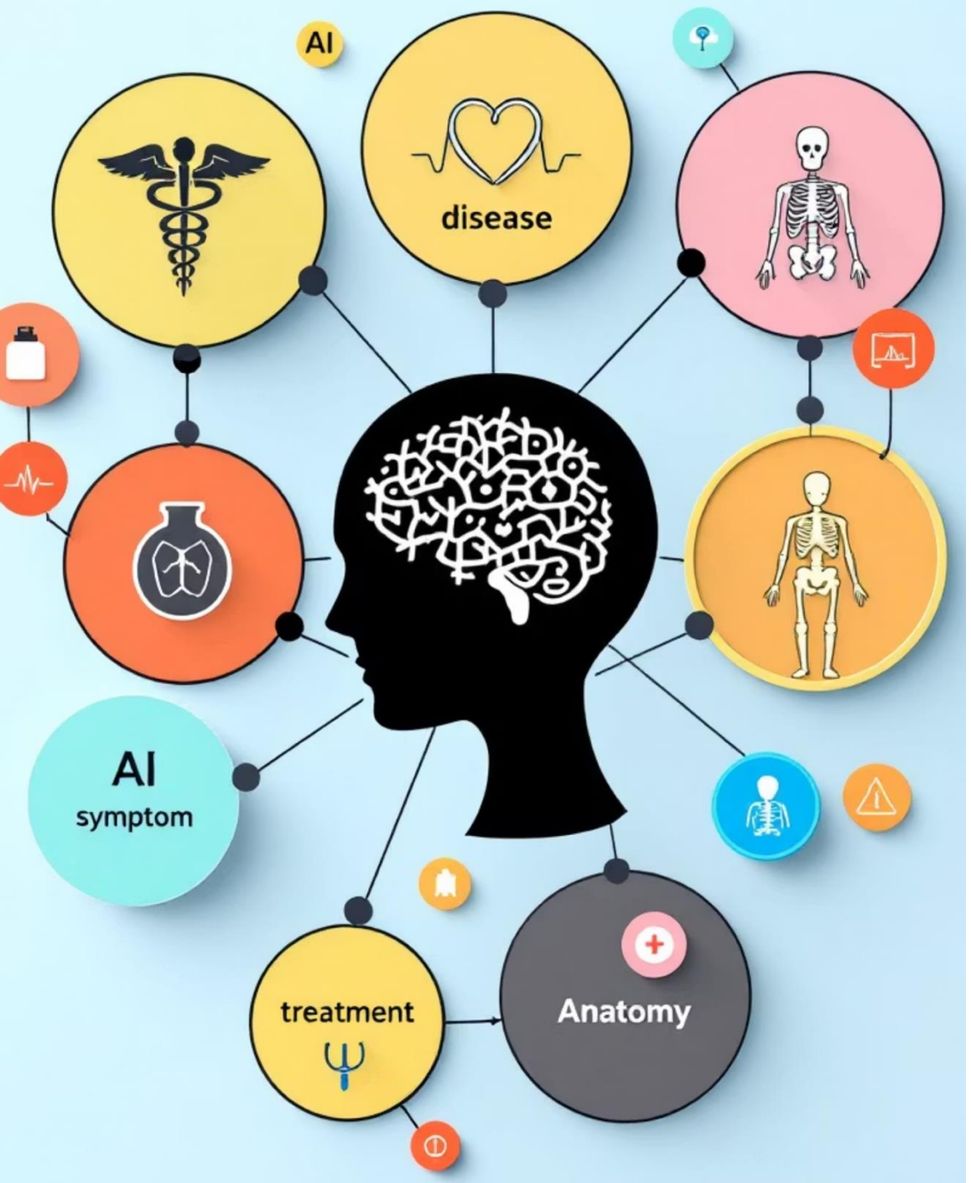


MediGraph AI - Intelligent Healthcare Knowledge Graph with LLM Integration

DAMG 7374 — Group 3 :

