# DSM DASHBOARD

## A Web based Interface for Real-Time NIDAS Data Quality Monitoring and Visualization

Ameen Hussain

Department of Computer Science & Department of Mathematics, Colorado State University
NCAR Earth Observing Laboratory SUPER Internship Program
ahuss12@rams.colostate.edu

## Background: What is a DSM?

The NCAR Earth Observing Laboratory (EOL) deploys the Integrated Surface Flux System (ISFS), a network of advanced weather stations used in remote scientific field campaigns.
At the heart of each station is the **Data Sampling Module (DSM)**, a ruggedized, field-deployable computer running the NCAR In-situ Data Acquisition Software (NIDAS). The reliability and performance of its monitoring dashboard are crucial for mission success.Critical functions include:

- **Data Acquisition**: Collects high-frequency data from a wide array of scientific instruments and sensors.
- **Real-time Processing**: Processes, calibrates, and quality-controls the incoming data streams on the fly.
- **Data Serving**: Serves the processed data and system status information via a web interface for real-time monitoring by scientists and technicians in the field.

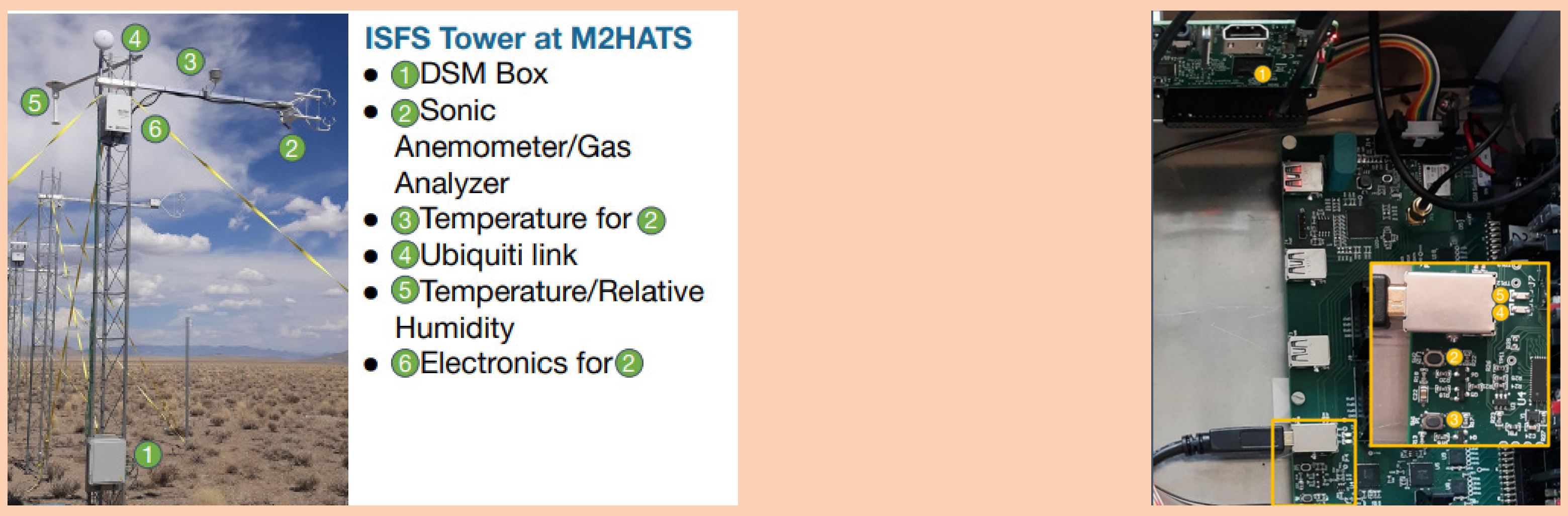
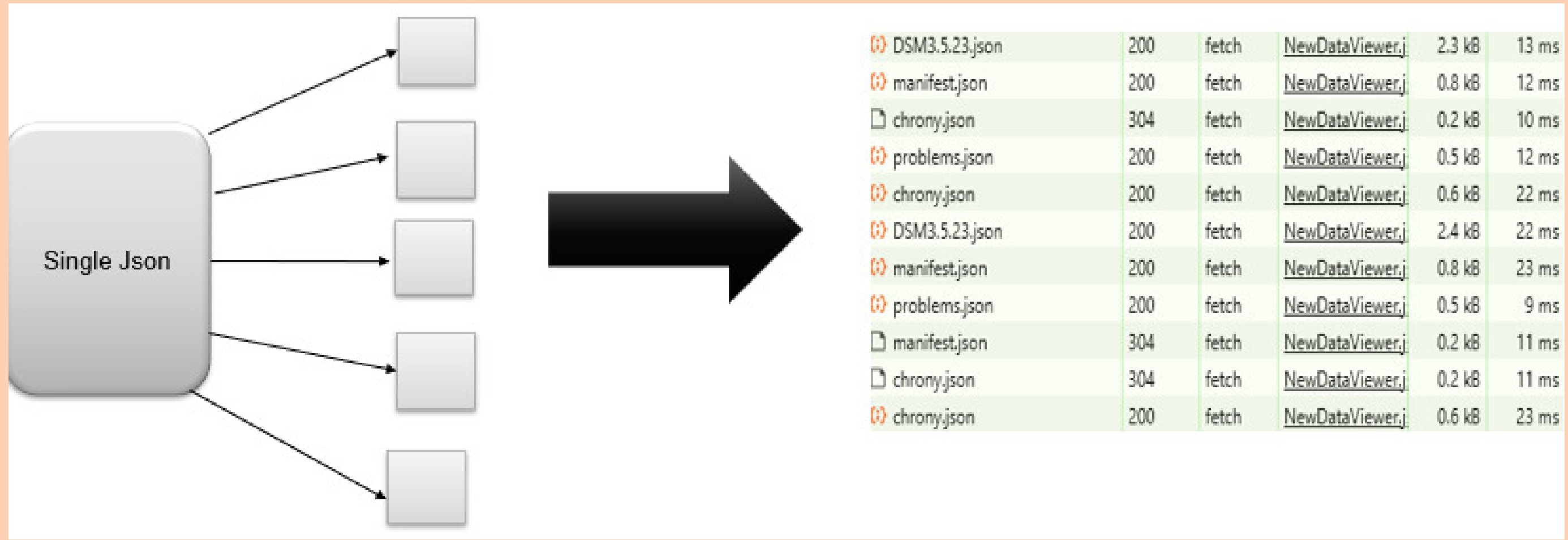


