ARTICLE ON

LAKEHOUSE ARCHITECTURE OF

‘CHALO TAXI’

-Arohi Vaghela

https://github.com/arohi-01/Articles

CONTENT

1. Introduction
2. Objective
3. Goals and Mission
4. Designing Phase
5. Description Of Data Sources
6. Initial Architecture
7. Final Architecture
8. Pipeline Ingestion and Failure Strategy
9. Conclusion
10. Appendix

INTRODUCTION:

Chalo Taxi is Calgary's innovative and efficient taxi service, designed to meet the diverse commuting needs of the city's residents and visitors. Inspired by the iconic NYC Taxi system, Chalo Taxi integrates cutting-edge technology, data-driven insights, and local expertise to provide a seamless transportation experience across Calgary's urban and suburban areas.

Why I opted for a Lakehouse Architecture?

- Combines data lake scalability with data warehouse performance.
- Unified platform for real-time insights, batch processing, and predictive analytics.
- Simplifies data management while ensuring cost-effectiveness and scalability.

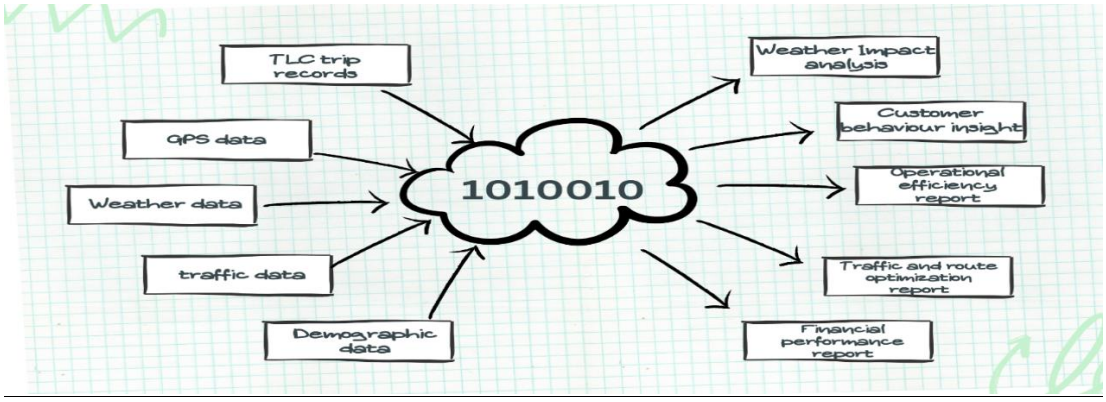
OBJECTIVE:

Leveraging a cloud based architecture to analyse and forecast CHALO TAXI operations.

GOAL AND MISSIONS:

- Streamline data ingestion and process for multiple resources.
- Perform predictive modeling for demand forecasting and traffic analysis.
- Generate actionable insights via dashboards.

DESIGNING PHASE:



Here there are five data sources such trip records, GPS data, weather data, traffic data as well as demographic data which can help in achieving desired outputs including weather impact analysis, customer behaviour, operational efficiency reports, traffic and route optimization report and financial performance report.

DESCRIPTION OF DATA SOURCES:

1. Trip records:

- Nature: Structured batch data.
- Format: CSV, Parquet, or other tabular formats.
- Usage: Ideal for analyzing historical trends, demand forecasting, and fare optimization.

2. GPS Data:

- Nature: Real-time, streaming data.
- Format: CSV, Parquet, or other tabular formats.
- Usage: Ideal for analyzing historical trends, demand forecasting, and fare optimization.

3. Weather Data:

- Nature: Semi-structured, real-time and historical data.
- Format: JSON, XML, or API streams.
- Usage: Used for route optimization, congestion mapping, and live tracking of taxis.

