Appendix A. Case Studies

Appendix A.1. Case B: Analytical Data Warehouse

The analytical data warehouse case involves a multinational retail company aiming to improve its reporting and dashboard capabilities by consolidating data from multiple sources. The key challenge was the integration of diverse data formats, including structured files (e.g., Excel and CSV) and semi-structured formats (e.g., JSON). The company also needed a scalable pipeline to handle batch and real-time data ingestion. Using the DAT framework, the architecture was modeled (Figure A.12) to include connectors for various data sources, a cloud-based ETL engine (Keboola), and a Snowflake data warehouse. This architecture facilitated seamless integration and transformation of data into a column-oriented format suitable for analysis. The ThoughtSpot analytics platform delivered interactive dashboards and key performance indicators (KPIs). The DAT framework reduced processing time and manual errors by automating Python code generation for data validation and integration, enabling the company to deliver faster and more accurate insights to stakeholders.

Appendix A.2. Case C: Errors Data Pipeline

This case focuses on a global technology company addressing the challenges of error reporting and analysis for its extensive fleet of printers. The error data, generated in JSON format, required processing into actionable insights to ensure timely troubleshooting and operational improvements. Key challenges included transforming raw error data into formats compatible with the company's analytics and reporting systems.

The DAT framework was applied to design an efficient Errors Data pipeline architecture (Figure A.13). JSON error logs from printers were collected and stored in AWS S3, processed in batches, and transformed into Parquet and CSV formats for downstream use. The framework's automated Python code generation ensured robust data quality checks, validating the completeness and consistency of error logs at each pipeline stage. This approach significantly reduced manual intervention, enhanced data reliability, and improved the company's ability to monitor and address system errors effectively, leading to better overall operational performance.

Appendix A.3. Case E: Hydre (Data Analytic System)

The Hydre case demonstrates the application of the DAT framework in modeling a data system that supports real-time and batch processing workflows, adhering to Lambda and Kappa architectural principles. The system was designed to handle complex data workflows requiring scalable storage, flexible reprocessing, and efficient analytics capabilities.

Using the DAT framework, the architecture was modeled (Figure A.14) to include key components such as a master dataset in Hadoop HDFS for storing raw data and a streaming ETL pipeline powered by Kafka consumers for ingesting and transforming

data. The model also integrated storage systems, including relational, graph, and time-series databases, which enable exploratory analyses conducted via tools like Jupyter notebooks.

A real-time insights component was modeled to extract and aggregate information, such as trending hashtags and user activity, using Kafka Streams. This component efficiently stores results in a time-series database with idempotent operations to ensure data consistency. The Hydre case highlights how the DAT framework can provide a comprehensive design for data-intensive systems by aligning the modeled architecture with Lambda and Kappa principles, ensuring scalability, adaptability, and robust real-time and batch data processing.

