Design Thinking for Data Scientist

GreenStream Energy – ETL Case Study

Designed By Eng:OLA EZZAT.

===

Task A: ETL Architecture Diagram (System Design)

Conceptual ETL Pipeline Overview
The ETL pipeline is designed as a serverless, event-driven architecture with clear separation between raw data ingestion, transformation, structured storage, and analytics archival.

ETL Architecture Components and Data Flow

1. Source Layer – Smart Meters

Smart meters installed in 50,000 households generate CSV-based time-series energy data. Readings may be reported in Watts (W) or Kilowatts (kW), may contain missing values, and may include faulty sensor readings. Data is continuously uploaded to raw storage.

2. Raw Data Storage (Landing Zone)

Raw CSV files are stored exactly as received in a data lake raw zone. No transformations are applied. This layer acts as a historical backup and supports auditing and reprocessing. Upload events trigger the orchestration layer.

3. Orchestration Layer

The orchestration layer coordinates the ETL workflow by triggering transformation jobs, monitoring execution, handling retries, and routing failed records. Successful runs continue the pipeline, while failures trigger retries, logging, and data isolation.

4. Transformation Layer (Serverless ETL)

This layer performs data cleaning, unit standardization, missing value handling, data validation, and rule-based faulty meter detection. Valid records follow the success path to structured storage, while invalid records are sent to error logs or quarantine storage.

5. Structured Storage Layer (Query & Validation)

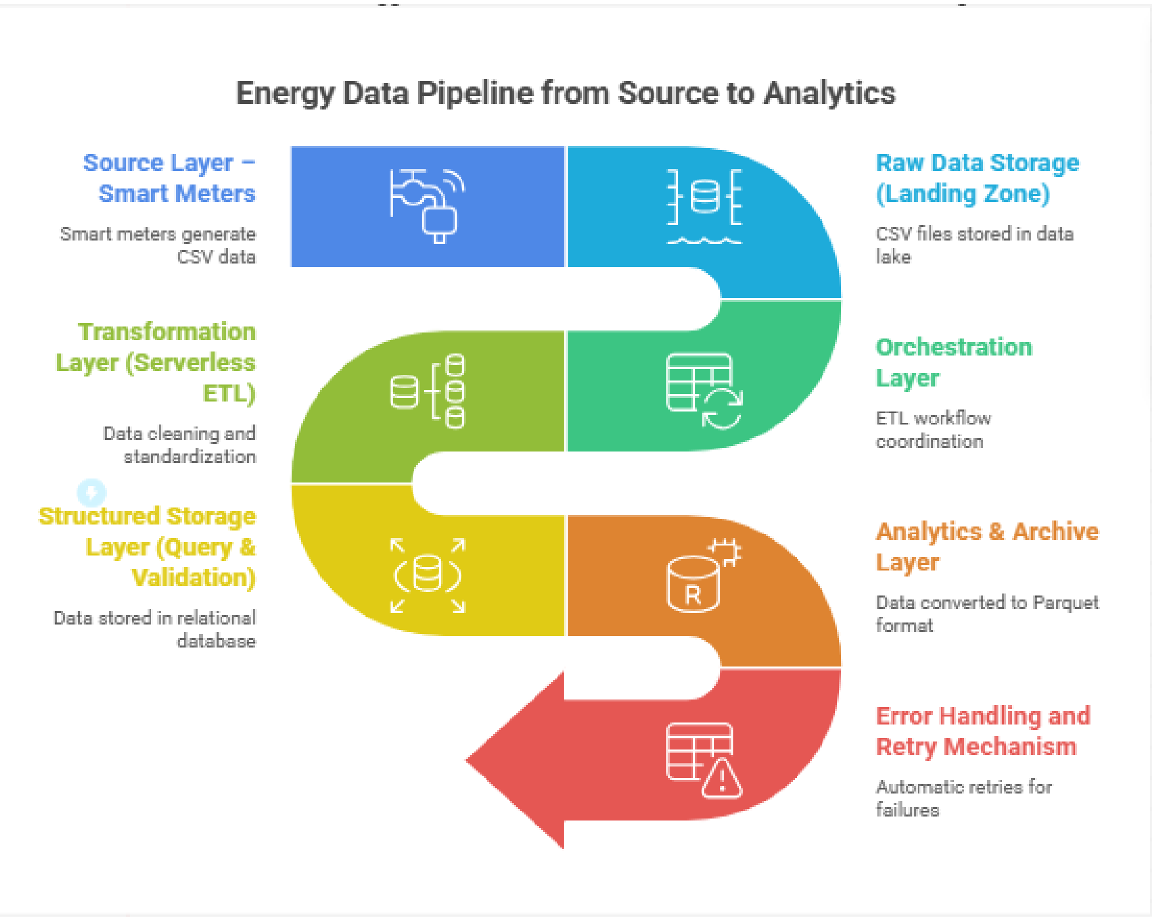
Clean, standardized, schema-enforced data is stored in a relational database. This layer supports SQL analytics, peak usage analysis, dashboards, and data quality validation.

