neo4j Graph Database & Analytics

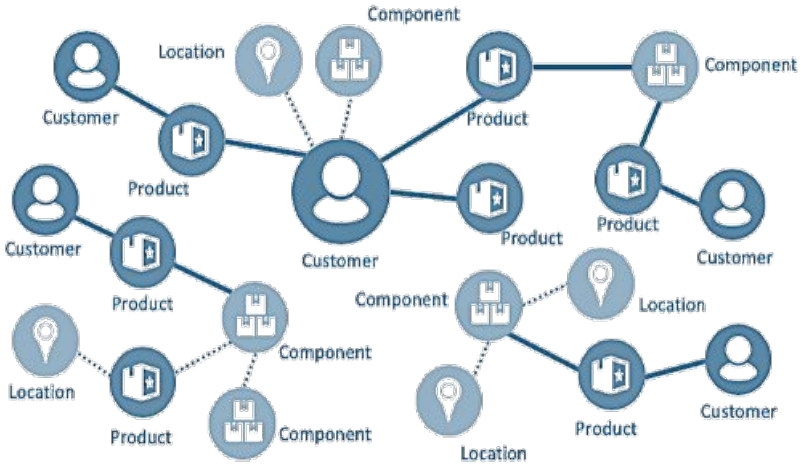
# nodes & network Gen AI Workshop

# Agenda

1. Graph Concepts & Neo4j Platform
2. GraphRAG and KG Construction architecture
3. MCP and Agentic workflows
4. Demos

# Why Graph

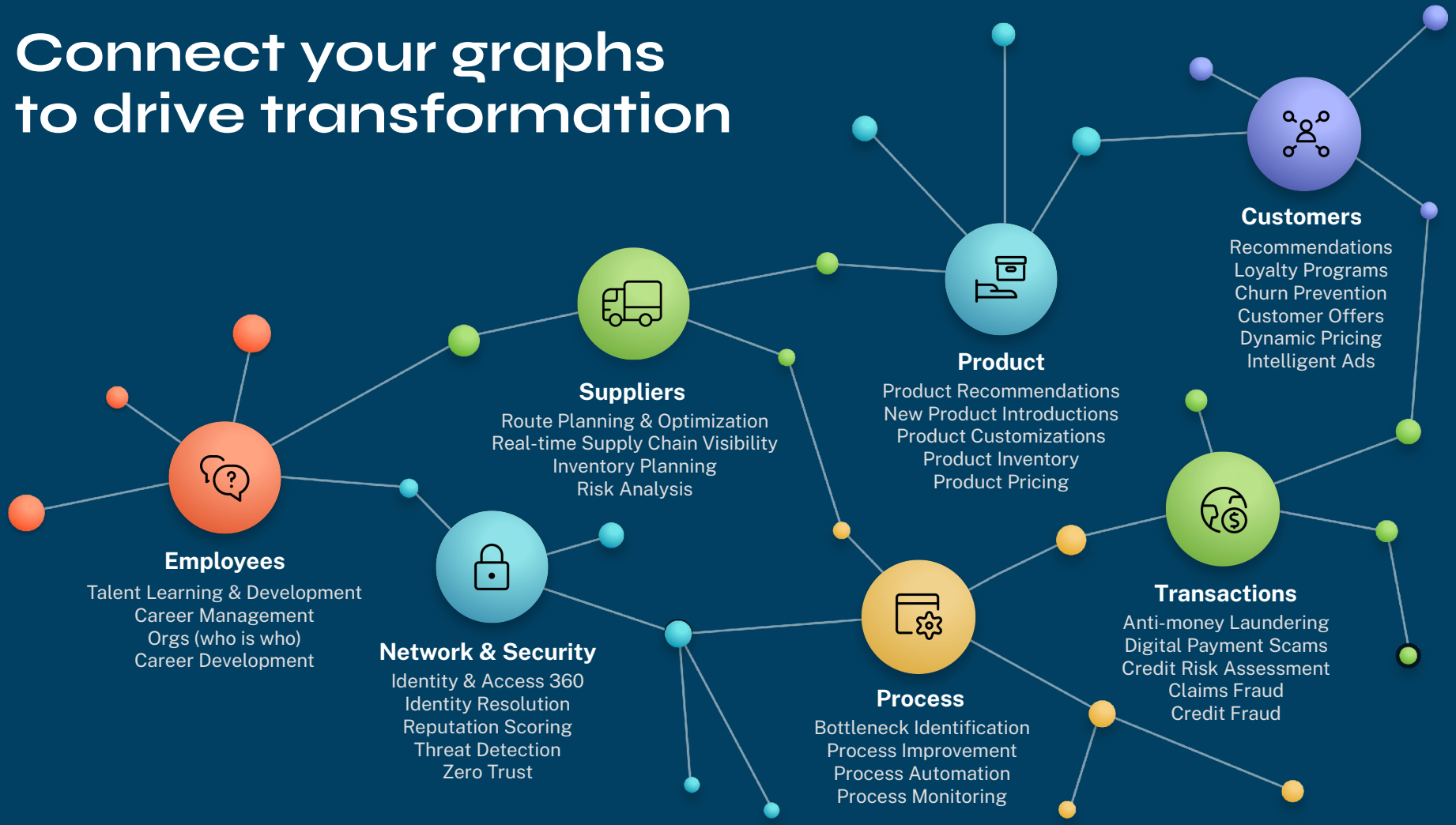
# Graphs




Graph creates a **connected** view of data that is intuitive and actionable. **Simple but powerful**

# Seven Graphs in an Enterprise

# Connect your graphs to drive transformation

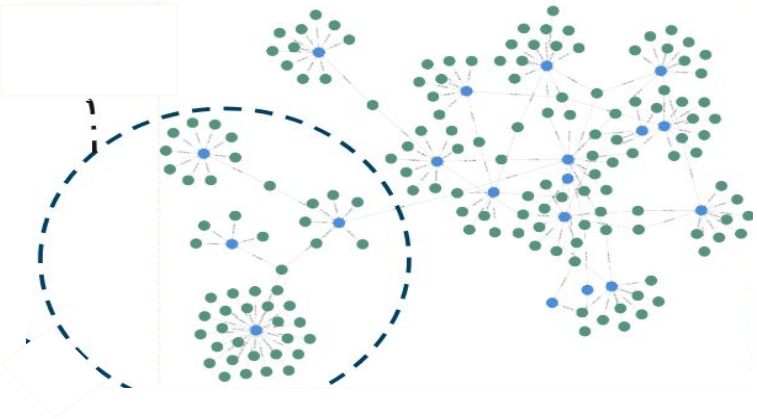


# When to use Graph

# Graph Benefits

## Innovation

Other solutions limit innovation by **constraining the datasets** available for real time processing

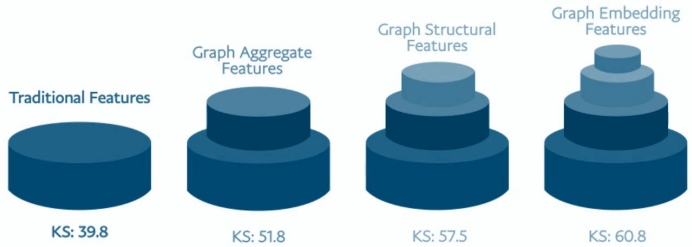


## Performance

Easier to iterate, introduce new features with less data footprint and lower TCO

## Understanding

**Fraud Analyst:** "With graph visualization I can see in 10 minutes what just took me a week to get to"




## Insights

Add **graph features** to traditional Machine Learning e.g.

- Centrality
- Community
- Similarity

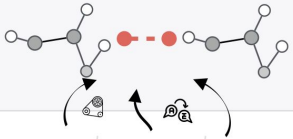
# Graph Data Analytics Patterns

- **Marshalling data around a single entity** → 360° views, deep context for GenAI
- **Tracing end-to-end flows** → Chain analyses, customer journeys, financial transactions
- **Understanding deep hierarchies** → Ownership & org structures, BOMs, product catalogs
- **Modeling dependency cascades** → Attack paths, POFs, gene regulation, isolation boundaries
- **Visualizing complex networks** → IT infra, BPM, supply chain, utility grids, social networks
- **Pathfinding & route optimizations** → logistics, shipping, least cost
- **Discovering hidden & buried relationships** → fraud detection, recommenders, communities
- **Enterprise data unification** → entity extraction, reconciliation & lineage across sources

**Anti-patterns:**

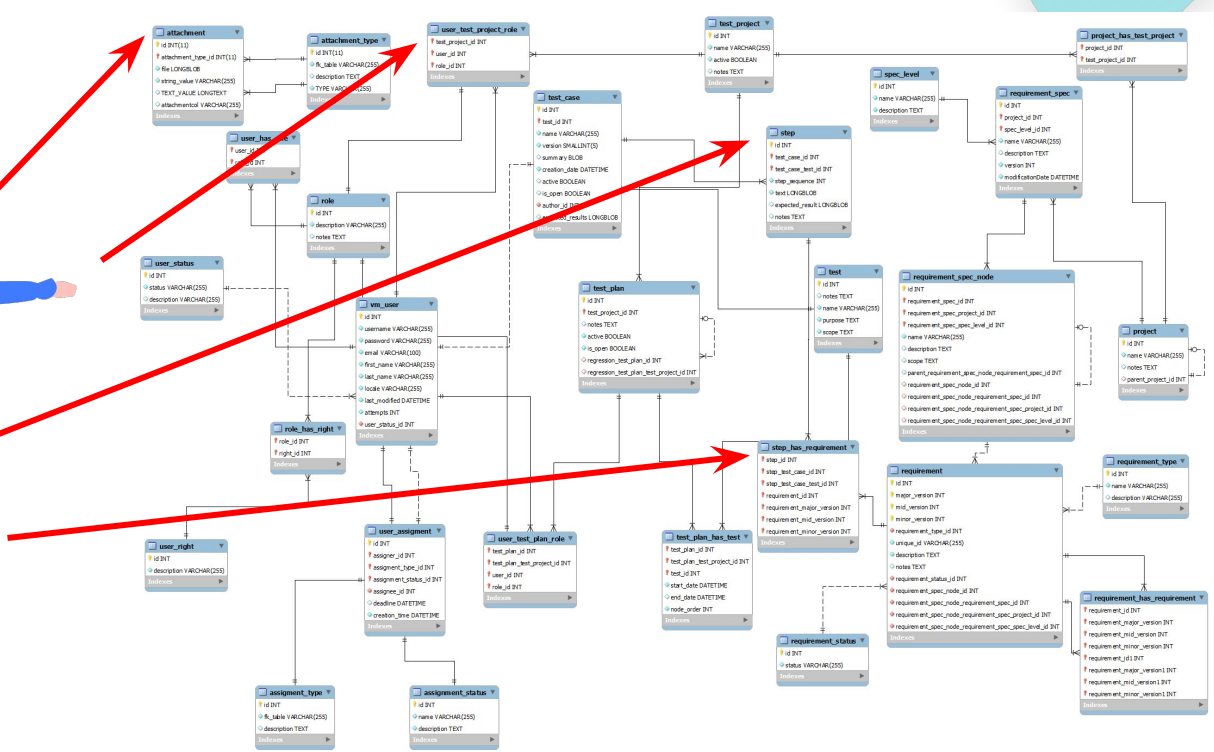
- Standard business reporting - columnar summations → *use aggregation nodes*
- Time series analysis - large volumes of high frequency events → *pre-process and extract alerts*
- Document storage - large binary files → *scrape document metadata and location pointers*

