# Neo4j Live: Entity Architecture for Efficient RAG on Graphs

Neo4j
57.5K subscribers

Subscribe

120 | Share | Ask | Download | ...

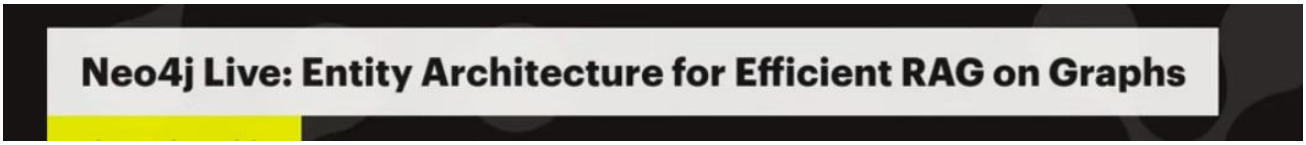
**3,184 views Streamed live on Jan 22, 2025** #entity #genai #graphrag

Join us live as we dive into the Entity Architecture for efficient Retrieval-Augmented Generation (RAG) on Knowledge Graphs. The architecture organizes RAG workflows into three distinct layers: an input layer for data ingestion, a middle layer for knowledge graph representation and reasoning, and an output layer for generating contextually accurate AI-driven results.
Learn how this innovative approach leverages fixed entities to enhance data retrieval, improve contextual understanding and boost AI performance.

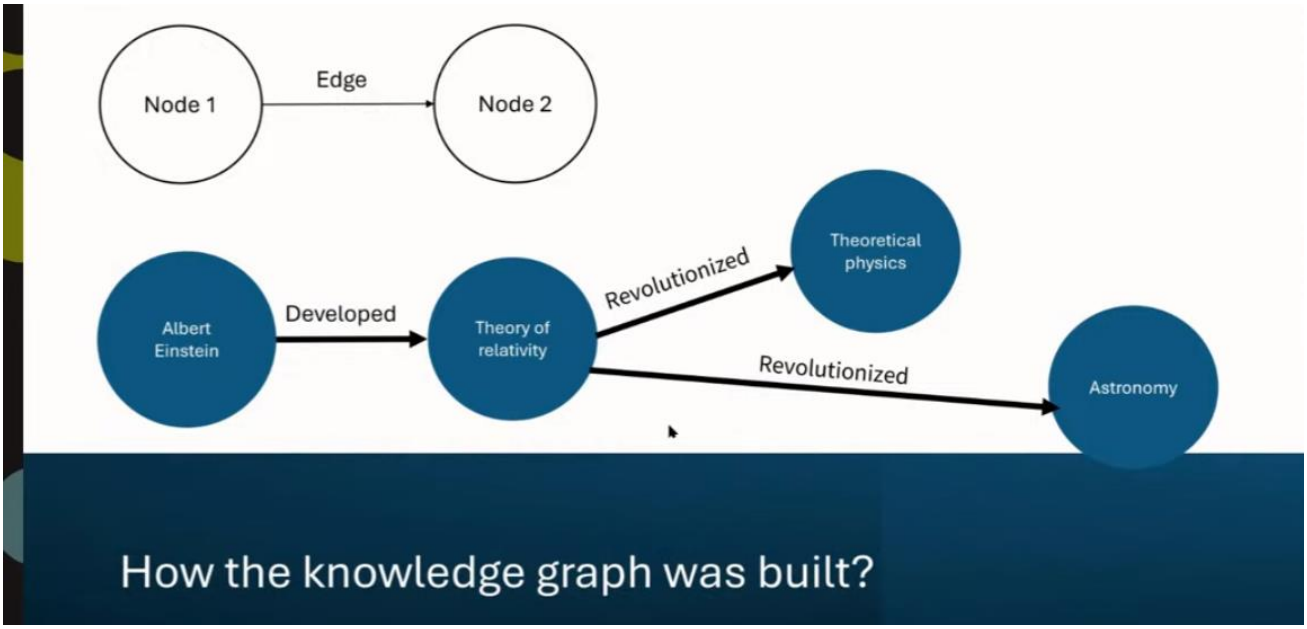
Guest: Irina Adamchic, PhD

From Local to Global: https://bit.ly/40tc4Ov
Blog: / three-layer-fixed-entity-architecture-for-...

0:00 - Welcome and introduction
0:32 - Overview of today's topic: Entity Architecture for Efficient RAG on Graphs
1:37 - Irina Adamchic - Introduction
3:37 - Discovering graph solutions for RAG use cases
6:05 - How GenAI projects evolved with graph technology
9:57 - Fixed Entity Architecture: Origin and challenges it solves
16:53 - Building the ontology layer: A practical approach
24:43 - Three-layer architecture and how it enhances scalability
31:04 - Comparing Microsoft GraphRAG vs. Fixed Entity Architecture
46:15 - Closing remarks, upcoming events, and resources







# Microsoft Approach to GraphRAG

**Research Paper**

"From Local to Global: A Graph RAG Approach to Query-Focused Summarization", Darren Edge et. al., 24 Apr 2024, Computer Science

**LLM derived Knowledge Graph**

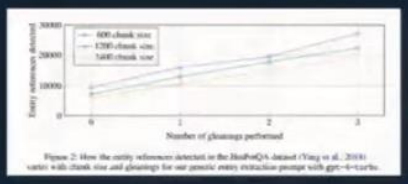
Utilizes LLM to build a knowledge graph, aggregating entities to communities and summarizing them

