PRESENTATION

Summer Intern In Parallel Computational Science (SIParCS)

Title: Microservice-Driven IoT Architecture for Atmospheric Sensonets and Real-Time Visualization

Date: July 29, 2025

Presented by: Isaac Oppong-Baah

Mentors: Agbeli Ameko, Keith Maull, John Schreck

About Me

Professional Background

- Software Engineer
- Manufacturing Controls Engineer

Academic Background

- MSc Electronics Engineering.
- BSc Electrical and Electronics Engineeering.

Research Interests: AI/ML for real-time fault monitoring, neuromorphic computing, memristors, and semiconductor process optimization.

Fun Facts

I love to play TV and board games.

SALESFORCE CERTIFIED

IEEE

THE NATIONAL SOCIETY OF BLACK ENGINEERS

Outline

Introduction & Background

Hello, this is Joe!

- Joe lives in a small town adjacent to an industrial site.
- He is concerned about his family's well-being.
- Local reports raised suspicions of airborne pollutants.
- He sought concrete evidence — but found none.

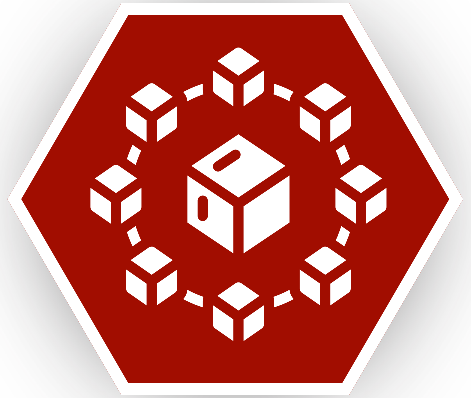
This sparked a grassroots effort: deploying low-cost, community-managed sensor (mesonets) to track environmental data in real time.

This project builds the technical foundation for that vision.

Project Objectives

Build scalable, low-cost mesonets

using **LoRa**-enabled sensonet stations and **RaspberryPi** gateways



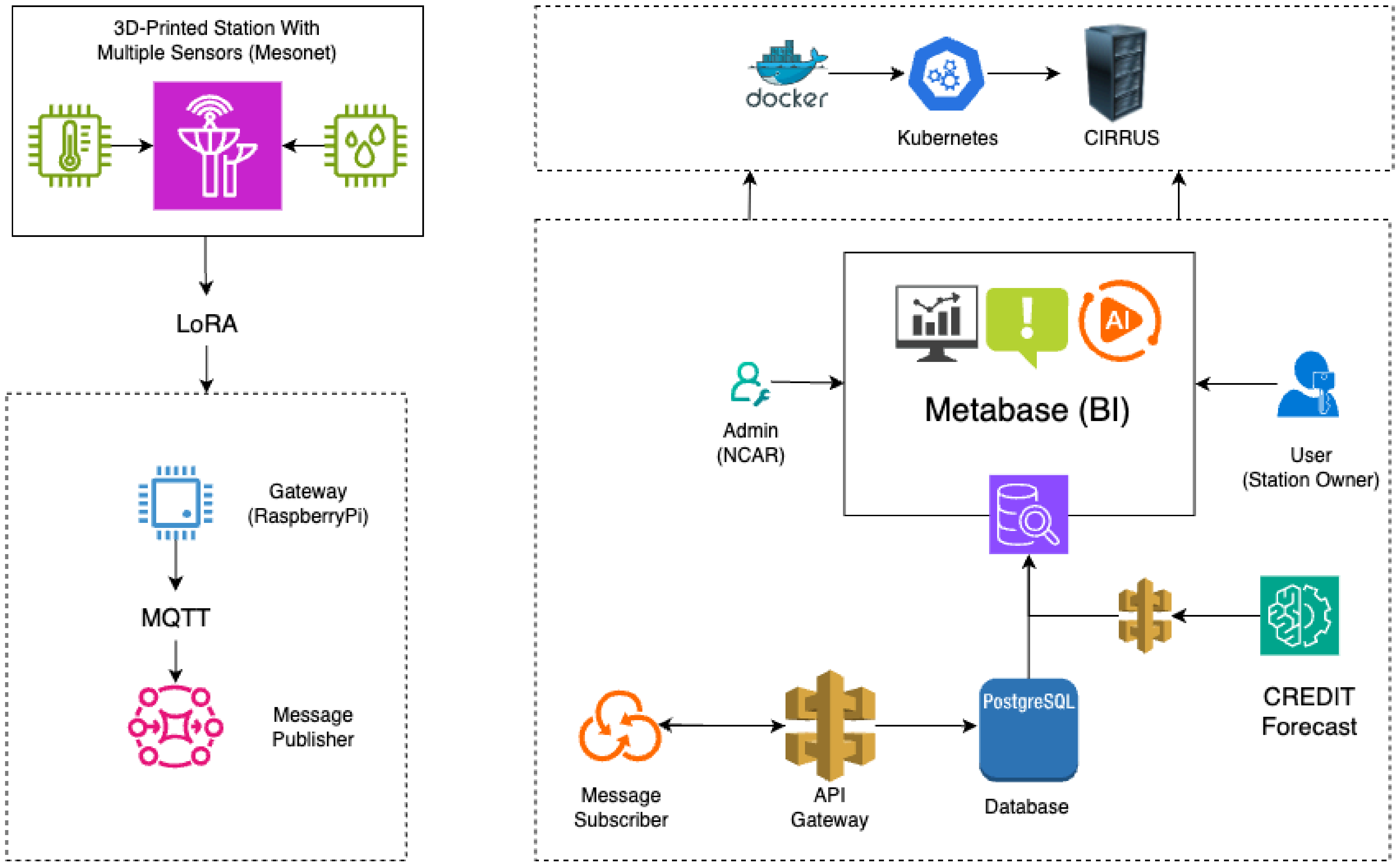
Develop a robust microservice architecture

to ingest, store, and serve sensor data through **MQTT**, **FastAPI**, and **PostgreSQL** using **Docker** containers and **Kubernetes**

