GreenStream Energy: Serverless ETL Pipeline Design

Design Thinking for Data Scientists

Student: Ahmed Abd Almenam Mokhtar Ahmed

Summary

GreenStream Energy collects vast amounts of operational data from 50,000 household smart meters, yet this data remains largely “dark”—unprocessed and inaccessible for strategic decision-making. To address this, we propose a serverless ETL (Extract, Transform, Load) pipeline designed to transform raw, noisy telemetry into high-quality, analytics-ready datasets.

This design leverages a cloud-native, event-driven architecture to automatically ingest, validate, clean, and store energy usage data. By addressing critical data quality issues such as inconsistent units and missing readings, this solution establishes the foundational data infrastructure required for peak load analysis, faulty meter detection, and future predictive analytics capabilities.

---

1. Case Study Overview

Background

GreenStream Energy is a smart utility provider managing a fleet of 50,000 smart meters. While data collection is active, the organization currently lacks the automated infrastructure to derive value from this data stream.

Strategic Goals

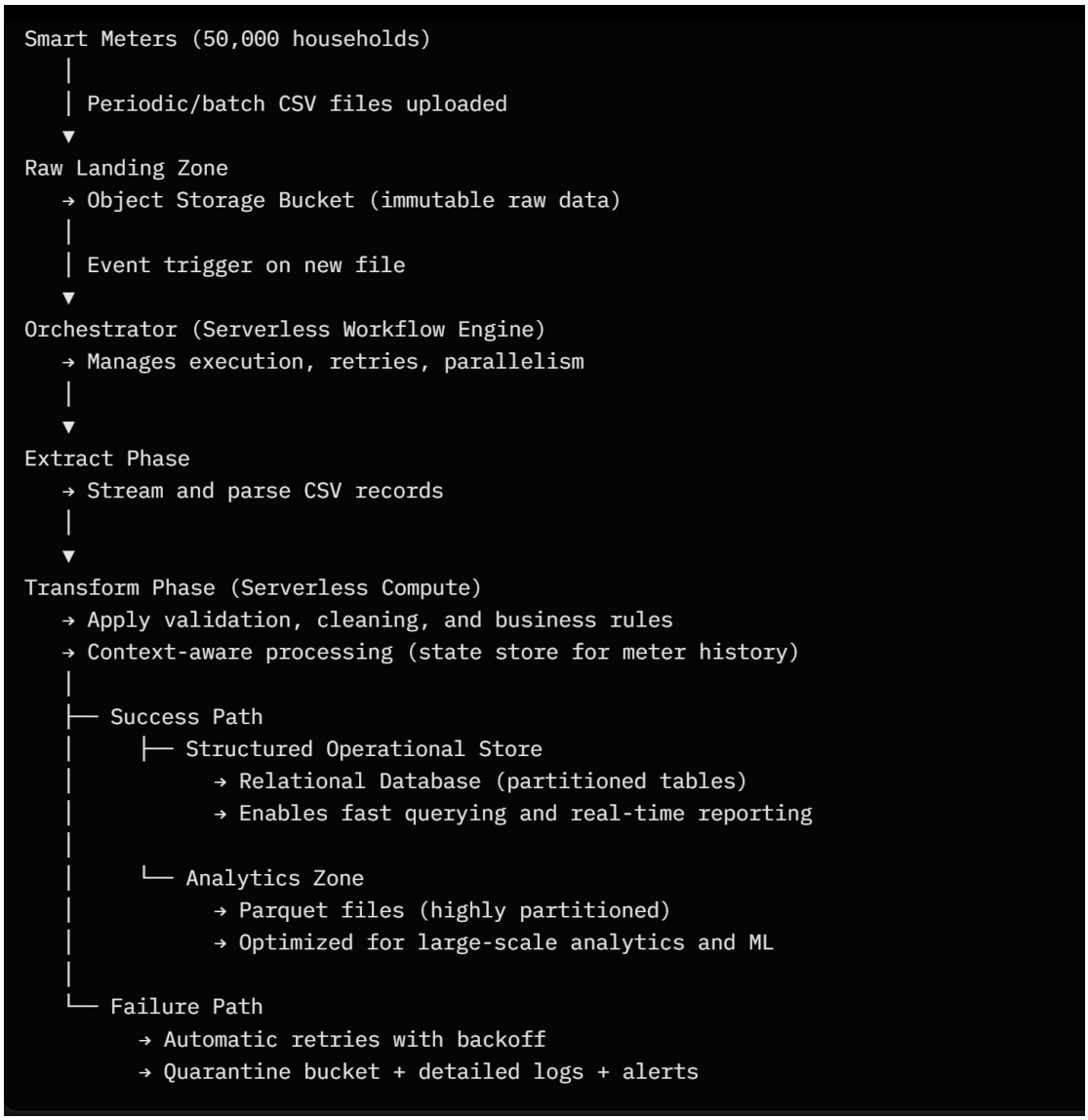
The pipeline is designed to support three primary business objectives:

- Identify peak energy consumption periods
- Detect abnormal or faulty smart meters
- Prepare data for future predictive analytics and forecasting

---

2. Task A: ETL Architecture (Conceptual Design)

High-Level Serverless Pipeline Workflow



Key Design Principles

- Event-driven and fully automated
- Idempotent operations for safe reprocessing
- Dual storage strategy: operational + analytical
- Built-in fault tolerance and monitoring

---

3. Task B: Transformation Logic & Business Rules

The Transform phase applies a strict sequential set of rules to ensure data quality and consistency.

1. **Schema & Mandatory Field Validation** Required fields: meter_id, timestamp, energy_value, unit → Any missing mandatory field → immediate rejection to error path
2. **Timestamp Processing**
    1. Convert to standardized UTC ISO-8601 format
    2. Detect and deduplicate exact duplicates (same meter_id + timestamp)
    3. Flag records that are out of expected sequence
3. **Unit Standardization**

    1. If unit = "W" → energy_value_kW = energy_value / 1000.0; unit = "kW"
    2. If unit = "kW" → no change
    3. Invalid or missing unit → flag invalid_unit = True → error path

4. **Value Validation & Outlier Detection**

    1. Negative values → reject to error path
    2. Maintain per-meter rolling 30-day maximum (from state store)
    3. Value > 1.5 × rolling maximum → flag outlier_high = True (retain with flag)

5. **Missing Value Handling**

    1. NULL energy_value → flag missing_reading = True
    2. Impute using linear interpolation between nearest valid previous/next readings
    3. Fallback: forward-fill from last known valid reading or meter-specific daily average
    4. Record gap duration in seconds for outage analytics

6. **Faulty Meter Detection (Context-Aware Logic)**

    1. Maintain per-meter counters (zero-streak, recent readings) in state store
    2. ≥ 48 consecutive near-zero readings (≤ 0.001 kW) → flag potential_faulty = True and generate alert record

3. Sudden drop > 90% compared to 7-day rolling average → flag sudden_drop = True
4. Reset counters upon valid non-zero reading

7. **Data Enrichment** Add derived attributes:

    1. reading_date, reading_hour, day_of_week, is_weekend
    2. is_peak_hour (configurable business hours)
    3. cumulative_daily_kWh (running total per meter per day)
    4. temperature_correlation_flag (placeholder for future external enrichment)

All transformations are idempotent. Flagged records are preserved with metadata flags for downstream analysis unless critically invalid.

---

4. Task C: Single Record Lifecycle (Detailed Example)

Sample Raw Record {meter_id: "M12345", timestamp: "2025-12-26T10:30:00+02:00", energy_value: 1500, unit: "W"}

1. **Upload** Record arrives within a CSV batch → stored unchanged in raw bucket at raw-meter-data/year=2025/month=12/day=26/batch_1030.csv
2. **Trigger** Object creation event → orchestrator instantiates new workflow execution for the file
3. **Extract** CSV streamed → record parsed into structured format
4. **Transform (Step-by-Step Application)**
    1. Schema valid
    2. Timestamp converted to UTC: "2025-12-26T08:30:00Z"
    3. Unit converted: energy_value_kW = 1.5, unit = "kW"
    4. Value validation: within historical range → no outlier
    5. Not missing → no imputation needed
    6. Faulty check: no long zero streak → no flag
    7. Enrichment: hour=8, day_of_week="Friday", is_weekend=False, etc.
5. **Load – Structured Operational Store** Cleaned record upserted into cleaned_readings table (fast indexed access)
6. **Load – Analytics Zone** Record appended to Parquet file in partitioned path: processed/year=2025/month=12/day=26/meter_id=M12345/part-XXXX.parquet

7. **Completion & Monitoring**

    1. Success: All steps complete → workflow status "SUCCEEDED"
    2. Any failure (e.g., parsing issue on another record): independent processing ensures valid records (including this one) are still loaded; failed ones quarantined with full diagnostic logs

---

Conclusion

This comprehensive serverless ETL pipeline transforms GreenStream Energy's raw smart meter telemetry into trustworthy, enriched, and purpose-optimized datasets.

Through meticulous data quality rules, resilient architecture, and dual-storage strategy, the solution not only resolves current "dark data" challenges but also positions the organization for advanced analytics, proactive maintenance, and data-driven innovation in energy management.