**Combined Information**

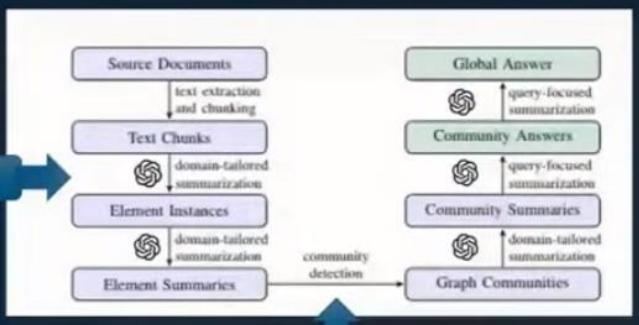
"The advantage of using a knowledge graph data representation is that it can quickly and straightforwardly combine information from multiple documents or data sources about particular entities." by Tomaz Bratanic.

# Microsoft Approach to GraphRAG

*"From Local to Global: A Graph RAG Approach to Query-Focused Summarization", Darren Edge et. al., 24 Apr 2024, Computer Science*



x Gleanings




- Source Documents to Text Chunks: Source documents are split into smaller text chunks for processing.
- Text Chunks to Element Instances: Each text chunk is analyzed to extract entities and relationships, producing a list of tuples representing these elements.
- Element Instances to Element Summaries: Extracted entities and relationships are summarized by the LLM into descriptive text blocks for each element.
- Element Summaries to Graph Communities: These entity summaries form a graph, which is then partitioned into communities using algorithms like Leiden for hierarchical structure.
- Graph Communities to Community Summaries: Summaries of each community are generated with the LLM to understand the dataset's global topical structure and semantics.

*"Implementing 'From Local to Global' GraphRAG with Neo4j and LangChain: Constructing the Graph" Tomaz Bratanic, medium.com*

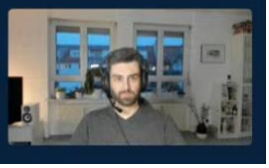
## Graph Construction Challenges

1. **Duplicates**
   Duplicate entities and relationships can clutter the graph and reduce efficiency.
2. **Sparse Data**
   Sparse data becomes a significant issue without entity resolution
3. **Incomplete Information**
   Incomplete or partial data from various sources can result in scattered and disconnected pieces of information, making it difficult to form a coherent and comprehensive understanding of entities -> needs LLM based entity resolution

"Implementing 'From Local to Global' GraphRAG with Neo4j and LangChain: Constructing the Graph" Tomaz Bratanic, Jul 9, 2024, Medium

## I want to use GraphRAG, but ...

1. **Too many duplicates!**
   Duplicate entities and relationships clutter the graph
2. **Too expensive LLM usage!**
   High computational costs for LLM calls
3. **Too complex, too cluttered for my use case...**
   Overwhelming complexity for specific applications
4. **Need for data control**
   I need to find the way to maintain control over my data in the GraphDB and be able to do RAG on it
5. **Cost-effective solution**
   Also to make it attractive for the stakeholders I wa
   still very good



# Fixed Entity Architecture

# The Ontological Fishbone

What is the ontology of you microworld?

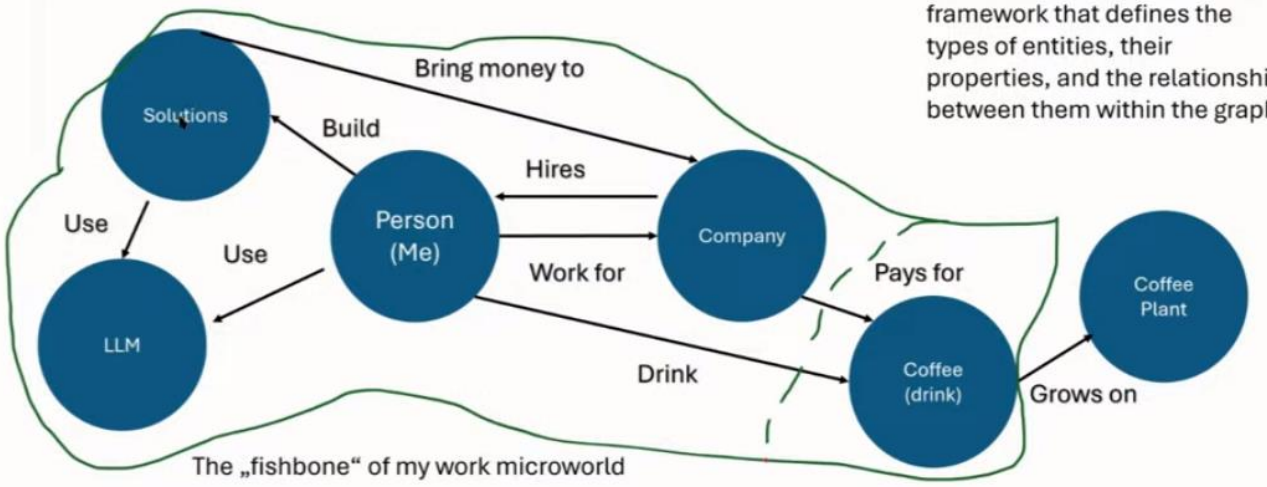
Ontologies are tools that help us understand the world around us. The "fishbone" metaphor encourages us to explore the fundamental building blocks and connections that make up our conceptual domains.