Aravind Balaji • Sai Manasa Karanam • Varun Tadimeti

Problems We Addressed

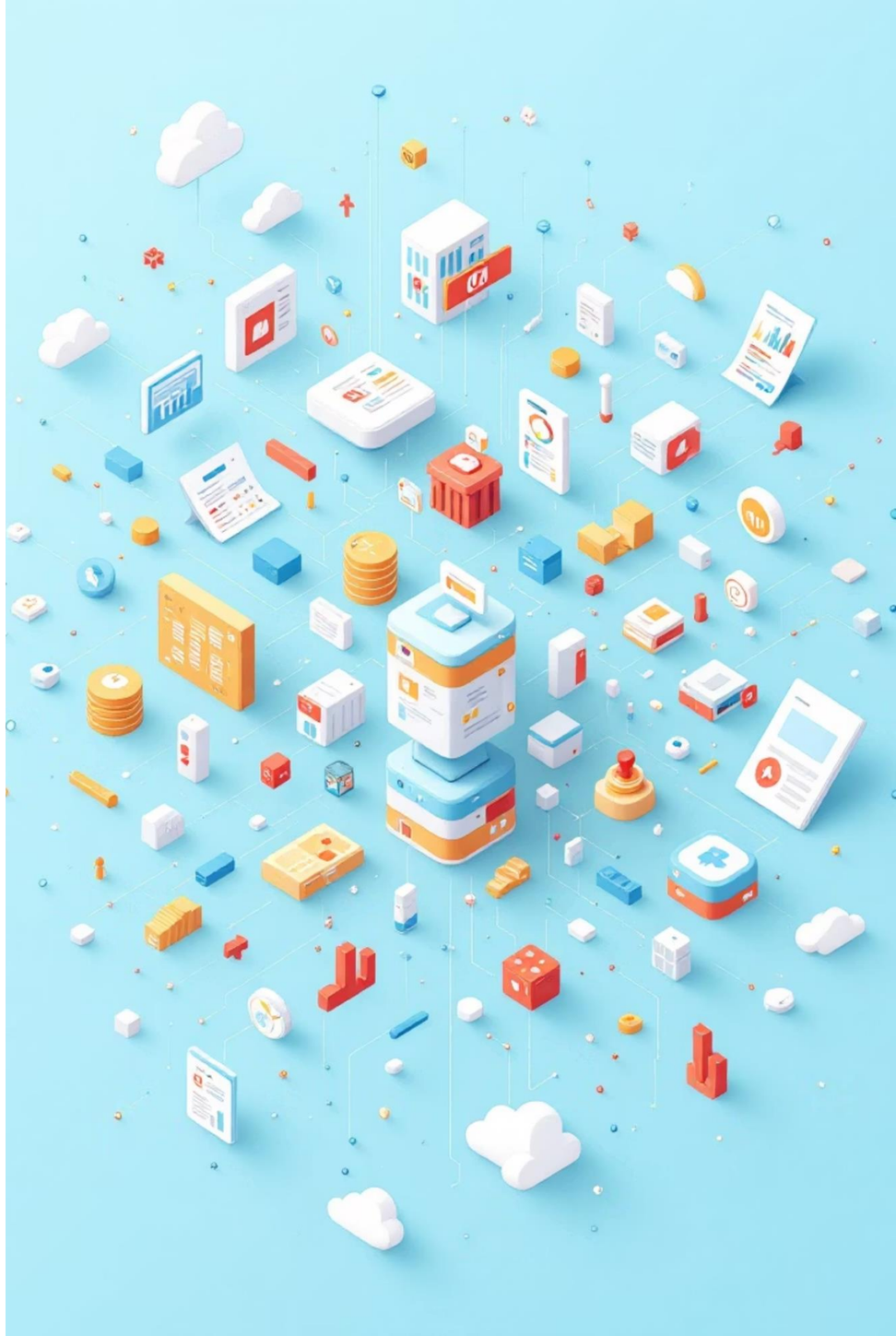
Healthcare data is scattered across tables:

- Patients
- Encounters
- Conditions
- Medications
- Observations
- Providers

This fragmentation makes it difficult to:

- See a full patient journey
- Identify clinical patterns
- Query relationships easily
- Support decision-making

Graph modeling + LLMs solve this.



Challenges & Solutions

Challenges:

- LLM hallucinations
- Missing GDS in Aura
- Large dataset ETL
- UI readability & theme
- Observations linking

Solutions:

- Added evaluations module
- Implemented safe Cypher wrapper
- Added incremental loading
- Fixed dark-theme alignment
- Added 112k observations correctly

Milestones Achieved (Phase-wise progress)

Phase 1 – Problem Definition & Planning
Defined the healthcare use-case and project scope covering patients, encounters, providers, conditions, medications, and observations.