Fig. 1: ISFS Tower at M2HATS (left) and the interior of a DSM (right).

## Methods: A Modern Architecture

- **Decomposed Data Delivery:** Replaced the legacy single, monolithic JSON file with a decoupled system of multiple, smaller, targeted files (manifest. Json, problems. Json, sensor-specific data).
- **Independent Polling:** The frontend now polls these lightweight files independently, fetching only the necessary data and improving performance.
- **Frontend:** Utilized react, material-ui and highcharts for various new frontend features.



| | | | | | |
|---|---|---|---|---|---|
| DSM3.5.23.json | 200 | fetch | NewDataViewer.j | 2.3 kB | 13 ms |
| manifest.json | 200 | fetch | NewDataViewer.j | 0.8 kB | 12 ms |
| chrony.json | 304 | fetch | NewDataViewer.j | 0.2 kB | 10 ms |
| problems.json | 200 | fetch | NewDataViewer.j | 0.5 kB | 12 ms |
| chrony.json | 200 | fetch | NewDataViewer.j | 0.6 kB | 22 ms |
| DSM3.5.23.json | 200 | fetch | NewDataViewer.j | 2.4 kB | 22 ms |
| manifest.json | 200 | fetch | NewDataViewer.j | 0.8 kB | 23 ms |
| problems.json | 200 | fetch | NewDataViewer.j | 0.5 kB | 9 ms |
| manifest.json | 304 | fetch | NewDataViewer.j | 0.2 kB | 11 ms |
| chrony.json | 304 | fetch | NewDataViewer.j | 0.2 kB | 11 ms |
| chrony.json | 200 | fetch | NewDataViewer.j | 0.6 kB | 23 ms |

Fig. 2: Network requests showing efficient fetching made possible through data decomposition.

