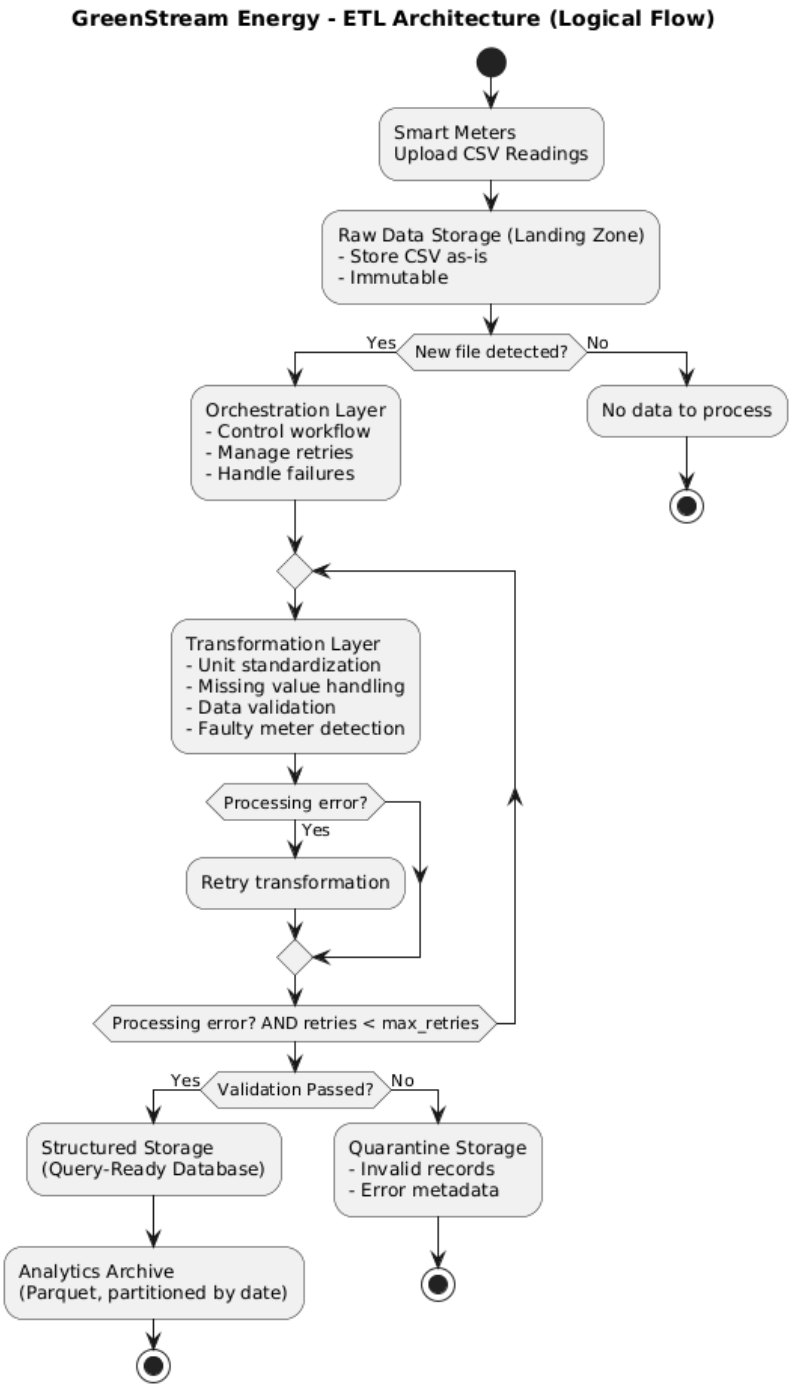
Selected Topics Course – Lecture Task

Name: Abdelrahman Ahmed Eldaba – **Academic ID:** 412200228

Task A: ETL Architecture Diagram (System Design)

Conceptual Serverless ETL Flow (Logical Diagram)



Failure & Retry Logic (Conceptual)

- **Transformation failure:** Automatically retried to handle transient processing issues.
- **Data rule violation:** Invalid records are routed to Quarantine Storage for inspection without blocking the pipeline.
- **Pipeline failure:** Orchestration halts downstream steps to prevent propagation of corrupted data.
- **Idempotency:** Successfully completed steps are not reprocessed during retries.

Task B: Transformation Logic & Business Rules Design

1. Unit Standardization Rules

Goal: Make all energy readings comparable.

- **Rule 1:**
  If `energy_unit = "W"` → divide `energy_value` by `1000` and convert unit to `"kW"`
- **Rule 2:**
  If `energy_unit NOT IN ("W", "kW")` → mark record as invalid and send to quarantine

2. Missing Values Handling

Goal: Preserve data integrity without creating misleading peaks.

- **Rule 3:**
  If `energy_value IS NULL` →
  - Flag record as `missing_reading = true`
  - Exclude from peak-consumption calculations
  - Keep record for data completeness analysis
- **Rule 4:**
  If missing duration < predefined threshold (e.g., 1 interval) →
  allow interpolation in future predictive models (not in core analytics)

3. Data Validation Rules

Goal: Prevent corrupt or illogical data from polluting analytics.

