



# **Software Requirements Specification**

**for**

## **Data for Policy in Food Systems Geospatial Platform**

Version 1

mistEO Private Limited

Submitted to



**India**

30.11.2021

# Privacy Policy

This document contains proprietary and confidential information. All data submitted to the recipient is provided in reliance upon the consent not to use or disclose any information contained herein except in the context of business dealings with mistEO Pvt. Ltd. (mistEO) and United Nations Development Programme - India, (UNDP). Proprietary and confidential information includes, but is not limited to, the following:

- Corporate, employee and infrastructure information about mistEO and UNDP-India
- mistEO and UNDP-India's project management and quality processes;
- Customer and project experiences provided to illustrate mistEO's and UNDP-India's capabilities;
- All commercial information;

Any disclosure of Confidential Information to, or use of it by a third party (i.e., a party other than to whom it is intended to) will be damaging to mistEO or UNDP-India. Ownership of all Confidential Information, no matter in what media it resides, remains with mistEO and UNDP-India. The recipient agrees not to duplicate or distribute or permit others to duplicate or distribute any material contained herein without the written consent of mistEO and UNDP-India.

# Table of Contents

|   |    |
|---|----|
| <b>Introduction</b>                       | 6  |
| Purpose                                   | 6  |
| Document and Intended Audience            | 6  |
| Project Scope                             | 7  |
| References                                | 7  |
| <b>Overall Description</b>                | 8  |
| Product Perspective and Features          | 8  |
| User Classes and Characteristics          | 8  |
| Operating Environment                     | 9  |
| Design and Implementation Constraints     | 9  |
| User Documentation                        | 10 |
| Assumptions and Dependencies              | 10 |
| <b>System Features</b>                    | 10 |
| Map view module                           | 10 |
| Data download                             | 12 |
| Use cases                                 | 12 |
| Data Preprocessing & Engineering pipeline | 13 |
| <b>Solution Architecture</b>              | 14 |
| Security                                  | 16 |
| 4.1.1 Application Security                | 16 |
| 4.1.2 Database Security                   | 16 |
| Implementation                            | 16 |
| Monitoring and Alerting                   | 17 |
| CI / CD                                   | 17 |
| Server Specification                      | 18 |
| <b>External Interface Requirements</b>    | 19 |
| User Interfaces                           | 19 |
| Software Interfaces                       | 19 |
| Communications Interfaces                 | 19 |
| <b>Other Nonfunctional Requirements</b>   | 19 |
| Performance Requirements                  | 19 |
| Safety Requirements                       | 20 |
| Security Requirements                     | 20 |

|                             |    |
|-----------------------------|----|
| Software Quality Attributes | 21 |
| <b>Appendix A: Acronyms</b> | 21 |
| <b>Appendix B: Datasets</b> | 21 |
| <b>Appendix B: UI</b>       | 21 |

## Revision History

| Version | Date       | Changes |
|---------|------------|---------|
| v0.1    | 30.11.2021 | Draft   |
|         |            |         |

# **1. Introduction**

## **1.1 Purpose**

UNDP has partnered with the Government of Telangana to jointly initiate the NextGenGov 'Data for Policy' initiative on Food Systems. The aim is to incorporate anticipatory governance models for future-fit food systems in Telangana using data-driven policymaking tools and ecosystem-driven approaches. UNDP is keen on augmenting learning capabilities, increasing the predictive or anticipatory capacity to feed-in to evidence-driven policies in the state, and create radical traceability and transparency across the system from producers to consumers by building provenance documentation around food that can help build trust in the system at the same time nurture sustainable and healthy practices. The goal is to design, develop and demonstrate anticipatory governance models for food systems in Telangana using digital public goods and community-centric approaches to strengthen data-driven policy making in the state.