Phase 2 – Snowflake Data Ingestion
Ingested Synthea EHR data into Snowflake, created analytical views, and enabled secure MFA/TOTP authentication.

Phase 3 – Graph Schema Design
Designed a clinically meaningful Neo4j knowledge graph schema with nodes and relationships reflecting patient journeys.

Phase 4 – ETL Pipeline Implementation
Built a Python-based ETL pipeline to transform Snowflake data into Neo4j AuraDB with scalable and demo-safe loading.

Phase 5 – Streamlit Application Development
Developed an interactive UI for data validation, graph visualization, and healthcare analytics.

Phase 6 – NL & LLM-based Querying
Implemented rule-based and LLM-powered natural language Q&A across all healthcare entities.

Phase 7 – Observations & Evaluations (Professor Feedback)
Integrated clinical observations and added evaluation metrics to assess graph richness and data coverage.

Phase 8 – Finalization & Demo Readiness
Completed all features within the initial scope and prepared the project for final presentation and demo.

Phase 9 - Completed • Evaluation-Ready • Final Demo Prepared

Project Objectives

1. Build a unified healthcare knowledge graph
2. Integrate Snowflake → Neo4j → Streamlit
3. Support live clinical queries using NL → Cypher (LLM)
4. Add automated observations, guidelines, and evaluations
5. Provide analytics using PageRank & communities

High-Level Architecture

Pipeline:

01

Snowflake Warehouse

ETL + curated EHR views

02

Python ETL

MERGE nodes & relationships

03

Neo4j AuraDB

Patient graph model

04

Streamlit UI

Live exploration and Q&A

05

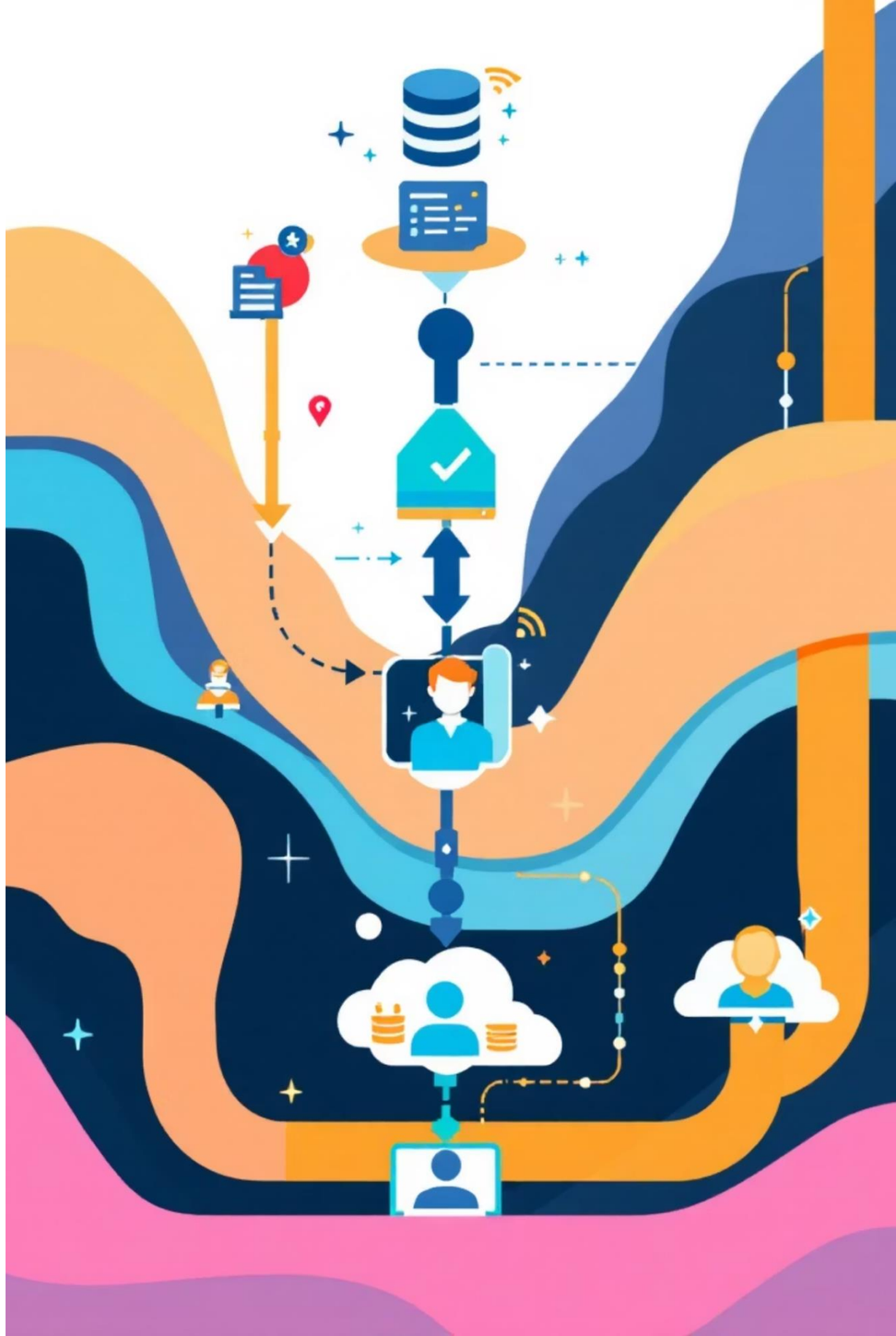
LLM Agent

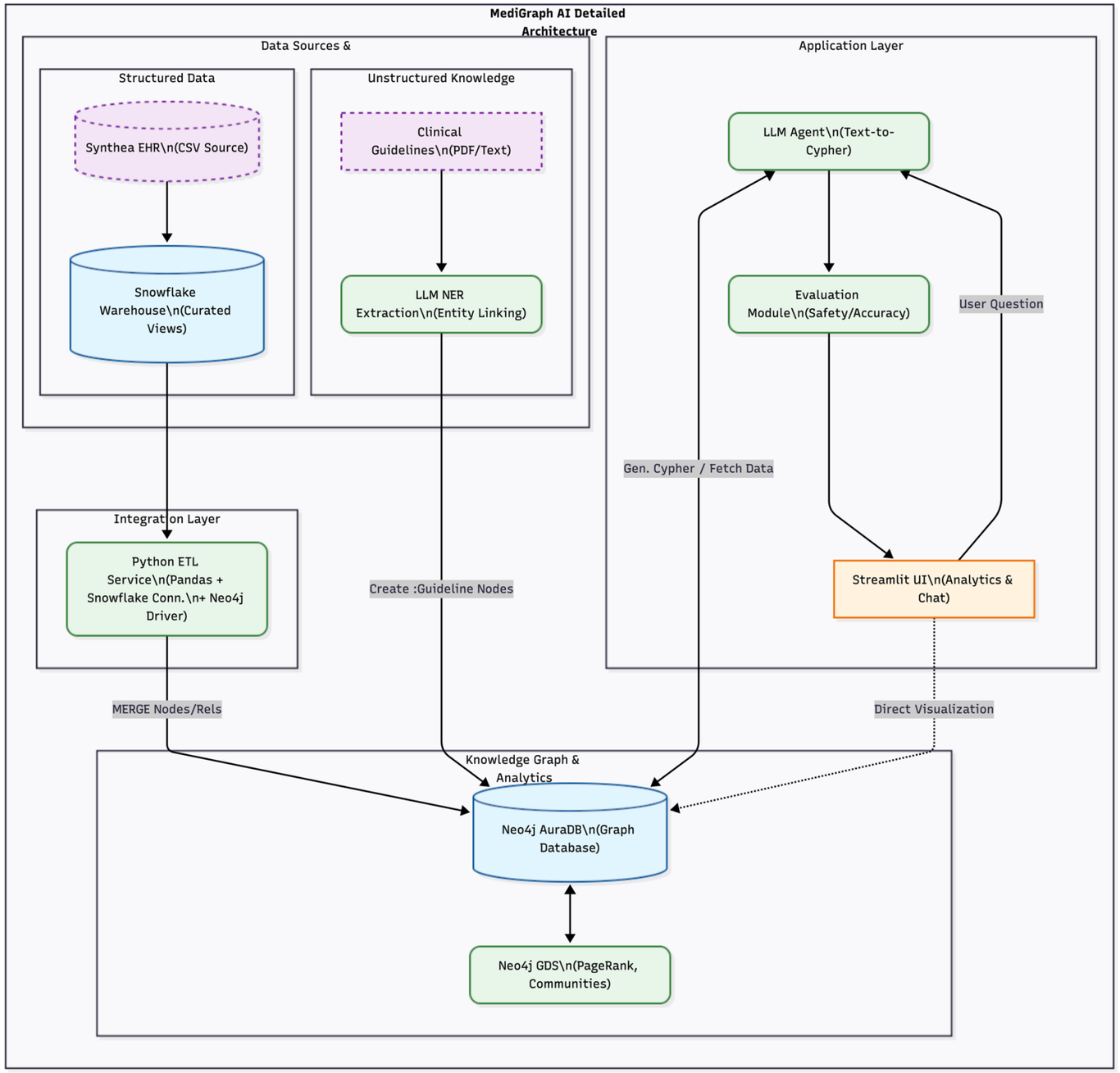
Cypher generation + grounded answers

06

Evaluations

Accuracy, completeness, safety scoring





Dataset (Synthea Synthetic EHR)

Loaded into Snowflake:

- 121 Patients
- 7,688 Encounters
- 4,750 Conditions
- 7,455 Medications
- 282 Providers
- 112,177 Observations

Structured & linked for graph modeling.