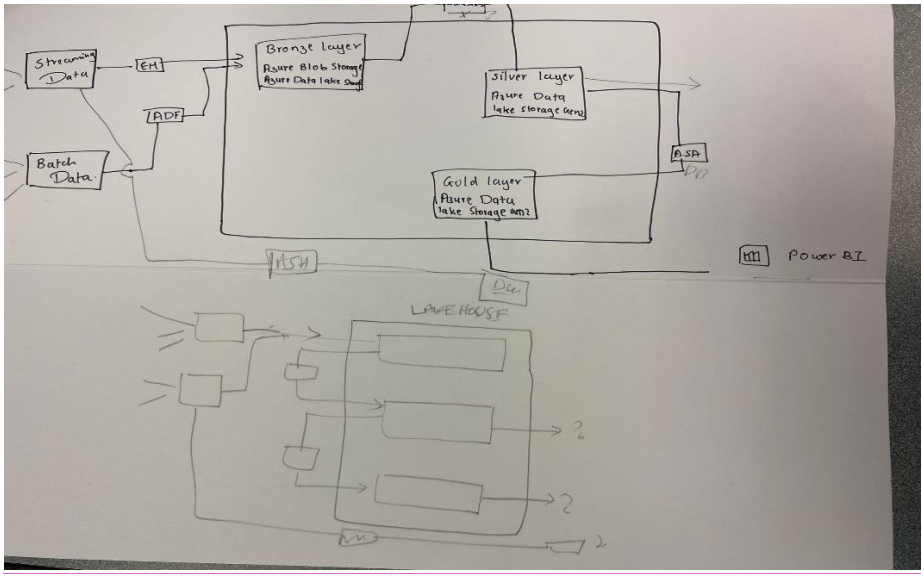
4. Traffic Data:

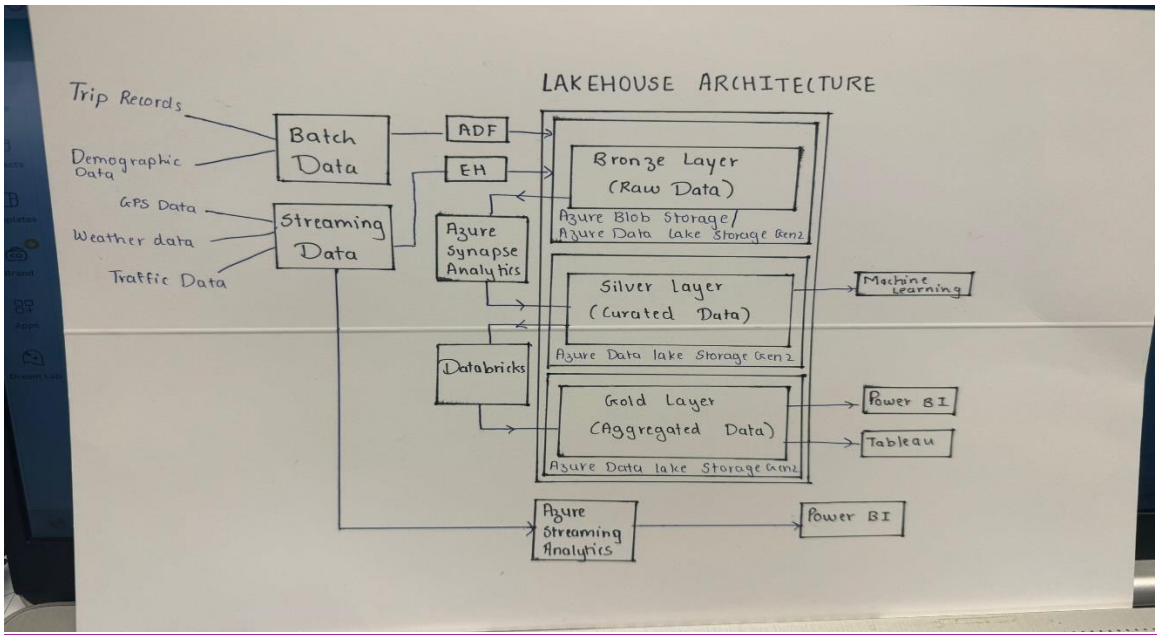
- Nature: Real-time, streaming data.
- Format: JSON or API streams, maps integration (e.g., Google Maps or Waze).
- Usage: Enhances route planning and reduces delays, improving operational efficiency.

5. Demographic Data:

- Nature: Structured, batch data.
- Format: CSV, JSON, or shapefiles for GIS integration.
- Usage: Identifies underserved areas, customer segmentation, and trip demand trends by neighborhood.

