

Project:	Web mapping of Rain and Temperature forecasts for Africa Report tender no. EC-JRC/IPR/2025/VLVP/3270
Meeting Date:	
Report:	This is the final report that gives an overall documentation of the project, its workflow, and deliverables. It is a formal output of the Project detailing the activities carried out from Task 1-Task 4 of the project.
Deliverables:	D1.1.1 Minutes of the workshop and strategic decisions for the development of the system. D2.1 Data model for storing the rainfall data, the long-term variables, and anomalies created. D2.2 Creation of 2 Python scripts to convert the source data to the database model. D2.3 Conversion of the archive and computation of the long-term characteristics. D2.4 Development, implementation, and release of the forecasting approach. D3.1 Data processing workflow that produces a comprehensive drought report D3.2 Program to compute the trend of the monthly and seasonal rainfall D3.3 Program to compute the correlations of the trend D3.4 Visualization module development and addition of an entry on the map viewer D3.5 Custom report analysis at different administrative levels, user-defined Area of interest (AOI), and user uploaded AOI D3.6 Registration and subscription function D3.7 Notification function D 4.1 Detailed report on system implementation and working system D 4.2 User engagement session to unveil the system to the public D.4.2.1 Minutes of the Presentation D 4.3 Final Report
Report ID:	

Report by:	Pierre H. KAMSU TAMO – ACMAD Hubert KABENGELA - ACMAD Collins ASEGA– ACMAD Wendlasida COMBERE - ACMAD
------------	--

Disclaimer:

“The information and views set out in this report are those of the author(s) and do not necessarily reflect the official opinion of the Commission. The Commission does not guarantee the accuracy of the data included in this study. Neither the Commission nor any person acting on the Commission’s behalf may be held responsible for the use which may be made of the information contained therein.”

Table of Contents

Introduction	3
Background Information and Purpose	3
Project Scope	4
Task 1	5
Figure 1 System proposed features	6
System Server Architecture at ACMAD	6
Task 2	7
Performance issues	9
Task 3	10
Task 4	11
Conclusions	12
Recommendations	12
Project Overview	14
ANNEX D1.0 - Document with a description of the Working system, deployment and the Continuous integration and Continuous delivery pipeline.	16
Introduction	16
System Documentation	16

4. GeoServer Integration	24
5. Task Processing (Celery + Redis)	25

Introduction

Background Information and Purpose

The JRC operates a Web Map server and integrated Content Management System to analyse and map geographic data in order to monitor and forecast drought in the world on two Drought Observatories (DO): European Drought Observatory (EDO) and the Global Drought Observatory (GDO). This work resulted in the development of a web application that manages monitoring, forecasting, and historic drought related data. In this project, the transfer of the Web Map server version based on Open Source technologies has been implemented by, ACMAD, through the installation of the system at its premises in Niamey, Niger and remote infrastructure in Johannesburg, South Africa.

In 2023, JRC's Drought group of the Natural Disaster Unit completed the transfer of drought web mapping technology to ACMAD within the intra ACP-ClimSA project. A web-mapping platform for Drought Monitoring was created: the African Drought Monitoring & Advisory (ADMA) <https://ada.acmad.org/home>. A near-real time system that uses Earth Observation and Weather information to monitor drought conditions in the African region is the final deliverable of this project. This system contains drought-relevant information such as maps of indicators derived from different data sources (e.g., precipitation measurements, satellite measurements, modelled soil moisture content). Different tools, like Graphs that allow for displaying and analysing the information and drought reports give an overview of the situation in case of imminent droughts.

A drought prediction component, based on seasonal forecasts, is now being developed and integrated to the current monitoring system ADMA to alert stakeholders and provide actionable information. This document describes the work done to reach that goal. The new product will be based on Copernicus input, thus allowing ACMAD to gain more knowledge on these products and to verify the validity and application of these products in the African context.

The system is an adaptation of the European Drought Observatory (EDO) to the conditions in the African region.

This document describes the work done to implement and adopt additional parameters most importantly the drought and precipitation forecast and their integration into the African Drought Monitoring and advisory system ADMA. It also documents the execution of the project from the beginning until the end, providing the chronological course of the project's activities and achievements ("project diary") organized by Tasks.

Project Scope

The scope of this project entails:

1. Improvement of the existing Drought system, by integration of drought forecasting, creation of user alerts on forecasted droughts. Addition of datasets including ERA5 SPI, bug fixes and enhancement of the user interface.
2. Assist in the documentation, automation and reproducibility of the seasonal forecast approach.