Empower communities and researchers

through automated **Metabase** dashboards and integration of **CREDIT** global weather forecast model and hosting on **CIRRUS**

System Architecture



01 Edge Sensing & Transmission

3D-printed weather stations equipped with LoRa radios collect environmental data and transmit it wirelessly to a Raspberry Pi gateway running an MQTT message broker

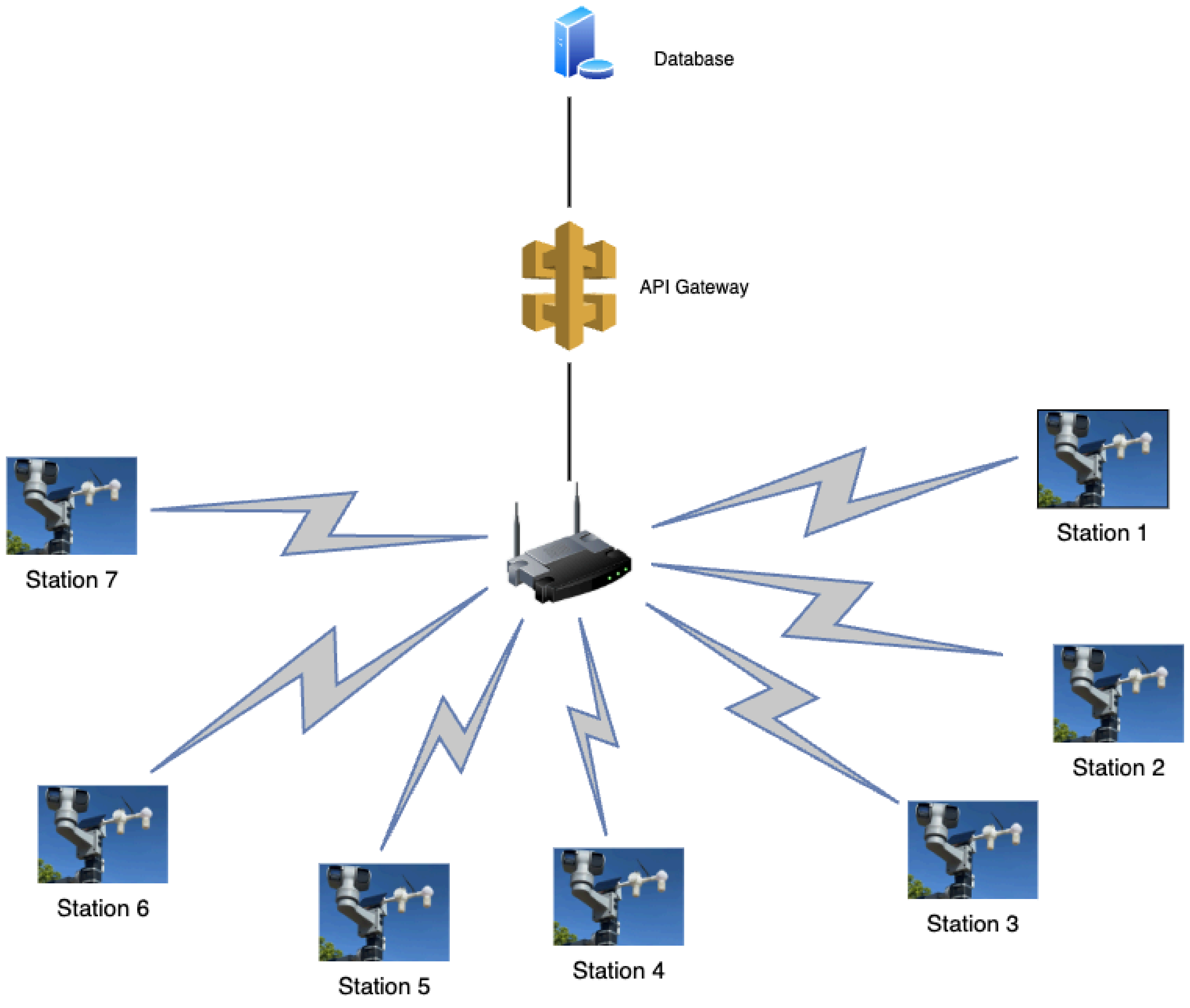
02 Microservice-Based Backend

Asynchronous message subscriber receives sensor payloads, processes them through FastAPI-based API services, and stores them in a PostgreSQL database alongside daily CREDIT forecast outputs.

03 Visualization & Orchestration

Data is visualized through auto-generated Metabase dashboards accessible by NCAR admins and station owners. The entire stack is containerized using Docker and orchestrated on Kubernetes (CIRRUS).

Data Flow via MQTT



01 Sensor Payload Publishing

Each Raspberry Pi gateway publishes readings to an MQTT broker using a lightweight protocol ideal for low-power devices and unstable networks