6. Analytics & Archive Layer

Validated data is converted into Parquet format and stored for long-term historical analysis, optimized analytics queries, and machine learning or forecasting workloads.

7. Error Handling and Retry Mechanism

Transient failures trigger automatic retries. Persistent failures are routed to error logs or dead-letter storage, ensuring pipeline reliability without manual intervention.


Energy Data Pipeline from Source to Analytics
Source Layer – Smart Meters
Smart meters generate CSV data
Raw Data Storage (Landing Zone)
CSV files stored in data lake
Transformation Layer (Serverless ETL)
Data cleaning and standardization
Orchestration Layer
ETL workflow coordination
Structured Storage Layer (Query & Validation)
Data stored in relational database
Analytics & Archive Layer
Data converted to Parquet format
Error Handling and Retry Mechanism
Automatic retries for failures
R

Task B: Transformation Logic & Business Rules Design

Rule 1: Unit Standardization

If the energy unit is W, divide the value by 1000 and convert it to kW. If the unit is already kW, keep the value unchanged. Any other unit marks the record as invalid.

Rule 2: Missing Value Handling

If the energy reading is NULL, the record is flagged as incomplete, excluded from peak usage calculations, and retained for auditing purposes.

Rule 3: Timestamp Validation

Records with missing timestamps or duplicate timestamps for the same meter are rejected. Chronological ordering is enforced per meter.

Rule 4: Range Validation

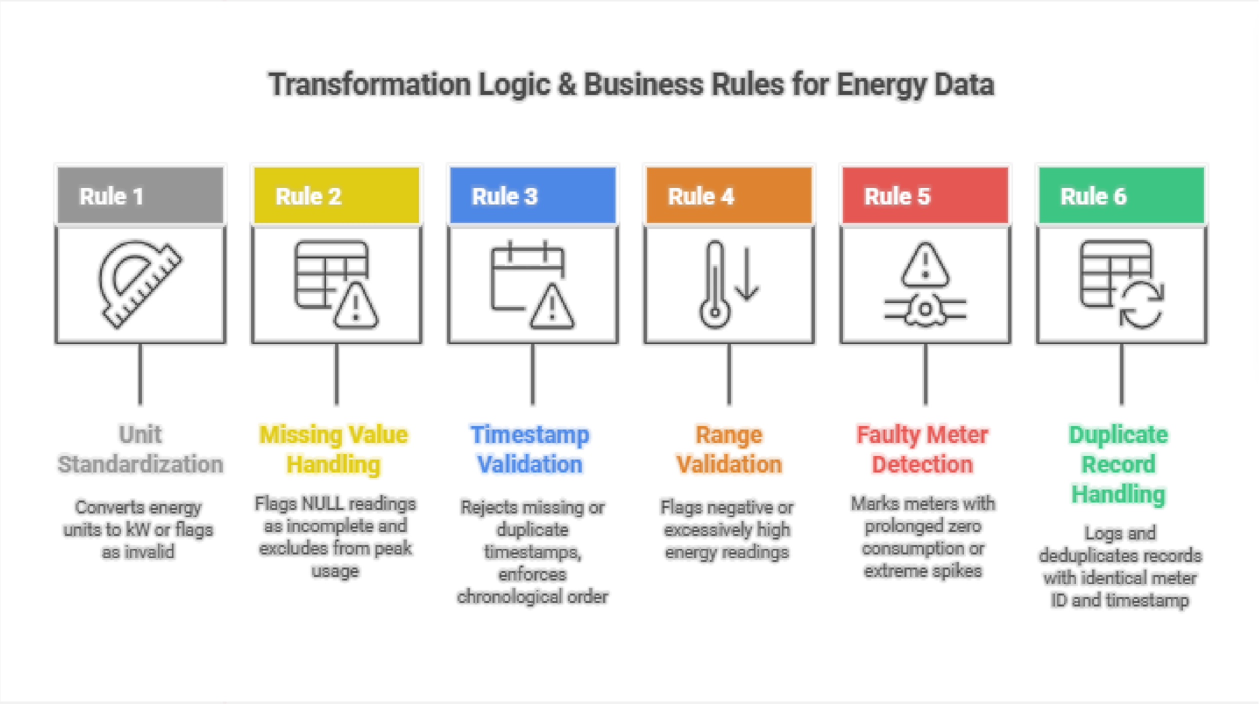
Energy consumption must be non-negative. Readings exceeding realistic thresholds are flagged as potential errors.

Rule 5: Faulty Meter Detection

Meters reporting zero consumption for unusually long periods or extreme spikes compared to historical averages are marked as potentially faulty.

Rule 6: Duplicate Record Handling

Duplicate records with the same meter ID and timestamp are logged and deduplicated using the latest valid reading.