Project Deliverables

Based on various tasks undertaken, the deliverables to be supplied in the course of the project implementation and execution are as below:

Task	Description	Deliverables
TASK 1	Decision workshop aimed to exchange possibilities and align on the forecast integration.	D.1.1, D 1.2
TASK 2	Development, implementation and release of the forecasting approach.	D2.1, D.2.2, D.2.3, D.2.4
TASK 3	Development, implementation, and release of the data-to-information approach.	D.3.1, D.3.2, D.3.3, D.3.4, D.3.5

TASK 4	Presentation of the system in an ACMAD user forum. This shall be done through an organized user session carried out online and may be extended to any upcoming in-person meeting that ACMAD or its partners organize.	D.4.1, D.4.2, D.4.3

Task 1

The activities of this task entailed organizing a kick off meeting to discuss the modalities of the implementation of the project. The meeting set the stage for recollection of the strong partnership between ACMAD and JRC, detailing the progress and success of the first phase of the project. Highlighting some of the challenges that the second phase seeks to address and the roadmap that the project seeks to achieve. The meeting concluded with deliberations to integrate the Unusually Wet and Dry and wet Conditions forecasting product from JRC, while ACMAD seeks to explore customization of this product to capture the specifics within the African context. The workshop also provided an opportunity for ACMAD to present the nine step approach they used for forecasting. Other critical aspects included development of software features that would improve the system, integration of social-economic layers and introduction of subscription modules that would assist in sending alerts to subscribed users.

Existing Features	Proposed Additional Features	Beyond the Scope
☒ Near Real-Time Drought Monitoring ☒ Integrated Data Platform (Multi-sources) ☒ Interactive Web Geoportal ☒ Location Search ☒ Time series data analysis ☒ Reporting ☒ Data Download (CDI) ☒ Backend & Automation ☒ API ☒ Integration with Datasets from Google Earth Engine	☐ Forecast & Scenario Analysis ☐ Automated Drought Advisory Generation ☐ Alerts & Dissemination • Threshold-triggered alerts via email • API endpoints for external integration with national systems (mainly documentation) • Bulletin scheduling for monthly/seasonal advisory releases ☐ Scenario overlays (rainfall deficits + vegetation health).	☐ Automated Drought Advisory Generation • AI-powered narrative summaries using LLM templates • Regional breakdown: country- or sub-region-specific reports ☐ Forecast integration: seasonal outlook with risk projections.

Figure 1 System proposed features

System Server Architecture at ACMAD

The section here describes high level Architecture used in the system as well:

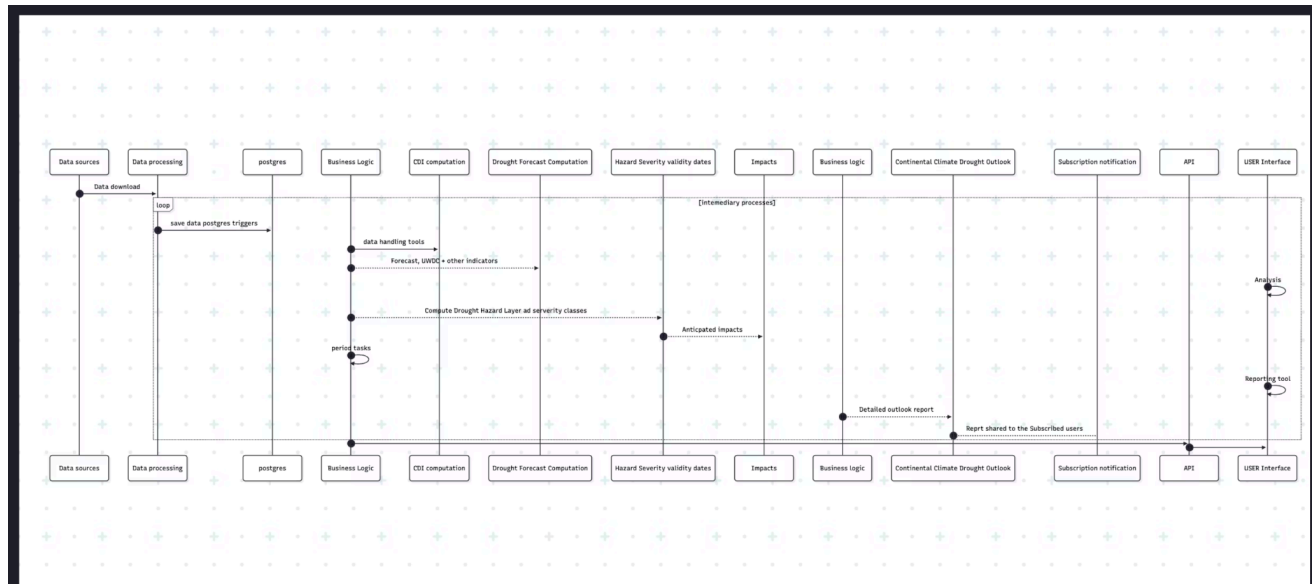


Figure 2: High level architecture

Task 2

This task focused on the development of a forecasting approach and adopted for use in the system. The Unusually wet and dry conditions (UWDC) dataset was explored including its research paper, the team then worked on a methodology to enhance the output by introducing bias correction, and observation data into the procedure of creating the dataset since it only relied on forecast dataset. ACMADs forecast polygons that are derived from the nine step expert driven approach were also explored and integrated the following activities were carried out:

1. Extension of the database to include a database model to store additional datasets, including the UWDC, ERA5 SPI datasets, population, land use land cover and temperature datasets.
2. Python scripts to process the datasets, this included modulization of the script that integrates the data to the database, where a python package was created, this package is installable and is run frequently to update the data. This ensures that we break down the system into functional

- parts that are easy to manage. We also modularized the python script that publishes the dataset to the tiling engine (Geo-server), creating styles and views where we chose to build a reusable python script that can be extended to accommodate any new datasets in the future.
3. Implementation of an additional CDI class where we check for the pixels in warning and watch and compare with the forecast dataset and if the forecast shows there will be unusually dry conditions, then we create a new class 'Drought forecasted to persist'.
 4. Integration of the forecast polygon.

```
class GeoDataPublisher:
    def __init__(self, dataset_name, store_name, table_name, data_table_name,
                 sld_template_path, output_dir):
        self.dataset_name = dataset_name
        self.store_name = store_name
        self.table_name = table_name
        self.data_table_name = data_table_name
        self.grid_id_column_name = grid_id_column_name
        self.data_id_column_name = data_id_column_name
        self.sld_template_path = sld_template_path
        self.output_dir = output_dir
        # Initialize other GeoServer and database settings
        self.database_user = env("POSTGRES_USER")
        self.database_pwd = env("POSTGRES_PASS")
        self.db_name = env("POSTGRES_DBNAME")
        self.db_host = env("PG_HOST")
        self.geoserver_workspace = env("Geoserver_Workspace")
        self.GEOSERVER_URL = "http://geoserver:8080/geoserver"
        self.USERNAME = env("Geoserver_user")
        self.PASSWORD = env("Geoserver_pass")
        # Initialize GeoServer connection
        self.geo = Geoserver(self.GEOSERVER_URL, username=self.USERNAME, password=self.PASSWORD)
        self.columns = get_columns(self.dataset_name)
        # todo include creation of geoserver workspace

    def create_view(self):
        """Create database views based on the dataset"""
        cur, conn = self.connect()
        if cur is None:
            return False
        for col in self.columns:
            view_name = f'{self.table_name}_{col}_view'
            sql_command = f'CREATE OR REPLACE VIEW {view_name} AS SELECT id, cell, {col}, year, xcol, yrow FROM {self.table_name}, {self.data_table_name}
                           WHERE {self.data_id_column_name} = {self.grid_id_column_name};'
            cur.execute(sql_command)
            conn.commit()
            log.info(f'created view name:{view_name}')
        conn.close()
        return True

    def handle_store_exists(self, store_name):
        """Check if store exists and create if not"""
        feature_store_exists = self.geo.get_featurestore(store_name=store_name, workspace=self.geoserver_workspace)
        if type(feature_store_exists) is not dict:
            self.geo.create_featurestore(store_name=store_name, workspace=self.geoserver_workspace,
                                         db=self.db_name, host=self.db_host, pg_user=self.database_user, pg_password=self.database_pwd)
            return {'message': 'complete'}

    def publish_data(self, prefix=None):
        """Publish data to GeoServer"""
        self.handle_store_exists(store_name=self.store_name)
        for col in self.columns:
            view_name = f'{self.table_name}_{col}_view'
            style_name = col if not prefix else f'{prefix}_{col}'
```

Figure 2: Geodata Publisher Modularised script

```
setup.py > ...
1  from setuptools import find_packages, setup
2
3  def parse_requirements(filename):
4      with open(filename, "r") as file:
5          return [line.strip() for line in file if line.strip() and not line.startswith("#")]
6
7  setup(
8      name='adma-data-pipeline',
9      version='0.0.5.dev.20250919',
10     description='ACMAD adma data pipeline package',
11     url='https://github.com/ACMAD-Niamey/adma-data-pipeline.git ',
12     long_description=open('README.md').read(),
13     long_description_content_type='text/markdown',
14     author='ACMAD',
15     author_email='',
16     license='',
17     packages=find_packages(),
18     install_requires=parse_requirements("requirements.txt"),
19     classifiers=[
20         'Development Status :: 1 - Planning',
21         'Intended Audience :: Science/Research',
22         'License :: OSI Approved :: BSD License',
23         'Operating System :: POSIX :: Linux',
24         'Programming Language :: Python :: >= 3.5',
25     ],
26 )
```

Python package for data processing, CDI computation and and database ingestion

Performance issues

With regards to performance of the system the overall implementation and modulization of the script and package development provided performance improvements. Where we gain speed in data processing and ingestion for example processing and ingestion of the Unusually wet and dry condition datasets takes approximately 0.8 seconds.