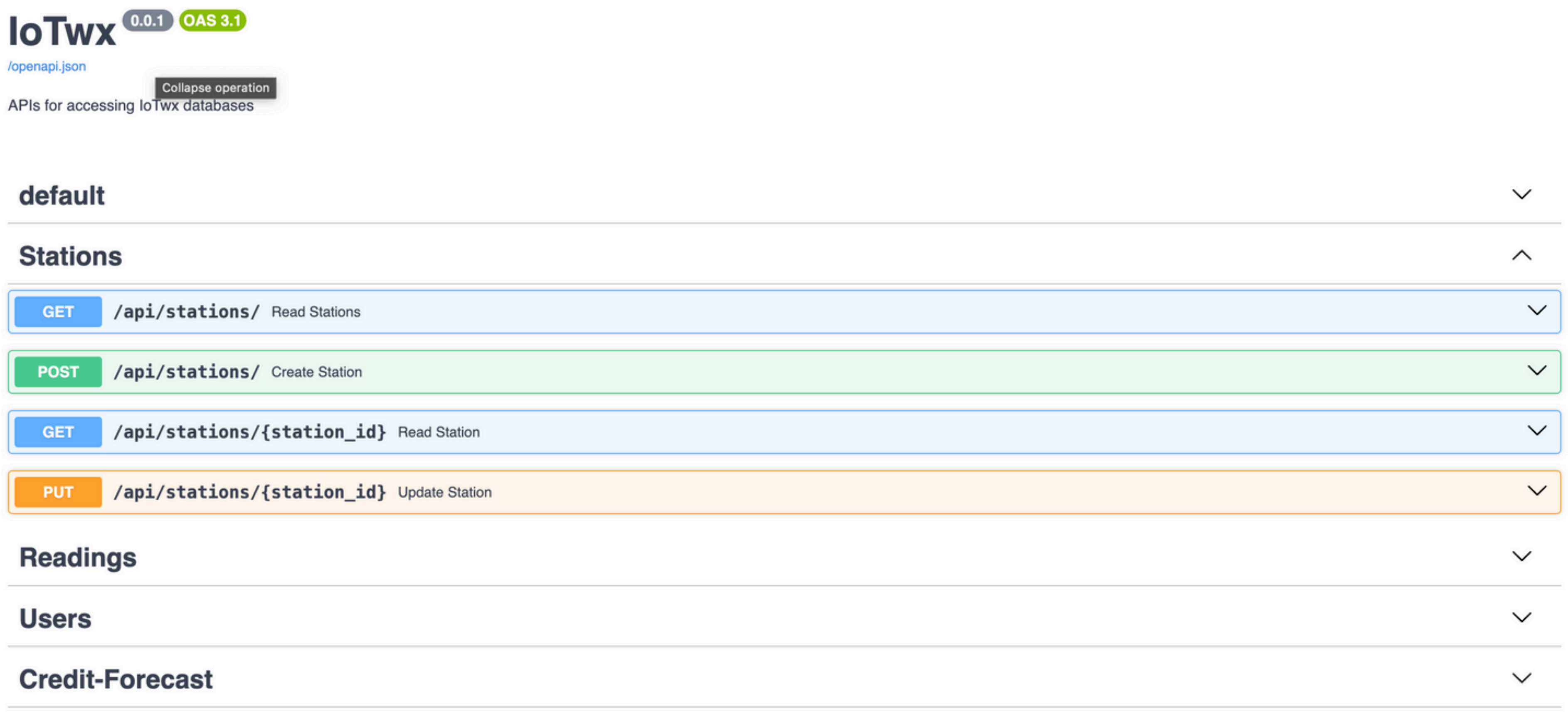
02 Custom Message Orchestrator

An asynchronous subscriber listens on relevant MQTT topics, decodes payloads, validates structure, and enriches data with metadata and timestamps.

03 Reliable Backend Handoff

The processed readings are sent to FastAPI services for insertion into the PostgreSQL database — ensuring fault tolerance and format consistency.

Backend Services and Data Modeling



01 FastAPI Microservices

RESTful API endpoints manage tasks such as station registration, sensor data insertion, and forecast ingestion — with clean async handling for scalability.