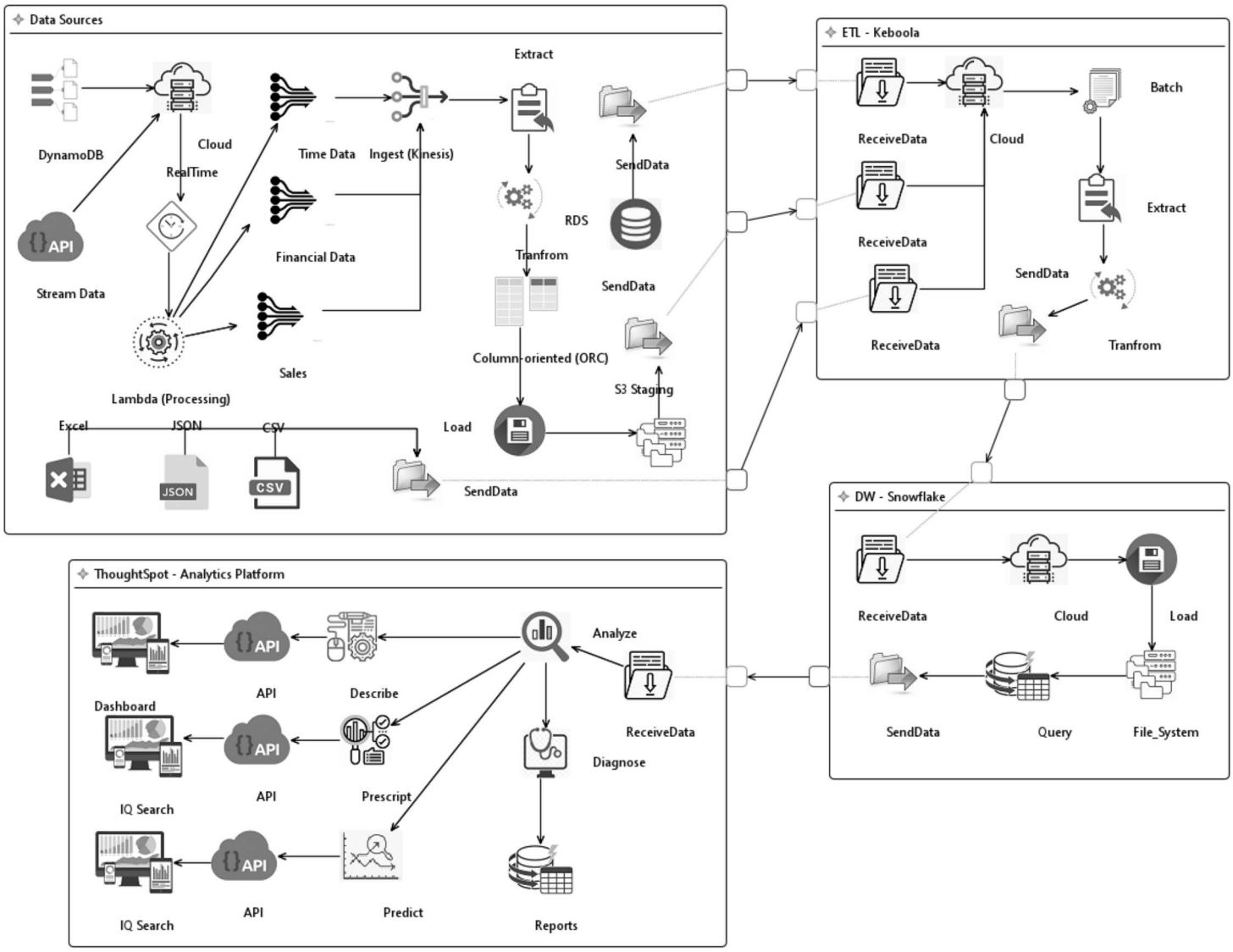

Figure A.12: Case B: DAT Application for Analytical Data Warehouse

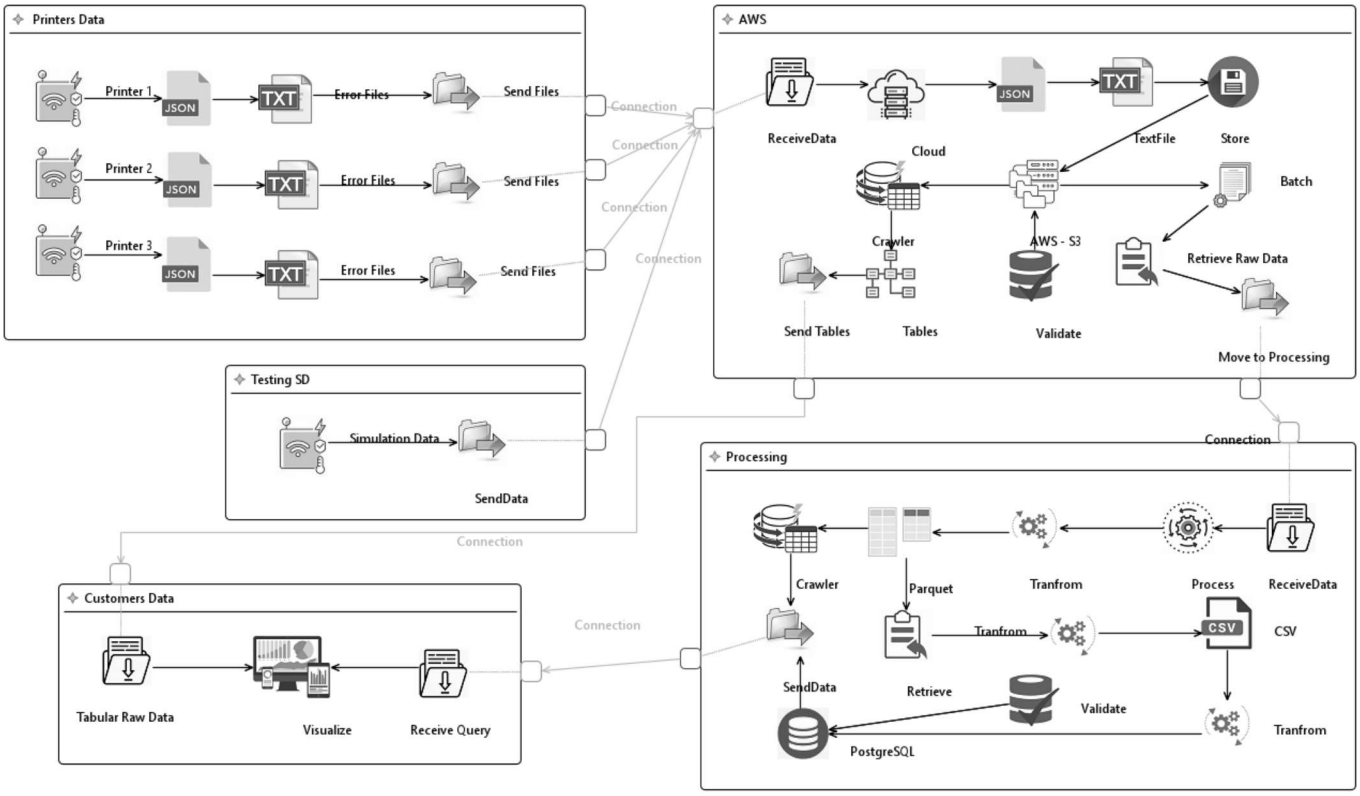