# How Neo4j Differentiates from other Databases

| | neo4j<br>Native Graph DB | Non-Native Graph DB | RDBMS |
|---|---|---|---|
| Visualization |  |  |  |
| Queries | Cypher<br>`(graphs)-[are]->(everywhere)` | Cypher/Gremlin<br>/Proprietary | SQL |
| Processing |  |  |  |
| Storage |  |  |  |

**Optimized for graph workloads**

# Finding a “Person” in a Sea of Tables




Join Orders… Join Demographics… Join Customer… Join Support…

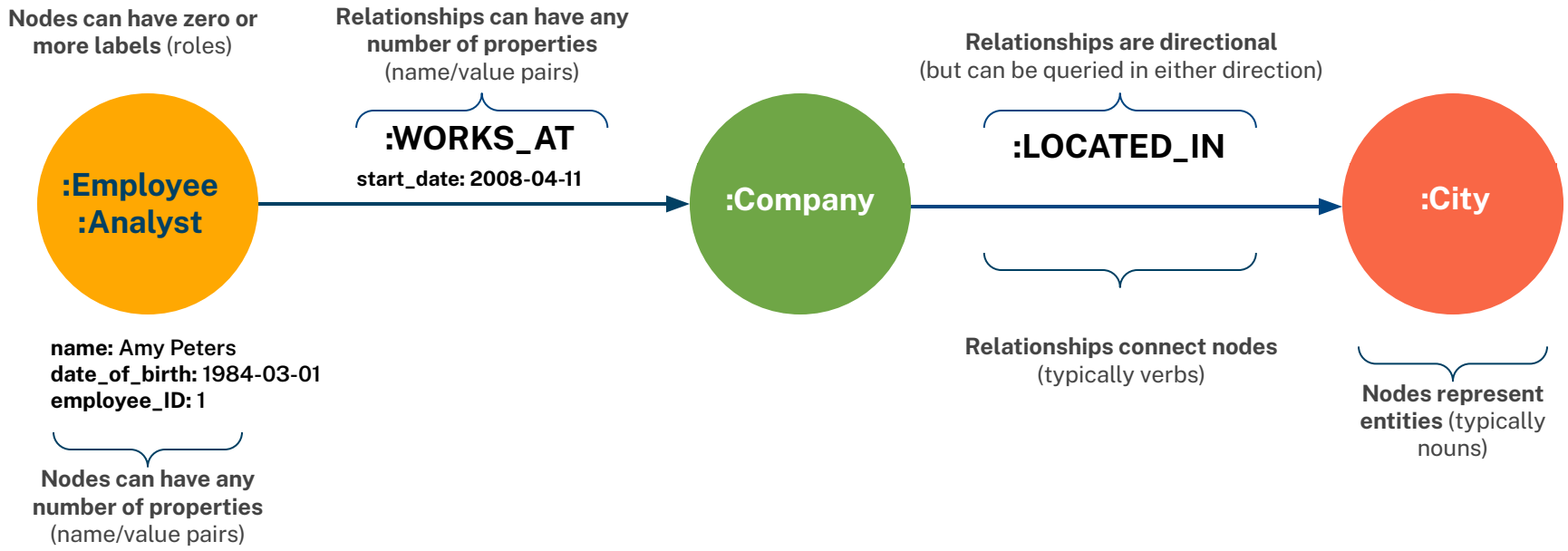
# “But I can do that in SQL”

*<postgres> Script-3

```
SELECT DISTINCT dep.deptype, dep.classid, dep.objid, cl.relkind, attr.attname,ad.adbin, ad.adsrc,
    CASE WHEN cl.relkind IS NOT NULL THEN cl.relkind || COALESCE(dep.objsubid::text, '')
        WHEN tg.oid IS NOT NULL THEN 'T'::text
        WHEN ty.oid IS NOT NULL THEN 'y'::text
        WHEN ns.oid IS NOT NULL THEN 'n'::text
        WHEN pr.oid IS NOT NULL THEN 'p'::text
        WHEN la.oid IS NOT NULL THEN 'l'::text
        WHEN rw.oid IS NOT NULL THEN 'R'::text
        WHEN co.oid IS NOT NULL THEN 'C'::text || contype
        WHEN ad.oid IS NOT NULL THEN 'A'::text
        ELSE ''
    END AS type,
    COALESCE(coc.relname, clrw.relname) AS ownertable,
    CASE WHEN cl.relname IS NOT NULL AND att.attname IS NOT NULL THEN cl.relname || '.' || att.
    ELSE COALESCE(cl.relname, co.conname, pr.proname, tg.tgname, ty.typname, la.lanname, rw.rul
    END AS refname,
    COALESCE(nsc.nspname, nso.nspname, nsp.nspname, nst.nspname, nsrw.nspname) AS nspname
FROM pg_depend dep
LEFT JOIN pg_class cl ON dep.objid=cl.oid
LEFT JOIN pg_attribute att ON dep.objid=att.attrelid AND dep.objsubid=att.attnum
LEFT JOIN pg_namespace nsc ON cl.relnamespace=nsc.oid
LEFT JOIN pg_proc pr ON dep.objid=pr.oid
LEFT JOIN pg_namespace nsp ON pr.pronamespace=nsp.oid
LEFT JOIN pg_trigger tg ON dep.objid=tg.oid
LEFT JOIN pg_type ty ON dep.objid=ty.oid
LEFT JOIN pg_namespace nst ON ty.typnamespace=nst.oid
LEFT JOIN pg_constraint co ON dep.objid=co.oid
LEFT JOIN pg_class coc ON co.conrelid=coc.oid
LEFT JOIN pg_namespace nso ON co.connamespace=nso.oid
LEFT JOIN pg_rewrite rw ON dep.objid=rw.oid
LEFT JOIN pg_class clrw ON clrw.oid=rw.ev_class
LEFT JOIN pg_namespace nsrw ON clrw.relnamespace=nsrw.oid
LEFT JOIN pg_language la ON dep.objid=la.oid
```