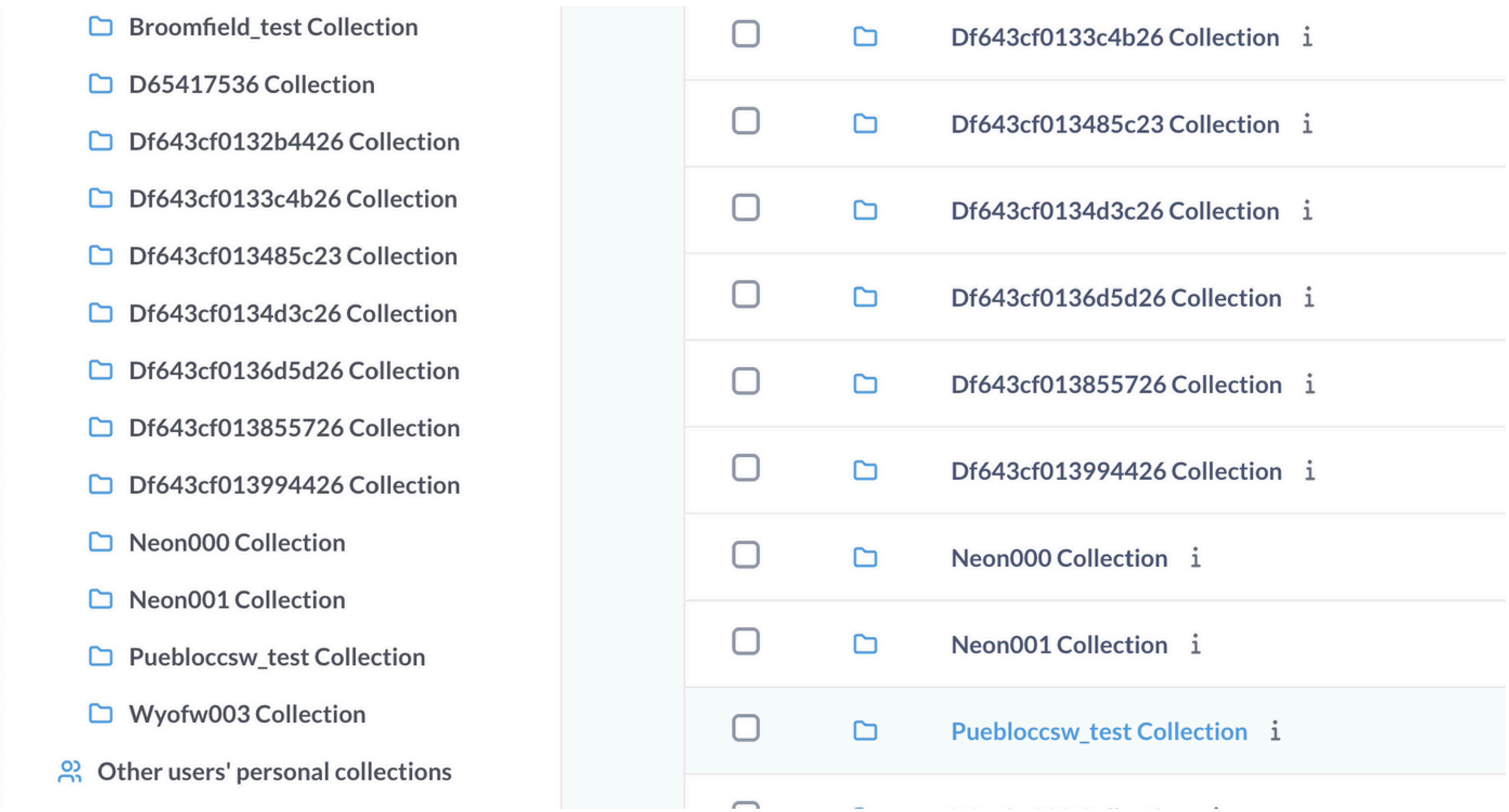
02 PostgreSQL Data Model

A schema stores payloads for Stations (location, owner, metadata, etc), Readtings(value, sensor type, measuremet, timestamp, etc), Users(email, name, etc), CreditForecast (forecast values, timestamp, etc)

03 Migration & Validation

Alembic manages schema evolution over time, while pydantic models enforce structure and validate incoming data before committing to the database.

Metabase Integration



01 Metabase Integration via API

Backend services automatically create users, collections, models, dashboards, and cards in Metabase — enabling real-time access to station data without manual setup.

02 Role-Based Access

Admins (e.g., NCAR staff) see all stations and forecast comparison dashboards.
Station Owners are only shown dashboards relevant to their station(s), improving privacy and focus.

03 Interactive Visualizations

Users can explore readings, time series trends, and CREDIT model forecasts via intuitive charts and maps — even with minimal technical experience.

Metabase was selected for its open-source flexibility, intuitive dashboards, which reduces the technical barriel between stakeholders and external station collaborators.

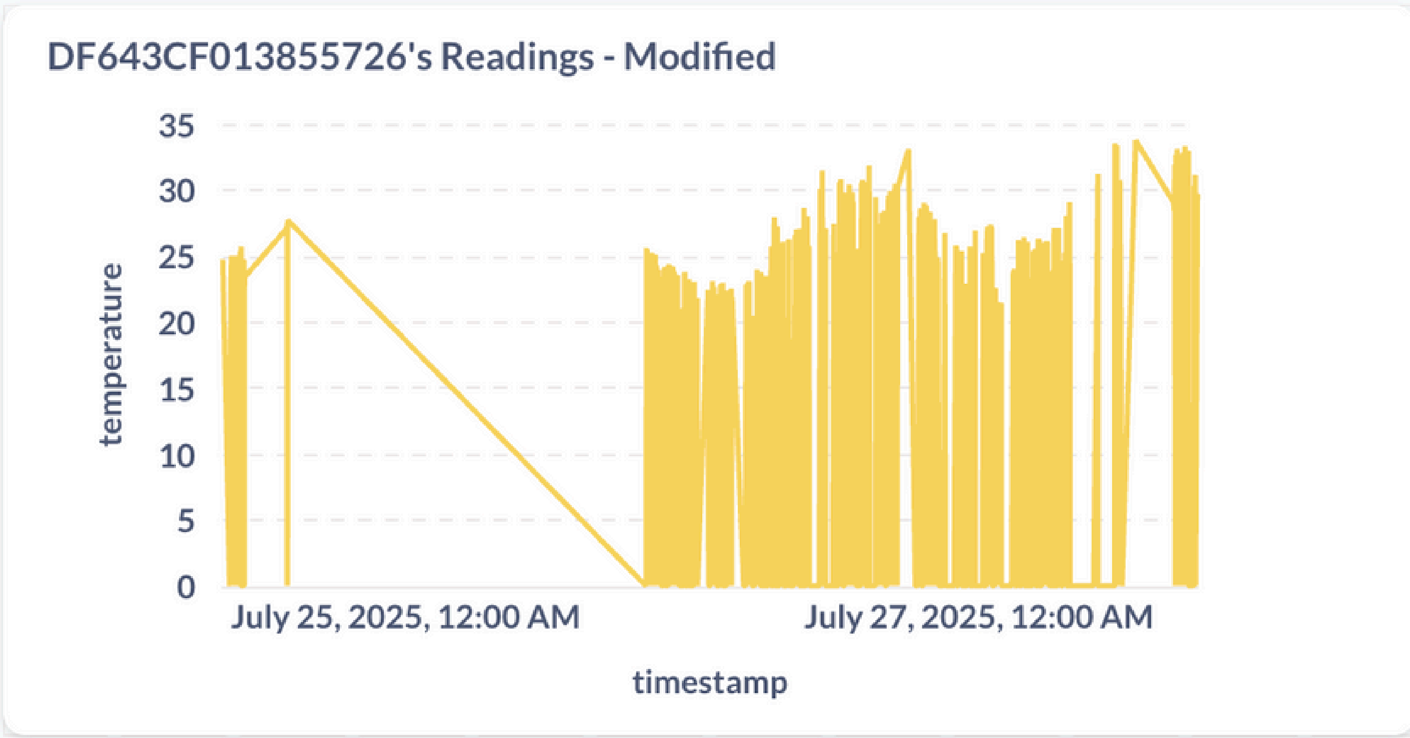
Metabase Visualization

DF643CF0133C4B26's Readings

This question is written in SQL.

