processing in a secure way.

### AWS Lambda

The AWS Lambda service is capable to take care of all the system’s administrative activities including operating system, scaling, code monitoring and logging.

The powerful Lambda service is thus chosen to persist the raw data from the IoT devices to S3 and compute to structure data for the client web browsers for displaying purpose. The lambda functions will be in charge of persisting structured data to DynamoDB and publishing to IoT MQTT broker.

### DynamoDB

In the vision of fully managed cloud database which is fast and flexible NoSql database service application, the Amazon DynamoDB will be deployed to persist structured data which is analysed by Lambda function.

### AWS S3

Amazon S3 is object storage built to store and retrieve any amount of data from anywhere – web sites and mobile apps, corporate applications, and data from IoT sensors or devices. It has the ability to simply and securely collect, store, and analyze their data at a massive scale. The system will deploy S3 storage in order to cover the massive data load from IoT devices.

### AWS Kinesis

Amazon Kinesis enables the system to process and analyse data as it arrives and responds instantly instead of having to wait until all your data is collected before the processing can begin.

### AWS EC2

Amazon Elastic Compute Cloud (Amazon EC2) provides scalable computing capacity in the AWS cloud. Using Amazon EC2 eliminates the need to invest in hardware up front, so the development and deployment of applications will be faster. It also allows to launch as many or as few virtual servers as needed, configure security and networking, and manage storage.

### Backend Web-Service API

Java web application using Spring Boot will perform business logic and create, update, delete and retrieve from Dynamo DB. The app will expose the business services as REST web services API.

### Front-end web application

User interfaces will be designed to develop in JavaScript using React platform. The JavaScript client application will also be calling to the web application which hosts REST web service API for the data processing to Dynamo DB. In order to stream live data to each of the front-end browser a web application is required. By subscribing to IoT MQTT broker topic, the web application clients will then process the live structured data sent from Lambda function and in-turn send it to the front-end browsers.

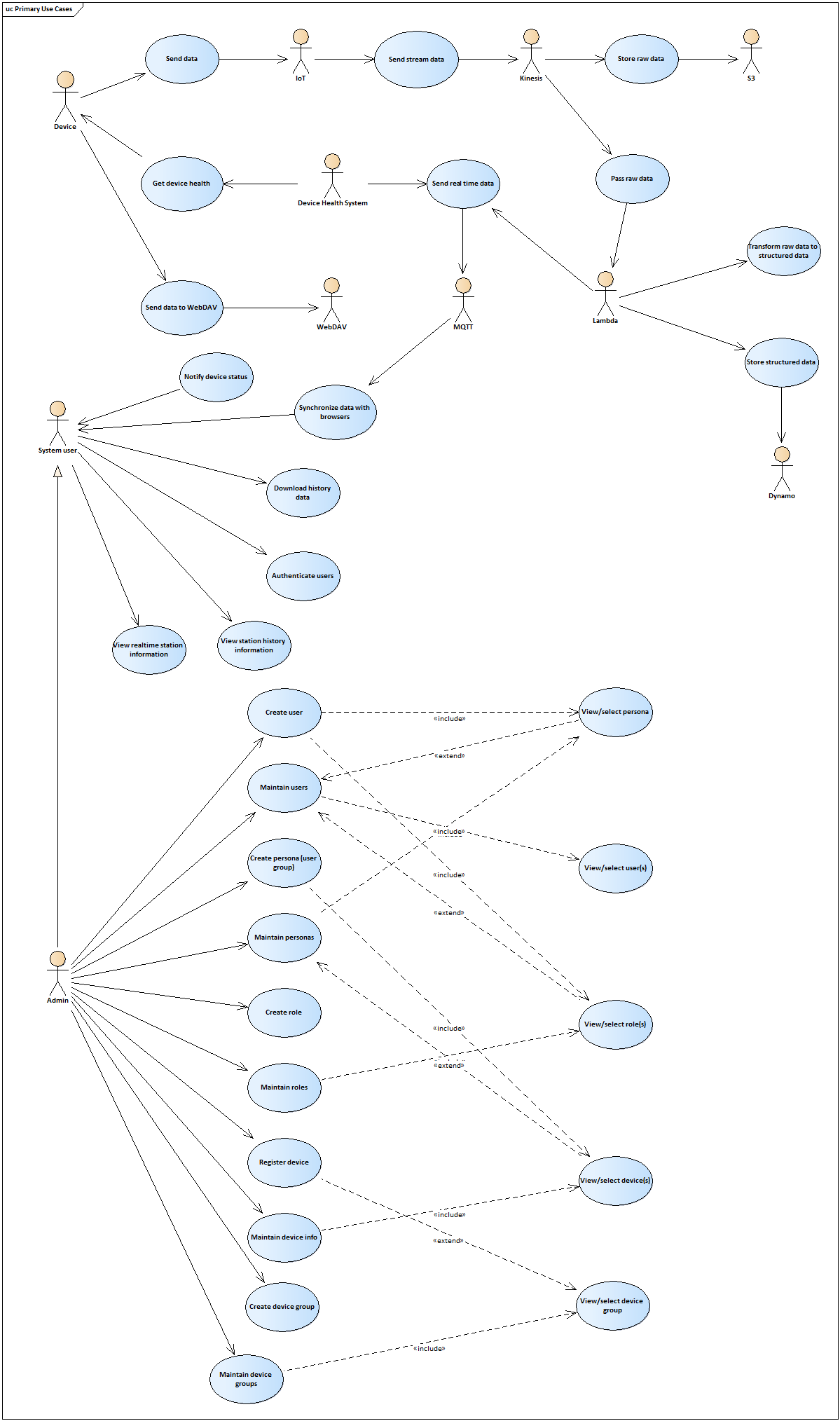
### Amazon Container

AWS Docker is a software platform that allows us to build, test, and deploy applications quickly. Docker packages software into standardized units called containers that have everything the software needs to run including libraries, system tools, code, and runtime. Using Docker, we can quickly deploy and scale applications inside EC2.

### Chart Libraries

JavaScript charting libraries will be used as tools to visualize live weather data parameters in the form of beautiful, easy to understand, and interactive client user interface.

## 2.3 Application architecture



## 2.4 System Interactions (Sequence Diagrams)

To be continued in design phase …

## 2.5 Database design

To be continued in design phase …

## 2.6 Coding standards and packaging

To be continued in design phase …

## 2.7 Performance

The system is required to collect data from all IoT devices tagged to the station. As of now each station is equipped with two devices. For the first implementation phase, the system needs to process data for a total of 5 stations. However, the number of devices tagged to a station and the number of stations to be serviced by the application is set to increase in the future. Therefore, it is necessary to design the system to perform efficiently taking into consideration the landscape of devices and stations to be served by the system in the future.