The Food Systems Innovation platform for Telangana is envisioned as a Digital Public Good that will strategically feed into data-driven decision making in the state. The platform will have the capability to visualize and analyze high resolution geospatial data (both vector as well as raster layers). The digital platform will need to curate, integrate and visualise such critical datasets and assets to answer the basic question of - What is growing where? How much is there and the spatial and temporal changes within the state across various indicators relevant to Agriculture and Food Systems. The platform should be able to visualize over time the changes that have happened to the agriculture ecosystem in terms of crop diversity, changes in soil/ground water, tree cover, and other indicators at higher resolution to support policy decisions. Such a synthesis of data and analytics can help identify farms which are doing exceptionally well (Positive Deviance) through which repositories of good practices and indigenous knowledge can be documented. This also helps in identifying farms that are not doing good as per the defined indicators (Negative Deviance).

## **1.2 Document and Intended Audience**

The system requirements specification document describes what the system is to do, and how the system will perform each function. The audiences for this document include the system developers and the users. The system developer uses this document as the authority on designing and building system capabilities. The users review the

document to ensure the documentation completely and accurately describes the intended functionality.

This document provides general descriptions of the system. The system developers should review the document to ensure there is adequate information for defining an initial design of the system. The users should review the document to affirm the features described are needed, to clarify features, and to identify additional features needed within the system.

### **1.3 Project Scope**

The project will be initially piloted for Telangana State. After the successful demonstration of this pilot, it is proposed to scale up to other states of India and has the potential to develop into a national Data for Policy in Food Systems.

In the first phase of digital technology platform development, a user-friendly web application must be developed using open software, open tools, open code and open APIs. The functionalities of the web application are as follows:

- Integrate different formats of geospatial data at high-resolution (provided by UNDP & partner organizations) with ability to have API based access to various databases and local storage
- Visualize geospatial data and render an interactive user experience and interface
- Ability to download the data after successful registration on the web application
- Ability to crowdsource use cases and projects that are actively using datasets curated in the platform
- An authentication and authorization module to screen the crowdsourced projects and approve them to be visualized on the platform

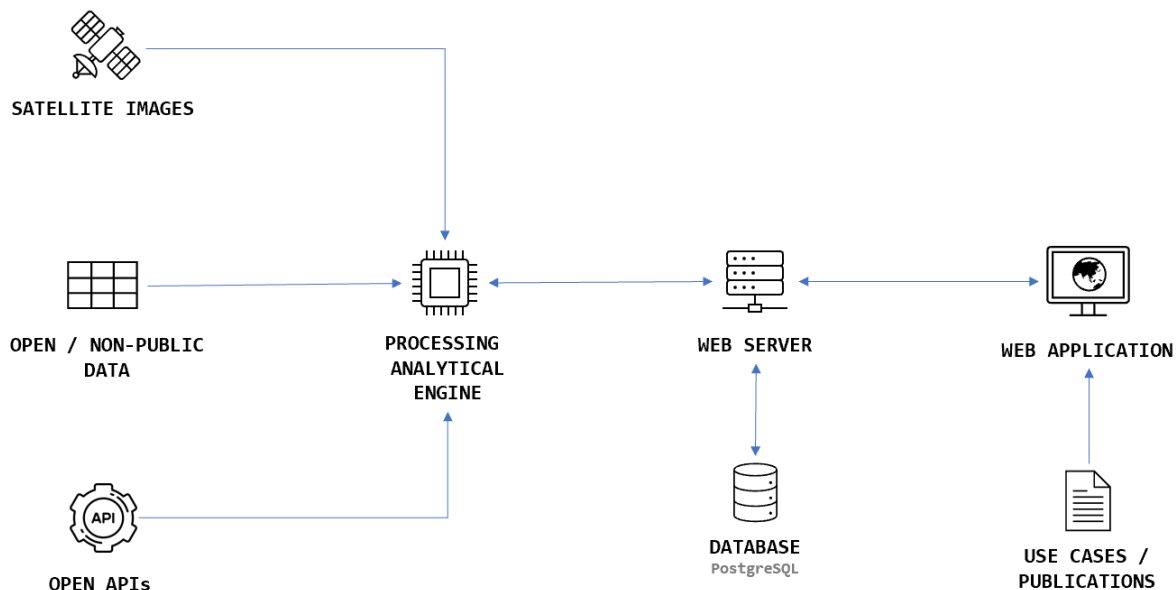
### **1.4 References**

- Terms of Reference document
- HungerMap by WFP - <https://hungermap.wfp.org/>
- Data4Policy project repository - <https://github.com/UNDP-India/Data4Policy>

