

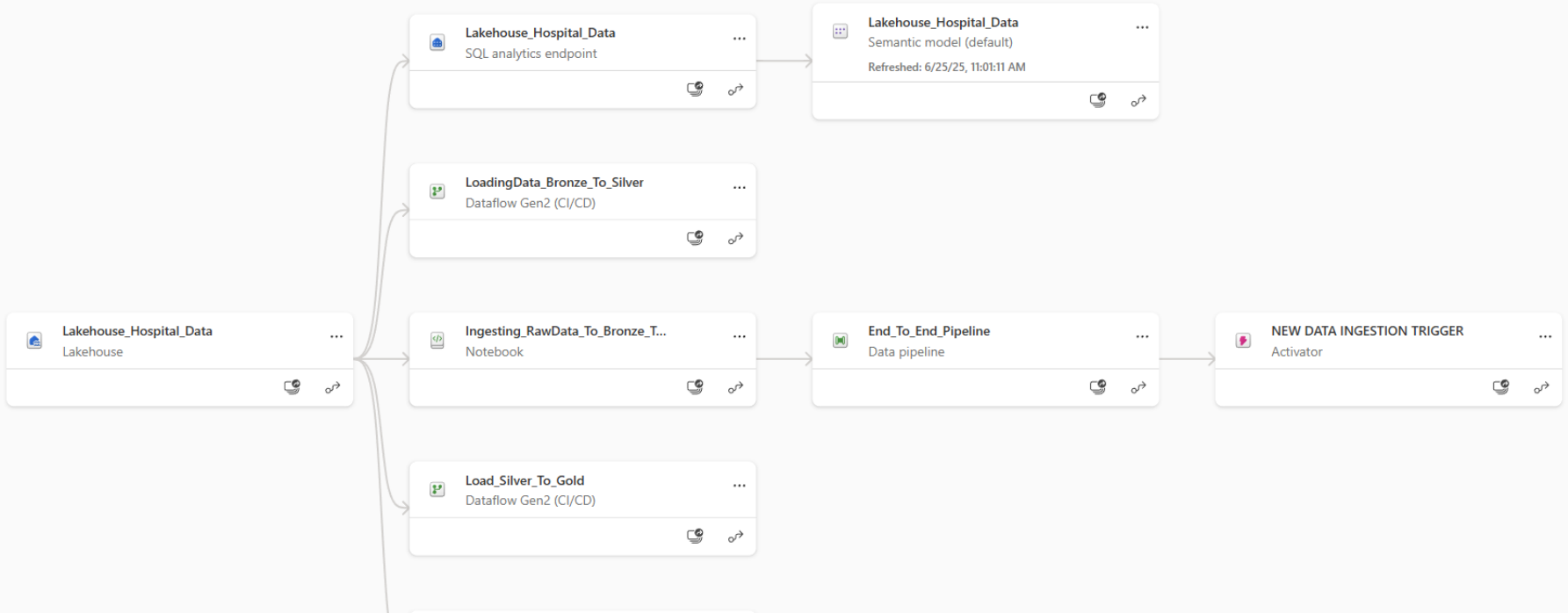
340B Data Integration & Analytics Pipeline – Microsoft Fabric

Asad Patel

Email: asadpatel517@gmail.com

GitHub: <https://github.com/apatel517>

LinkedIn: <https://www.linkedin.com/in/asad--patel>



Executive Summary

- This project was developed in a hospital setting where six pharmacy-related datasets are received monthly in raw CSV format. These datasets are loaded into a Microsoft Fabric Lakehouse and processed using the Medallion Architecture (Bronze → Silver → Gold).
- The pipeline is fully automated:
 - Raw data is ingested into Bronze tables using a PySpark Notebook.
 - Dataflow Gen2 transforms and cleans the Bronze data into Silver tables.
 - A second Dataflow processes Silver tables into Gold tables into star-schema tables using fact and dimension modeling.
 - Gold tables are connected directly to a Power BI dashboard for visualization.
 - A Microsoft Fabric Data Pipeline is configured to automatically trigger the entire flow when new files arrive.