# The Property Graph: Simply Powerful

A more human way to create data relationships

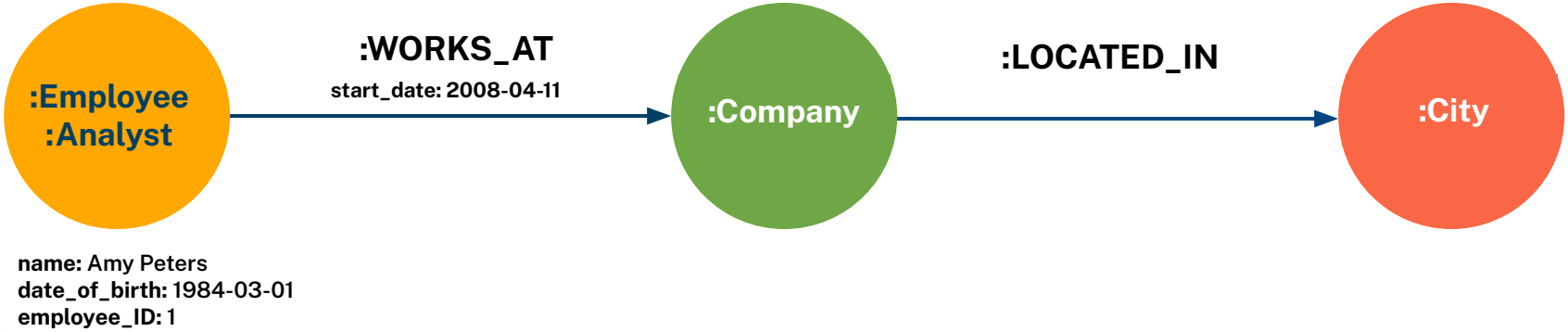


# Cypher/GQL Query Language

A pattern matching query language made for graphs

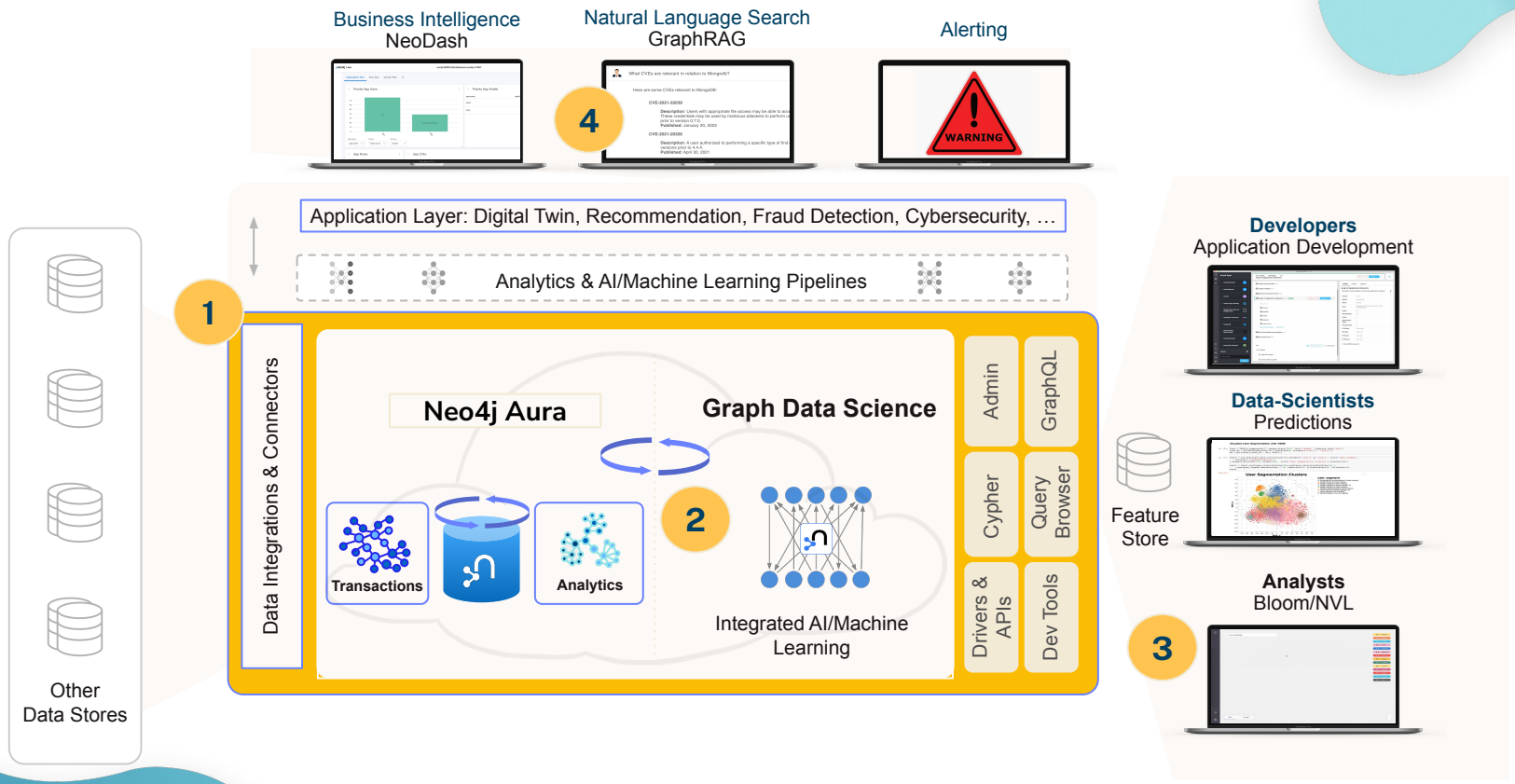
Declarative & Expressive *(what to find, not how to find it)*

```
MATCH (:Employee {name:'Amy Peters'})-[worked:WORKS_AT]->(comp:Company)-[:LOCATED_IN]->(city:City)
RETURN comp.name, worked.start_date, city.name
```



# Neo4j Graph Platform

# Neo4j - Complete Graph Platform



# Graph Data Science

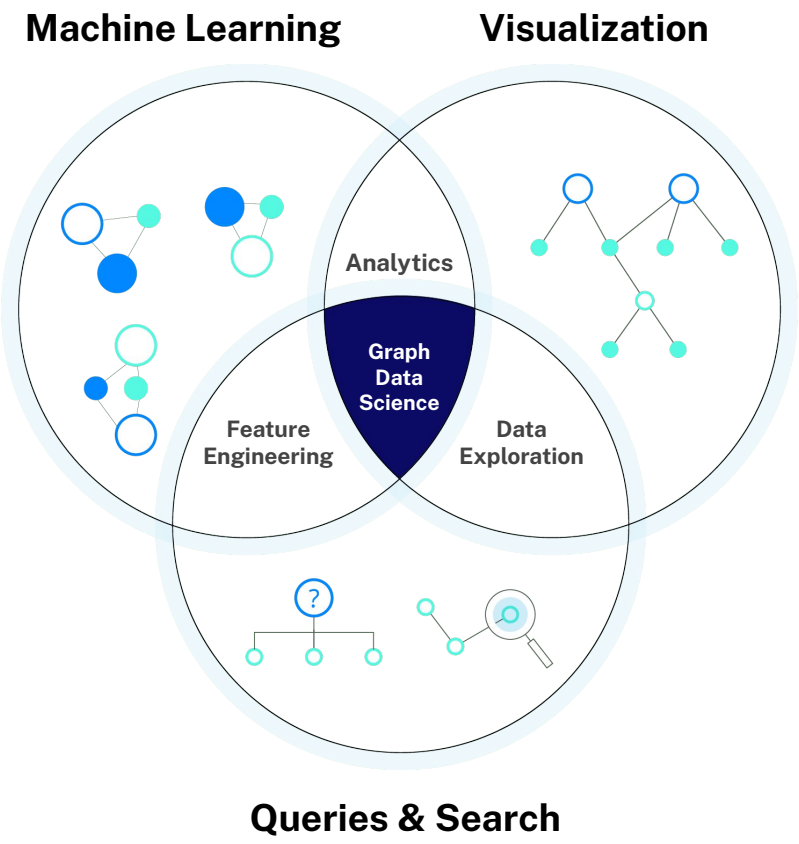
# Neo4j Graph Data Science And Machine Learning

## WHAT?

Use context and relationships between data points to enhance analytics and ML

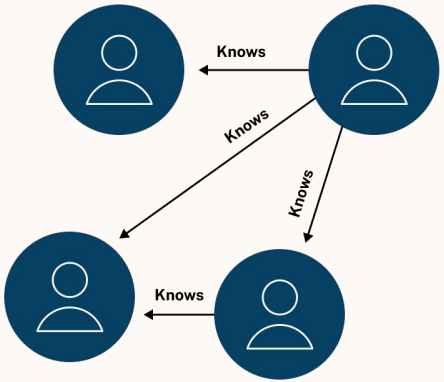
## WHY?

Faster, simpler, more accurate predictions and models whenever context matter.

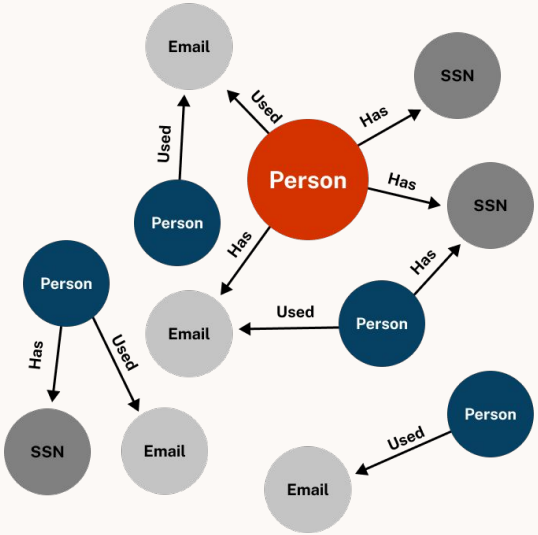


# Graph Data Science
# Make Sense Of Data Relationships

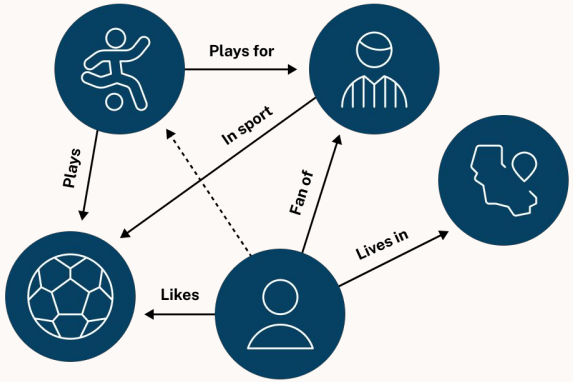
## What's important?




## What's unusual?




## What's next?



# Graph Data Science Maturity Model

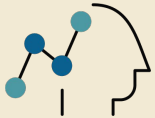
## Knowledge Graphs

Find the patterns you're looking for in connected data

## Graph Algorithms

Identify associations, anomalies, and trends using unsupervised machine learning

## Graph Native ML

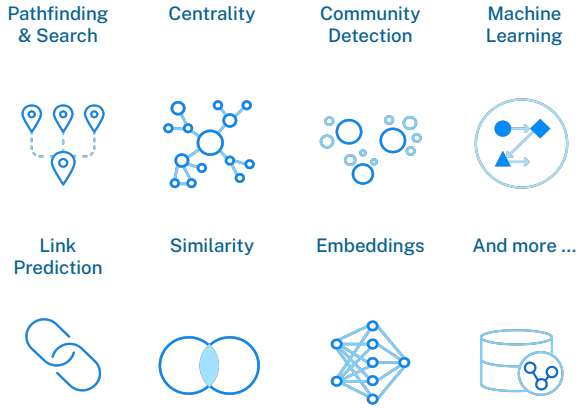


Learn features in your graph that you don't even know are important yet

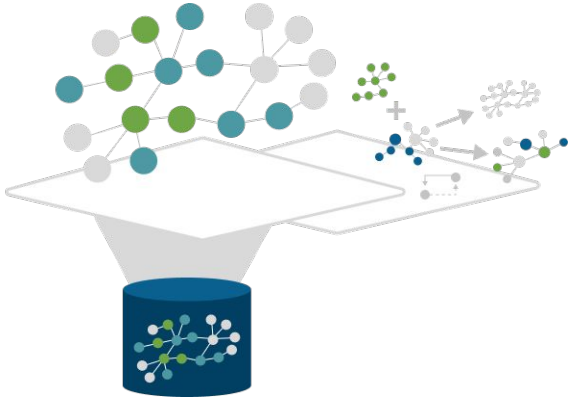
# Improve Models And Answer Big Questions

## The Largest Catalog of Graph Algorithms



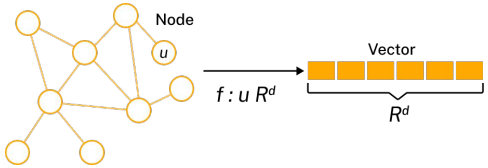

Over 65 pretuned, parallelized algorithms.

## Native Graph Catalog and Analytics Workspace



Iterate fast with different data sets, models, and version trained models.

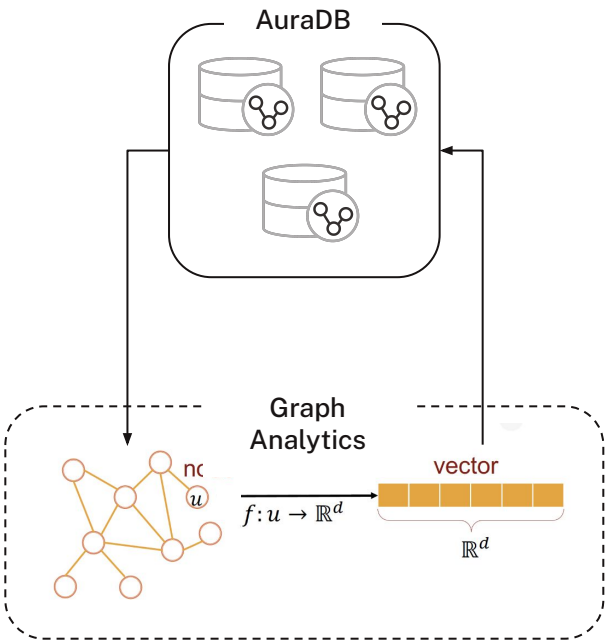
## Graph Embeddings for Machine Learning




Bring the context of your connected data into a format that other pipelines can ingest.