station_id	timestamp	gas_resistance	pressure	relative_humidity	temperature
DF643CF0133C4B26	July 27, 2025, 2:53 PM	30.75	815.26	32.26	25.09
DF643CF0133C4B26	July 27, 2025, 2:52 PM	30.35	815.25	0	24.45
DF643CF0133C4B26	July 27, 2025, 2:51 PM	0	0	32.25	0
DF643CF0133C4B26	July 27, 2025, 2:49 PM	0	0	0	0
DF643CF0133C4B26	July 27, 2025, 2:48 PM	0	0	0	0
DF643CF0133C4B26	July 27, 2025, 2:47 PM	31.33	815.2	32.23	25.1
DF643CF0133C4B26	July 27, 2025, 2:46 PM	0	0	0	24.45
DF643CF0133C4B26	July 27, 2025, 2:45 PM	0	0	32.23	25.1
DF643CF0133C4B26	July 27, 2025, 2:44 PM	0	815.25	32.23	25.1
DF643CF0133C4B26	July 27, 2025, 2:43 PM	31.33	815.26	0	24.45
DF643CF0133C4B26	July 27, 2025, 2:42 PM	0	0	0	24.45
DF643CF0133C4B26	July 27, 2025, 2:41 PM	31.07	815.27	32.22	24.45
DF643CF0133C4B26	July 27, 2025, 2:40 PM	0	0	0	24.45
DF643CF0133C4B26	July 27, 2025, 2:39 PM	0	0	0	0
DF643CF0133C4B26	July 27, 2025, 2:38 PM	30.6	815.27	32.19	25.1
DF643CF0133C4B26	July 27, 2025, 2:37 PM	30.31	815.28	0	24.45
DF643CF0133C4B26	July 27, 2025, 2:36 PM	0	815.26	32.18	0

Visualization



Miles CREDIT Integration

station_id	prediction_time	forecast_date	forecast_temperature	observed_temperature
wyofw003	July 23, 2025, 12:00 AM	July 27, 2025, 12:00 PM	10.15	19.03
wyofw003	July 23, 2025, 12:00 AM	July 29, 2025, 6:00 AM	14.49	19.03
wyofw003	July 23, 2025, 12:00 AM	July 28, 2025, 12:00 PM	9.72	19.03
wyofw003	July 23, 2025, 12:00 AM	July 26, 2025, 12:00 PM	8.69	19.03
wyofw003	July 23, 2025, 12:00 AM	July 26, 2025, 6:00 AM	13.35	19.03
wyofw003	July 23, 2025, 12:00 AM	July 24, 2025, 12:00 PM	9.52	19.03
wyofw003	July 23, 2025, 12:00 AM	July 29, 2025, 12:00 PM	10.83	19.03
wyofw003	July 23, 2025, 12:00 AM	July 28, 2025, 6:00 PM	23.31	19.03
wyofw003	July 23, 2025, 12:00 AM	July 24, 2025, 6:00 AM	11.72	19.03
wyofw003	July 23, 2025, 12:00 AM	July 26, 2025, 12:00 AM	24.88	19.03
wyofw003	July 23, 2025, 12:00 AM	July 26, 2025, 6:00 PM	23.81	19.03
wyofw003	July 23, 2025, 12:00 AM	July 28, 2025, 12:00 AM	24.73	19.03
wyofw003	July 23, 2025, 12:00 AM	July 27, 2025, 12:00 AM	25.87	19.03
wyofw003	July 23, 2025, 12:00 AM	July 25, 2025, 12:00 PM	8.3	19.03
wyofw003	July 23, 2025, 12:00 AM	July 24, 2025, 6:00 PM	18.11	19.03
wyofw003	July 23, 2025, 12:00 AM	July 23, 2025, 6:00 AM	17.59	19.03

01 CREDIT Forecast Integration

Each day, a forecasting service runs the CREDIT mode four times(00:00, 06:00, 12:00, 18:00 UTC) as a job to generate predictions for each station’s location and forecast date.

02 CREDIT Grid Analysis

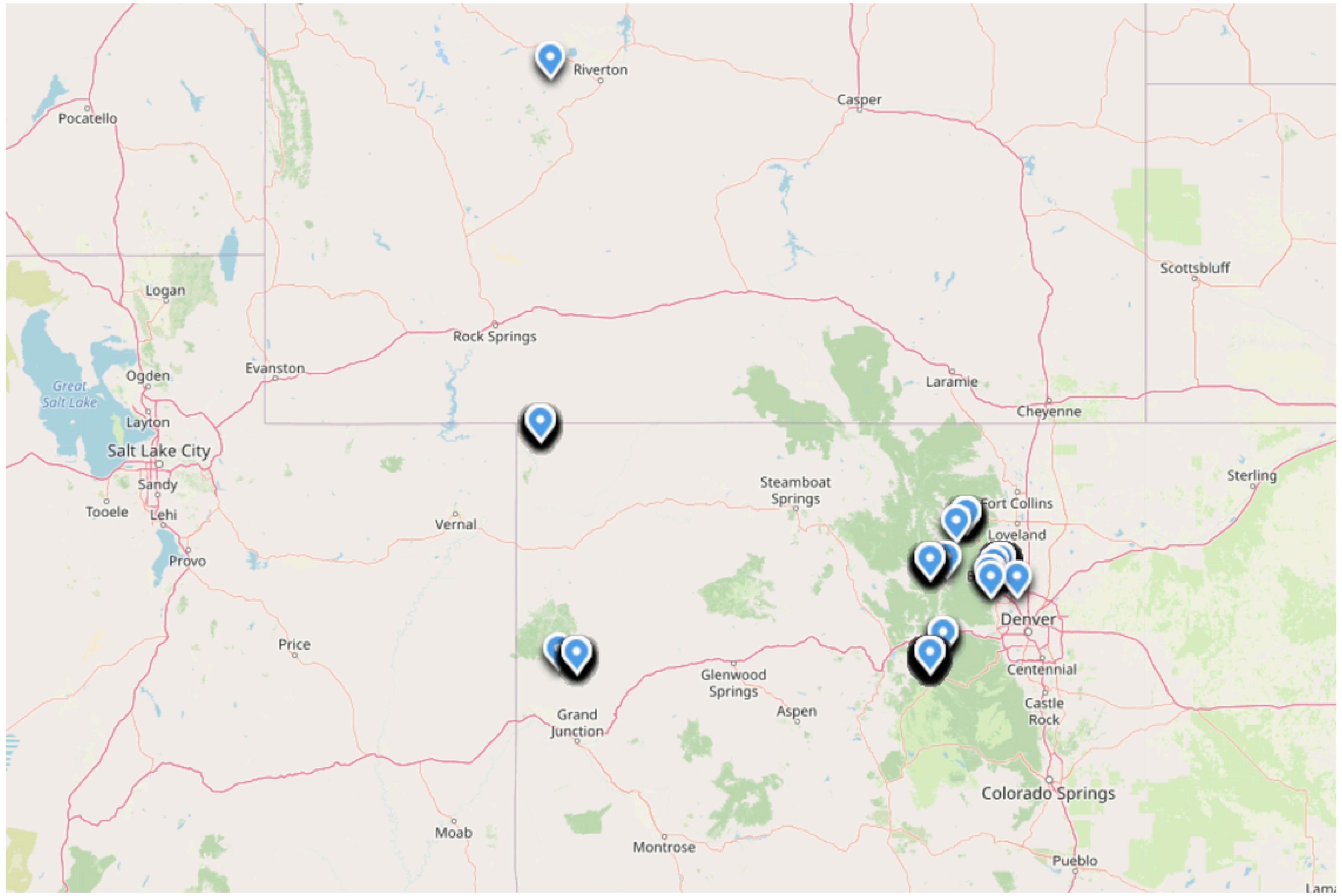
Each station’s location is matched to the closest grid point in the forecast model using Euclidean distance