## 2. Overall Description

### 2.1 Product Perspective and Features

The Food Systems Innovation platform for Telangana is envisioned as a Digital Public Good that will strategically feed into data-driven decision making in the state. The platform will have the capability to visualize and analyze high resolution geospatial data (both vector as well as raster layers). The digital platform will need to curate, integrate and visualise such critical datasets and assets to answer the basic question of - What is growing where? How much is there and the spatial and temporal changes within the state across various indicators relevant to Agriculture and Food Systems. The platform should be able to visualize over time the changes that have happened to the agriculture ecosystem in terms of crop diversity, changes in soil/ground water, tree cover, and other indicators at higher resolution to support policy decisions. Such a synthesis of data and analytics can help identify farms which are doing exceptionally well (Positive Deviance) through which repositories of good practices and indigenous knowledge can be documented. This also helps in identifying farms that are not doing good as per the defined indicators (Negative Deviance).



### 2.2 User Classes and Characteristics



System shall have only one type of user, ie Public User / Citizen

| User                  | Roles  |
|-----------------------|--|
| Public User / Citizen | <ul style="list-style-type: none"><li>• These users includes citizens as well as policy makers</li><li>• These users will have complete access to the system</li></ul> |

## 2.3 Operating Environment

The rule for selecting hardware and software is that the components/application must be functionally efficient, capable of interfacing with other software, and easy to maintain.

- The System shall operate with all the leading Web browsers.
- The System shall operate from servers on Amazon Web Services (AWS)

## 2.4 Design and Implementation Constraints

The goal of the web application is to be platform independent on the client side wherever possible. Therefore, the web applications will be implemented to run on the server side as much as possible. Also, it is required to test the application using different platforms, connection speeds, screen settings, colors/graphics, and browsers.

**DIC-1 :** The system shall be in compliance with all Accessibility, Web Design, and Security policies applicable.

**DIC-2 :** As part of standard operating procedures, a testing plan will be documented during the Design phase. The testing plan will be based on user roles, modules or use cases, required tasks and expected outcomes.

**DIC-3 :** The database structure should be complete as quickly as possible during the design stage but there should be room for modification without a large overhaul during later phases.

**DIC-4 :** The data/layers to be used should be finalised as fast as possible to avoid unexpected issues related to data processing.

**DIC-5 :** It may take some time to develop a data pipeline process for new datasets.

**DIC-6** : Every component/libraries/data used in this project should be available as open source.

## **2.5 User Documentation**

User documentation will consist of the several components usually expected of a modern web-based software application, including a tutorial, help pages, FAQs with an online request form, and a complete user's manual.

**UDN-1** : A tutorial will provide a quick start, a walk-through of major system features, and a further reference source for the complete system features.

**UDN-2** : A video explanation of the portal will be provided.

**UDN-3** : The user's guide (or user manual) will contain sufficient information and instructions required to access and use the data system. It will include:

**UDN-3.1** : Overview of the system features and architecture.

**UDN-3.2** : Instructions for accessing the system.

**UDN-3.3** : Samples of screens, where appropriate.

## **2.6 Assumptions and Dependencies**

- It is anticipated that the data/layers required for this project will be provided by partner organizations of UNDP-India.
- The content required for the portal related to the project will be provided by UNDP-India.
- It is anticipated that there are no other user roles than public user / citizen.
- No authentication is required to access this platform.

# **3. System Features**

## **Functional Requirements**

### **3.1 Map view module**

The main section on the portal.

**FR-3.1-01** : Layer selection and view spatial data

The user should be able to select different layers to see it on top of the base map.

#### **FR-3.1-02 : Location search**

The user should be able to search the location through 3 different ways

- a) Address Search - Nominatim API can be used.
- b) Coordinate Search - Lat, Long Search
- c) Administrative Name search

#### **FR-3.1-03 : Option to choose Raster or Vector**

The user should be able to switch between the raster or vector of a parameter layer as per its availability.

#### **FR-3.1-04 : Map controls**

The basic map control features like zoom-in, zoom-out & pan options should be available.