Scale out in Graph Analytics

# Aura Graph Analytics



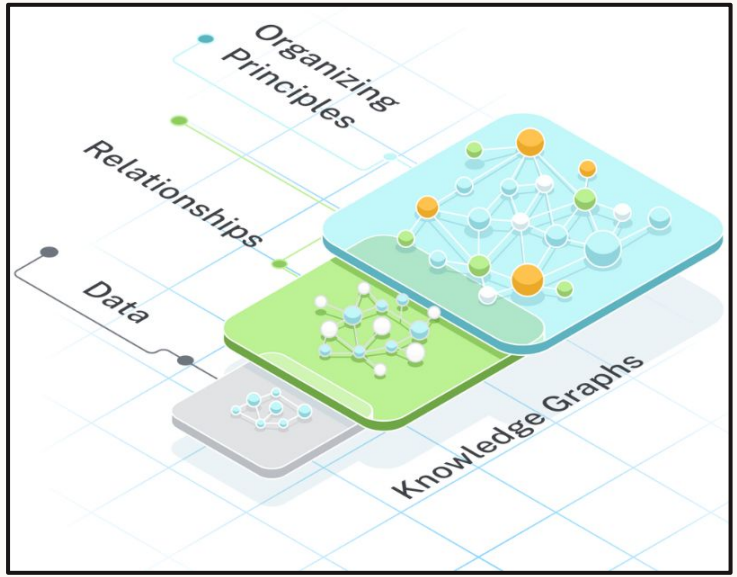

- Connect to a Neo4j DB anywhere
- Create a Session (separate instance)
- Project data with a Cypher query from DB
- Run Graph Analytics asynchronously
- Write back output to DB
- Shut down Session

# Unlocking the GenAI Opportunity

# What Value Does GraphRAG Provide?

## Graph Based Use Cases

- Data contains rich relationships
- Data is structured or semi-structured
- Advanced search required
- Need higher accuracy results




***GraphRAG value is grounded in high accuracy and user trust***

# Why Ground your LLM?

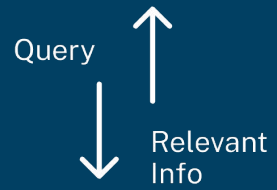
Find me all the three bedrooms near me


Nearest bedrooms are..

# Why Ground your LLM?

What are the 3 bedrooms closest to me that are affordable?

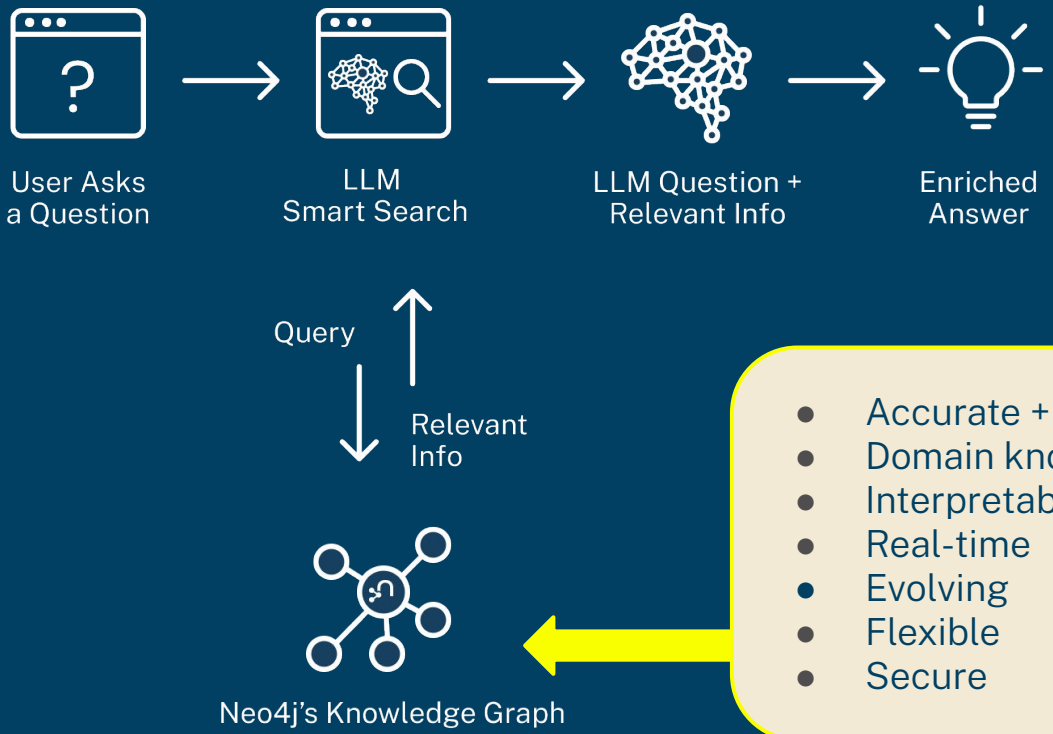

50 listings that match the query




Neo4j's Knowledge Graph

# Leveraging the Knowledge Graph

Find all 3-bedroom properties within 2km of top-rated schools and hospitals with cardiology speciality




Here are 3-bedroom properties near ***Burlingame*** *Elementary School* and *Mills-Peninsula Medical Center* .....

- Accurate + Contextual
- Domain knowledge
- Interpretable
- Real-time
- Evolving
- Flexible
- Secure

**with the caveat that you have this data

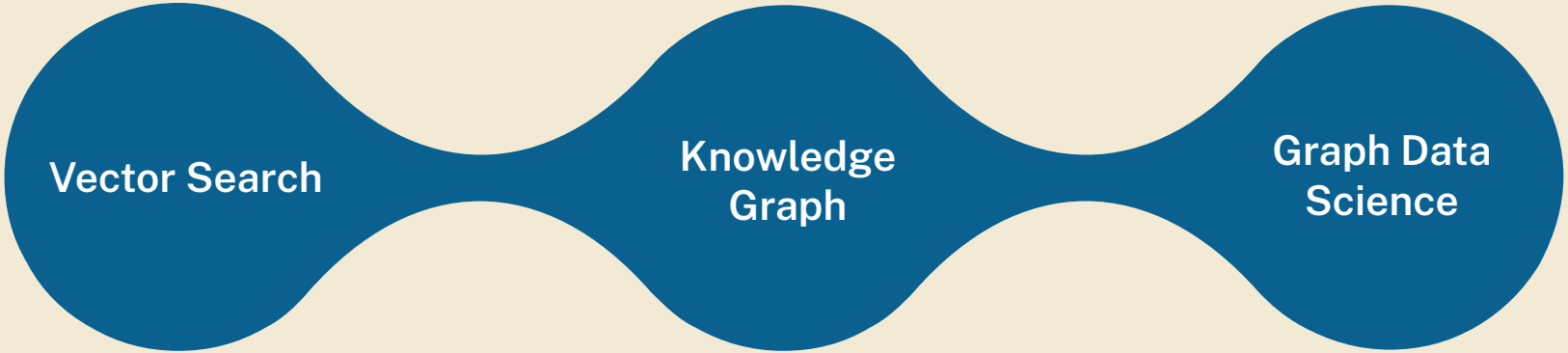


neo4j

# Demo

# RAG with Neo4j - GraphRAG

Unify vector search, knowledge graph and data science capabilities to improve RAG quality and effectiveness




Find similar documents and content

Identify entities associated to content and patterns in connected data

Improve GenAI inferences and insights. Discover new relationships and entities

# GraphRAG Advantage

Semantic Similarity alone is insufficient for rich enterprise reasoning

**Leverage ALL data:** capture relationships from unstructured, structured, and semi-structured data

1

2

**Capture critical context:** Highlighting connections between facts and documents enables answers to multi-step, domain-specific, questions

**Expanded Similarity for Relevance:** Semantic similarity is valuable, but doesn't address every question. Graph traversals help capture business logic, and graph embeddings encode knowledge from new perspectives.

3

4

**Explainability:** Neo4j provides traceable and auditable data through a visual interface.

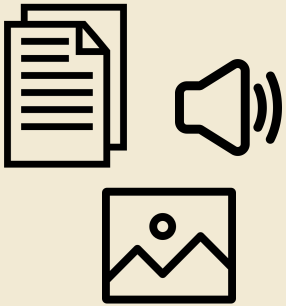
The Limitations of Text Embeddings in RAG Applications

# Data Sources for RAG

## Unstructured

**Text blobs, Images, Audio, etc.**



Research Papers

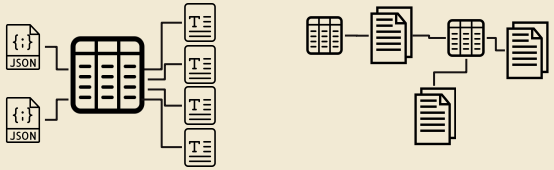
Internal Documentation

Most Use Cases

## Mixed

**Linked Structured & Unstructured**

Table with unstructured fields | Documents linked to tables | Interconnected metadata

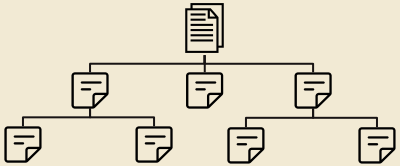



Operational Stores

Security & Project Management Tools

Data Lakes

**Documents with Lexical Structure**



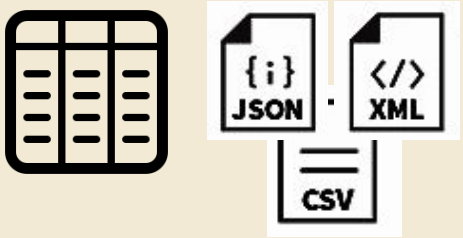
User Manuals

Legal Contracts

Product Catalogs

## Structured

**Tables & flat files (CSV, JSON)**




CRM/ Historic Customer Data

Supply Chain & Inventory Data

Financial Transactions

# GraphRAG: Higher Accuracy

GraphRAG delivers up to 3x higher accuracy than traditional RAG, with better multi-hop reasoning for context- rich AI applications.

| | GraphRAG Performance | RAG Performance |
|---|---|---|
| **Lettria Analysis[1]** | 81.67% | 57.50% |
| **Writer Knowledge Graph[2] (RobustQA Benchmark)** | 86.31% | 32.74%–75.89% |
| **RAG vs. GraphRAG: Multi-hop Question Answering[3]** | 77% | 66% |
| **GenUI Experiments (MultiHop-RAG Dataset)[4]** | Successfully answered complex, multi-step queries | Struggled integrating data from multiple sources |