Unit Standardization: Converts energy units to kW or flags as invalid

Missing Value Handling: Flags NULL readings as incomplete and excludes from peak usage

Timestamp Validation: Rejects missing or duplicate timestamps, enforces chronological order

Range Validation: Flags negative or excessively high energy readings

Faulty Meter Detection: Marks meters with prolonged zero consumption or extreme spikes

Duplicate Record Handling: Logs and deduplicates records with identical meter ID and timestamp

Task C: Single Record Lifecycle Explanation

Step 1: Upload to Raw Storage

A smart meter record (MeterID 12345, Timestamp 2025-01-10 08:15, Energy 4500 W) is uploaded as part of a CSV file to raw storage without any transformation.

Step 2: Transformation Trigger

The arrival of the file triggers the orchestration workflow, initiating the ETL process.

Step 3: Data Cleaning and Standardization

The unit is converted from W to 4.5 kW. The energy value and timestamp are validated and confirmed as valid.

Step 4: Data Validation and Fault Detection

The reading falls within acceptable limits, and no abnormal behavior is detected. The record is marked as valid.

Step 5: Storage in Structured Database

The cleaned record is stored in a relational table with fields MeterID, Timestamp, Energy_kW, and Status, making it immediately available for analytics.

Step 6: Conversion and Archival

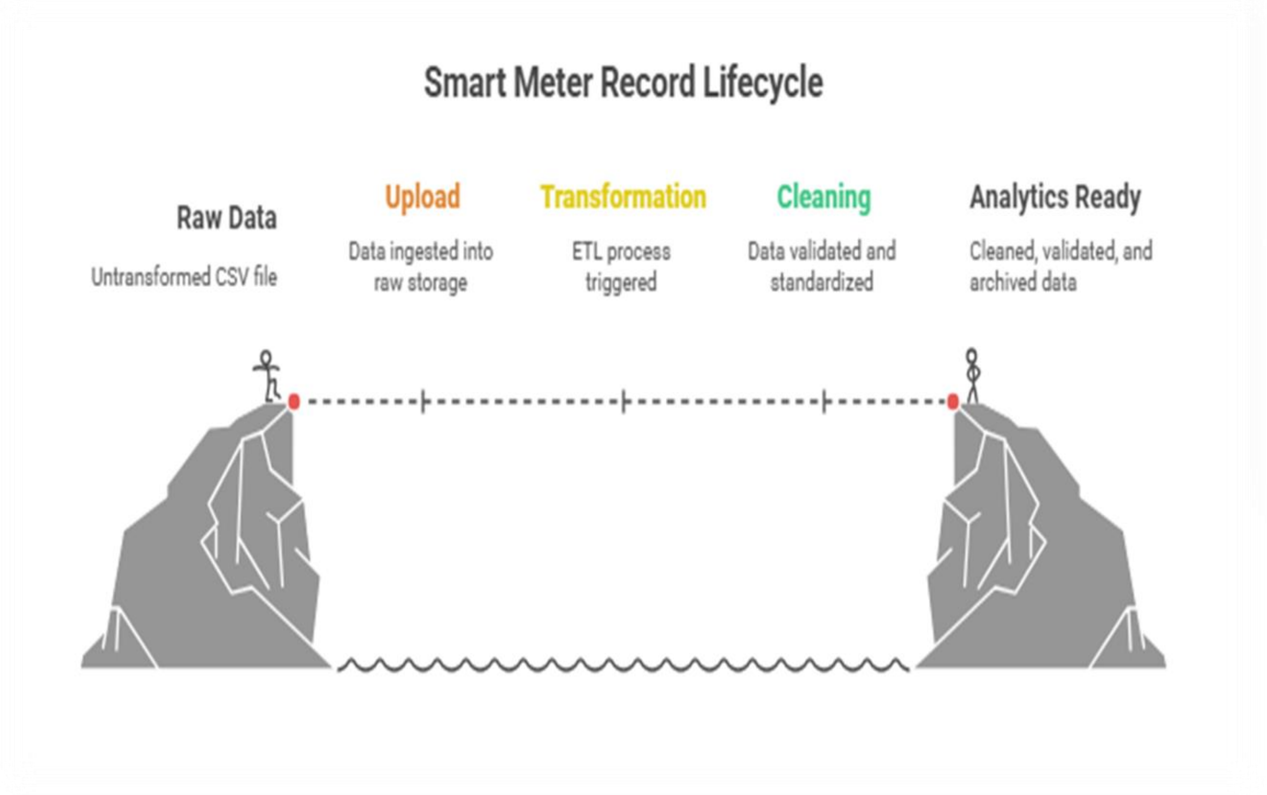
The record is batched with others, converted into Parquet format, and stored in the analytics archive zone.

Step 7: Success and Failure Handling

Successful records complete the pipeline with logged metadata. Failed records trigger retries, and persistent failures are routed to error storage.

Final Outcome

This serverless ETL design transforms dark data into actionable insights, ensures data quality and consistency, supports scalable analytics and machine learning workloads, and reflects strong data science thinking rather than cloud configuration details.


Smart Meter Record Lifecycle
Raw Data
Untransformed CSV file
Upload
Data ingested into raw storage
Transformation
ETL process triggered
Cleaning
Data validated and standardized
Analytics Ready
Cleaned, validated, and archived data