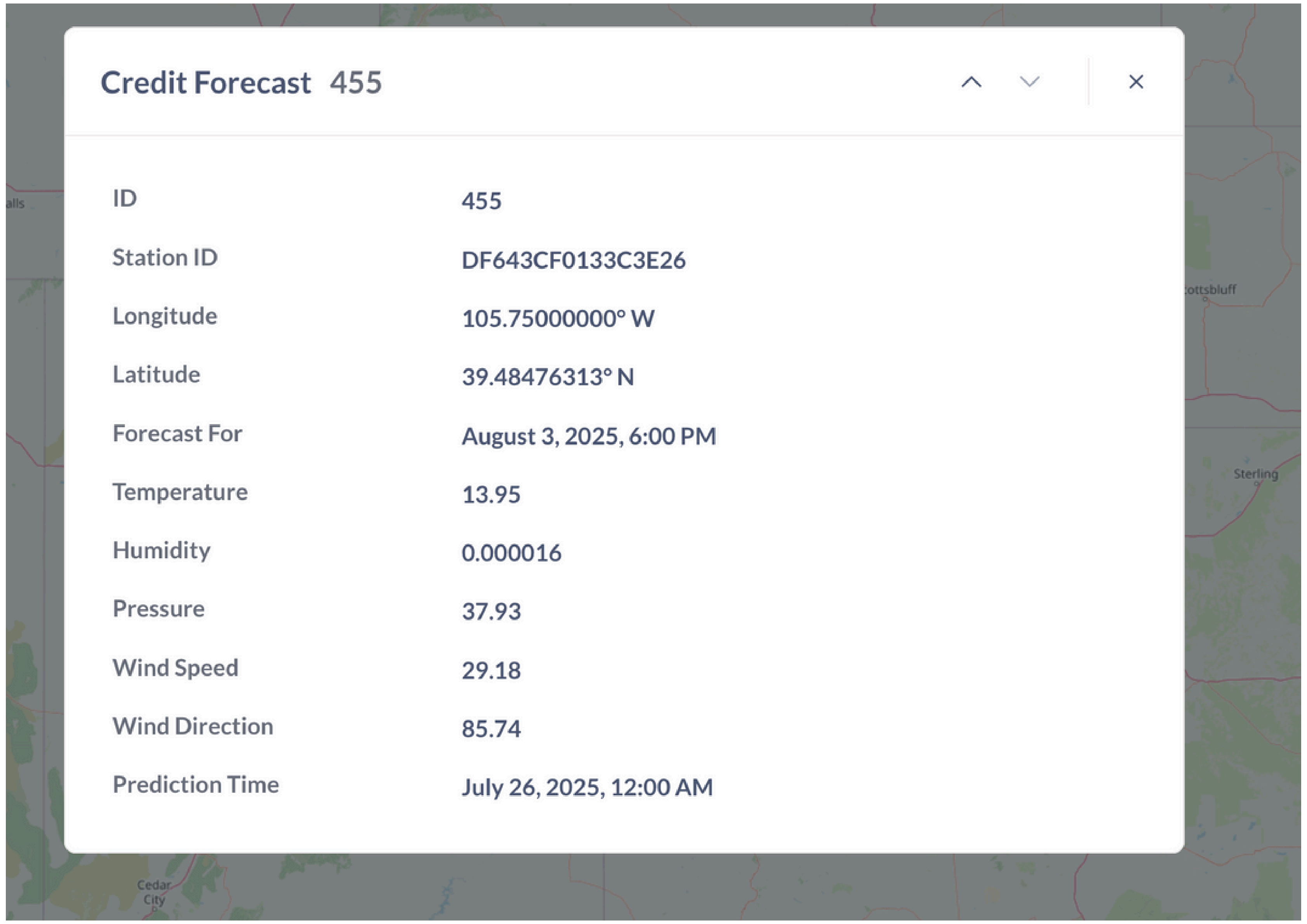
$$d = \sqrt{(lat_{\text{grid}} - lat_{\text{station}})^2 + (lon_{\text{grid}} - lon_{\text{station}})^2}$$