1) https://writer.com/engineering/rag-benchmark/ 2) https://www.lettria.com/blogpost/vectorrag-vs-graphrag-a-convincing-comparison
3) https://arxiv.org/abs/2502.11371 4) https://www.genui.com/resources/graphrag-vs.-traditional-rag-solving-multi-hop-reasoning-in-llms

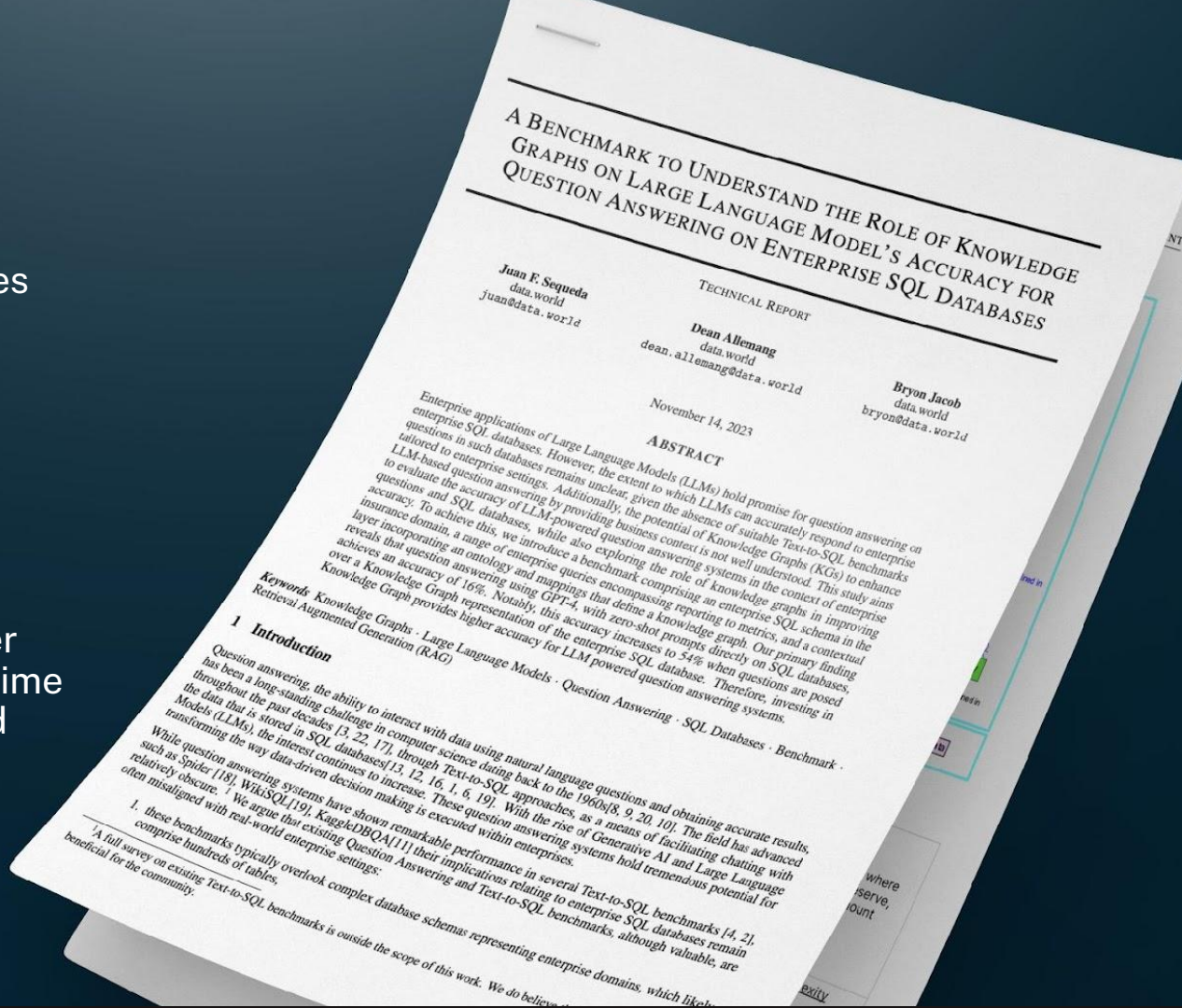
# Benefits

A Knowledge Graph improves the accuracy of LLM responses by 54.2%, an average of 3x

## LinkedIn

Reduced the median customer support per-issue resolution time by 28.6% over 6 month period

https://arxiv.org/pdf/2404.17723



A BENCHMARK TO UNDERSTAND THE ROLE OF KNOWLEDGE GRAPHS ON LARGE LANGUAGE MODEL'S ACCURACY FOR QUESTION ANSWERING ON ENTERPRISE SQL DATABASES

TECHNICAL REPORT

**Juan F. Sequeda**
data.world
juan@data.world

**Dean Allemang**
data.world
dean.allemang@data.world

**Bryon Jacob**
data.world
bryon@data.world

November 14, 2023

ABSTRACT

Enterprise applications of Large Language Models (LLMs) hold promise for question answering on enterprise SQL databases. However, the extent to which LLMs can accurately respond to enterprise questions in such databases remains unclear, given the absence of suitable Text-to-SQL benchmarks tailored to enterprise settings. Additionally, the potential of Knowledge Graphs (KGs) to enhance LLM-based question answering by providing business context is not well understood. This study aims to evaluate the accuracy of LLM-powered question answering systems in the context of enterprise questions and SQL databases, while also exploring the role of knowledge graphs in improving accuracy. To achieve this, we introduce a benchmark comprising an enterprise SQL schema in the insurance domain, a range of enterprise queries encompassing reporting to metrics, and a contextual layer incorporating an ontology and mappings that define a knowledge graph. Our primary finding reveals that question answering using GPT-4, with zero-shot prompts directly on SQL databases, achieves an accuracy of 16%. Notably, this accuracy increases to 54% when questions are posed over a Knowledge Graph representation of the enterprise SQL database. Therefore, investing in Knowledge Graph provides higher accuracy for LLM powered question answering systems.

***Keywords*** *Knowledge Graphs · Large Language Models · Question Answering · SQL Databases · Benchmark · Retrieval Augmented Generation (RAG)*

## 1 Introduction

Question answering, the ability to interact with data using natural language questions and obtaining accurate results, has been a long-standing challenge in computer science dating back to the 1960s[8, 9, 20, 10]. The field has advanced throughout the past decades [3, 22, 17], through Text-to-SQL approaches, as a means of facilitating chatting with the data that is stored in SQL databases[13, 12, 16, 1, 6, 19]. With the rise of Generative AI and Large Language Models (LLMs), the interest continues to increase. These question answering systems hold tremendous potential for transforming the way data-driven decision making is executed within enterprises.

While question answering systems have shown remarkable performance in several Text-to-SQL benchmarks [4, 2], such as Spider [18], WikiSQL[19], KaggleDBQA[11] their implications relating to enterprise SQL databases remain relatively obscure. [1] We argue that existing Question Answering and Text-to-SQL benchmarks, although valuable, are often misaligned with real-world enterprise settings:

1. these benchmarks typically overlook complex database schemas representing enterprise domains, which likely comprise hundreds of tables.

[1] A full survey on existing Text-to-SQL benchmarks is outside the scope of this work. We do believe ... beneficial for the community.

# Support Ticket Knowledge Graph

LinkedIn Cut Down Ticket Resolution Time from 40 hrs to 15 hrs

## Parse Tickets →

Extract entities from tickets to create knowledge graph with unstructured and structured data