Figure A.13: Case C: DAT Application for the Errors Data Pipeline

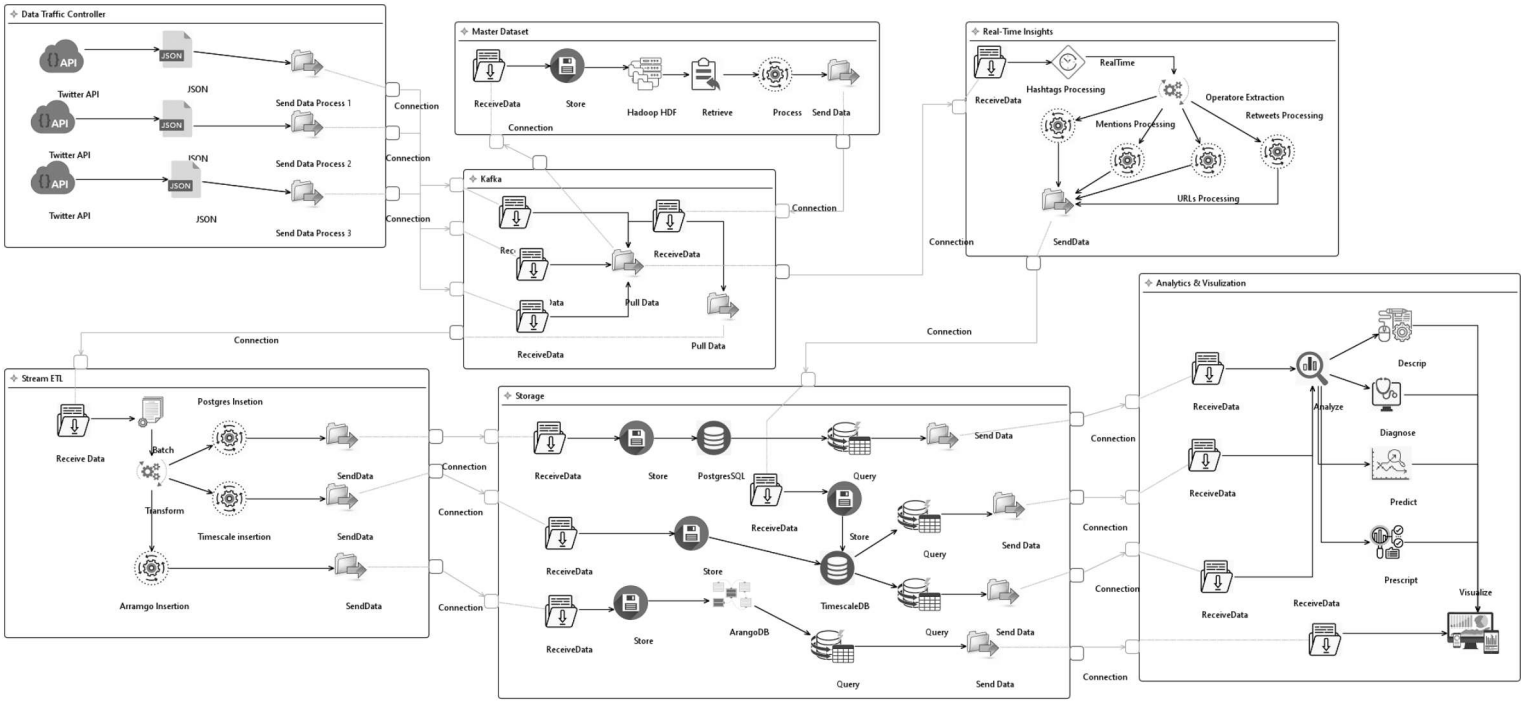

