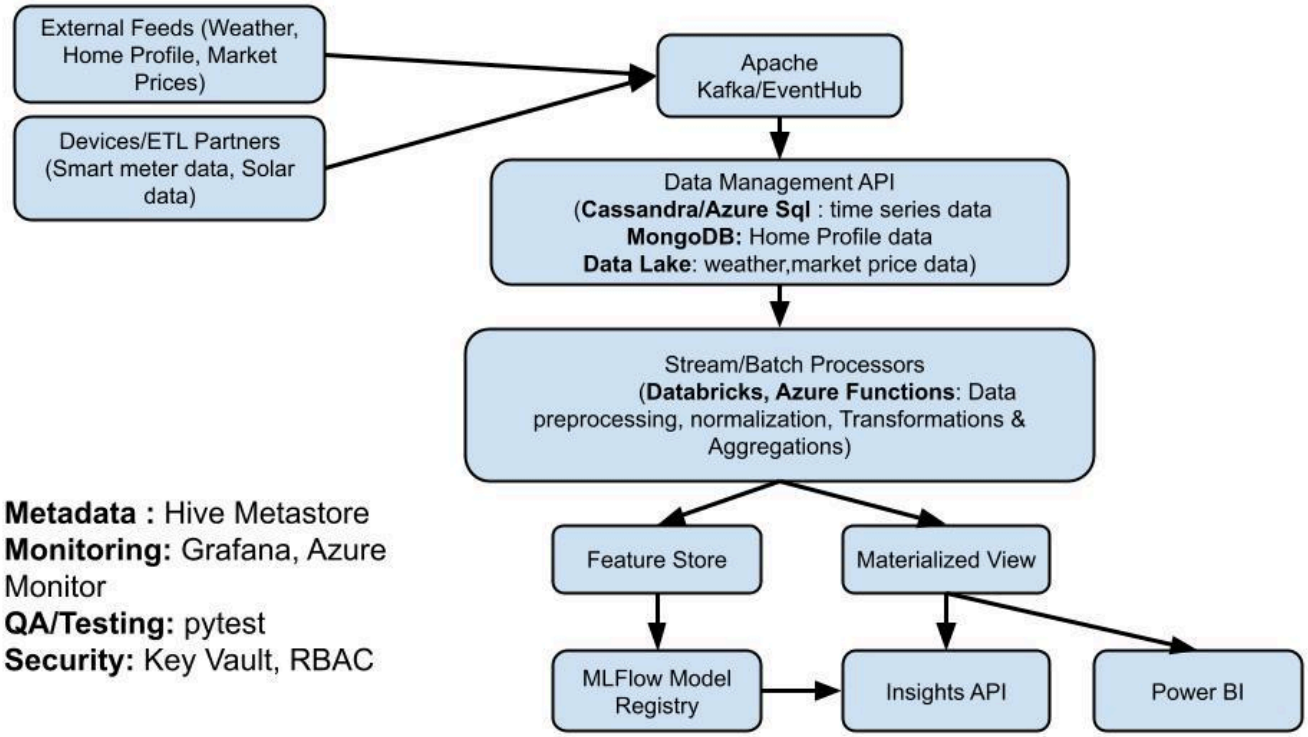
Data Platform Architecture



Data :

- **External Feeds** (Weather, Electricity prices, Market data)
- **Devices / ETL Partners** (Smart meters, Solar installers, Aggregators)

Apache Kafka / Azure Event Hub

- decouples producers and consumers, enables both real-time processing and buffering for batch (Guarantees ordering per key (e.g., meter ID) and retention for re-processing)

Data Management API

- validates, writes canonical records to storage, enforces schema, access control, and hides internal storage details from producers. Stores metadata about sources and ingest events.

Datastores

- **Time-series DB (e.g., Cassandra or Azure SQL)**

    - Stores high-frequency meter readings, indexed by device & time (low-latency reads for time-window queries, efficient retention/TTL policies)

- **Document DB (e.g., MongoDB / Cosmos DB)**
    - Stores Home Profile documents (flexible schema for attributes: heating type, household members, meters).