Creating ontologies requires careful consideration of what to include, how things are related, and what is most important. The fishbone structure helps us identify core elements while acknowledging the complex web of connections that bring our ideas to life.

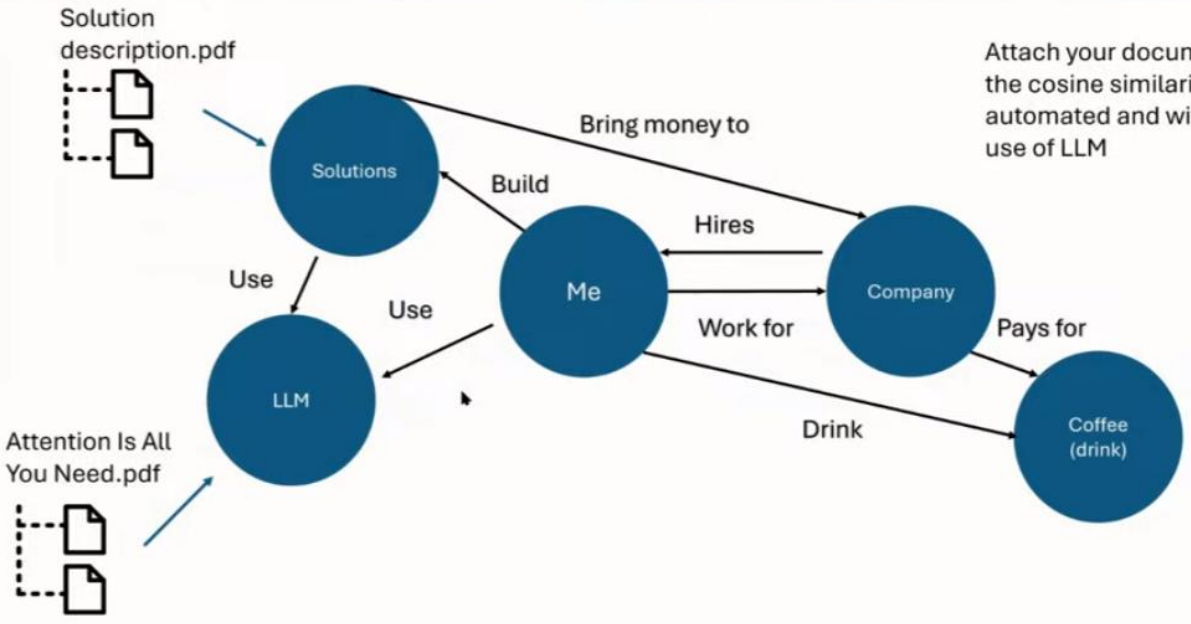
# Fixed Entity Approach

## Ontology is a meta-model of a microworld of your domain

The ontology of a knowledge graph refers to the structured framework that defines the types of entities, their properties, and the relationships between them within the graph.