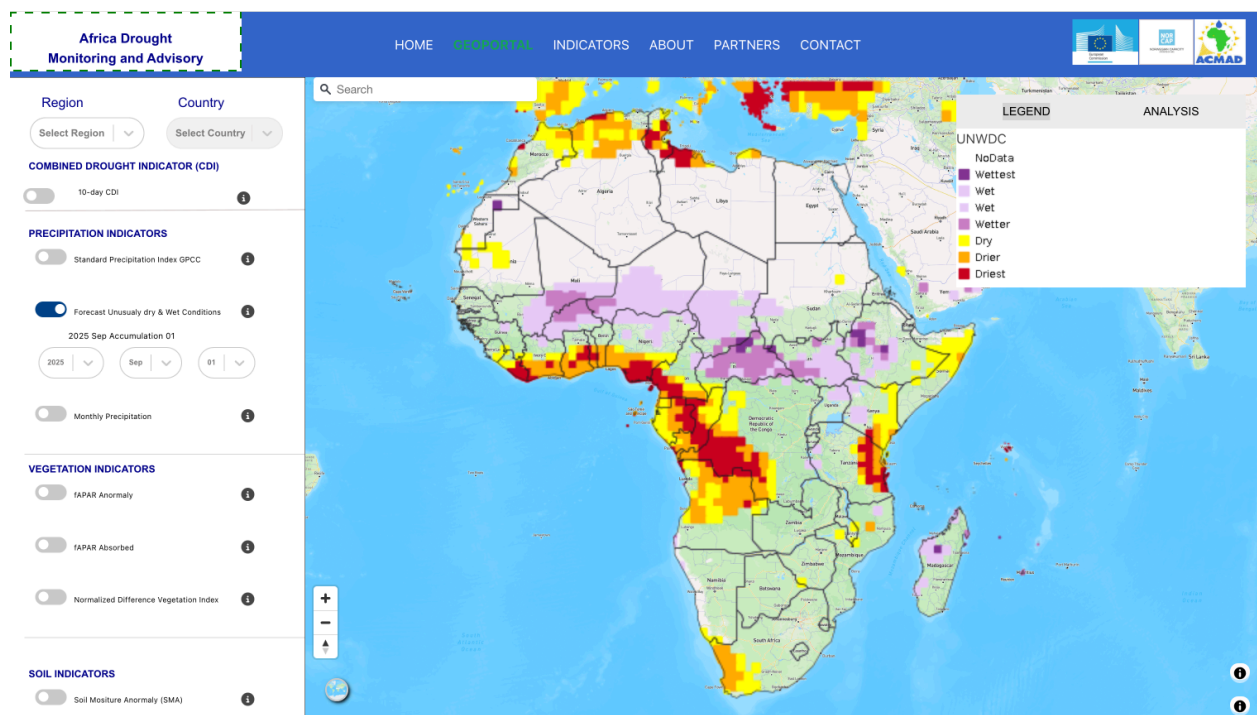
```
INFO:adma_data_pipeline.utils:Inserted 0 records and updated 3939 records. in 0.39824414253234
INFO:__main__:Data ingested successfully for unwdc pararms: grid grid_1dd, year 2025, month 7,
INFO:__main__:Total time taken: 0.7954740524291992 seconds
```

Task 3

Task three mainly entailed working to create a data processing workflow that produces a comprehensive drought report, computation of trend for monthly and seasonal rainfall. Integration of the new datasets as new entry to the user interface, Custom reporting at administrative levels as well as user defined area of interest. It further included implementation of user registration form and notification for registered users.

The activities undertaken for this task included:

- Extension of the python scripts to include threshold mechanisms that trigger a drought alert event.
- Scripts to produce the report
- Program that produces the monthly and seasonal rainfall outputs
- Entry of the datasets into the user interface
- implementation of map drawing tools to allow users to draw custom areas of interest (AOI) and upload their own shapefile.
- User registration module and notification functionality



Unusually wet and dry conditions forecast

List of data to be included in ADMA system

1. **Combined Drought Indicator:** A high level index that combines the precipitation soil moisture and vegetation indicators. Includes the existing combined drought indicator computed using GPCC, additional index computed using ERA 5 SPI, and the additional class that for 'Drought Projected to Persist'
2. **Monthly precipitation dataset**
3. **Standardized Precipitation Index (SPI)** - This indicator measures anomalies of accumulated precipitation during a given period (e.g., 1, 3, 12 months), and is the most commonly used indicator for detecting and characterizing meteorological droughts. Resolution is 1 decimal degree. ERA5 SPI dataset, that enhances the resolution to 0.25 decimal degrees and gives the user additional options on the platform due provision of updates every 10 days instead of monthly.
4. **Forecast Datasets**- The forecast dataset include the Unusually wet and dry conditions, the forecast polygons and the enhanced UNWDC dataset.
5. **Anomaly of Vegetation Condition (FAPAR Anomaly)** - This indicator measures anomalies of satellite-measured FAPAR (Fraction of Absorbed Photosynthetically Active Radiation), and is used to highlight areas of relative vegetation stress due to agricultural drought.
6. **Anomaly of Vegetation Condition (FAPAR Absorbed)** - The indicator captures vegetation productivity
7. **Soil Moisture Anomaly (SMA)** - This indicator measures anomalies of daily soil moisture (water) content, and is used to measure the start and duration of agricultural drought conditions.
8. **Normalized difference vegetation index** : A monthly product that captures the vegetation condition from Modis at 500 meters resolution.
9. **Protected Areas**: The dataset captures locations that are protected, this includes locations such as forests, National parks, wetlands and other protected areas.
10. **Geographic Background datasets**: Includes the Koppen Climate classification layer, land use land cover layer and Population dataset.