## Link Tickets →

Link tickets on

- Context
- shared dependencies
- references

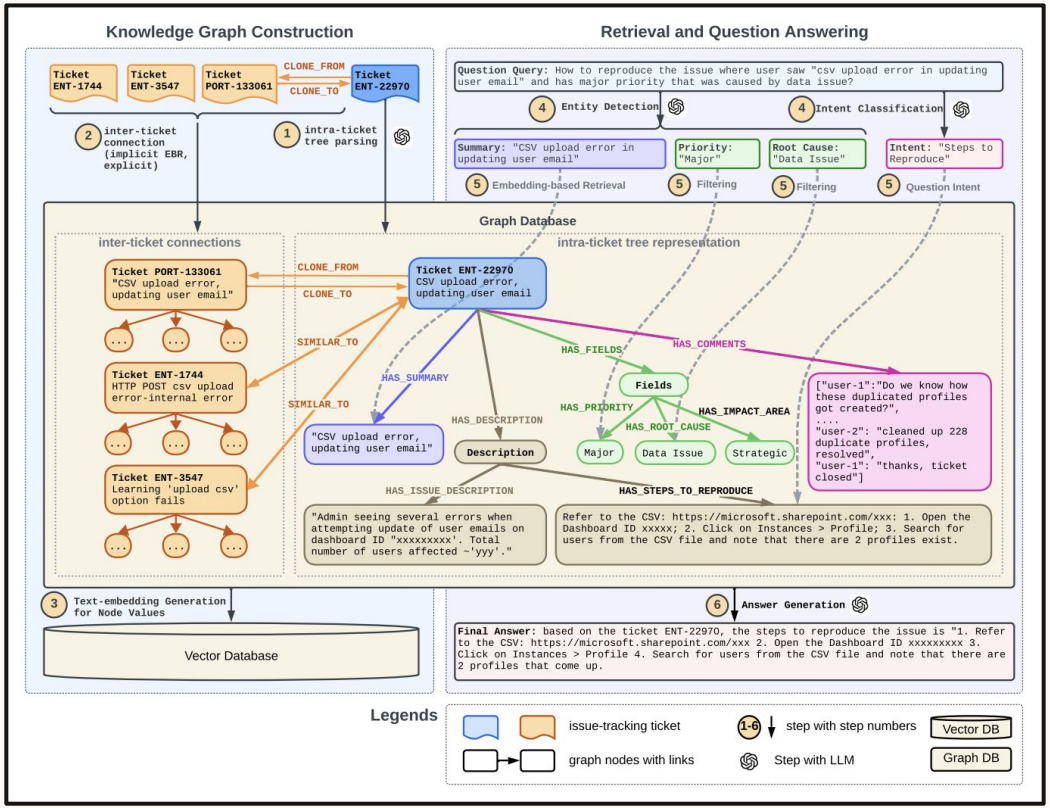
## Traverse Graph →

Semantic similarity
+
graph traversals

## Return Insights

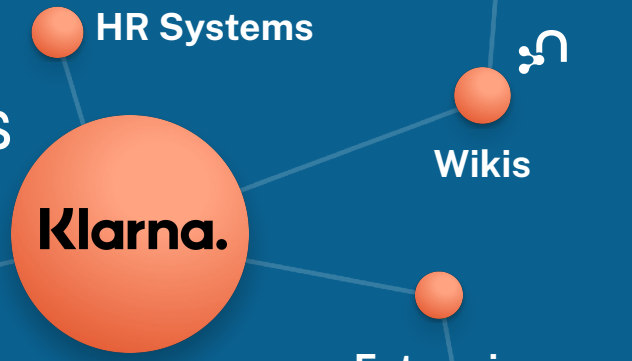
Efficient and accurate answers

# Support Ticket Knowledge Graph



## FROM SEARCHING TO KNOWING

# Klarna transforms knowledge access with GraphRAG



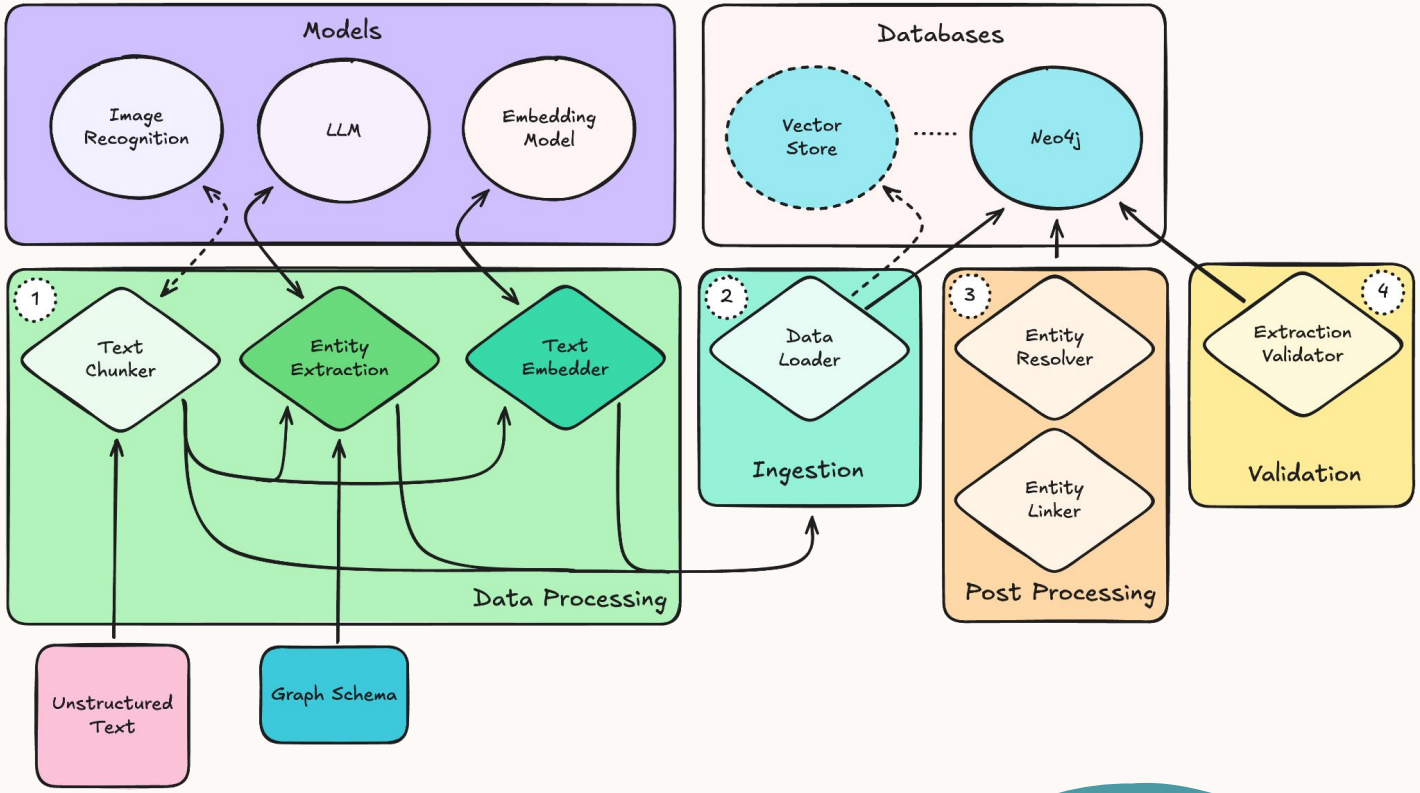

**250K** Employee questions answered in first year

**2,000** Daily queries processed

**85%** Employee adoption

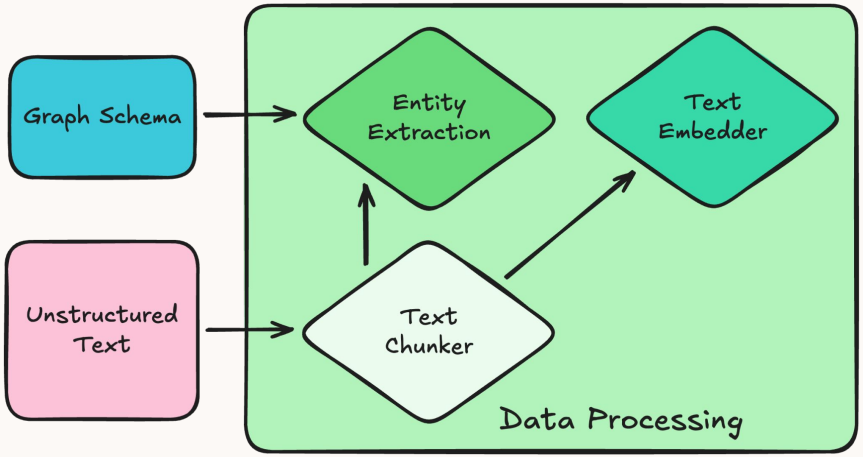
# Knowledge Graph Generation

# Knowledge Graph Generation



# Knowledge Graph Generation | Data Processing

| | Text Chunking | Entity Extraction | Text Embedding |
|---|---|---|---|
| GraphRAG Package | X | X | X |
| APOC | | | X |
| LLM Graph Builder | X | X | X |
| Unstructured | X | X | |
| LangChain | X | X | X |
| LlamaIndex | X | X | X |
| LangChain4J | X | X | X |
| HayStack | X | X | X |
| Semantic Kernel | | | X |



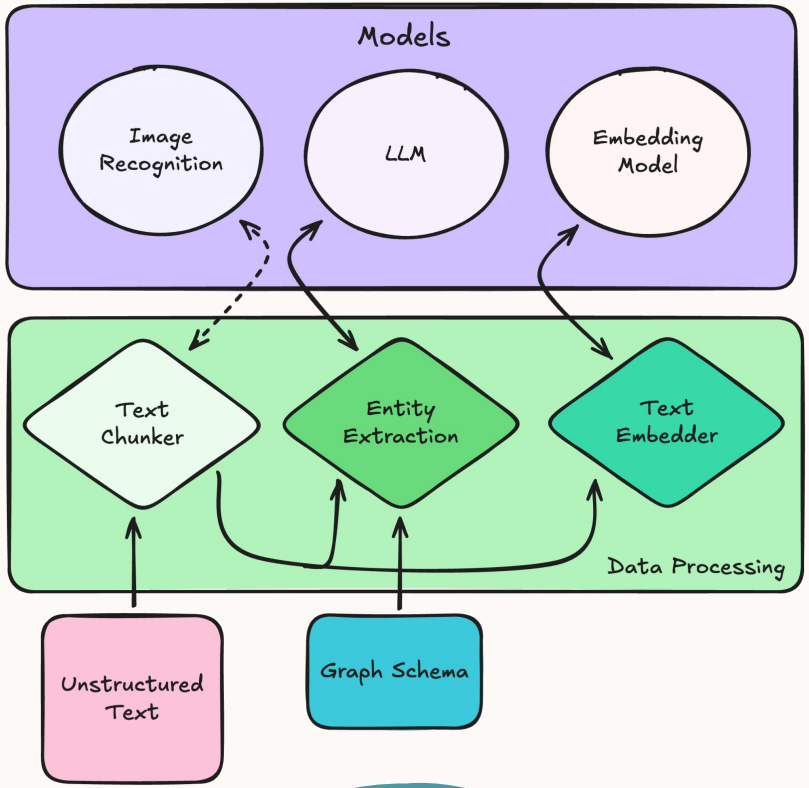
# Knowledge Graph Generation | Data Processing

Unstructured
- Text Chunking
- Image Recognition

Pydantic
- Graph Schema Definition
- Extraction Validation

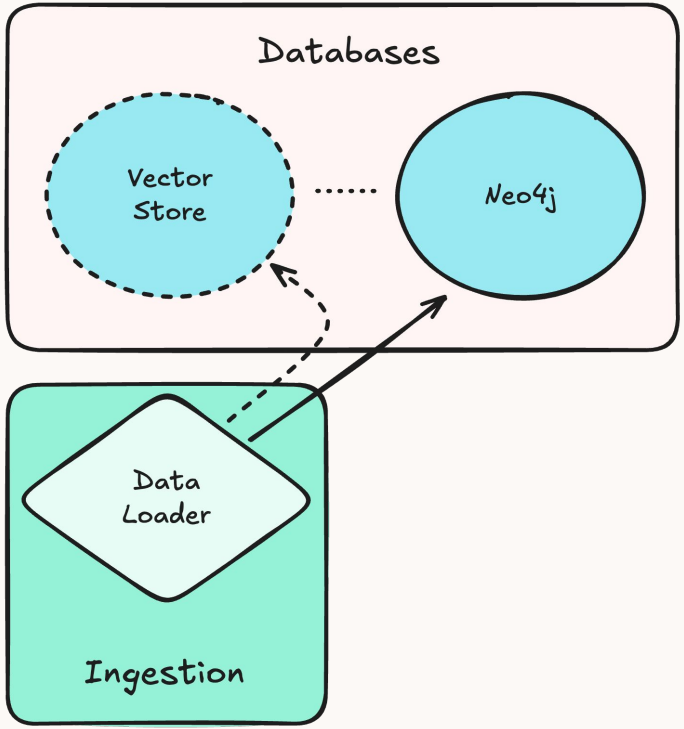
**Instructor**
- LLM requests
- Response parsing
- Retry handling