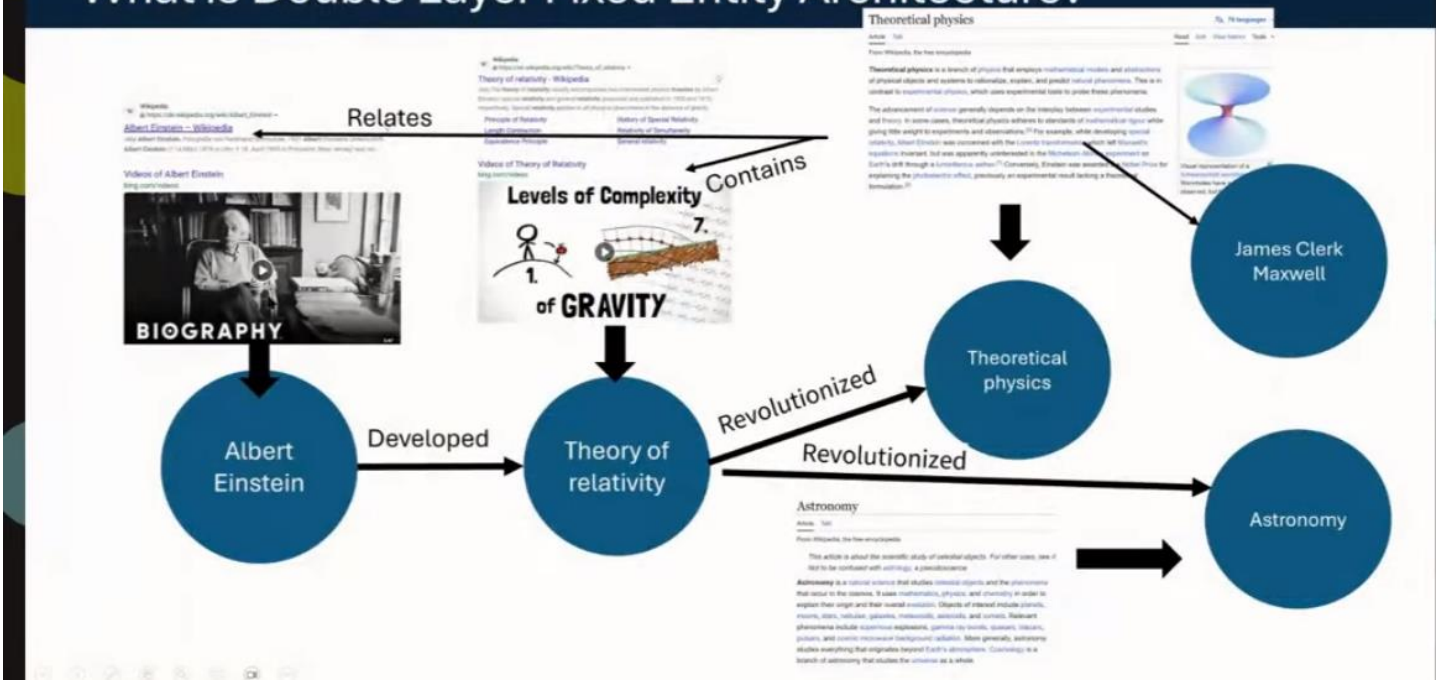


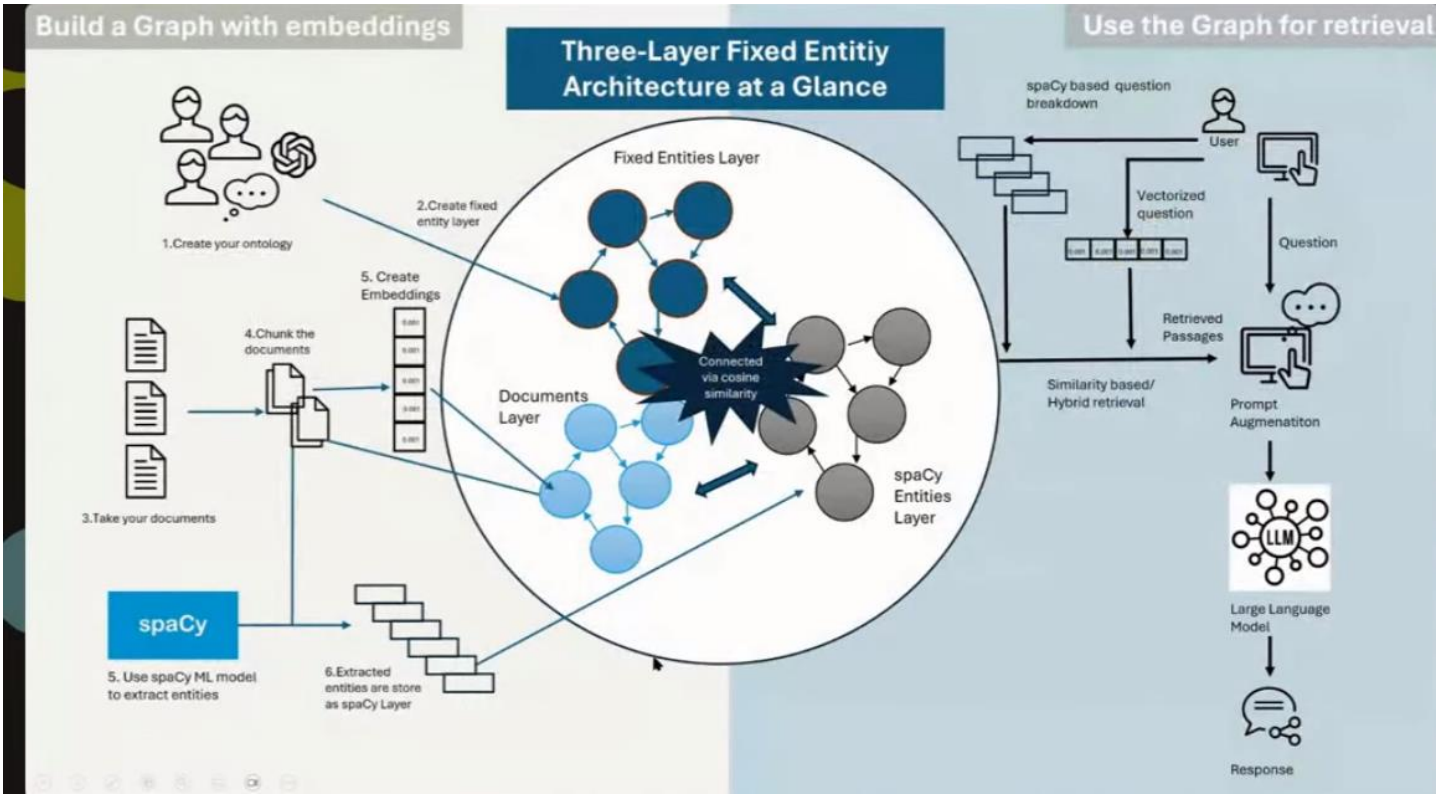

# Fixed Entity Approach



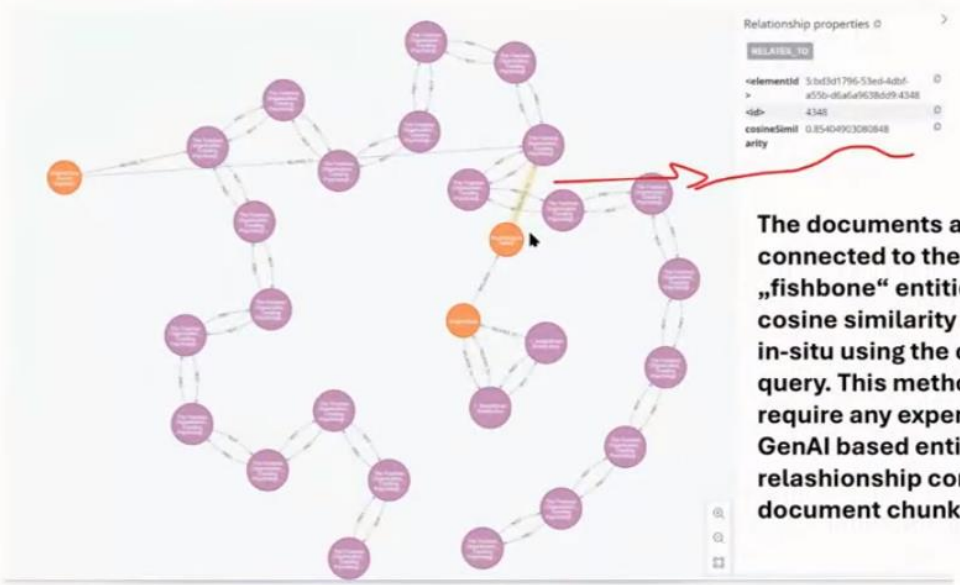

Attach your documents using the cosine similarity – fully automated and without any use of LLM

# What is Double Layer Fixed Entity Architecture?



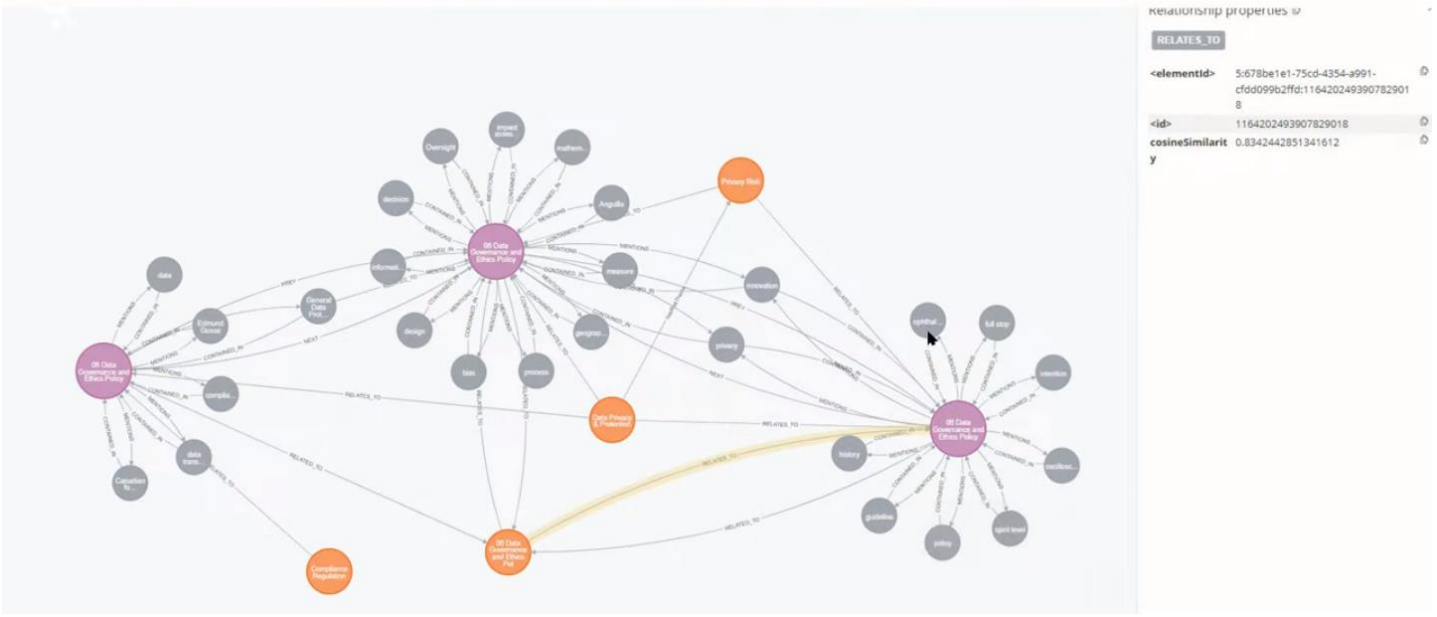



# Connection of the documents to each other and the „fishbone“ entities




**The documents are connected to the „fishbone“ entities by the cosine similarity calculated in-situ using the cypher query. This method do not require any expensive GenAI based entity-relashionship convertion of document chunks!**

# Three-Layer Fixed Entitiy Architecture at a Glance



# Graph Based Hybrid Search

## Nodes and Relationships Vector Index

The entity and edge vector indexes find the closest documents on similarity properties using Neo4J built-in functions.

## Full Text Index

The full text index allows additional search across the entire text content of documents using keyword search.

## Fixed Entity Smart Search

The fixed entity smart search represents additional possibilities to tweak the cypher query code performing retrieval.

## Topic Related Retrieval On Graph

1.Ask your question:
"Find the main research fields of Albert Einstein that revolutionized the world?"

spaCy

2. Extract related topics
information (1) : research
process (1) : research
human (1) : Albert Einstein
concept (1) : world
object (1) : world

3. Extract ALL nodes from ALL 3 node levels, having the name of ALL extracted related topics. Build a subgraph with all of them

Use case specific: more customization

4. Filter on ONLY Fixed Entity (FE) Layer and Documents Layer and retrieve the FE and the document chunks than are only as first chunk connected to the FE's.