Python ETL Pipeline

- Snowflake Connector + Neo4j Driver
- Incremental loading
- 7,000-row cap for demo speed
- Loads:
  - Patient
  - Provider
  - Encounter
  - Condition
  - Medication
  - Observation
- Auto-progress logging
- Skips categories already loaded

Snowflake Warehouse

- CSV ingestion → curated views
- MFA / TOTP-secured access
- Analytical views:
  - V_PATIENTS
  - V_ENCOUNTERS
  - V_CONDITIONS
  - V_MEDICATIONS
  - V_PROVIDERS
  - OBSERVATIONS

Enables clean ETL into Neo4j.

Neo4j Graph Schema

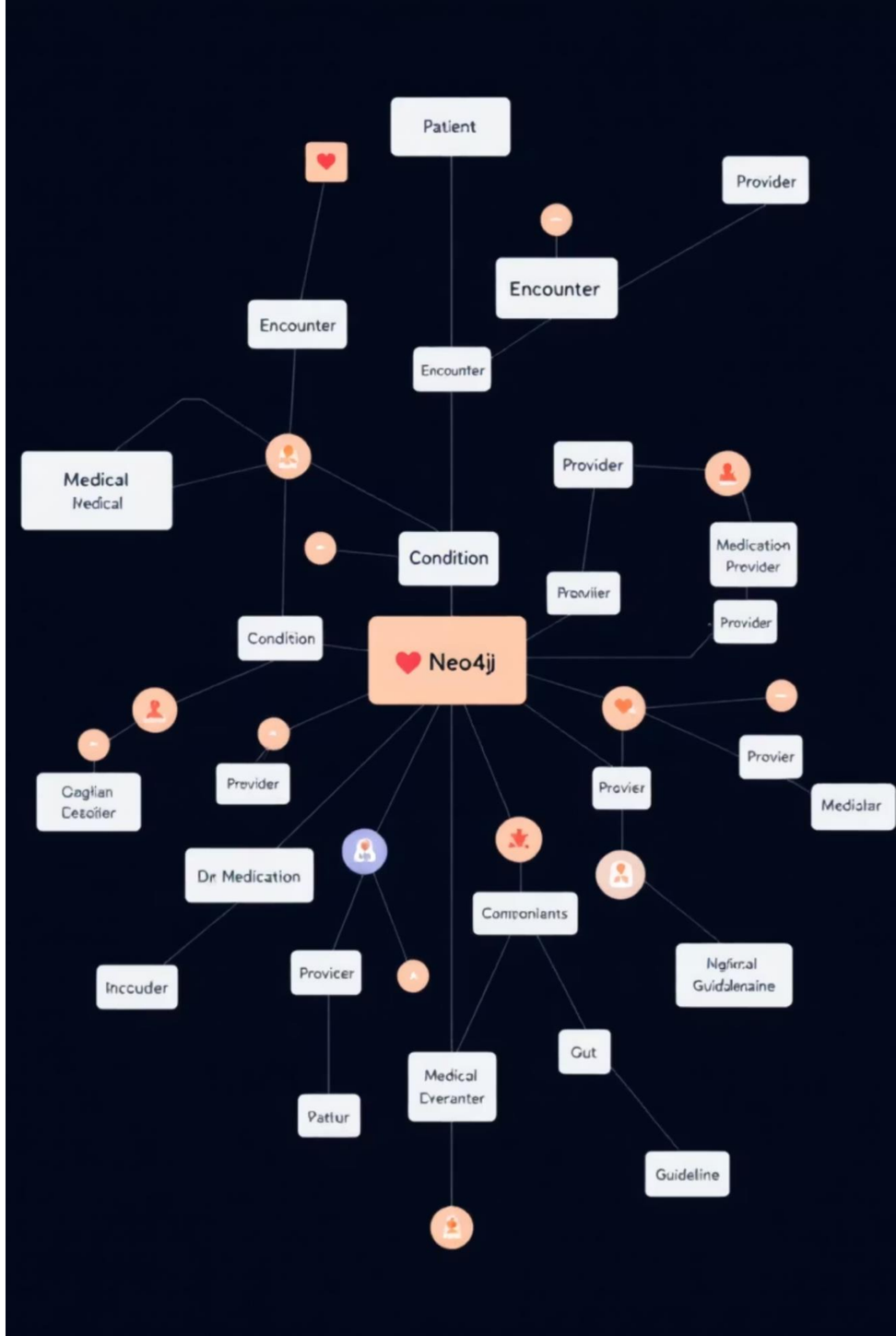
Nodes:

- Patient
- Encounter
- Condition
- Medication
- Observation
- Provider
- Guideline

Relationships:

- HAS_ENCOUNTER
- HAS_CONDITION
- TAKES_MEDICATION
- HAS_OBSERVATION
- HAS_PROVIDER
- MENTIONS_CONDITION
- MENTIONS_MEDICATION

Graph model supports multi-hop reasoning.



Streamlit UI (End-to-End)

Tabs included:

- Product Overview
- Snowflake Views
- AuraDB Graph Explorer
- AuraDB Analytics (GDS-like)
- Guidelines & NER
- LLM Q&A (Cypher Generator)

Features:

- Live Snowflake + AuraDB connection
- Dark sleek UI
- Patient graph visualization

Observations Integration

Added full support for:

- Vitals
- Labs
- Scores
- Measurements

Observation nodes connected to:

- Patients
- Encounters
- Codes

Enables queries like: "Show observations for Patient X "

Analytics (Simulated GDS)

- Provider PageRank
- Condition clusters
- Encounter network density
- Patient-provider connectivity

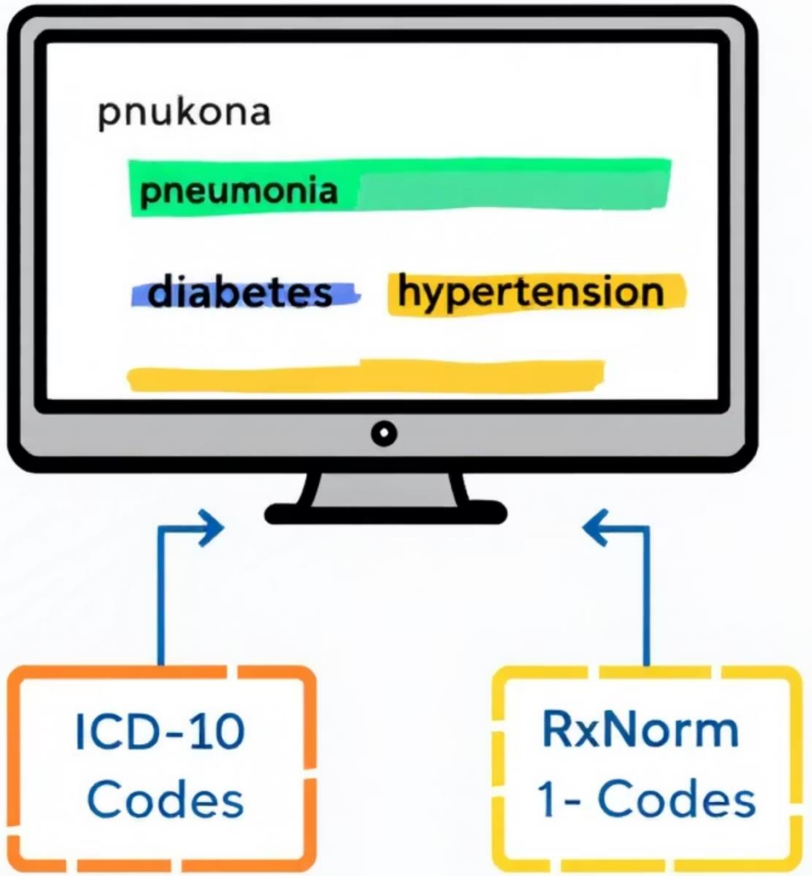
Displayed in UI under AuraDB Analytics tab.