- **Data Lake (Azure Data Lake Storage / S3)**
    - Stores raw ingested data, enriched parquet files, historical exports, and market/weather archives (inexpensive long-term storage, reprocessing, analytics, and model training)

- **Materialized Views(Delta Lake /BigQuery)**
    - Precomputed aggregates, join results needed by BI and Insights API.

Stream / Batch Processors (Databricks, Azure Functions)

- jobs and streaming pipelines for preprocessing, normalization, transformations (reads into consistent, analysis-ready formats, resample, impute, up/down-sample), feature engineering, aggregations, and quality checks, compute daily/weekly aggregates, and detect anomalies.

Feature Store

- central store of features (per household/device/time) for models and online serving (e.g., Databricks Feature Store).

MLFlow Model Registry

- model lifecycle: versioning, packaging, stage (dev/staging/prod), deployment hooks.

Insights API

- services exposing computed insights (consumption forecast, savings suggestions, anomaly alerts) to apps and partners.

Power BI / BI Tools

- dashboards for product, CS, sales, and executive KPIs.

Metadata / Catalog (Hive Metastore)

- dataset, table schemas, column-level metadata, and lineage.

Monitoring & Observability (Grafana, Azure Monitor)

- pipeline health, latency, errors, job success/failures, data quality metrics.

Security & Secrets (Key Vault, RBAC)

- secret management, keys, encryption at rest/in transit, role-based access (regulatory compliance, protect PII / credentials)

QA / Testing (pytest)

- CI pipelines running tests for data validations, transformations, and APIs.

Responsibilities & Interactions Across Teams

Data Engineering

- **Responsibilities**
    - Build & maintain ingestion (Kafka/EventHub, Data Management API).
    - Operate storage systems (Time-series DB, Data Lake, Document DB, Materialized Views).
    - Own stream/batch processing pipelines (Databricks, Azure Functions).
    - Ensure data quality (schema validation, normalization, and imputation).
    - Set up monitoring (Grafana, Azure Monitor) & CI/CD for ETL pipelines.
- **Interactions**
    - Collaborates with the Data Science team to derive new features for the Feature Store.
    - Supports Customer Success with curated datasets and dashboard-ready materialized views.
    - Works with Sales/Product to provide aggregated datasets for reports and CRM integrations.

Data Science / ML Engineering

- **Responsibilities**
    - Define, engineer, and test new features (with support from the Data Engineer).
    - Develop, train, evaluate, and document ML models.
    - Manage model lifecycle via MLFlow Registry (Dev → Staging → Prod).
    - Define KPIs for models (accuracy, drift, explainability).
- **Interactions**
    - Reads/writes features through the Feature Store maintained by the Data Engineer.
    - Exposes models via APIs (used by Backend Engineering & Customer Success).
    - Provides Sales/CS teams with interpretable model outputs (forecasts, anomalies, savings potential).

Backend / Application Engineering

- **Responsibilities**
    - Own the Insights API, integrating models and materialized views into product-facing endpoints.
    - Ensure API performance, security (RBAC, Key Vault), and stability.
    - Provide data access to front-end apps and external partners.
- **Interactions**
    - Consumes ML models through MLFlow endpoints.
    - Supplies Customer Success with real-time insights for end users.
    - Works with Sales/Product on integrations for partner platforms.

Customer Success (CS) / Product Operations

- **Responsibilities**
    - Define customer-facing reporting needs (dashboards, alerts, KPIs).
    - Use Power BI dashboards to monitor customer energy usage, trends, and engagement.
    - Collect feedback on insights to improve models and dashboards.
- **Interactions**
    - Uses Insights API and Power BI to deliver insights to customers.
    - Works with the Data Scientist to request new insights or feature improvements.
    - Provides Sales with evidence-based customer success stories and benchmarks.

Sales / Commercial

- **Responsibilities**
    - Define reporting needs for lead scoring, churn risk, and upsell opportunities.
    - Use dashboards and aggregated datasets to understand customer behavior and market opportunities.
    - Provide business requirements for new data products.
- **Interactions**
    - Consumes Power BI dashboards fed by materialized views.
    - Works with CS to align customer success outcomes with commercial messaging.
    - Requests Data Engineer and Data Scientist to generate datasets or model-based scoring to support campaigns.

