Design Thinking for Data Scientists

ETL Pipeline Conceptual Design

Prepared by:
Mustafa Ahmed Attia Oun
Academic ID: **410022378**

Date: 2024-05-27

1. Case Study Background

GreenStream Energy is a smart-utility provider that collects electricity consumption data from 50,000 households via smart meters. Currently, this valuable data remains largely untapped—classified as 'dark data'—stored without being transformed into actionable business intelligence.

Strategic Objectives:

- Identify peak energy consumption periods to optimize grid operations
- Detect abnormal or malfunctioning smart meters proactively
- Prepare clean, structured data for predictive analytics and forecasting models

2. Data Quality Challenges

Challenge	Impact
Inconsistent measurement units (W vs kW)	Prevents direct comparison and aggregation
Missing readings due to connectivity issues	Creates gaps in time series analysis
CSV format inefficiency for large datasets	Slows down historical analysis and queries
No automated validation process	Risk of corrupt data affecting analytics

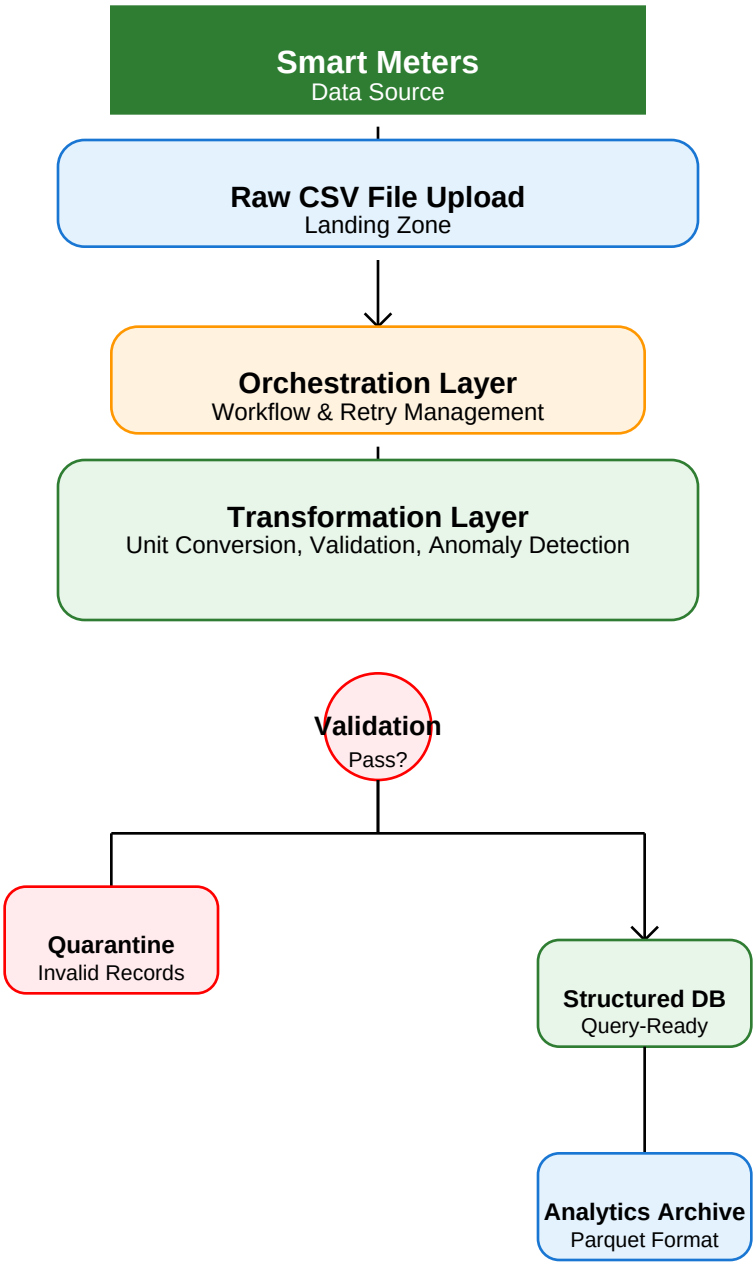
3. Design Goal

The objective is to **conceptually design**—not implement—a serverless ETL pipeline that addresses the identified challenges while supporting GreenStream's strategic objectives. The design should reflect data science thinking rather than cloud configuration details.