Architecture – Medallion Data Flow

- Medallion Architecture Flow:
- - Monthly Raw CSVs (OneLake)
 - ↓ Triggered Pipeline
 - Notebook (Ingest to Bronze Layer)
 - ↓
 - Dataflow Gen2 (Transform to Silver Layer)
 - ↓
 - Dataflow/SQL (Aggregate into Gold Layer)
 - ↓
 - Power BI Dashboard (Connected to Gold Tables)

```
1 from pyspark.sql.functions import input_file_name, to_date, col
2 from pyspark.sql.types import DoubleType, IntegerType
3
4 # -----
5 # 1. Claims Data
6 # -----
7 claims_path = [redacted]
8 df_claims = spark.read.option("header", True).csv(claims_path)
9 df_claims = (
10     df_claims
11     .withColumn("source_file", input_file_name())
12     .withColumn("payment_amount", col("payment_amount").cast(DoubleType()))
13     .withColumn("processed_date", to_date("processed_date", "yyyy-MM-dd"))
14 )
15 df_claims.write.format("delta").mode("append").option("mergeSchema", "true").
16
17 # -----
18 # 2. Prescriptions Data
19 # -----
20 prescriptions_path = [redacted]
21 df_prescriptions = spark.read.option("header", True).csv(prescriptions_path)
22 df_prescriptions = (
23     df_prescriptions
24     .withColumn("source_file", input_file_name())
25     .withColumn("rx_date", to_date("rx_date", "yyyy-MM-dd"))
26
27     .option("mergeSchema", 'true').write.format("delta").mode("append").option("mergeSchema", 'true')
```

Name	Date modified
June_2025_claims.csv	6/11/2025, 8:56:00 PM
May_2025_Claims.csv	6/11/2025, 7:17:17 PM