Platform / Infra / SRE

- **Responsibilities**
    - Provision and maintain infrastructure (Databricks clusters, Kafka, storage, monitoring stack).
    - Enforce security & compliance (RBAC, Key Vault).
    - Optimize cost and ensure scalability & availability.
- **Interactions**
    - Supports Data Engineer and Data Science team with reliable compute/storage environments.
    - Works with Security & Compliance to enforce policies.
    - Provides visibility to CS/Sales/Product on system uptime & performance SLAs.

Security & Compliance

- **Responsibilities**
    - Define and enforce data governance (metadata catalog, PII handling, retention).
    - Ensure encryption at rest/in transit and secure key management.
    - Perform audits and certify compliance (GDPR, ISO, SOC2).
- **Interactions**
    - Works with Infra/Eng to enforce RBAC & access controls.
    - Provides guardrails to CS & Sales for how customer data may be shared/used.
    - Ensures Data Scientists adhere to privacy-by-design when using sensitive data.

Implementation plan/phases

Phase 1: Prep & Foundation (Weeks 0–2)

Goal: Set up the minimal infrastructure, schemas, and governance to start ingestion.

- **Tasks**
    - Kick off with a cross-functional team; finalize data contracts (meter, weather, prices, profiles).
    - Choose core tech stack: Kafka/EventHub, Databricks, Time-series DB, Feature Store, MLFlow.
    - Provision minimal dev infra (Kafka, Data Lake buckets, Databricks cluster).
    - Define initial security & retention policies
    - Create schema registry and base templates

Phase 2: Reliable Ingestion & Canonical Storage (Weeks 2–8)

US-1: As a Data Engineer, I want meters to reliably stream readings into Kafka so downstream jobs can consume them.

- Tasks:
    - Define ingestion schema for meter readings (device_id, timestamp, fuel_type, direction, value, granularity).
    - Build Kafka topic topology (partition by device_id, retention rules).
    - Implement producers for a sample smart meter vendor & a test harness.
    - Implement Data Management API (batched uploads, auth, validation).
    - Add unit tests & contract tests.

Phase 3: Transformations, Resampling & QA (Weeks 6–14)

US-3: As a Data Scientist, I want a standardized 15-minute resampled timeseries for each meter to train models.

- Tasks:
    - Implement a Databricks notebook to resample mixed-granularity inputs to 15-minute (up/down-sample rules).
    - Define imputation rules & quality flags.

- Add integration tests with synthetic data.

US-4: As Customer Success, I want daily aggregate consumption per household in Materialized Views for dashboards.

- Tasks:
  - Design materialized view schema (household_id, date, total_kWh, peak_kW, fuel_breakdown).
  - Implement daily aggregation jobs writing to the analytical DB.
  - Connect basic BI reporting to Materialized Views.

Phase 4: Feature Store + ML Lifecycle (Weeks 12–20)

US-5: As an ML engineer, I want features stored in a Feature Store and models versioned in MLFlow.

- Tasks:
  - Select Feature Store tech (E.g, Databricks Feature Store).
  - Create a feature ingestion pipeline from resampled & aggregated data.
  - Add MLFlow integration, CI for model packaging & versioning.
  - Train and register a **baseline forecasting model**

Phase 5: Insights API & Dashboards (Weeks 18–26)

US-6: As Product manager, I want an Insights API that returns forecast + top 3 actionable tips per household.

- Tasks:
  - Define API contract with Product/CS input.
  - Implement a simple scoring endpoint (baseline model or rules-based).
  - Build an Insights API service that calls models & aggregates.

    - Integrate Insights API into **Power BI dashboards** for demo use.
    - Conduct **user acceptance testing** with the CS team.

Phase 6: Harden, Scale & Automate (Weeks 24–36)

Goal: Scale ingestion to more sources, enforce governance, and automate deployments.

- **Tasks**
    - Add RBAC, encryption at rest/in transit
    - Expand ingestion pipelines to more device partners & external feeds.
    - Implement CI/CD for ETL pipelines and ML deployments.
    - Add SLA monitoring (latency, freshness, quality).
    - Cost monitoring & infra autoscaling.