Task 4

The task denoted the presentation of the tool to the user community and official reporting on the functional system.

The detailed report of the system is annexed below, while additional information is also in the git repository of the system. The unveiling of the system to the public at the time of writing this report has not been done due to logistical constraints, we hope to conduct the same in the coming weeks. This requires planning and bringing the stakeholders together.

Conclusions

The implementation of the second phase was a success bringing together the collaborative effort from both teams to build a great product that drives impact and value proposition. We appreciate the resources afforded to ACMAD both time, expertise (technical and non technical), data access, archive datasets and database models design. During the project implementation phase which was very rapid we had a chance to review the expert led forecasting process. Where details of how sub-seasonal forecasts are done, the process opens up an opportunity to pursue this in a reproducible way.

The project extended the functionality of the system to address forecasting of drought and create early warning by enhancing the CDI class for areas under watch and warning to have a class for 'Drought projected to persist' this empowers downstream tasks and users empowering them in many areas, including energy, humanitarian, Agriculture among others.

Recommendations

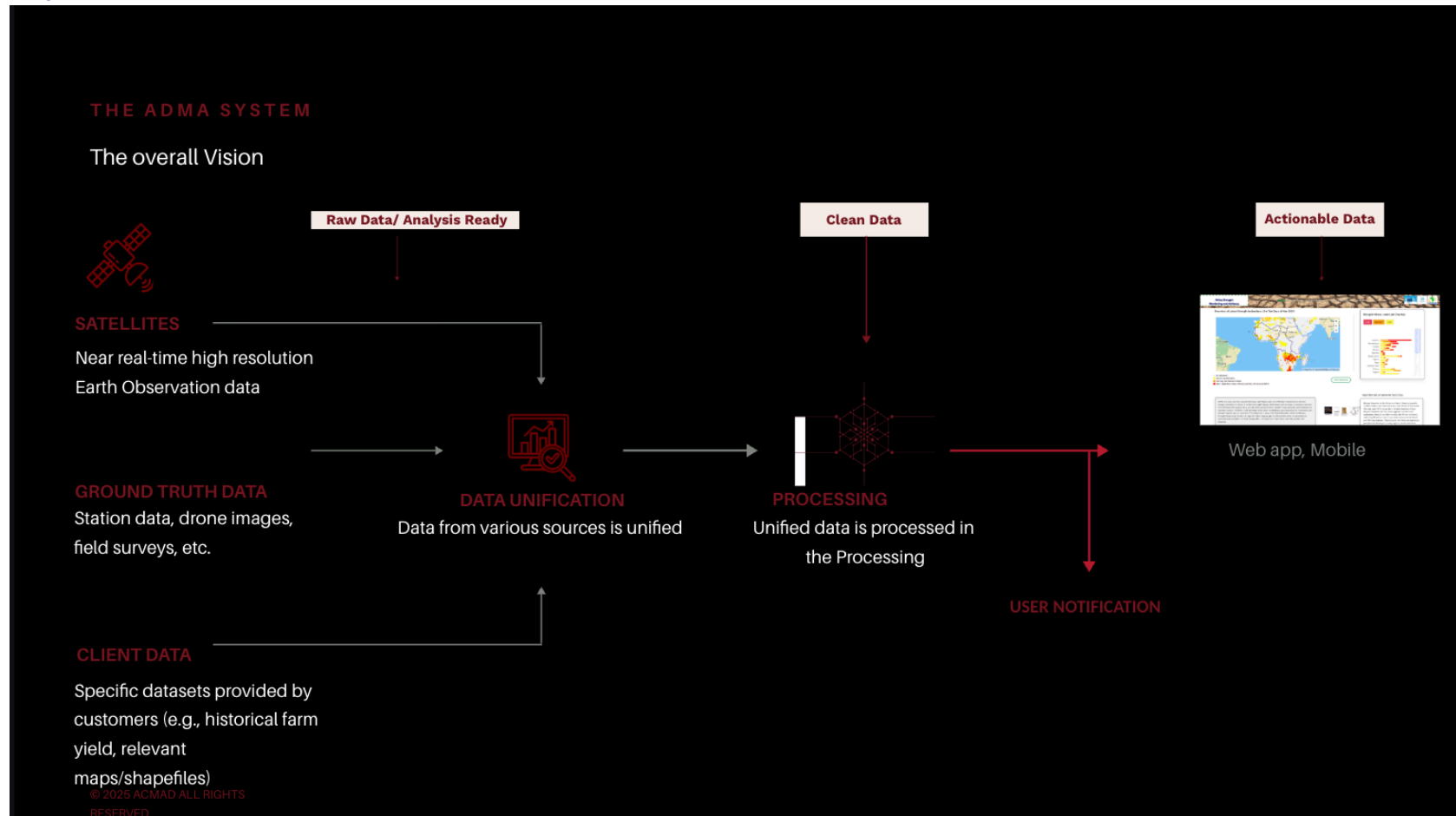
Future work as regards to the system should consider, development of and end to end process aimed and optimising the process of producing the sub-seasonal forecast, even though efforts at ACMAD have already begun where climate monitoring, the teleconnections datasets are housed in a dashboard integrated with the drought system. There is a need to develop a QGIS plugin that can allow for visualization, statistical analysis and generation of all the datasets that are used in the expert-led sub-seasonal forecast. The plugin shall be equipped with the capability to synchronize the output of the forecast with the system for realtime dissemination and updates. This will greatly enhance the process and act as a tool for training to other institutions, researchers or individuals who are interested in the process. Bringing about scalability and reusability of tools developed.