#### **FR-3.1-05 : Map Legend**

A colour palette like legend for a selected layer should be there. Min & Max Value or Low & High indication can be provided based on the layer.

#### **FR-3.1-06 : Option to display partner logos**

The logos of UNDP and Government of Telangana are considered as primary logos and will be placed at the top-right of the page. The logos of other partners can be placed at the bottom of the dashboard or at the 'about project' section.

#### **FR-3.1-07 : Option to know more about the layer**

The user will be able to see the details of the layer such as description, source, frequency etc by clicking on an info icon.

#### **FR-3.1-08 : Option to download the data of selected layer**

The download of selected layer can be triggered directly from the layer selection option.

#### **FR-3.1-09 : Display value on hover**

The display option of an attribute value on hover of the selected layer should be there.

#### **FR-3.1-10 : Details section for a selected shape**

There will be a modal visible on the right side of the dashboard. This will get triggered on click of a selected shape (active layer). The below mentioned information will be available.

- Details of selected shape
- Attribute value of selected shape
- Min, Max value
- Description of the layer
- Data Source
- Trend chart where it shows the temporal changes of the selected region of the parameter. (eg. NDVI value of a selected region)
- Option to download the current

#### **FR-3.1-11 : Draw custom polygon (farm boundary)**

The user should be able to draw a custom polygon and the details of the same should be displayed. The trend also needs a download option if available.

### **3.2 Data download**

#### **FR-3.2-01 : Data**

The user should be able to download the selected data in different formats.

### **3.3 Use cases**

#### **FR-3.3-01 : Upload use cases**

Option to upload use cases is required. The information like project details, data used, url and uploader's contact details will be collected.

#### **FR-3.3-02 : Listing of Use cases**

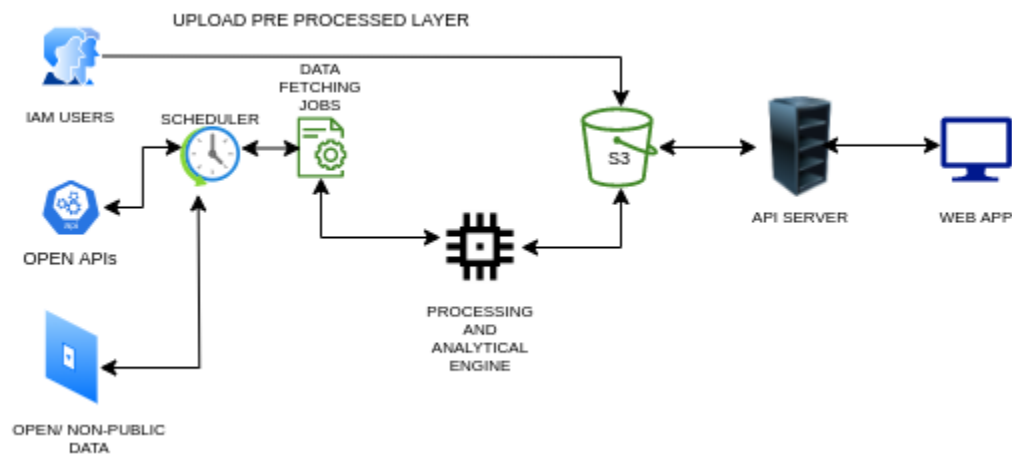
Option to list all approved use cases is required. The approval process for the same TBD.

#### **FR-3.3-03 : Use Cases approval**

The use case approval can be handled through updating the status on a configuration file. Based on the file, frontend loading can be decided.

### **3.4 Data Preprocessing & Engineering pipeline**

The data preprocessing pipeline is dependent on the type & source of the data. The required process to be written to get the data. The data flow is explained below.



#### **FR-3.4-01 : Fetch data functionality**

Data/Layers come from different sources. So data extraction methods may differ and the different approaches to be handled as per the data.

#### **FR-3.4-02 : Option to incorporate preprocessed data for partner organisations**

There should be an option to update the required data/layers (preprocessed) to the storage. So that it can be consumed for the application.