OpenAI
- LLM Provider
- Embedding Model Provider



# Knowledge Graph Generation | Ingestion

| | Neo4j | Vector Store |
|---|---|---|
| GraphRAG Package | X | X |
| LLM Graph Builder | X | |
| Unstructured | X | X |
| LangChain | X | X |
| LlamaIndex | X | X |
| LangChain4J | X | X |
| HayStack | X | X |



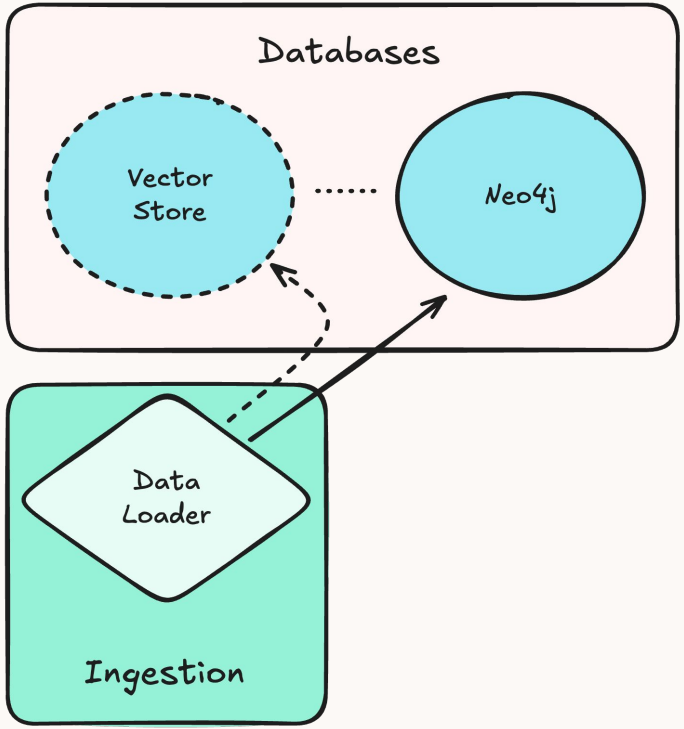
# Knowledge Graph Generation | Ingestion

Neo4j

- Text
- Text Metadata
- Text Embeddings
- Entities
- Entity Relationships

Vector Stores ( Optional )

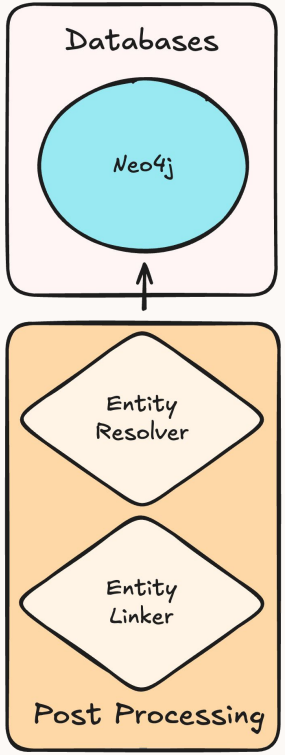
- Text
- Text Metadata
- Text Embeddings



# Knowledge Graph Generation | Post Processing

Methods

- Cypher Queries
- GDS Algorithms
- LLMs
- Named Entity Linking Models



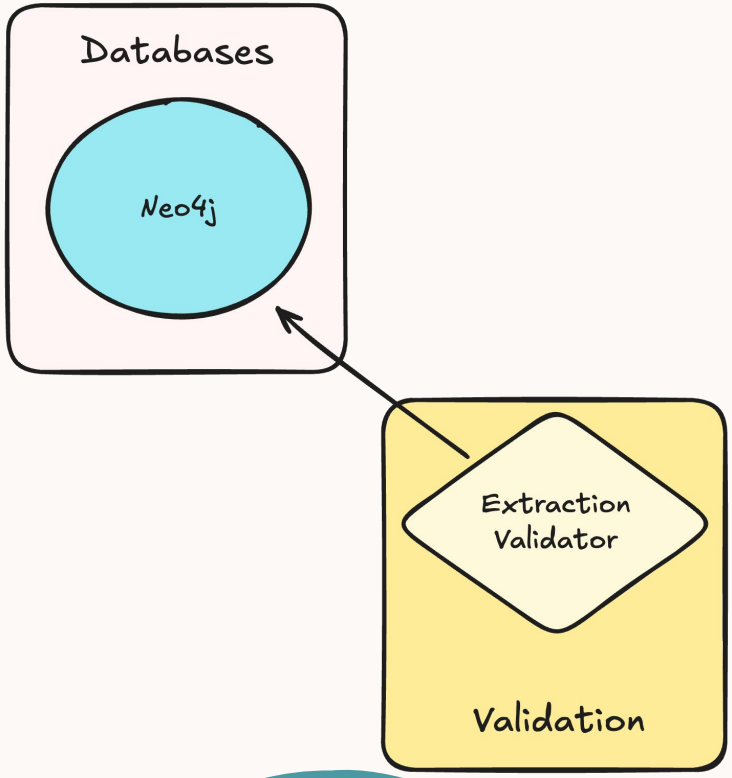
# Knowledge Graph Generation | Validation

## Methods

- Cypher Queries
- GDS Algorithms
- LLMs
- Named Entity Linking Models

## Services

- LangSmith
- Arize
- Patronus AI
- Braintrust



# 5 min break

# Neo4j + MCP

Simplifying your Neo4j developer experience

# MCP Overview

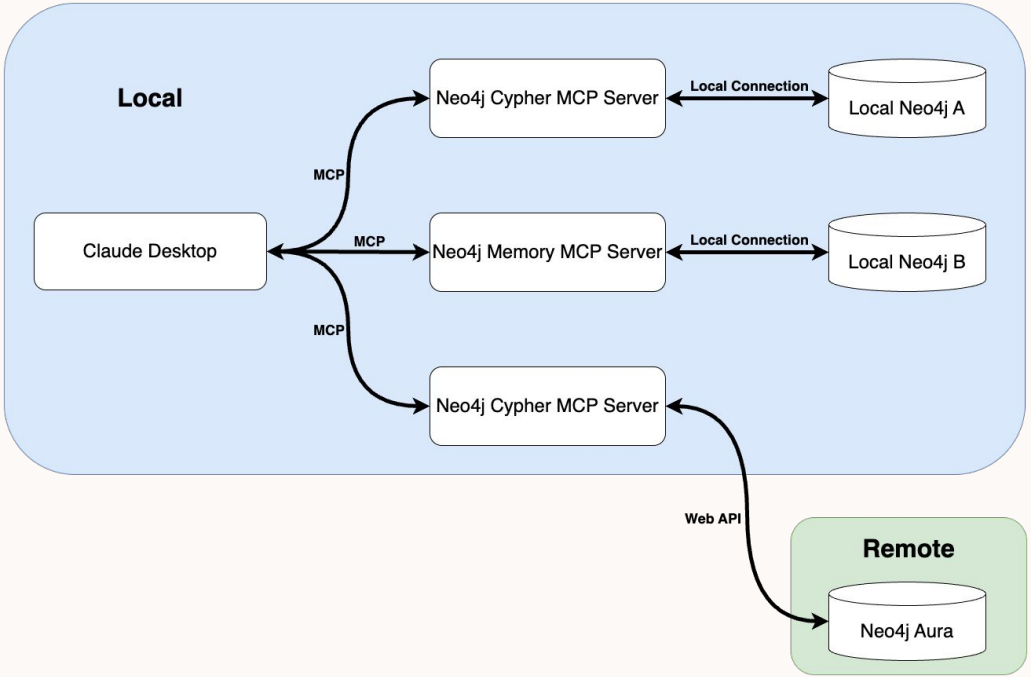
## Standardizing how applications provide context to LLMs

MCP servers provide

- Tools → *Execute Actions*
- Resources → *Static Content*
- Prompts → *Guide Users*

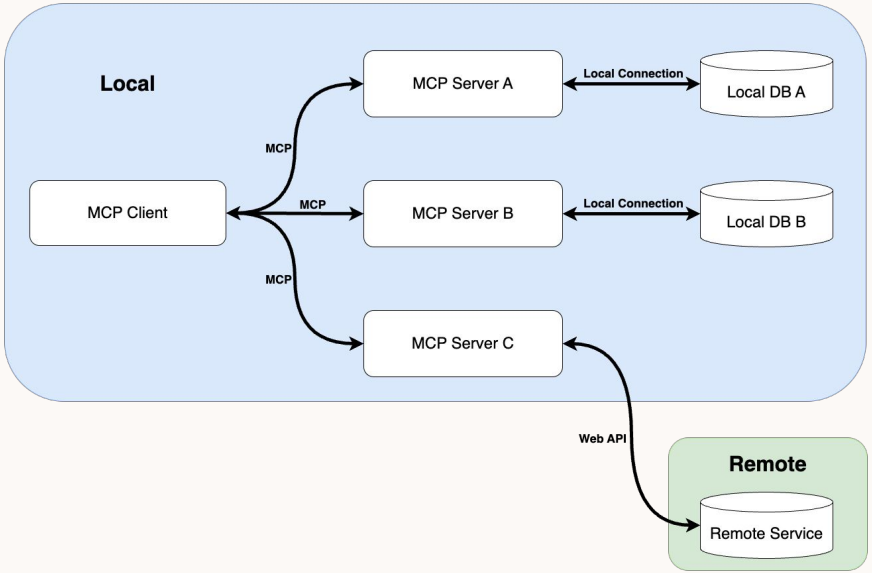
MCP Clients include

- Cursor IDE
- Claude Desktop
- And many others...