INITIAL ARCHITECTURES:

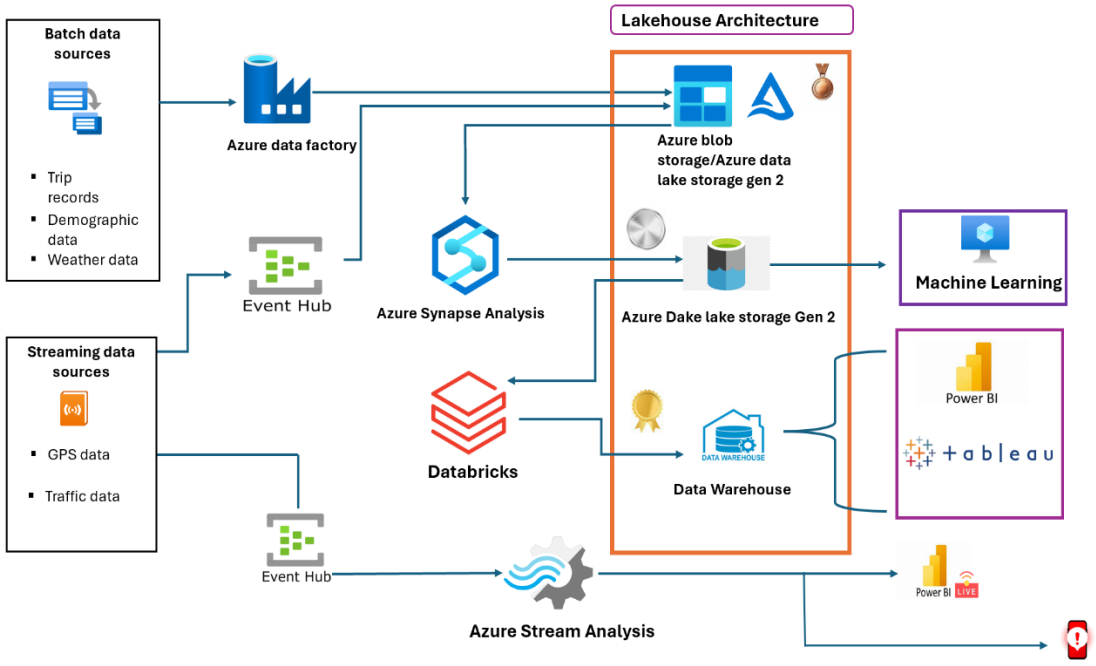




As one can see these rough architectures had minor issues such as in first photo, I did not use certain service at its right place which were data bricks and azure synapse analytics and I even did include multiple options towards sink side. In the second photo, though everything was at right place but I missed to introduce mobile alerts and live dashboards for the drivers. Apart from this, added weather data as a streaming data which is again not a good approach.

FINAL ARCHITECTURE:

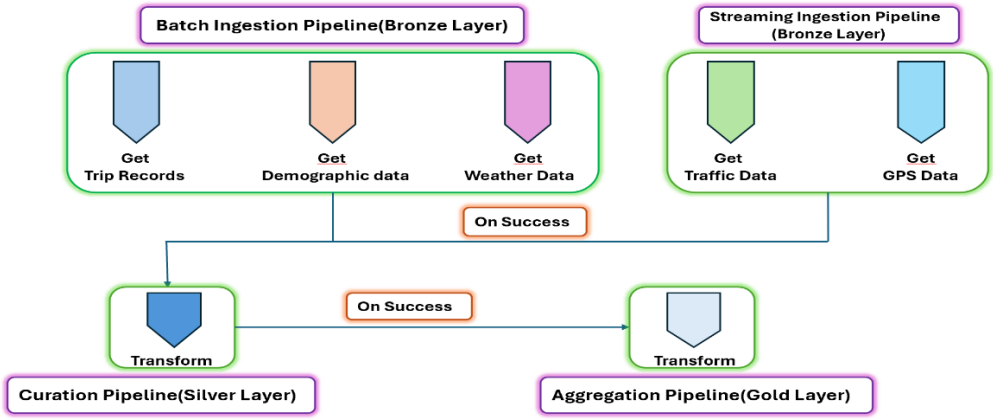
This is the enhanced Lakehouse architecture.



Here in my architecture there has batched as well as streaming data, where I used azure data factory to ingest the batched data in the bronze layer and for the streaming data I have used azure event hub. The streaming data travels in two ways, in first way the data travel directly through the azure streaming analytics which is helpful for producing live dashboards and mobile alerts for the drivers so that drivers can get information regarding traffic in the nearby areas and in the second way the data is directly ingested into bronze layer through event hub. I have used a hybrid approach for storage in bronze layer

where data like location at every now and then which is not that useful can be stored in azure blob storage and rest of the data in azure data lake storage gen 2. The reason for using delta lake in the bronze layer is transactional data as Lakehouse architecture supports ACID transactions . To process data from bronze to silver layer I have used Azure synapse analytics because the raw data is huge and azure synapse analytics can process the data well. The comprehensive data of silver layer is store in azure data lake storage gen 2. This data directly goes to machine learning for creating predictive models and reports. To process the data from silver to gold I have chosen data bricks as the data is big and the storage service for the gold layer is data warehouse as warehouse supports SQL query it will be easy to retrieve aggregated data. Lastly, data travels from gold layer to power bi and tableau where we can visualize the data which can help in analysing trends.