Figure A.14: Case E: DAT Application for The Lambda architecture

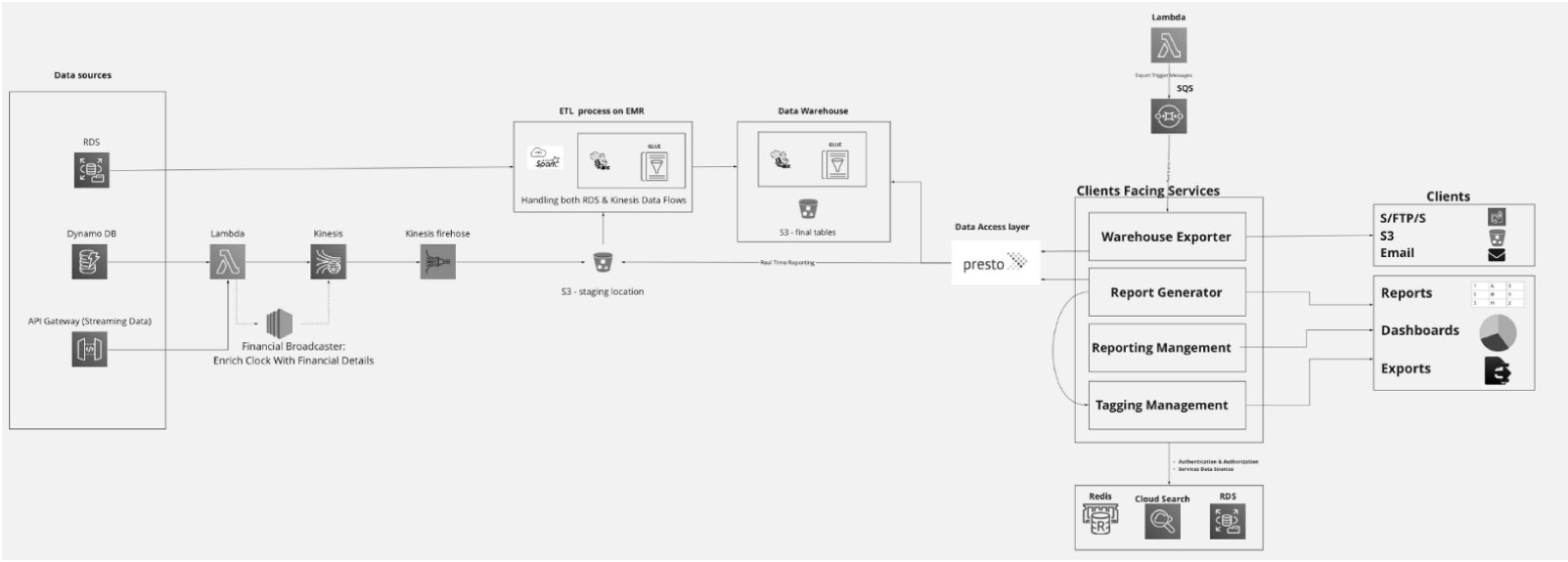

Figure A.15: Operational Data Warehouse