03 Synchronized Storage & Evaluation

Forecasts and observed readings are stored with consistent timestamp and station references

Forecast Visualization

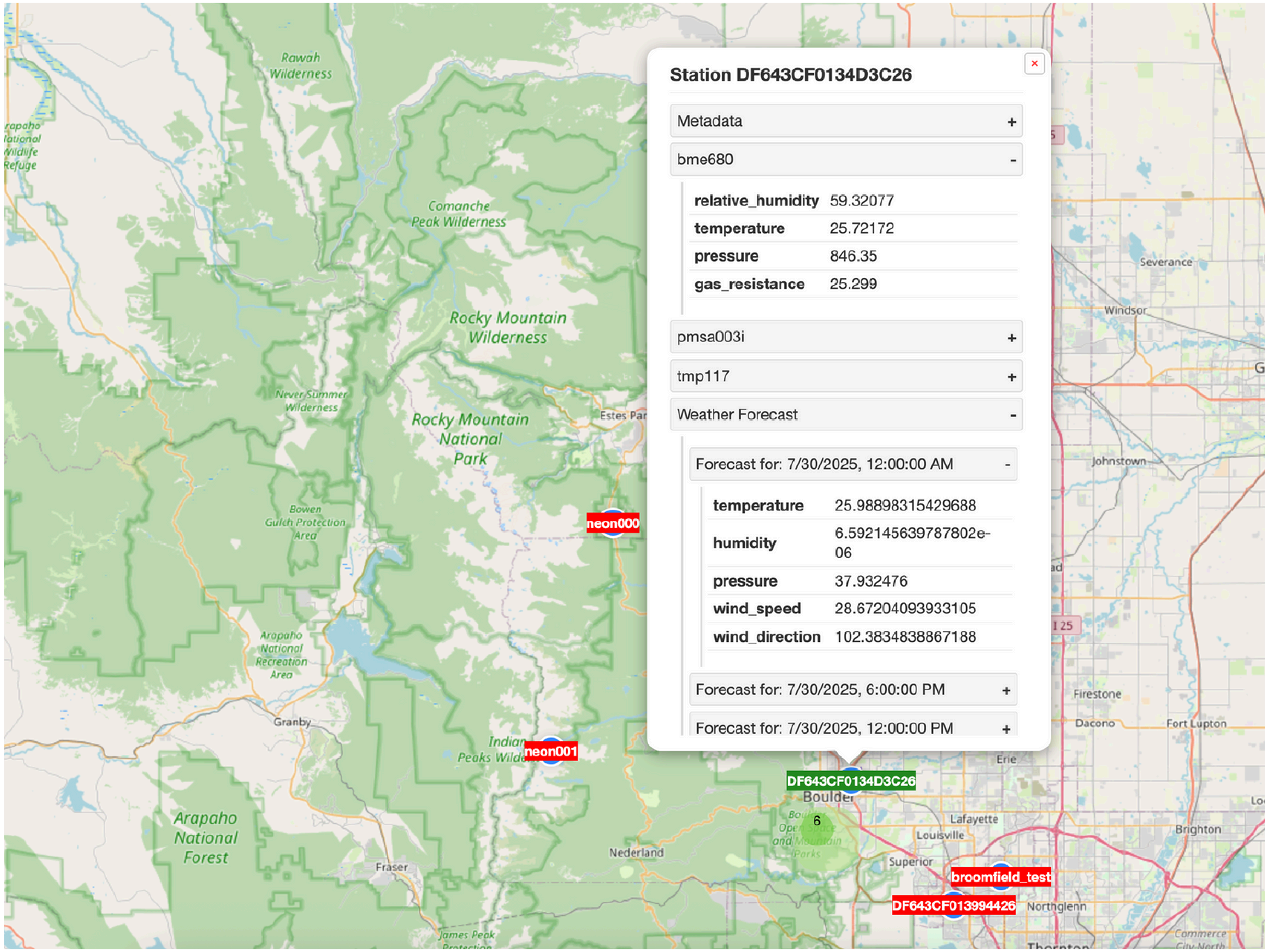




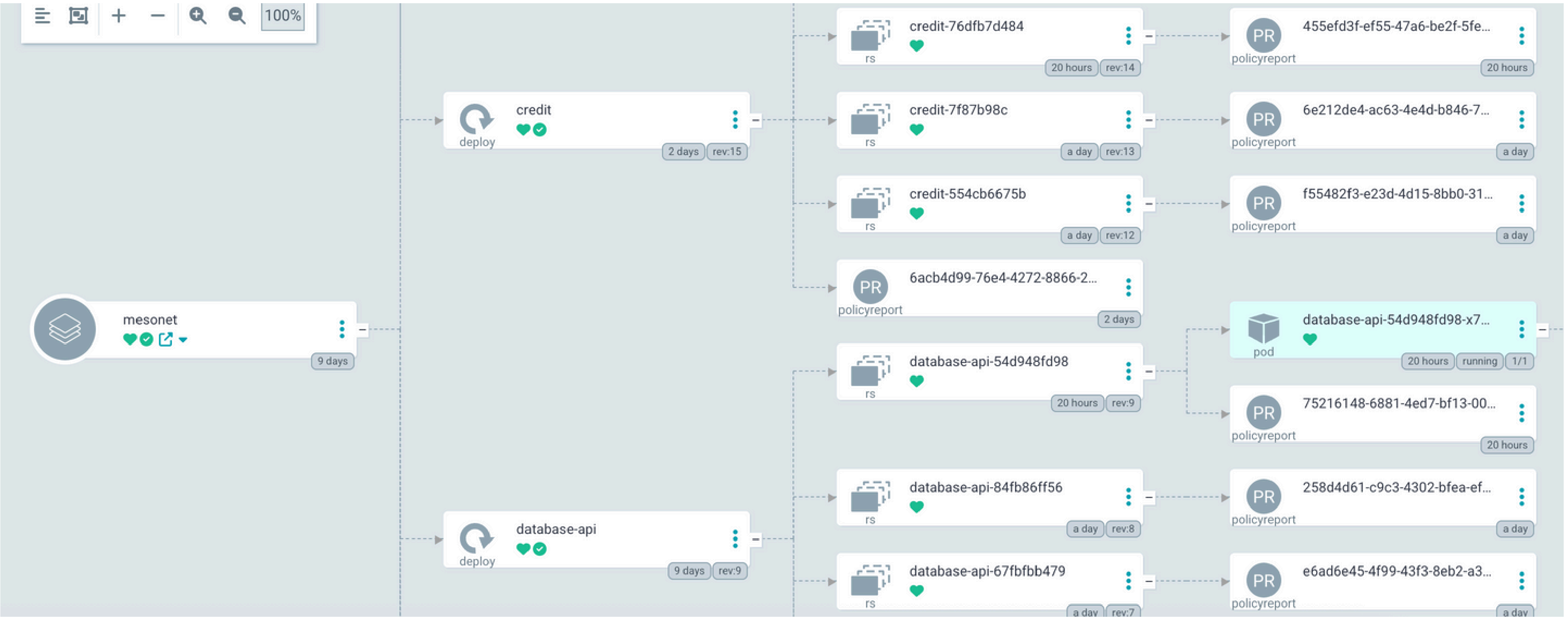
Credit Forecast 455

ID	455
Station ID	DF643CF0133C3E26
Longitude	105.75000000° W
Latitude	39.48476313° N
Forecast For	August 3, 2025, 6:00 PM
Temperature	13.95
Humidity	0.000016
Pressure	37.93
Wind Speed	29.18
Wind Direction	85.74
Prediction Time	July 26, 2025, 12:00 AM

Forecast Visualization



Deployment on CIRRUS



Pods & Services	Ingresses	Persistence Volumes	Config Maps
34	8	6	3

01 Dockerized Microservices

Services are packaged as an isolated Docker container for reproducibility and portability across environments. The containers are also packaged in a docker compose file for easy reproduceability.

02 Kubernetes-Based Orchestration

Services are deployed and managed using Helm charts on the CIRRUS cluster, supporting automated scaling, service discovery, and health monitoring.

03 CI/CD for DevOps

The architecture supports continuous integration and delivery pipelines — enabling rapid updates to code and configuration, and aligning with UCAR's internal deployment standards.

Impact & Significance

Empowering Communities Through Data Transparency

Democratized Environmental Monitoring

Enables individuals, schools, and underserved communities to deploy affordable weather stations, generate real-time data, and independently assess local atmospheric conditions.

Actionable Insight for Advocacy & Research