Guidelines & NER

Pipeline:

1. Paste clinical guideline text
2. LLM NER extracts ICD-10, RxNorm, key phrases
3. Guideline nodes created
4. MENTIONS_* relationships attached to graph

Enables evidence-based reasoning.



Evaluations

Every LLM response evaluated on:

- Correctness
- Completeness
- Safety
- Grounding to data

Scores stored & displayed inside UI. Enhances trustworthiness.

LLM-Powered Q&A

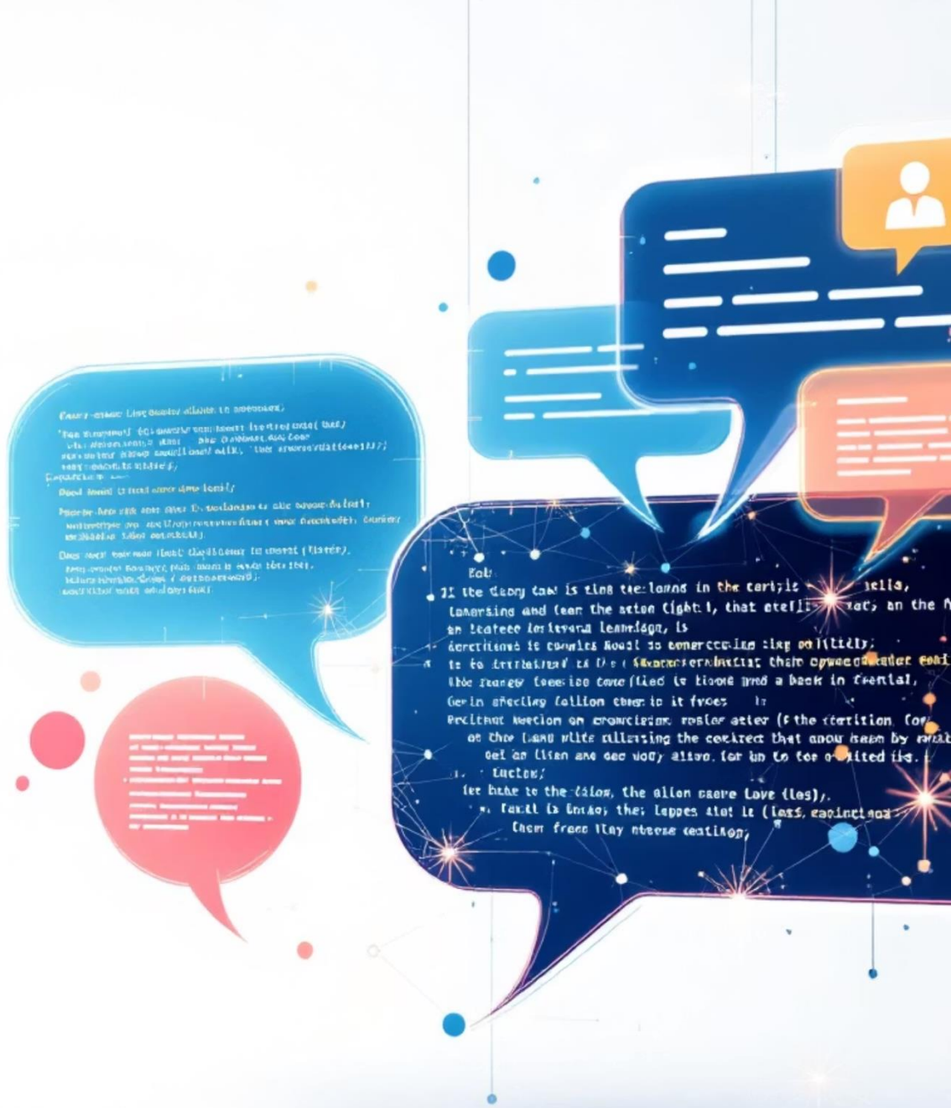
Ask natural questions:

- "List 5 patients with diabetes."
- "Show encounters for patient John."
- "Which medications are prescribed for hypertension?"
- "Show observations for diabetic patients."