1. Clean and standardize energy data for consistency
2. Store structured data for efficient querying and validation
3. Archive data in analytics-optimized format for long-term analysis
4. Automatically handle failures and retries for reliability

Task A: ETL Architecture Diagram

The following logical diagram illustrates the serverless ETL pipeline flow from data ingestion to analytics-ready storage:


Smart Meters
Data Source
Raw CSV File Upload
Landing Zone
Orchestration Layer
Workflow & Retry Management
Transformation Layer
Unit Conversion, Validation, Anomaly Detection
Validation
Pass?
Quarantine
Invalid Records
Structured DB
Query-Ready
Analytics Archive
Parquet Format

Key Flow Characteristics:

- New file detection automatically triggers the ETL workflow
- Failed transformations are retried (up to a configurable limit)
- Invalid records are quarantined without blocking pipeline flow
- Pipeline halts on critical failures to prevent data corruption
- Idempotent operations ensure no duplicate processing during retries

Task B: Transformation Logic & Business Rules

The following business rules are applied during the transformation phase to resolve data quality issues:

1. Unit Standardization Rules

Goal: Ensure all energy readings are comparable by standardizing measurement units.

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 route to quarantine storage

2. Missing Values Handling

Goal: Preserve data integrity without creating misleading analytical artifacts.

Rule 3:	If energy_value IS NULL → Flag as missing_reading = true and exclude from peak consumption calculations
Rule 4:	If missing duration < threshold (e.g., 1 interval) → Allow interpolation in predictive models (not core analytics)

3. Data Validation Rules

Goal: Prevent corrupt or illogical data from polluting analytical datasets.

Rule 5:	If energy_value < 0 → Invalid (send to quarantine)
Rule 6:	If timestamp outside ingestion window or duplicated → Flag and deduplicate
Rule 7:	If meter_id is missing or unknown → Reject record

4. Faulty Meter Detection Rules

Goal: Early identification of hardware or connectivity issues for proactive maintenance.

Rule 8:	If meter reports zero consumption for >24 continuous hours → Mark as "potentially_faulty"
Rule 9:	If extreme consumption spikes compared to historical average → Flag as "anomalous_reading"

Task C: Single Record Lifecycle

The following illustrates the end-to-end journey of a single smart-meter reading through the ETL pipeline:

1.	**Upload to Raw Storage**	CSV file from smart meter stored unchanged in landing zone for auditability
2.	**Trigger Transformation**	File arrival activates orchestration workflow automatically
3.	**Data Cleaning & Validation**	Unit conversion, null handling, business rule validation, anomaly detection
4.	**Structured Storage**	Validated records stored in query-ready database (indexed by meter_id, timestamp)
5.	**Archival in Parquet**	Same records converted to Parquet format for long-term, partitioned storage
6.	**Success/Failure Handling**	Automatic retry for transient errors, quarantine for rule violations

Handling Scenarios:

Success:	Workflow completes, record available for analytics immediately
Recoverable Failure:	Automatic retry (e.g., temporary network issue)
Rule Violation:	Record sent to quarantine with detailed error metadata
Critical Pipeline Failure:	Downstream processes halted to prevent data corruption

Design Principles Applied:

Modularity:	Each component has a single, well-defined responsibility
Fault Tolerance:	Automatic retries and quarantine prevent total pipeline failure
Data Integrity:	Invalid data is isolated rather than discarded
Scalability:	Serverless architecture handles variable data volumes efficiently
Auditability:	Full traceability from raw data to analytics-ready output