- **Rule 5:**
  If `energy_value < 0` → invalid (send to quarantine)
- **Rule 6:**
  If `timestamp` is outside ingestion window or duplicated → flag and deduplicate
- **Rule 7:**
  If `meter_id` is missing or unknown → reject record

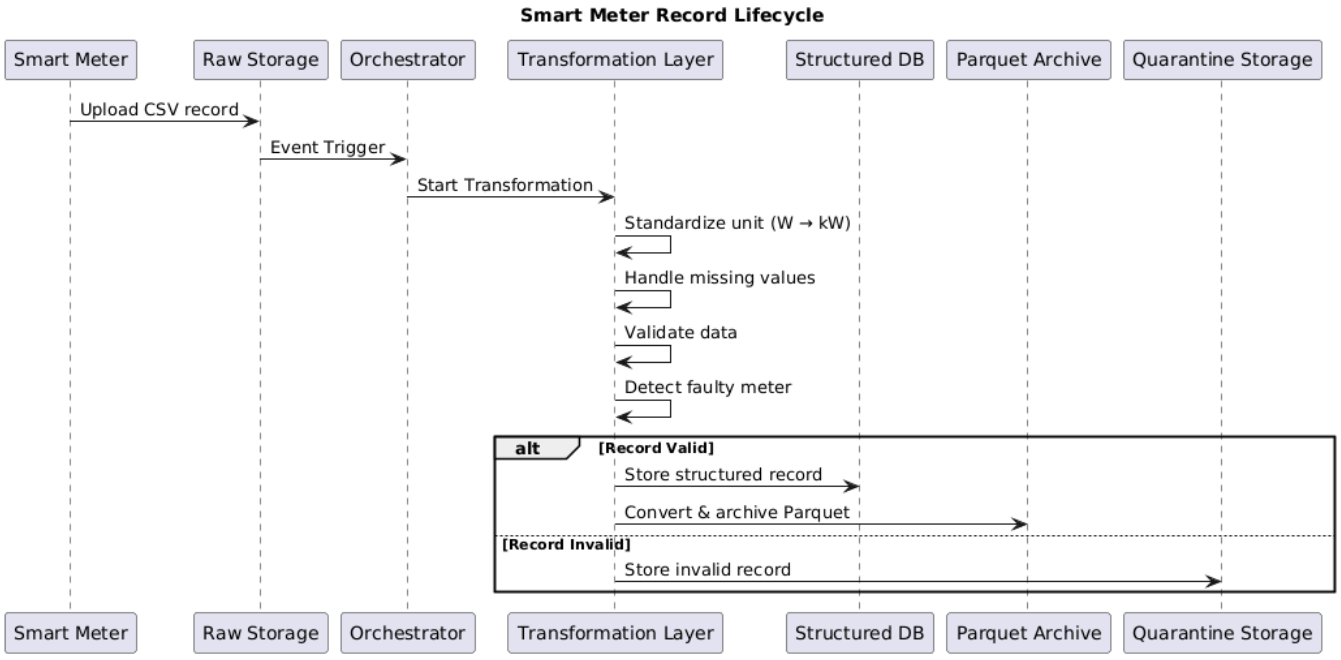
4. Faulty Smart Meter Detection

Goal: Early identification of hardware or connectivity issues.

- **Rule 8:**
  If a meter reports **zero consumption for an unusually long continuous period** (e.g., 24+ hours) →
  mark `meter_status = "potentially_faulty"`
- **Rule 9:**
  If meter produces extreme spikes compared to its historical average →
  flag as `anomalous_reading` (not removed, only labeled)

Task C: Single Record Lifecycle Explanation

Example: One Smart-Meter Record



1. Upload to Raw Storage

- A CSV file containing the record is uploaded from the smart meter
- The raw record is stored **unchanged**
- This preserves auditability and traceability

2. Triggering the Transformation Process

- Arrival of the new CSV triggers the orchestration workflow
- The workflow initiates the Transform phase automatically

3. Data Cleaning & Validation

The record goes through sequential checks:

1. **Unit check**
    - If unit = W, value is converted to kW
2. **Missing value check**
    - If value is null → flagged and excluded from peak usage
3. **Validation**
    - Negative values, unknown units, invalid timestamps → rejected
4. **Fault detection**
    - Compared with meter's recent history for zero or abnormal patterns

4. Storage in Structured Format (Query-Ready)

- If the record passes validation:
    - Stored in a structured database
    - Indexed by meter_id and timestamp
- This enables:
    - Fast peak-hour analysis
    - Fault monitoring dashboards
    - Data validation queries

5. Conversion & Archival (Parquet)

- The same validated record is transformed into **Parquet format**
- Stored in partitioned long-term storage (e.g., by date)
- Optimized for:
    - Historical analysis
    - Forecasting models
    - Batch analytics

6. Success & Failure Handling

- **Success:**
  Workflow completes, record is available for analytics
- **Recoverable failure:**
  Automatic retry (e.g., temporary processing error)
- **Rule failure:**
  Record sent to quarantine with error reason logged
- **Pipeline failure:**
  Downstream steps are halted to avoid corrupt outputs