## Conclusions

The new architecture dramatically reduces the initial and subsequent data fetches, leading to significantly faster page load times as shown below.The myriad of ui improvements has also greatly increased reliability and accessibility. Future works might include:

- **Hardware Port Control:** Implement controls within the UI to remotely toggle power to specific hardware ports, aiding in remote troubleshooting.
- **User Authentication:** Introduce a login system to secure the dashboard and provide a foundation for user-specific permissions and settings.
- **Manual Data Ingestion:** Allow technicians in the field to manually add contextual notes or supplementary data points directly through the interface.

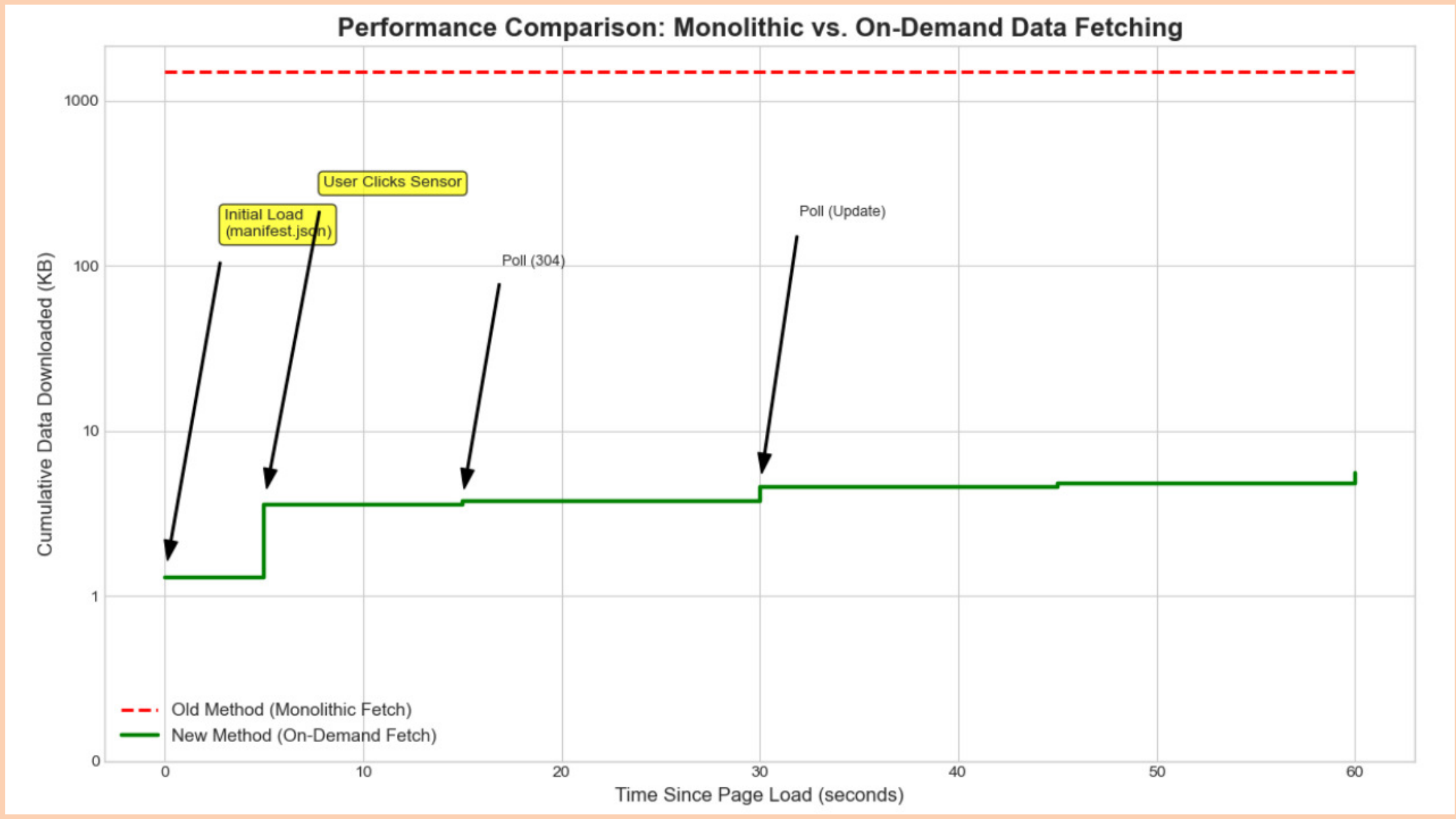


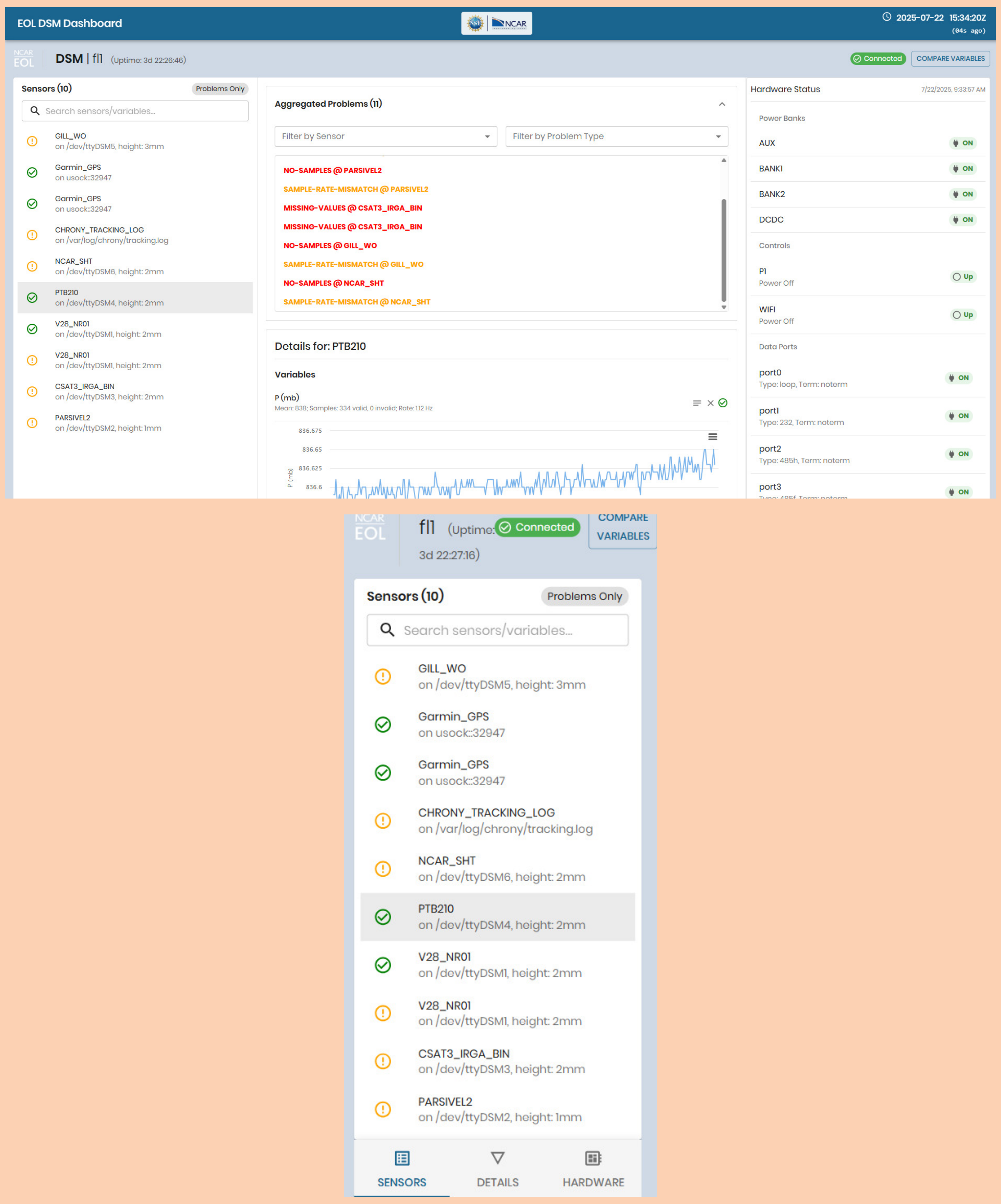
Fig. 3: Performance comparison showing the dramatic reduction in initial data load with the new on-demand fetching model.