5. Take all retrieved nodes' full_text or description and make the dot product on the user's query and the extracted text, and retrieve first x passages for the answer generation.

Albert Einstein's main research fields include:

1. **Theory of Relativity**: This encompasses b
forces of nature. These theories revolution
2. **Theoretical Physics**: Einstein made signifi
theory, such as explaining the photoelectr
3. **Astronomy and Cosmology**: Through his
gravitational waves.

Albert Einstein's main research fields that revolutionized the world include:

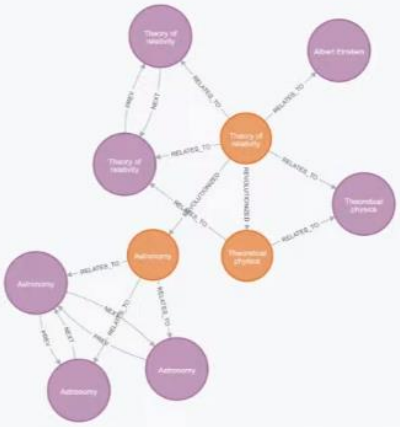
**1.Theory of Relativity**:
1. **Special Relativity**: Introduced in 1905, this theory addressed the inability of classical mechanics to acco
electromagnetic field. It led to the famous mass-energy equivalence formula ( E = mc^2 ).
2. **General Relativity**: Proposed in 1915, this theory extended the principles of special relativity to include
understanding of gravity and the structure of the universe.

**2.Quantum Mechanics**:
1. **Photoelectric Effect**: Einstein's explanation of the photoelectric effect, for which he received the 1921 N
in the development of quantum theory.
2. **Quantum Theory of Radiation**: His work on the quantum physics of radiation, including the concept that light consists of particles called photons, was significant in the field.
3. **Bose-Einstein Statistics**: Collaborating with Indian physicist Satyendra Nath Bose, Einstein laid the groundwork for Bose-Einstein statistics, which describes the statistical distribution of identical particles with integer spin.

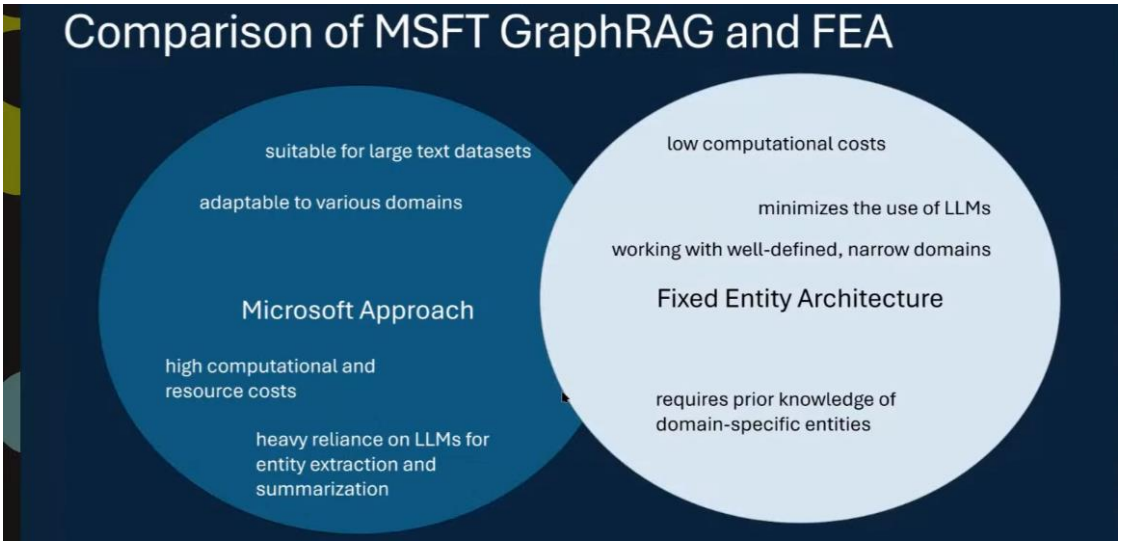
**3.Statistical Mechanics**:
1. Einstein made important contributions to statistical mechanics, particularly in explaining Brownian motion, which provided empirical evidence for the existence of atoms and molecules.

**4.Cosmology and Astrophysics**:
1. His theories of relativity predicted extraordinary astronomical phenomena such as neutron stars, black holes, and gravitational waves, significantly advancing the fields of cosmology and astrophysics.