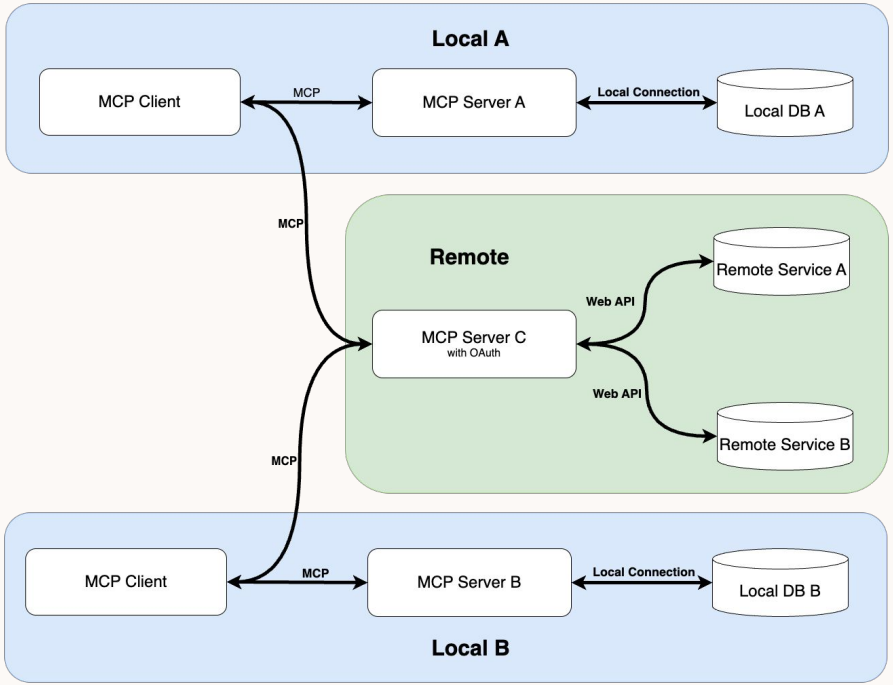


# MCP Hosting

## Local




## Remote

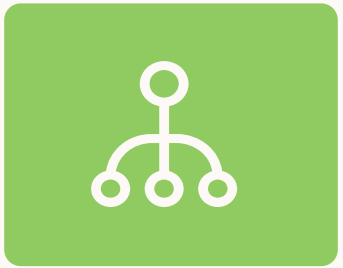


# MCP Servers (from Neo4j Labs)

## Querying

No-Code analysis of your Neo4j data

Documentation



## Data Modeling

Quickly create and augment your graph data models

Documentation



## Agentic Memory

Generate and query knowledge graphs from agent interactions

Documentation

## DB Management

Automate the management of your Neo4j Aura instances

Documentation

Official Neo4j MCP server in Beta! - https://github.com/neo4j/mcp

# What is AgenticRAG?

Agentic RAG is an evolution of RAG that incorporates **autonomous agents to enhance retrieval, reasoning, and generation capabilities.**

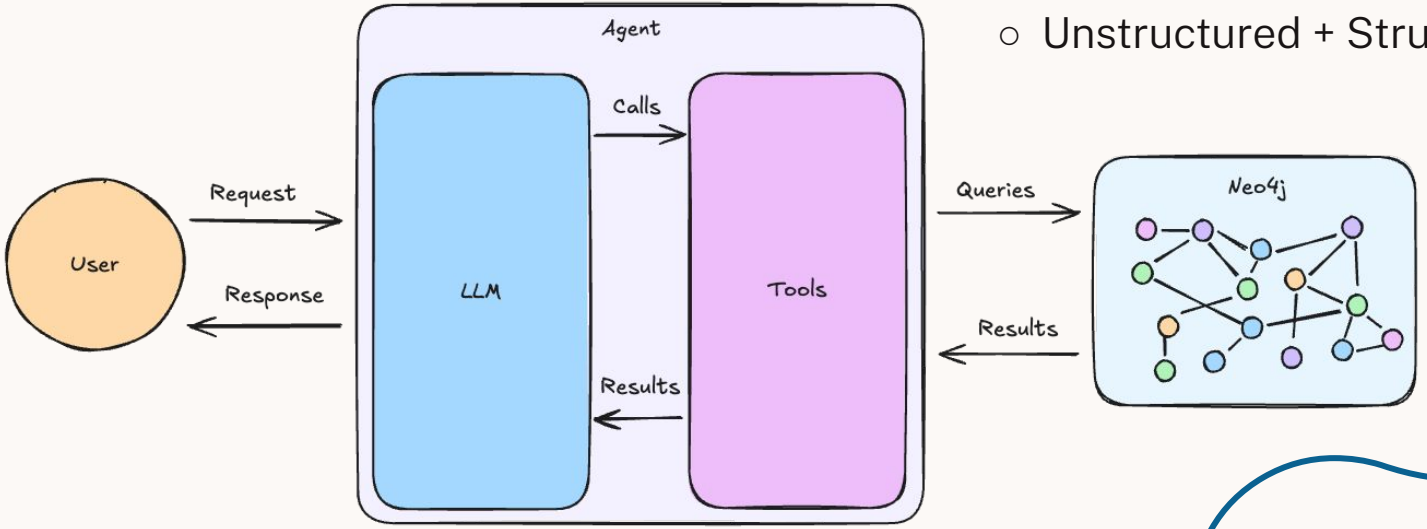
In Agentic RAG, the retrieval path **involves adaptive, goal-driven agents that dynamically explore, refine, and structure information retrieval.**

# GraphRAG with Agents

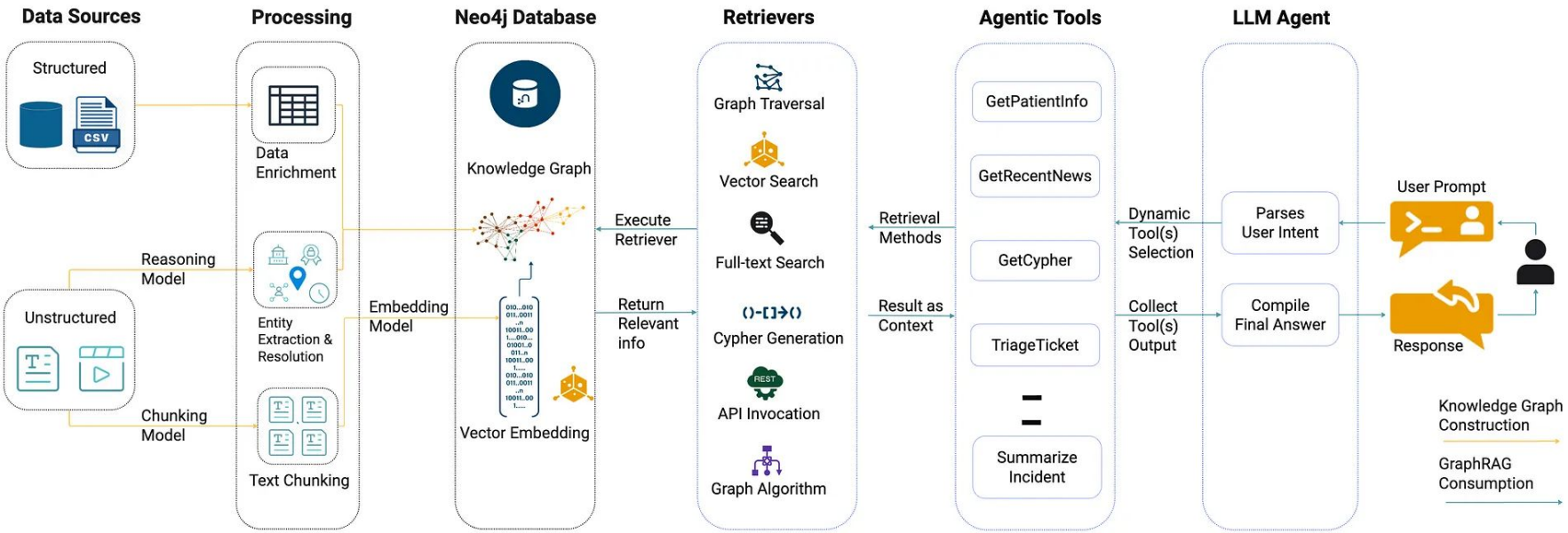
Context retrieval that implements graph traversals

- Complex filtering
- Improved explainability

- Integrate disparate sources
- Nuanced responses
  - Unstructured + Structured Data



# Agentic Tools



# Enterprise Agentic Memory

Zep AI's GraphRAG memory improves accuracy by 18% and reduce latency by 90%

Features

- Manage temporal changes in entities
- Resolve incoming data with existing knowledge graph
- Combine semantic search, BM25 and graph traversal
  - P95 latency of 300ms

# Wrap Up

Next Steps / Q&A

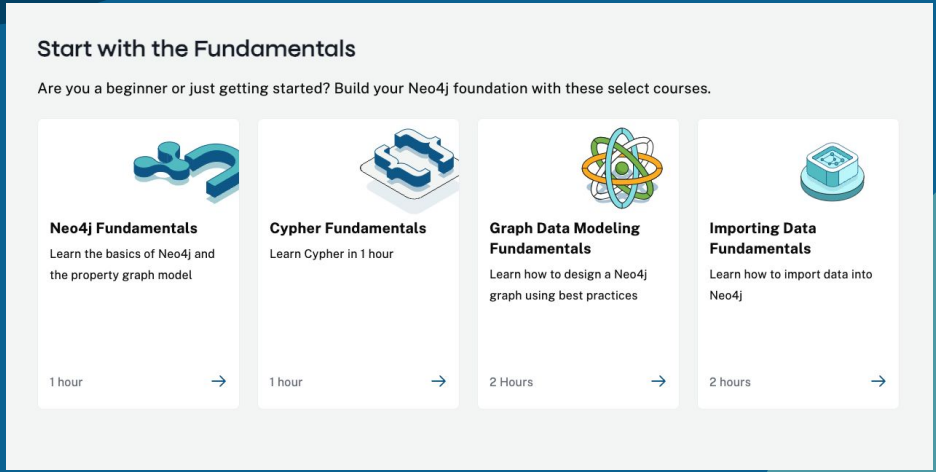
# Questions?

## Resources and Training:

- Neo4j Aura
- Graph Academy
- Example GenAI Projects

## Next Steps:

- Use case ideation through pilots
- Graphathon/Hands-On workshop
- Future Sessions:



# Neo4j Resources

**Cypher Style Guide**

- The cypher style guide provide recommendations for building clean, easy to read Cypher queries
- Link to Cypher style guide

**Cypher Cheat Sheet**

- This page gives quick examples of how to write different queries within Cypher
- Link to Cypher cheat sheet

**Neo4j Cypher Manual**

- The Cypher manual can be used to get more information about Cypher commands
- Link to cypher manual

APOC Documentation

- APOC is a great plugin to level up your cypher
- This documentation outlines different commands one could use
- Link to APOC documentation

Neo4j Graph Data Science Documentation

- Neo4j Graph Data Science documentation is a great reference to see which algorithms to use
- Show how to use different algorithms
- Link to Graph Data Science documentation

Neo4j Driver Manual

- The driver manual provides the official drivers that are supported by Neo4j
- Link to Neo4j driver manual

**GraphGists**

- GraphGists has many different use cases and examples for specific industries
- Link to GraphGists

**Arrows App**

- The Arrows app allows one to design a graph without using Cypher
- Link to Arrows app

**Neo4j Sandbox**

- The Neo4j sandbox provides a quick deployment of a Neo4j server
- It does not require a download
- Comes with example projects
- Link to Neo4j Sandbox