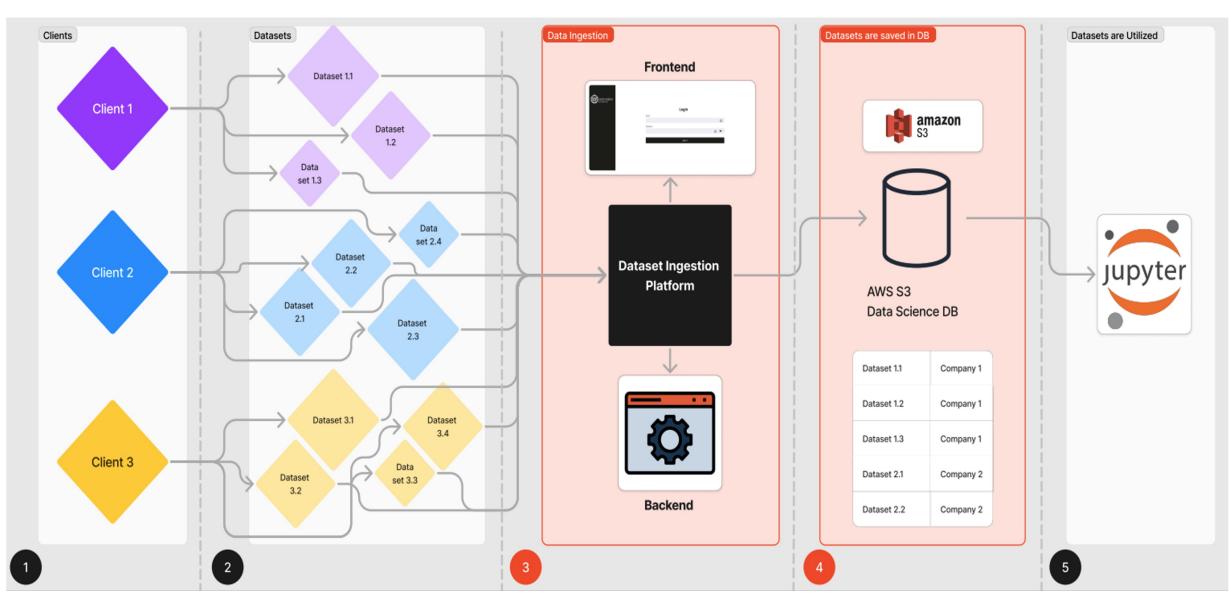


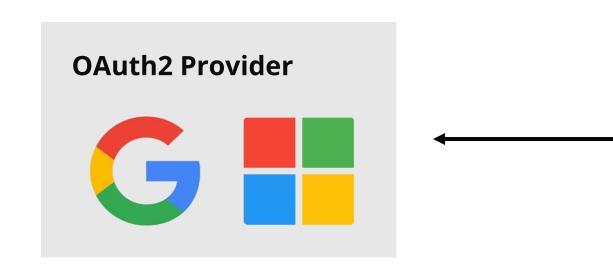
A Cloud-Based Multimodal Data Ingestion Platform for Time Series, Cycling, and Open-Circuit Battery Data STUDENTS: Winona Chu, Wenjun Fan, Jessica Liu, Nesa Marvi

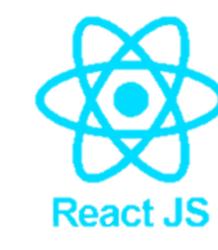
Problem and Objective

Battery technology companies struggle with inconsistent, inefficient data from multiple sources. Our web-based portal standardizes battery data through an intuitive interface and powerful backend, enabling seamless upload, processing, and insight extraction.



Architecture and Requirements





- Frontend React-powered web portal offering an intuitive user experience
- Backend Python-based data processing pipeline with FastAPI for seamless database connectivity
- Database AWS PostgreSQL for web portal data management coupled with S3 for efficient dataset storage



(47)

ELECTRICAL & COMPUTER ENGINEERING

UNIVERSITY of WASHINGTON



Component Design

Extractor.py – Unifies CSV, Excel, JSON, and MAT files into a single DataFrame for processing. **standardization.py** – Standardizes units, handles missing data, and renames columns using fuzzy matching.

Transformer.py – Enhances data with time-series features, cycle segmentation, and custom transformations.

Visualizer.py – Creates high-quality cycling plots using Seaborn and Matplotlib for analysis and reporting.

User-Friendly A clean

dashboard with quick access to projects, profiles, and instant dataset search.

User Interface

astrolabe	Dashboard / Project Project
Data Ingestion Platform	
HomeProfile	Dataset A Last updated on 5/10/2025
O Frome	
Projects ~	
Project1	
Project2	
Project3	
	© 2025 Astrolabe Analytics. All rights reserved.