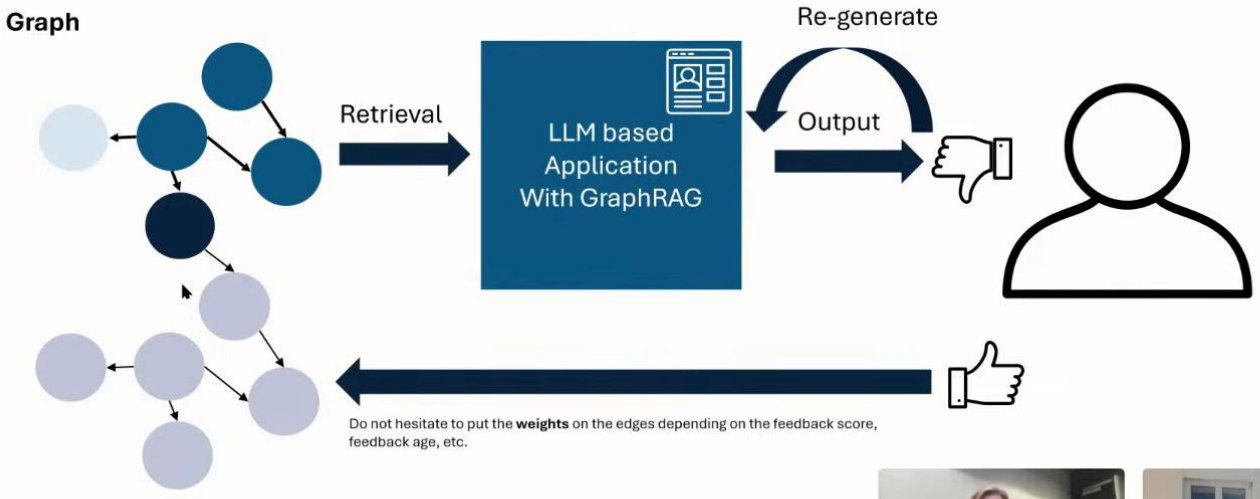
Einstein's work not only transformed theoretical physics but also had profound implications for the philosophy of science and the development of modern technology.

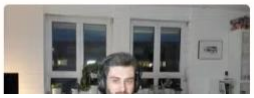
# Comparison of MSFT GraphRAG and FEA




# Example of feedback data integration




**Best Produced Result Feedback Nodes**



# Example of Fixed Layer Architecture for a semantic layer

The KG is a Fixed Entity Architecture (FEA)* type property graph. It is a lexical graph, containing information on relational data schema, overall domain information, and specific information of the use case.

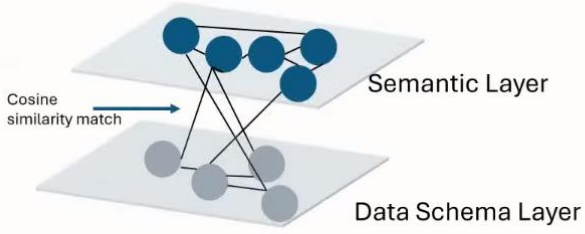
Using the KG the system can perform a semantic reasoning on user questions.

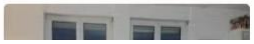
The KG is fully vectorized with the built-in vector indexes allowing to perform fully-fledged Graph RAG for semantic reasoning.

The FEA-KG has (at least) two layers:

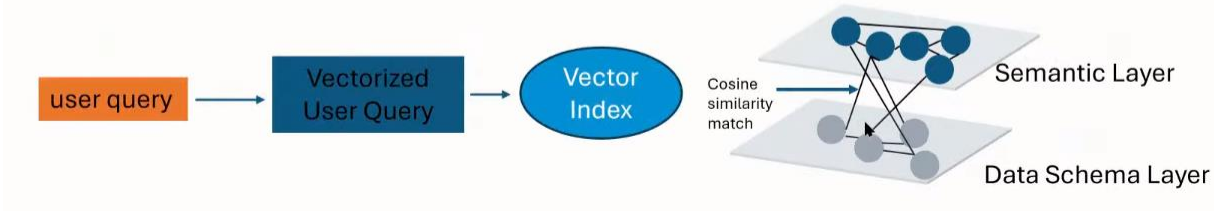
- **Relational Data Schema**
- **Domain specific Information**

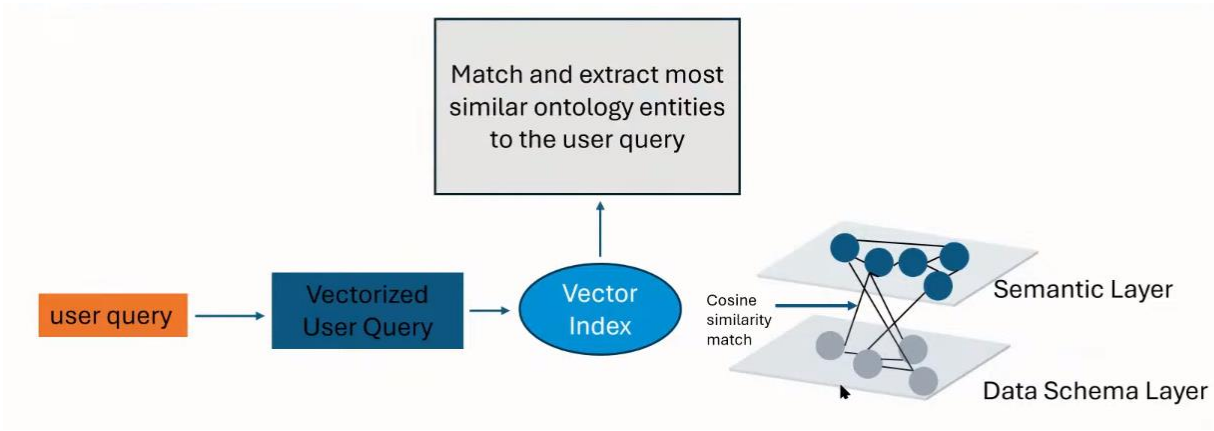
The Layers an be connected either by a domain logic or a cosine similarity match.