Integration of AI workflows, with the current wave of AI there is an opportunity to integrate AI both in Drought prediction and reporting functionalities, it may also be extended to provide custom advice,



foster education and effectively provide summaries of previous events of the same magnitude and the effects at that time, conversely helping the decision makers to have detailed information on what to do.

Project Overview





ANNEX D1.0 - Document with a description of the Working system, deployment and the Continuous integration and Continuous delivery pipeline.

Introduction

This section describes the working system, providing details of the function and non functional capabilities as well as the building blocks of the system.

System Documentation

The Africa Drought Advisory and Monitoring System is a comprehensive web-based platform designed to monitor, analyze, and report on drought conditions across Africa. The system provides near real-time drought monitoring capabilities, historical data analysis, and automated reporting features for decision-makers, researchers, institutions and other users. The second phase primarily extends the capabilities to drought forecasting to empower anticipatory action and disaster risk reduction.

Purpose and Goals

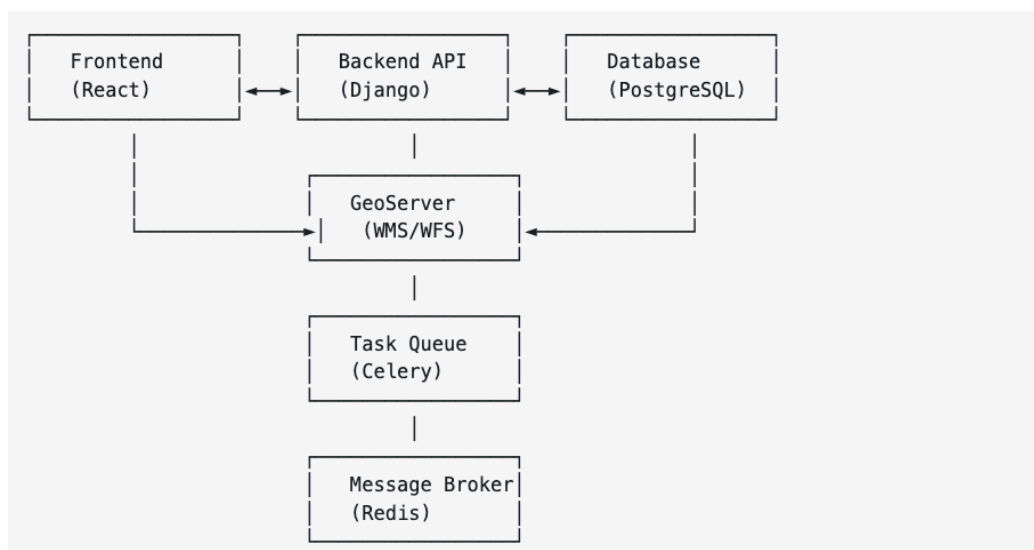
- Real-time Monitoring: Continuous monitoring of drought conditions using satellite-based remote sensing data
- Early Warning System: Provide timely alerts for potential drought conditions
- Historical Analysis: Analyze long-term drought patterns and trends
- Decision Support: Generate reports and visualizations for policy makers and researchers
- Data Integration: Integrate multiple data sources including precipitation, soil moisture, and vegetation indices

Key Features

- Interactive web-based dashboard with map visualizations
- Automated data ingestion from multiple data sources including JRC, google earth engine.
- Multi-level administrative boundary analysis (country and sub-national levels)
- Standardized Precipitation Index (SPI) calculations
- Combined Drought Index (CDI) generation
- PDF report generation capabilities
- RESTful API for data access
- User subscription and alert
- User authentication and access control (not in use currently)

Architecture Design

The system follows a microservices-oriented architecture with containerized components:



Design Principles

1. Separation of Concerns: Clear separation between frontend, backend, data processing, and visualization layers
2. Scalability: Horizontal scaling capabilities through containerization and load balancing
3. Modularity: Independent, loosely-coupled components that can be developed and deployed separately
4. Data-Driven: Centralized data management with multiple access patterns
5. Service-Oriented: RESTful API design for external integrations
6. Automated Processing: Background task processing for data ingestion and **analysis**