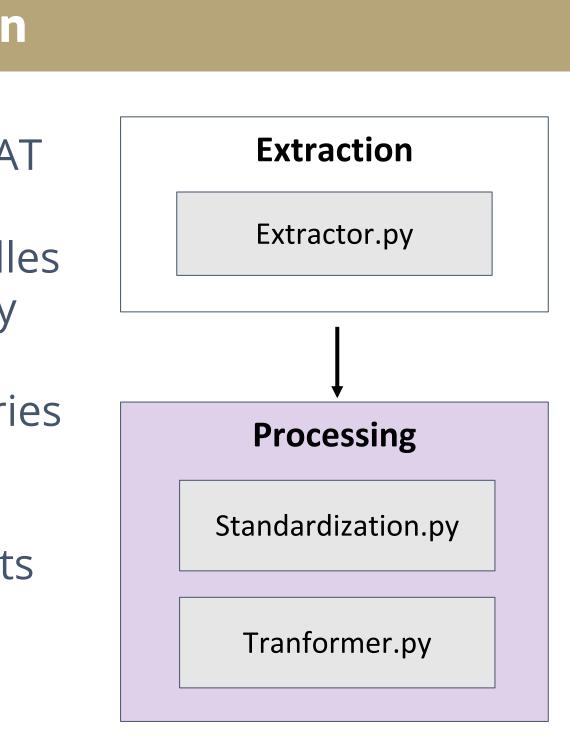


Matching Table

A visual interface that streamlines data integration by combining automated column mapping with intuitive customization for flexibility and efficiency.

Upload Dataset >	Matching Columns >	Results	>	Download
Original Columns	Standardized Columns		Unit	
Timestamp	date_time	~		
Time	time	✓ s		
V_in_v	Missing	^ mV		
Cell current	date_time time	mA		
Temperature	voltage	С		
State of charge	temperature state_of_charge			
	Missina			





Standardization of Columns

Input Data set (type: Pandas Dataframe)

	Time	Current	Voltage	Capacity	Energy	SOC	SOE
0	1	-3450.1	4.1152	0.000000	0.000000	0.000000	0.000000
1	2	-3450.8	4.1052	0.958583	3.938639	0.041710	0.047859

Output Data set (type: Pandas Dataframe)

	time (s)	Current(a)	Voltage(v)	discharge capacity(Ah)	discharge_energy(Wh)	state_of_charge
0	1	-3450.1	4.1152	0.000000	0.000000	0.000000
1	2	-3450.8	4.1052	0.958583	3.938639	0.041710

Validated Results & Key Features

Clean Output Visualization: Cycle 101 State of Charge plot shows a distinct V-shape, confirming that our preprocessing effectively preserved meaningful

transitions and removed noise.

Time-Series Segmentation: Efficiently identifies charging, discharging, and idle cycles for targeted analysis.

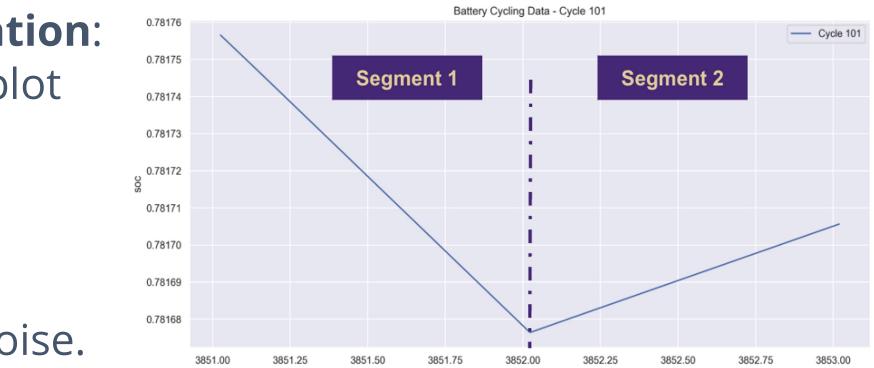
W High-Quality Plotting: Ready-to-use visualizations for reports using Seaborn/Matplotlib.

Scalable AWS Storage: Backed by S3 and PostgreSQL for secure, expandable data handling.

- public accessibility and real-time use.
- health, usage trends, and remaining life.
- mappings.

ADVISERS: David Beck SPONSOR: Astrolabe Analytics





Automated Column Mapping: Smart mapping with user prompts for consistent schema across formats (CSV, Excel, JSON, MAT).

Future Work

EC2 Deployment: We plan to host the platform on Amazon EC2 for

Battery Usage Prediction: ML models integrated to predict battery

Cross-Domain Application: The pipeline will be adapted for other fields like EV telemetry or solar monitoring, using custom schema