#### **FR-3.4-03 : Data download & process Scheduler**

Develop a scheduler to fetch data in different frequencies and different sources.

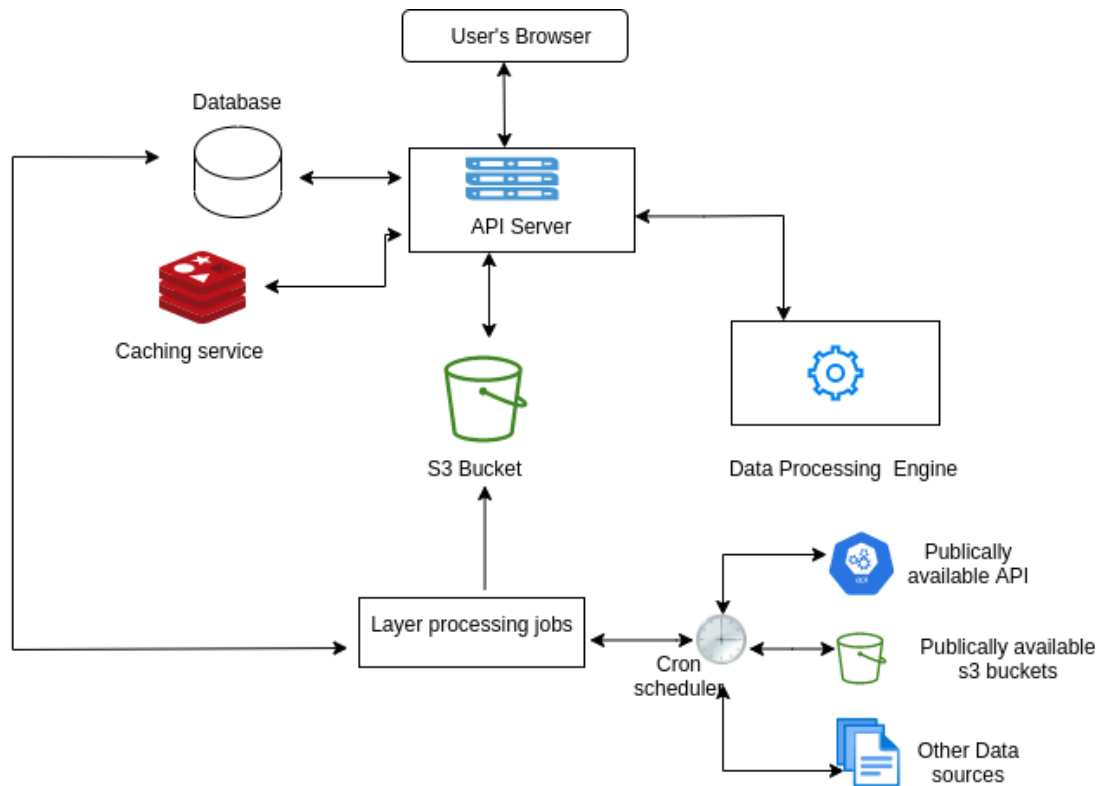
#### **FR-3.4-04 : Data preprocessor**

- Validation - Validating the format & projection
- Conversion - convert to the required format.

#### **FR-3.4-04 : API Development**

- API for add use cases
- API for list use cases
- API for getting zonal statistics against a shape
- API for getting zonal statistics against a shape - temporal trend
- API for getting layers for a user drawn location
- API for getting layer for current selected parameter
- API for download data for admin

## **4. Solution Architecture**



The cloud architecture consists of the following components.

- WAF
- EC2
- RDS
- S3
- Cloudfront
- CloudWatch
- SNS

The application will be hosted under AWS infrastructure (Asia Pacific-Mumbai). The application deployed and the other services will be monitored using Cloud Watch.

Regarding the datastore, RDS for PostgreSQL will be used. The database is configured as global write and local read.

## 4.1 Security

The AWS services used for the application security

- WAF
- Security Group

### 4.1.1 Application Security

- Inbound traffic to the application will be whitelisted using the WAF and security groups. WAF will be only accepting the traffic from the pre-defined IPs.
- The Inbound traffic to the application will be used standard HTTP/HTTPS Ports for inbound traffic.
- All frontend forms will have captcha verification enabled.
- The data communication happens through an API that will have an encryption layer.
- The API server will verify the domain from which the request comes and act accordingly.