Component Interaction Patterns

- Request-Response: Frontend ↔ Backend API communication
- Publish-Subscribe: Task scheduling and execution via Celery
- Data Pipeline: ETL processes for satellite data ingestion
- Service Mesh: Inter-service communication through Docker networks

Technology Stack

The technology used in this implementation can be classified into backend and frontend:

Backend Technologies

- Framework: Django 4.1+ (Python web framework)
- API Framework: Django REST Framework (DRF)
- CMS: Wagtail (Content Management System)
- Database: PostgreSQL 12.0 with PostGIS extension

- Task Queue: Celery with Redis broker
- Geospatial: PostGIS, GDAL, Rasterio
- Python package for data processing and ingestion
- External APIs: Google Earth Engine API

Africa Drought Monitoring and Advisory Api 1.0.0 OAS 3.0

/api/schema/

API Drought Monitoring, forecasting and Advisory

ACMAD Dev Team - Website
Send email to ACMAD Dev Team
MIT

Authorize 

data_api

GET	/api/data_api/api_add_data/	 
POST	/api/data_api/api_add_data/	 
GET	/api/data_api/api_add_data/{id}/	 
PUT	/api/data_api/api_add_data/{id}/	 
PATCH	/api/data_api/api_add_data/{id}/	 
DELETE	/api/data_api/api_add_data/{id}/	 
POST	/api/data_api/api_add_data/add_api_data/	 
GET	/api/data_api/api_update/	 
POST	/api/data_api/api_update/	 
GET	/api/data_api/api_update/{id}/	 
PUT	/api/data_api/api_update/{id}/	 

```
Body Cookies Headers (13) Test Results 200 OK 139 ms 12.87 KB Save Response
{} JSON Preview Visualization
1 {
2   "cdi": {
3     "latest_year": 2025,
4     "latest_month": 9,
5     "latest_dekad": 11,
6     "start_year": 2015,
7     "2015": {
8       "months": [
9         "01",
10        "02",
11        "03",
12        "04",
13        "05",
14        "06",
15        "07",
16        "08",
17        "09",
18        "10",
19        "11",
20        "12"
21      ],
22      "dekads": [
23        "1",
24        "11",
25        "21"
26      ]
27    },
28    "2016": {
29      "months": [
30        "01",
31        "02",
32        "03",
33        "04",
34        "05",
35        "06",
```

Frontend Technologies

- Framework: React 18.2+ (JavaScript library)
- Mapping: MapLibre GL JS
- Styling: Bulma CSS framework
- Charts: Chart.js, ECharts, Recharts
- Build Tool: Create React App (CRA)
- HTTP Client: Axios

Infrastructure & DevOps

- Containerization: Docker & Docker Compose
- Web Server: Nginx (reverse proxy and static file serving)

- SSL/TLS: Certbot for Let's Encrypt certificates
- Caching: Redis (in-memory data structure store)
- Monitoring: Flower (Celery monitoring)
- Geospatial Server: GeoServer 2.26.1

Development Tools

- Version Control: Git
- Testing: Django TestCase, React Testing Library
- Documentation: OpenAPI/Swagger (drf-spectacular)
- Code Quality: Built-in Django and React linting

System Components

1. Backend API (Django Application)

Core Modules

- `DroughtAdvisory/`: Main project configuration
 - Settings management (development/production)
 - URL routing and middleware configuration
 - Celery configuration for asynchronous tasks
- `data_api/`: Primary data processing and API module
 - Models for geospatial data and metadata
 - API viewsets for data access
 - Google Earth Engine integration
 - Report generation utilities

- `users/`: User authentication and management
 - Custom user model with profile support
 - Authentication views and forms
 - Permission and access control

- `common_apps/`: Shared functionality
 - Metadata management
 - Common utilities and helpers

Key Features

- Geographic Data Models: Admin boundaries, grid systems, satellite imagery URLs
- API Endpoints: RESTful endpoints for data retrieval and analysis
- Background Processing: Celery tasks for data ingestion and processing
- Report Generation: Automated PDF report creation
- User Management: Authentication, authorization, and user profiles

2. Frontend Application (React)

Component Structure

components/: Reusable UI components

- Geoportal.js: Main mapping interface
- reporting/: Report generation and visualization
- sidepane/: Control panels and filters
- home/: Landing page components

Key Features

- Interactive Maps: MapLibre-based geospatial visualization
- Data Visualization: Charts and graphs for drought indices
- Report Generation: Client-side PDF generation with html2canvas
- Responsive Design: Mobile-friendly interface with Bulma CSS
- Real-time Updates: Dynamic data loading and filtering

3. Database Layer (PostgreSQL + PostGIS)

Core Tables

- Administrative Boundaries:
 - data_api_adminlevelzero: Country-level boundaries