LLM converts NL → Cypher → Neo4j results.

Includes:

Bright Cypher display



Demo Workflow

01

Connect Snowflake (TOTP)

02

Connect Neo4j AuraDB

03

Explore tables

04

Visualize patient graph

05

Run LLM questions

06

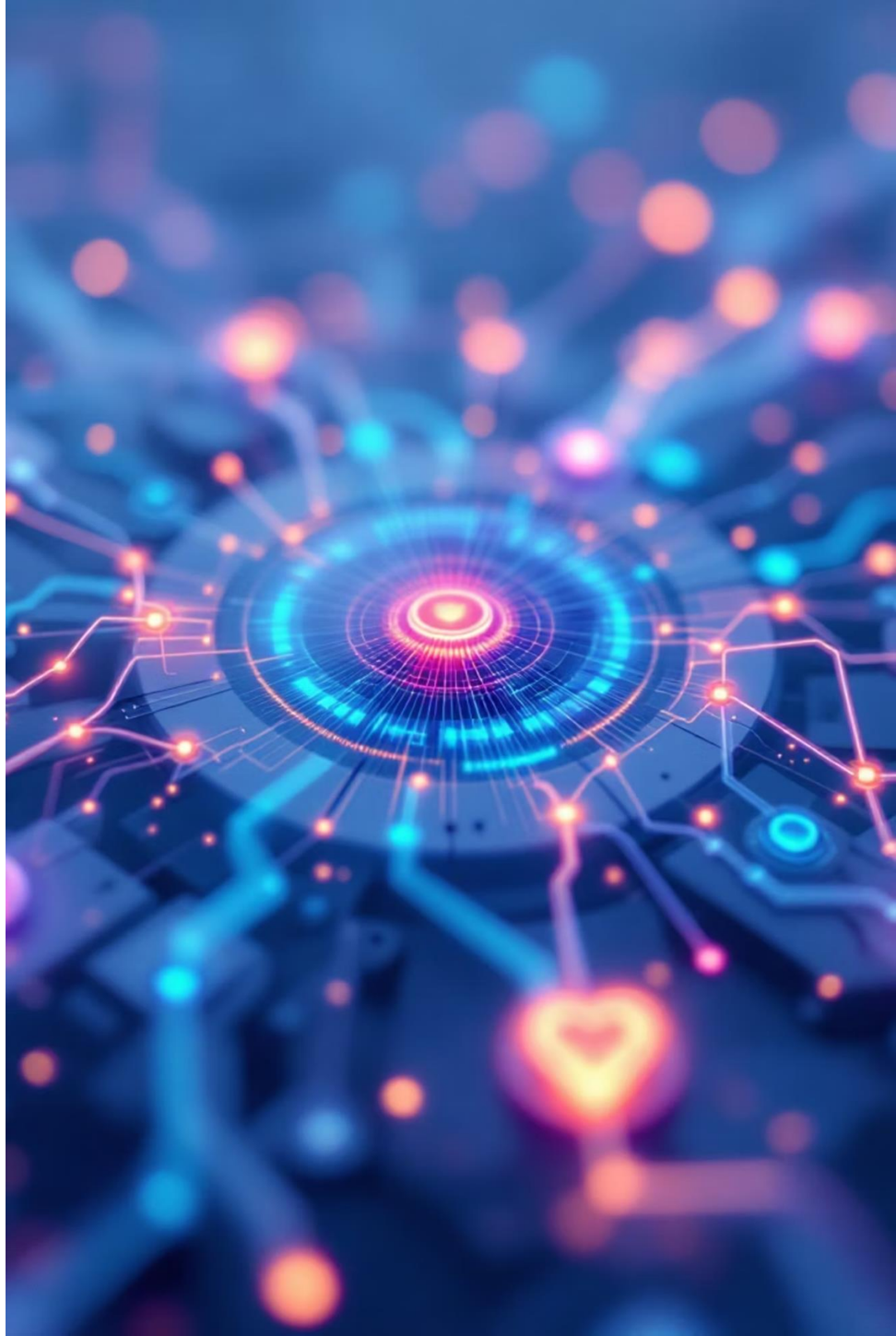
Extract guidelines

07

View evaluations

Future Work

- GraphRAG clinical decision support
- Temporal reasoning (time-based events)
- FHIR API integration
- Embeddings + vector search
- Provider-ranking engines
- Auto-explanations of clinical queries



Thank You