### 4.1.2 Database Security

- The database access will be restricted to authenticated users.
- The database default PORT will be changed.
- TLS implementation.
- Encryption at rest is not considered since the application collects limited PII data.
- DB instance login will be restricted.

## 4.2 Implementation

The entire infrastructure will be provisioned as per the requirement. And the CI / CD will be handled using GitOps.

- EC2 t2.2xlarge is used to handle the data processing and API in the production environment.
- EC2 t2.xlarge is used for the backend in the development environment.



- The frontend deployment will be handled through AWS S3 Static hosting and Cloudfront.
- S3 Storage class: Amazon S3 Standard (S3 Standard). Lifecycle policies are not needed for the project.
- RDS for PostgreSQL will be used as the database

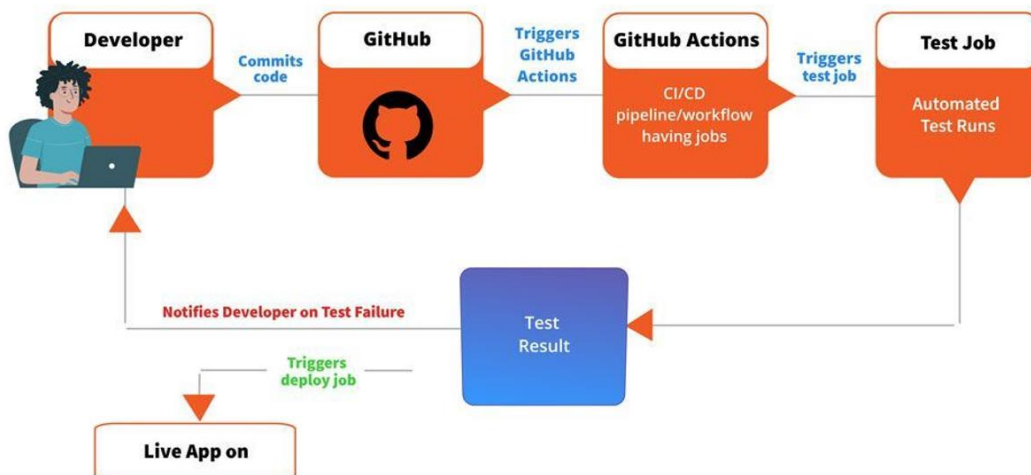
### 4.3 Monitoring and Alerting

The Different services used for monitoring and alerting will be.

- CloudWatch
  - Detailed monitoring
- SNS
- Grafana

Cloudwatch and Grafana will be used for monitoring the status of the server, SNS will be used to create an alarm and notify the System Admin through SMS and emails.

### 4.4 CI / CD



The deployment of the apps to the infrastructure will be handled using the GitOps methodology. Github Actions will be involved in it.

Whenever a developer moves a feature for deployment by placing a merge requests (MR) , Github CI pipeline will be triggered and it will run a set of pipelines.

The pipeline includes the

- **Lint test** : Check the code is properly formatted, syntax error etc.
- **Unit test** : Runs predefined test cases.
- **Dependency audit** : This step will check whether there are any vulnerabilities in the libraries we are using.
- **Build** : This step creates the build of the application and pushes to the server.

If any of the above steps fails , the whole process will be aborted.

Incase of production , the pipeline won't be triggered automatically, we need to trigger it manually.