- `data_api_adminlevelone`: Sub-national administrative units
- Grid Systems
 - Multiple resolution grids (1°, 0.1°, 0.083°, 005) for spatial analysis
 - Grid-to-admin unit mapping for aggregation
- Satellite Data
 - Image URLs and metadata from Google Earth Engine
 - Product information and temporal data
- User Management:
 - User profiles and authentication data
 - Access control and permissions
 -

4. GeoServer Integration

a. Configuration

- i. Workspace: `africa_drought_datasets`
- ii. Data Sources: Direct PostGIS connection
- iii. Services: WMS and WFS for map visualization
- iv. Extensions: Query layer plugin for advanced filtering

b. Features

- i. Map Services: Styled map layers for drought indices
- ii. Data Access: Web Feature Service (WFS) for geospatial data
- iii. Performance: Optimized rendering and caching
- iv. Integration: Seamless connection with PostgreSQL database

5. Task Processing (Celery + Redis)

Task Categories

- Data Ingestion: Automated satellite data
- Index Calculation: Background processing of drought indices
- Report Generation: Scheduled report creation and distribution
- Database Maintenance: Data cleanup and optimization tasks
- Ssl certificate renewal

Other systemwide detailed documentation may be found on the systems github repository where we describe the development and production environments, the security and networking components. We also endeavored to create a continuous integration and continuous delivery pipeline in this implementation. Where code changes are automatically tested and deployed to the server instances if peer review is done and approved.

ACMAD-Niamey / Drought_advisory_system

Q Type to search

<> Code Issues Pull requests Actions Projects Security Insights Settings

Actions

New workflow

All workflows

Copilot

drought-system-ci

Management

- Caches
- Attestations
- Runners
- Usage metrics
- Performance metrics

All workflows

Showing runs from all workflows

Filter workflow runs

Help us improve GitHub Actions

Tell us how to make GitHub Actions work better for you with three quick questions.

Give feedback

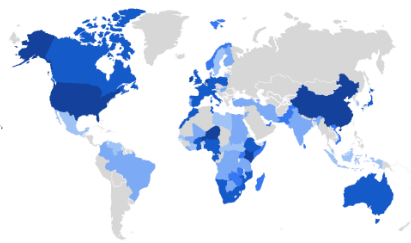
35 workflow runs

	Event	Status	Branch	Actor
✓ Merge pull request #25 from ACMAD-Niamey/copil... drought-system-ci #37: Commit 4b8c473 pushed by Transega	develop	51 minutes ago 5m 24s		
✓ Running Copilot Copilot #1: by Copilot AI	copilot/fix-d3670898-dd11...	Sep 29, 3:55 PM GMT+3 13m 4s		
✓ Merge pull request #24 from ACMAD-Niamey/in... drought-system-ci #33: Commit be7f665 pushed by Transega	develop	Sep 25, 2:58 PM GMT+3 6m 23s		
✓ handle legend unwdc drought-system-ci #32: Pull request #24 opened by Transega	integrate-data-ui	Sep 25, 2:57 PM GMT+3 5m 38s		
✓ Merge pull request #23 from ACMAD-Niamey/in... drought-system-ci #31: Commit f59412f pushed by Transega	develop	Sep 25, 2:39 PM GMT+3 6m 19s		
✓ handle unwdc on the ui for dem drought-system-ci #30: Pull request #23 opened by Transega	integrate-data-ui	Sep 25, 2:38 PM GMT+3 4m 9s		

CI/CD pipeline

Web traffic

Active users by Country



COUNTRY	ACTIVE USERS
United States	98
Netherlands	70
Ireland	62
Kenya	53
China	42
Niger	41
France	22

[View countries →](#)

Active users by City

CITY	ACTIVE USERS
Amsterdam	64
Dublin	63
Nairobi	50
Niamey	41
Libreville	22
Paris	14
London	13

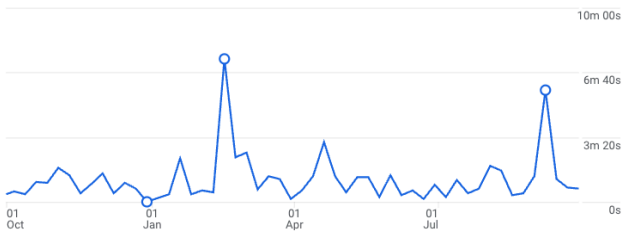
[View cities →](#)

Average engagement time per active user

1m 41s

Engaged sessions per active user

1.9



Event count by Event name

EVENT NAME	EVENT COUNT
page_view	8.3K
scroll	1.8K
session_start	1.5K
user_engagement	1.2K
first_visit	626
click	18
form_start	16

[View events →](#)

Views by

PAGE TITLE AND S...	VIEWS
Africa Drought Advis...	2.4K
/	2.2K
geoportal	2.2K
/home	580
home	557
reporting	199
Alerta de seca na Áfr...	26

[View pages and screens →](#)

Users tracking per country