## Objectives

- **Improve Performance:** Drastically reduce page load times and improve data refresh speeds by re-architecting the backend data delivery.
- **Enhance User Interactivity:** Replace the static interface with a dynamic one, allowing users to actively filter and explore system problems.
- **Increase Data Visibility:** Display crucial, context-rich metadata—such as device port and installation height—directly in the main interface.
- **Modernize the User Experience:** Develop a professional, branded, and fully responsive user interface that is functional on desktop and mobile devices.

## Results: The Modernized Dashboard

The new dashboard provides a fast, robust, and intuitive interface for real-time monitoring. Key results include a dramatic reduction in initial load time, interactive data filtering, hardware status bar, ability to compare variables, and the display of critical real-time status indicators that were previously unavailable.


EOL DSM Dashboard

2025-07-22 15:34:20Z (04s ago)

DSM | fl1 (Uptime: 3d 22:26:46)

Sensors (10)

GILL_WO
on /dev/ttyDSM5, height: 3mm

Garmin_GPS
on usock::32947

Garmin_GPS
on usock::32947

CHRONY_TRACKING_LOG
on /var/log/chrony/tracking.log

NCAR_SHT
on /dev/ttyDSM6, height: 2mm

PTB210
on /dev/ttyDSM4, height: 2mm

V28_NR01
on /dev/ttyDSM1, height: 2mm

V28_NR01
on /dev/ttyDSM1, height: 2mm

CSAT3_IRGA_BIN
on /dev/ttyDSM3, height: 2mm

PARSIVEL2
on /dev/ttyDSM2, height: 1mm

Aggregated Problems (11)

NO-SAMPLES @ PARSIVEL2
SAMPLE-RATE-MISMATCH @ PARSIVEL2
MISSING-VALUES @ CSAT3_IRGA_BIN
MISSING-VALUES @ CSAT3_IRGA_BIN
NO-SAMPLES @ GILL_WO
SAMPLE-RATE-MISMATCH @ GILL_WO
NO-SAMPLES @ NCAR_SHT
SAMPLE-RATE-MISMATCH @ NCAR_SHT

Details for: PTB210

Variables

P (mb)
Mean: 838; Samples: 334 valid, 0 invalid; Rate: 1.12 Hz

Hardware Status 7/22/2025, 9:33:57 AM

Power Banks
AUX ON
BANK1 ON
BANK2 ON
DCDC ON

Controls
P1 Power Off Up
WIFI Power Off Up

Data Ports
port0 Type: loop, Term: noterm ON
port1 Type: 232, Term: noterm ON
port2 Type: 485h, Term: noterm ON
port3 ON

fl1 (Uptime: 3d 22:27:16)

Sensors (10)

GILL_WO
on /dev/ttyDSM5, height: 3mm

Garmin_GPS
on usock::32947

Garmin_GPS
on usock::32947

CHRONY_TRACKING_LOG
on /var/log/chrony/tracking.log

NCAR_SHT
on /dev/ttyDSM6, height: 2mm

PTB210
on /dev/ttyDSM4, height: 2mm

V28_NR01
on /dev/ttyDSM1, height: 2mm

V28_NR01
on /dev/ttyDSM1, height: 2mm

CSAT3_IRGA_BIN
on /dev/ttyDSM3, height: 2mm

PARSIVEL2
on /dev/ttyDSM2, height: 1mm

Fig. 4: The main desktop UI (top) and the responsive mobile view (bottom).

## Acknowledgements

I would like to extend my sincere gratitude to my mentor Gary Granger for his invaluable guidance and support throughout this project. Thank you to the entire EOL-ISF team and the SUPER program coordinators for this incredible opportunity.

## Tools and Technologies

Fig. 5: Core frontend technologies used in development.