## 4.5 Server Specification

The application will be hosted on Amazon Web Services (AWS) and will also use other managed services from AWS

1. Amazon EC2 Instance t2.xlarge (vCPU : 8, Memory : 32GB, Storage : EBS-Only, Network Performance : Moderate ) - for Backend Production Server
2. Amazon EC2 Instance t2.xlarge (vCPU : 4, Memory : 16GB, Storage : EBS-Only, Network Performance : Moderate ) - for Backend Development/staging Server
3. Amazon S3 Static Hosting - FrontEnd
4. Amazon CloudFront - CDN
5. Amazon S3 - Data Storage
6. Amazon Relational Database Service (RDS) - Database
7. AWS WAF (Web Application Firewall) - Security
8. AWS SNS - Alert
9. AWS CloudWatch - Monitoring
10. AWS Elastic IP - IP
11. S3 Glacier - Data Backup

Ref:

<https://calculator.aws/#/estimate?id=c7fcfd61be73b10bfd3bd5a3fa921c01a1fb039b>

## 5. External Interface Requirements

### 5.1 User Interfaces

The user interface will be simple and consistent, using terminology commonly understood by the intended users of the system. The system will have a simple interface, consistent with industry standard interfaces, to eliminate the need for user training of infrequent users. For additional details see Appendix .... User testing will be used to ensure the user interface is clear (simple, commonly understood vocabulary, intuitive to use without training), complete (users can perform all functions from the interface), and consistent (buttons and wording are the same throughout the system).

### 5.2 Software Interfaces

The system will use the standard OpenSource tools/libraries. This includes, but is not limited to, ReactJS, Python, Fast API, PostgreSQL, and Apache & NGINX server.

### 5.3 Communications Interfaces

The system will use open communication protocols. This includes, but is not limited to, HTTP protocol for communication with the web browser and the web server and TCP/IP network protocol with HTTP protocol.

## 6. Other Nonfunctional Requirements

### 6.1 Performance Requirements

**Response time:** The application shall show no visible deterioration in response time as the number of persons increases. Response times seen by end users for querying metadata should be on the order of a few seconds or less.

**Loading speed:** The application shall load as quickly as comparable productivity tools on whatever environment it is running in.

## **6.2 Safety Requirements**

AWS is responsible for protecting the infrastructure that runs all of the services offered in the AWS Cloud. This infrastructure is composed of the hardware, software, networking, and facilities that run AWS Cloud services.

## **6.3 Security Requirements**

The security requirements will have four primary components. They are authentication, confidentiality, integrity, and availability.

### **SCR-1: Authentication**

Since it is an open source public platform, user authentication is not used. But all other security features such as captcha, encryption and blocking of bulk api requests will be enabled.

### **SCR-2: Confidentiality**

Confidentiality security requirements from the client will be addressed.

### **SCR-3: Data Integrity**

The integrity of data will be critical to its success as a product. The data used only from validated sources. Therefore, extensive data validation and review will be performed both before data are uploaded to the system and as part of the upload process. The system will need policy and procedures protecting the data from intentional or unintentional modifications, and to ensure accurate data are made available.

### **SCR-4: Availability**

The fourth consideration for security requirements is availability. The system must be available to the intended audience 24 hours per day, 7 days a week with 99% availability and a tolerance of -5% (not less than 50% of working hours in any week). For this system, availability will be concerned with the reliability of the software and network components. Intentional “denial of service attacks” is not foreseen as a significant concern.

## 6.4 Software Quality Attributes

### SQA-01: Portability

This database will be built for a particular system and may not be portable but results to queries will be portable between many environments.

### SQA-02: Adaptability

Implementation of the application software/code and design of database structure should be flexible enough for the necessary change in the later phase.

### SQA-03: Availability

Availability is defined here to mean the ability to use the system during its intended period of operation.