PIPELINE INGESTION:



Batch pipeline:

This pipeline is responsible for ingesting historical or accumulated data in batches and transforming it into a usable form for further processing.

The batch pipeline is chosen for handling larger, pre-existing datasets that can be processed at intervals rather than in real-time. This method is suitable when data is not time-sensitive and can be processed in chunks.

Streaming pipeline:

This pipeline handles real-time, continuous data streams. These are necessary for scenarios where real-time data is critical. The continuous influx of real-time traffic and location data makes this pipeline ideal for applications such as real-time traffic monitoring or navigation systems that require timely data to provide insights.

Curation Pipeline:

A data curation pipeline organizes, cleans, and enriches raw data to ensure it is high-quality, relevant, and ready for analysis. In summary, a curation pipeline turns raw data into actionable insights, helping taxi services optimize operations, improve customer experience, and support strategic growth.

Aggregation Pipeline:

An aggregation pipeline helps taxi services derive critical insights from data, supporting decision-making and improving efficiency, revenue, and customer satisfaction.

What if the pipeline fails?

Here are few measure which can be taken on such circumstances:

1. Proactive Monitoring: Implement real-time monitoring to detect issues early using tools like Grafana or Datadog.
2. Retry Mechanism: Automatically retry failed jobs a set number of times.
3. Alerting and Notifications: Send immediate alerts to the responsible teams when failures occur.
4. Checkpoints: Save progress at intermediate stages to enable recovery from the last successful point.
5. Incident Management: Establish a clear escalation and resolution process for critical failures.
6. Logging and Auditing: Maintain detailed logs for traceability and debugging.

7. Failover Systems: Use backup or redundant systems for critical components to ensure continuity.

Trigger to run a pipeline: -

Event based and schedule based = Real-Time Alerts & Notifications

- *Proactive Decision-Making*
- **Driver Alerts**: Route suggestions during traffic jams.
- **Fleet Manager Notifications**: Alerts for demand spikes during peak hours.
- **Tools**:
    - SMS/Email: Twilio, SendGrid.
    - Live APIs: Azure Maps or Google Maps for navigation.

CONCLUSION:

-Unified data platform for real-time insights and scalability.

-Simplifies data workflows while enabling advanced analytics.

-Future-proof solution for growing business demands.

APPENDIX:

Description of outputs of designing phase:

1. Operational Efficiency Reports:

Delivery Method:

Interactive Dashboards (Power BI):

Purpose: Provide real-time insights into driver performance, idle times, and fleet utilization.

Example: Heatmaps showing high-demand zones or a line graph tracking idle time trends.

2. Customer Behavior Insights:

Delivery Method:

Mobile-Friendly Dashboards (Power BI):

Purpose: Allow fleet managers and marketing teams to view customer trends on-the-go, enabling targeted advertising or service adjustments.

Details: Dashboards displaying pickup/dropoff trends or underserved areas.

3. Demand Forecasting Reports

Delivery Method:

Predictive Models Embedded in Applications:

Purpose: Provide real-time demand predictions for operational planning.

4. Traffic and Route Optimization Reports:

Delivery Method:

1. Live Navigation Integration (Azure Maps):

Purpose: Deliver optimized routes directly to drivers in real-time to reduce trip duration.

Details: Drivers receive route recommendations that avoid traffic congestion, improving efficiency and customer satisfaction.

2. Traffic Reports in Power BI:

Purpose: Help managers analyze traffic patterns over time for planning purposes.

Details: Visualizations such as congestion heatmaps and average trip duration charts.

5. Weather Impact Analysis:

Delivery Method:

1.Dynamic Dashboards (Power BI):

Purpose: Fleet managers use weather dashboards to prepare for demand spikes during adverse conditions.

Details: Correlation graphs between weather patterns and trip volumes, combined with real-time weather alerts.

2. SMS/Email Alerts:

Purpose: Notify drivers and fleet managers about expected demand surges due to weather.

Details: Short, actionable messages providing information on anticipated ride demand increases in specific areas.