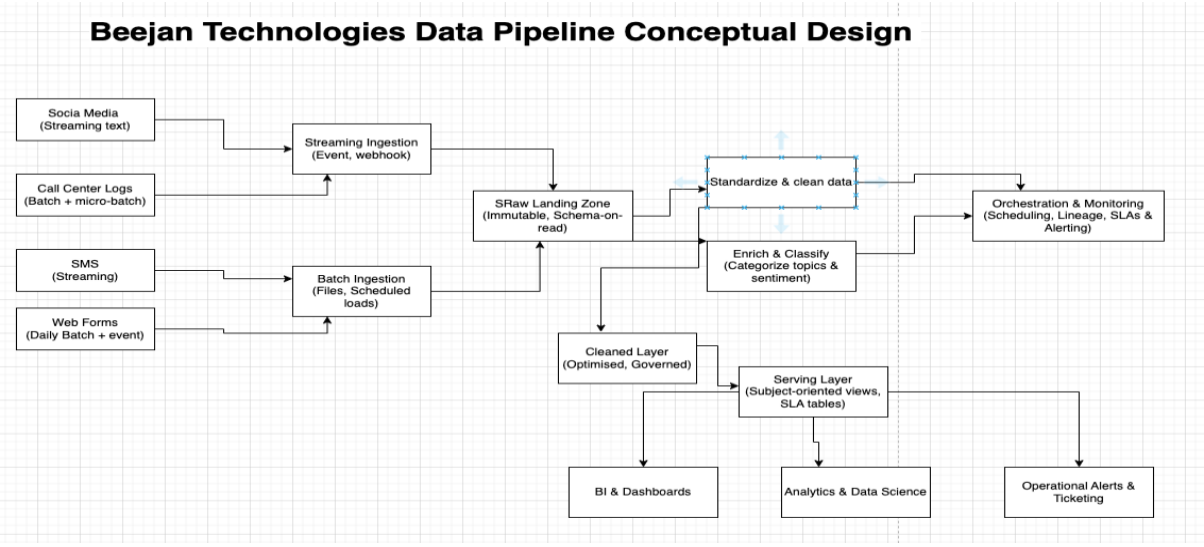
Beejan Technologies Conceptual End-to-End Pipeline for Customer Complaints.

**Purpose:** unifying multi-channel complaints, cleaning and making them for accurate actionable insights.




**Assumptions**

- **Volume:** 40k complaints per day across channels.
- The data contains sensitive account details.
- Business needs near real-time alerting (minutes) and daily executive reporting (batch).
- **Downstream users:** analytics, BI, operations, finance, and compliance.

1. **Source Identification**
    - **Channels:** Social media, call-center logs, SMS, website forms.
    - **Formats:**
    1. Social media & SMS: unstructured/short text, continuous events.
    2. Web forms: structured events + daily export for reconciliation
    3. Call-center logs: semi-structured text/CSV, periodic files.
2. **Ingestion Strategy**
    - **Real-time feeds:** Events endpoints/webhooks for social, SMS, and web-form events.
    - **Batch:** scheduled file drops for call-center logs and the web-form daily export.
3. **Processing/Transformation**
    - **Standardize:** Mapping all inputs to a common complaints record (Customer/account keys, channel, timestamps with time zone normalization, geo
    - **Clean:** Deduplicate near-duplicates, fix encodings, and handle missing/invalid values.
    - **Classification:**

1. L1 (High-level buckets): Network, Billing, Service >> L2 (Sub-categories): “Coverage, Speed, Outage”:” Overcharge, Refund”: “Agent experience”.
2. Combine precise rule patterns with statistical.

- **Enrichment:** Joining of reference data (store/site/cluster, cell IDs, tariff plans, campaigns, lightweight geo(city/region), and customer segments.

4. **Storage Options**
   - Lake (raw layer): Immutable, schema-on-read for replay/audit.
   - Warehouse (Structured + Curated Layers): Clean data, conformed tables in columnar format for analytics and BI.
   - Date formats: Columnar for analytics and compact JSON for event replay.
   - Layout: Partition by event date and channel.
5. **Serving**
   - **Querying:** Stable views(incremental) over clean data for self-service BI and ad-hoc analysis.
   - **Usage:**
     1. Dashboard/KPIs: Daily/weekly metrics and trend analysis.
     2. Low-latency aggregates: Live widgets (complaints per region in the last 15-20 minutes).
     3. Operational contracts: Well-defined schemas for alerting and ticket creation.
6. **Orchestration & Monitoring**
   - **Cadence** (When things run): Some data comes in continuously (social media and SMS), while other data runs on a schedule (call logs or daily web form files)
   - **SLA (**Service promises**):**
     1. Real-time data (streams) should be available within about 10 minutes.
     2. Daily batch data should be processed and ready by 6 AM each morning.
   - **Monitoring** (watchdogs)
     1. Tracking where data came from and how it moved (lineage)
     2. Watching for problems like format changes (schema drift), missing or late data, or unexpected spikes/drops.
     3. Keeping an eye on system health and backlog
     4. If something goes wrong or runs late, alerts are sent to the team immediately.
7. **DataOps** (How the pipeline runs in production, day to day)
   - **Runtime:** The system runs on managed infrastructure. It can scale up automatically to handle bursts of streaming data and has sufficient capacity scheduled for daily batches.
   - **Promotion** (Moving changes safely): Build and test logic in a separate environment using sample data. Once tested, promote it to production with version-controlled configurations and schema (changes can be traceable and reversible
   - **Support** (keeping it reliable):
     Engineers are on-call with documented steps (runbooks) to handle problems like reprocessing data, filling gaps, or fixing errors quickly.
     The system also injects synthetic test messages (canary events) to make sure the pipeline is working end-to-end even when real data is quiet.