## 7. Appendix A: Acronyms

UNDP - United Nations Development Programme

MPL - mistEO Private Limited

GIS - Geographical Information System

UI - User Interface

HTML - HyperText Markup Language

HTTP - HyperText Transfer Protocol

RDS - Relational Database Service

WAF - Web Application Firewall

NDVI - Normalized difference vegetation index

AWS - Amazon Web Services

API - Application Programming Interface

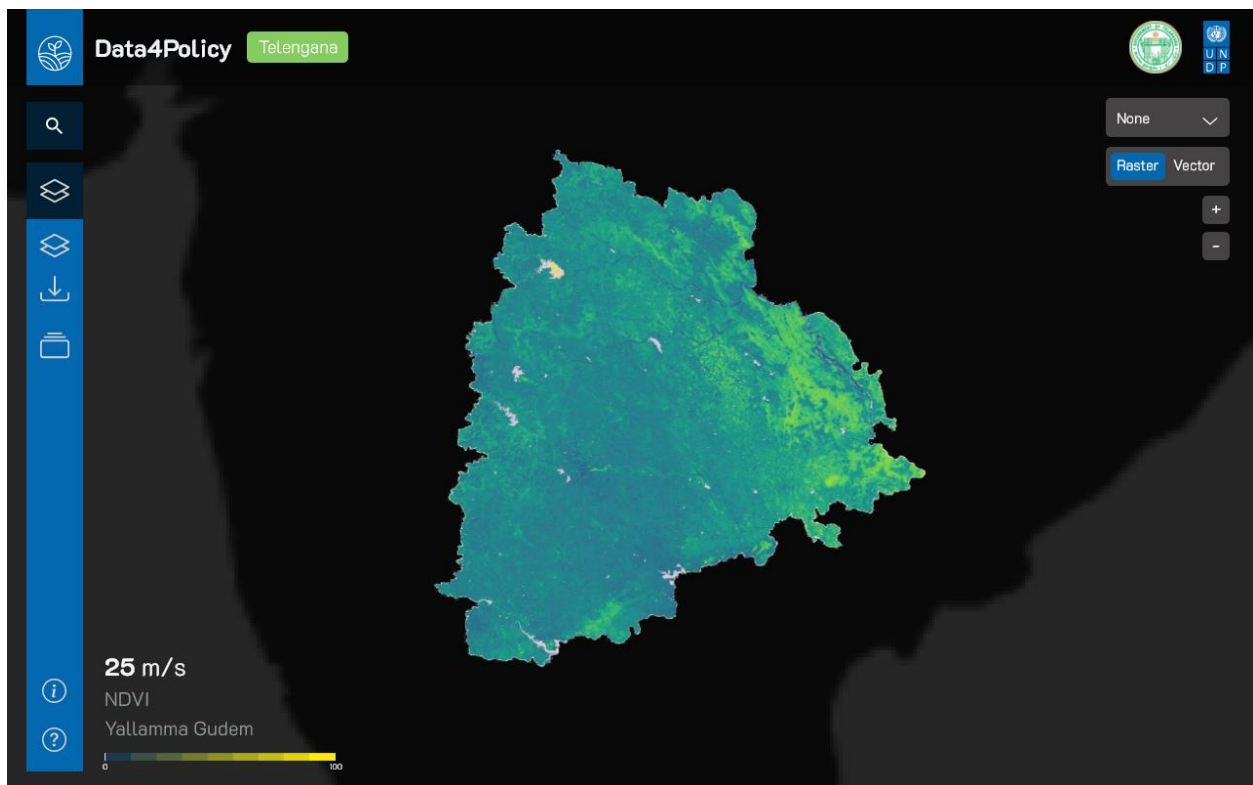
CI - Continuous Integration

CD - Continuous Delivery/Continuous Deployment

## 8. Appendix B: Datasets

- NDVI
- Soil Moisture
- Fire Events
- 

## 9. Appendix B: UI





## Use cases

+ Add



20 March 2021

[www.deccanherald.com](http://www.deccanherald.com)

## What is 'Data for Policy' initiative on Food Systems ?

Lorem ipsum dolor sit amet, consectetur adipiscing elit,  
 sed do eiusmod tempor incididunt ut labore et dolore  
 magna aliqua. Ut enim,

[Find out More](#)



ANALYTICS

20 March 2021

[www.tableau.com](http://www.tableau.com)

How the vegetation changes over last 5 years ?

Lorem ipsum dolor sit amet, consectetur adipiscing elit,  
 sed do eiusmod tempor incididunt ut labore et dolore  
 magna aliqua. Ut enim,

[Find out More](#)



PROJECT

20 March 2021

[www.deccanherald.com](http://www.deccanherald.com)

Apply best practices - Agriculture

Lorem ipsum dolor sit amet, consectetur adipiscing elit,  
 sed do eiusmod tempor incididunt ut labore et dolore  
 magna aliqua. Ut enim.

[Find out More](#)



ARTICLE



ANALYTICS

+ view



PROJECT