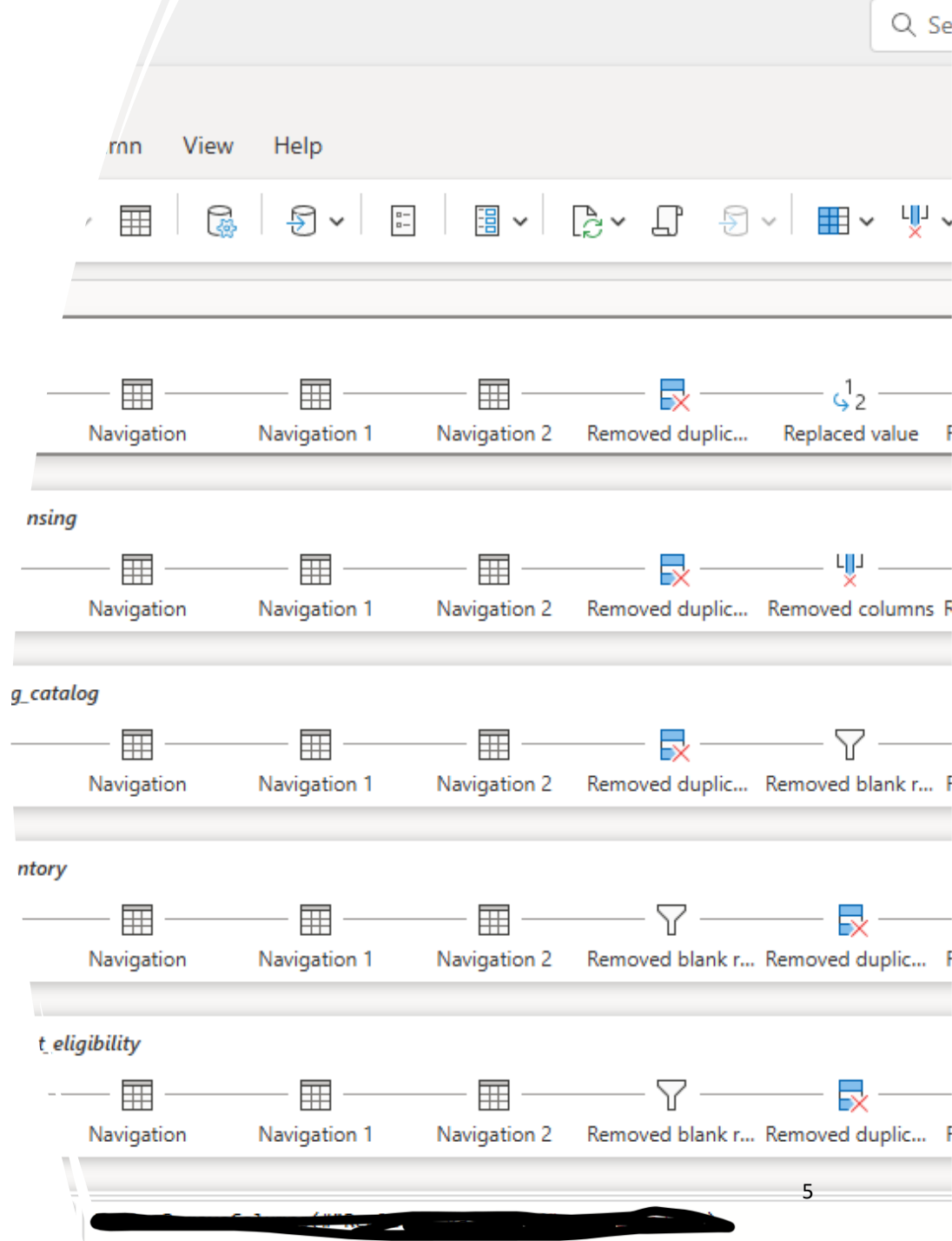
Bronze Layer

– Raw Ingestion

- Tool: PySpark Notebook
- - Ingests raw CSV files from OneLake
 - Applies schema validation and basic data profiling
 - Stores raw but structured data in Lakehouse Bronze tables

