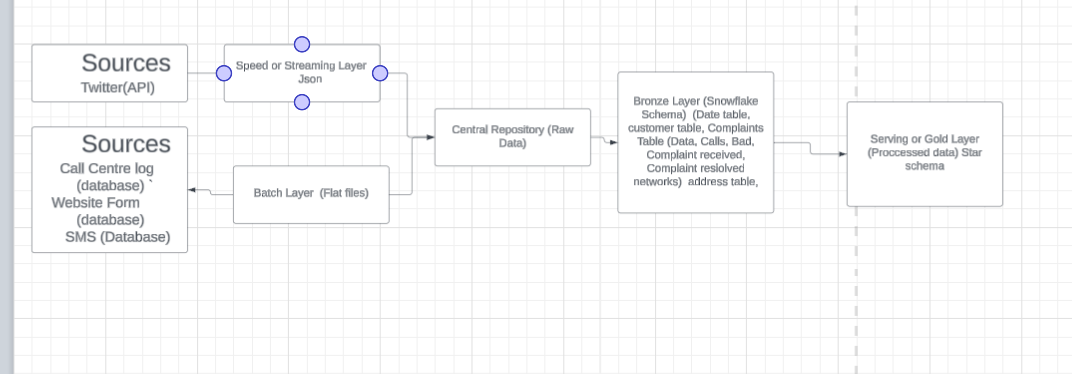
**Beejan Technologies – Conceptual Data Pipeline (Lambda Architecture)**



**Design Choices**

I designed the pipeline using a Lambda Architecture approach to handle both real-time (streaming) and batch data sources.

Streaming Layer: Social media (e.g., Twitter) is ingested continuously in JSON format to capture customer complaints as they happen.

Batch Layer

Call center logs, website forms, and SMS complaints are processed as batch flat files or database extracts.

Both layers feed into a Central Repository (Raw Data), ensuring all incoming data is collected in one place.

From there, the data flows into a Bronze Layer, where data is structured into base tables such as customer, complaints, and address.

Serving/Gold Layer

Provides cleaned and processed data in a star schema for reporting, dashboards, and advanced analytics.

This layered design ensures that real-time monitoring and historical analytics can be supported simultaneously.

Assumptions / Thought Process:

There will be a central repository where all data—regardless of whether it arrives via batch or stream—lands first. This avoids silos and ensures consistent governance.

Data will be progressively refined

Raw Data → Bronze Layer → Gold Layer.

SMS data will first land in a database or gateway before being extracted into the pipeline.

The business needs both real-time insights (e.g., sudden spike in complaints) and batch insights (e.g., weekly or monthly trends).

Standard tables such as date, customer, complaints, and address form the foundation of reporting.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Challenges / Unknowns

SLA (Service Level Agreements): There is currently no clarity on required latencies (e.g., “how quickly must a Twitter complaint show up in dashboards?”). This affects whether to prioritize near real-time or batch-first ingestion.

Data source details: It is not fully clear where all data will arrive from (for example, SMS may come from an SMS gateway or directly from a database). This will affect ingestion design.

Data quality: Unstructured text (e.g., social media posts or SMS) may be messy and require additional natural language processing (NLP) or classification.

Integration complexity: Combining structured call logs with unstructured social media may pose schema alignment challenges.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Other Information:

The design ensures flexibility: batch systems can be scaled independently of streaming systems.

By centralizing data early, the company reduces duplication and makes governance easier.

Using a layered approach (raw → bronze → gold) allows both data engineers (who need raw data) and business analysts (who need cleaned, reliable data) to work from the same ecosystem.

Future extensions could include ML-based complaint classification and real-time alerts to operations teams when complaint spikes are detected.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

In summary

This conceptual pipeline leverages Lambda Architecture with both batch and streaming layers feeding into a central repository, followed by structured bronze and gold layers. While it provides a strong foundation, further clarity is needed on SLAs, source system integration, and data quality handling.