Residents affected by industrial or environmental risks can collect evidence-backed data to support policy discussions, while researchers benefit from scalable, open-source sensor infrastructure.

Scalable, Replicable Framework

The architecture supports rapid onboarding of new stations and users, making it suitable for regional-scale deployments — from citizen science to institutional research networks.

Challenges & Future Works

Automating Metabase

Creating users, assigning collection-specific permissions, and linking dashboards to the correct roles via Metabase's REST API required deep exploration of undocumented behaviors and manual API workflows.

Running & Containerizing CREDIT

Porting the Miles CREDIT model to macOS with MPS GPU acceleration and containerizing it in Docker and deploying it on CIRRUS with Kubernetes marked a major milestone.

Toward Scalable, Self-Served Deployments

Future improvements include a self-serve onboarding portal for new station owners, enhanced auth via API tokens, and more robust logging & retry mechanisms in the data pipeline.

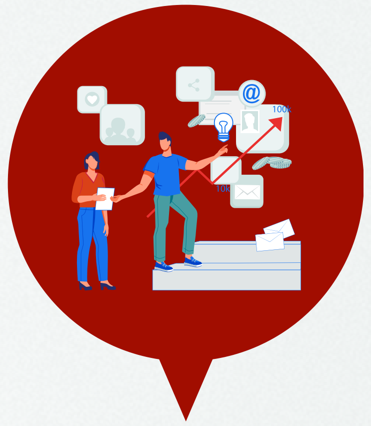
Acknowledgments

Adebowale Adelekan

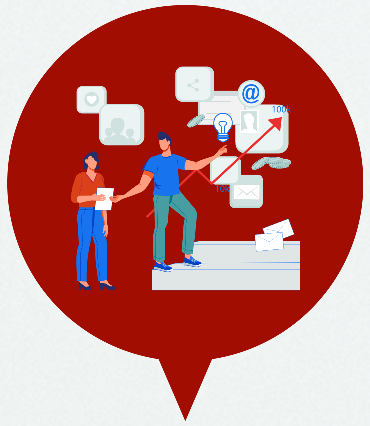
Agbeli Ameko

Keith Maull

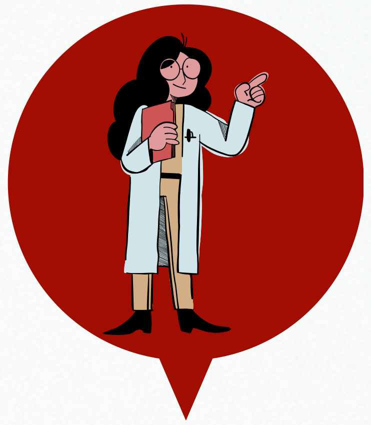
John Schreck



Nick Cote



Nathan Hook



Virginia Do

Jessica Wang

Reva Golden

Wayne Chestnut

Chris Patton

HESS & Facilities Team

DISCLAIMER

This material is based upon work supported by the U.S. National Science Foundation National Center for Atmospheric Research, which is a major facility sponsored by the U.S. National Science Foundation under Cooperative Agreement No. 1755088. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the U.S. National Science Foundation.

Q & A

in

GitHub