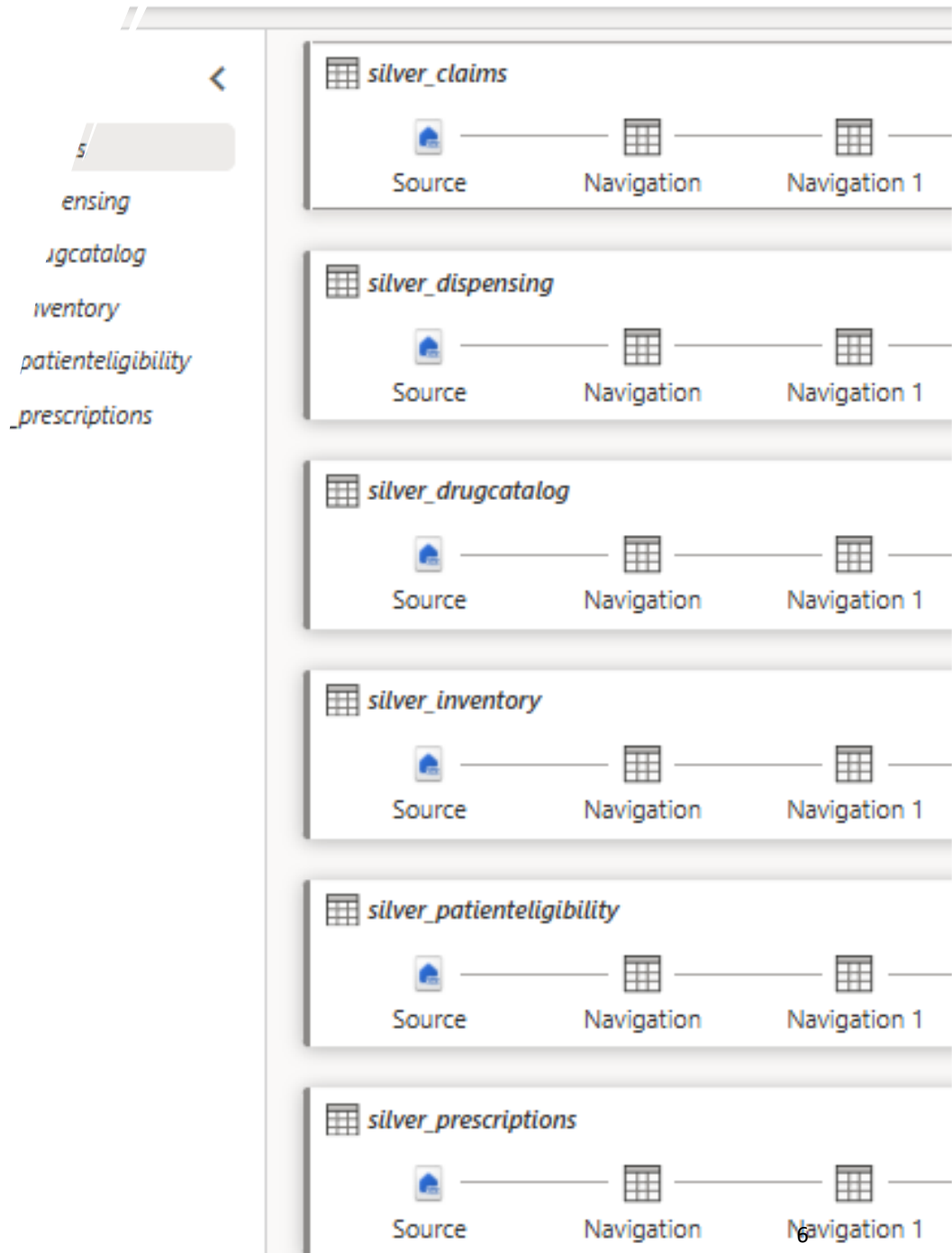
Silver Layer – Cleaning & Joins

- Tool: Dataflow Gen2
- - Cleans and standardizes Bronze table data
- - Deduplicates monthly entries using keys/timestamps
- - Performs joins across datasets (e.g., eligibility, claims, inventory)
- - Outputs structured, analytics-ready tables



Gold Layer Details

- Tool: Dataflow Gen2
- Builds a **star schema** using fact and dimension tables for analytics.(Include schema or SQL screenshot)
- Provides clean, joinable data for each subject area (Eligibility, Claims, Prescriptions, Inventory, etc.)



- Connected to curated Gold tables
- A single, unified entry point to navigate across dashboards for Eligibility, Prescription Monitoring, Dispensing, Claims & Revenue, and Inventory. Designed for pharmacy leadership and compliance teams to drive actionable insights with a few clicks





7

Pipeline Automation – Trigger Flow

- Microsoft Fabric Data Pipeline:
- - Triggered when new files arrive in OneLake folder
- - Executes PySpark Notebook to load Bronze layer
- - Automatically runs Dataflows to build Silver and Gold layers
- - No manual steps – monthly refresh runs end-to-end

The screenshot displays the Microsoft Fabric Data Pipeline interface. The top section shows the pipeline definition for 'End_To_End_Pipeline', which consists of a Notebook activity 'Ingesting_RawData_To_Bronze' followed by two Dataflow activities: 'Bronze_To_Silver' and 'Silver_To_Gold'. The bottom section shows the 'History' tab, which is currently empty, displaying the message 'No data to show. Try changing parameters, population sample, or time range.' The right sidebar contains the 'Definition' panel with sections for 'Monitor', 'Condition', and 'Action', and the 'Advanced settings' section at the bottom.

Outcomes & Business Value

- -  Reduced manual data entry and errors
-  Fully automated monthly processing
-  Real-time analytics and compliance tracking
-  Modular and scalable design for other hospital datasets

Contact Information

- To request a walkthrough or access to sample visuals:



Email: asadpatel517@gmail.com



GitHub: <https://github.com/apatel517>



LinkedIn: <https://www.linkedin.com/in/asad--patel>