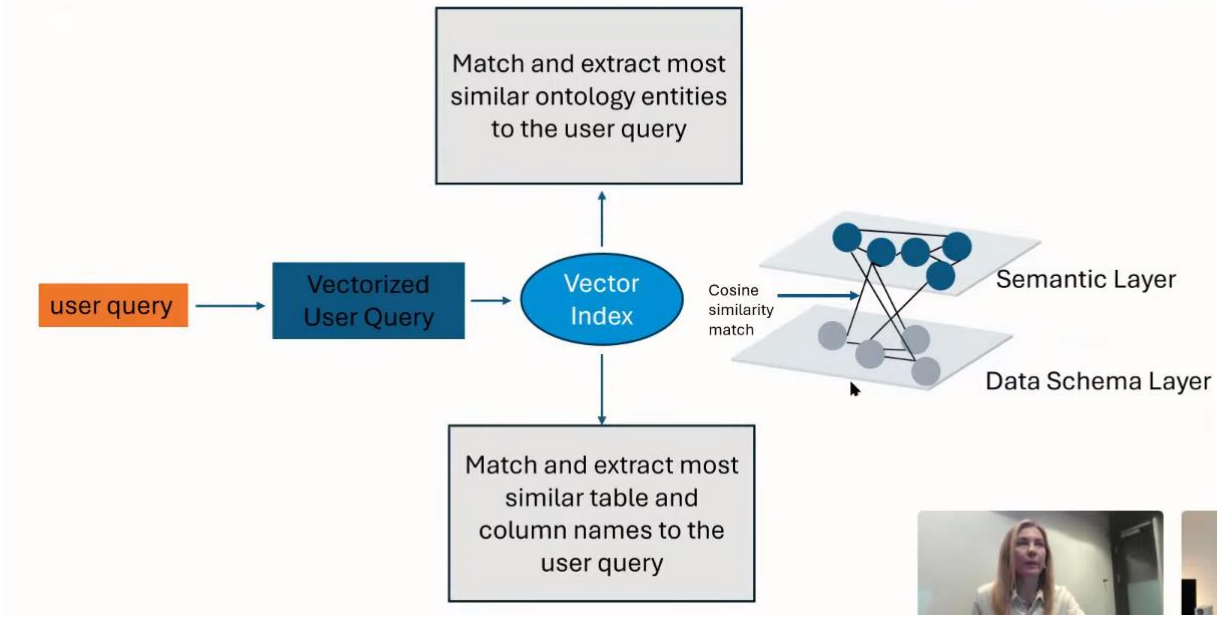


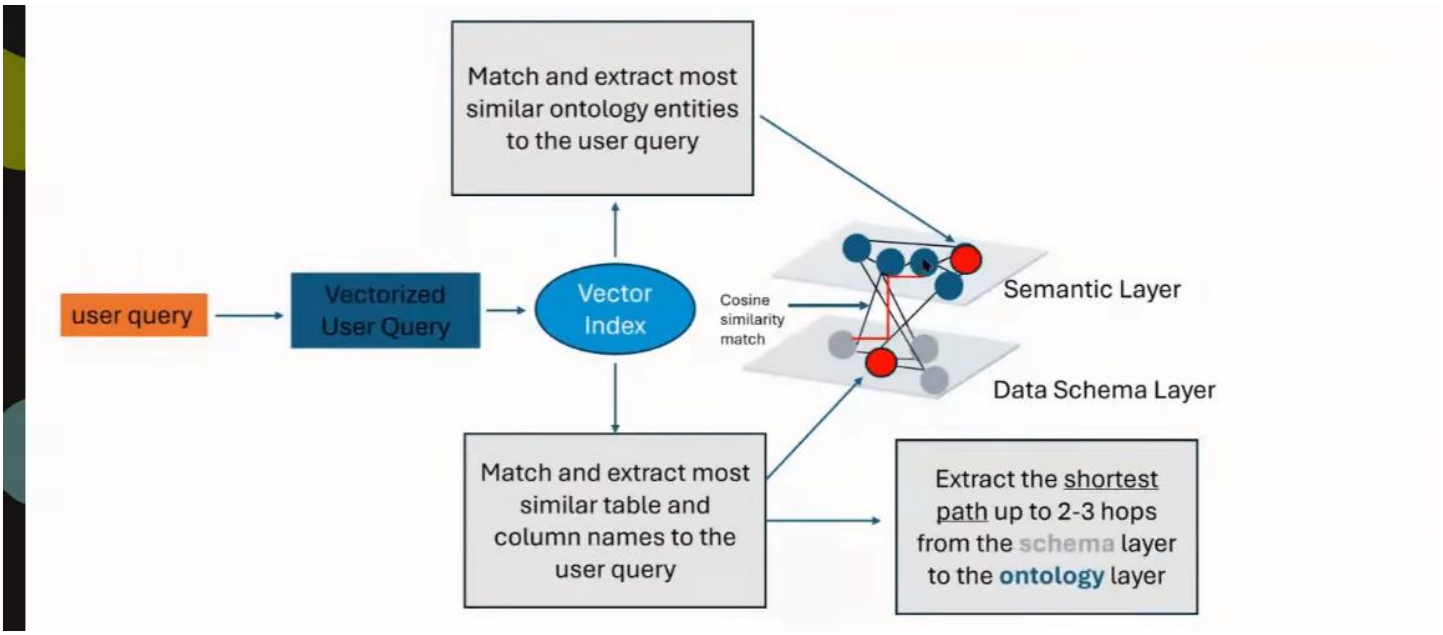



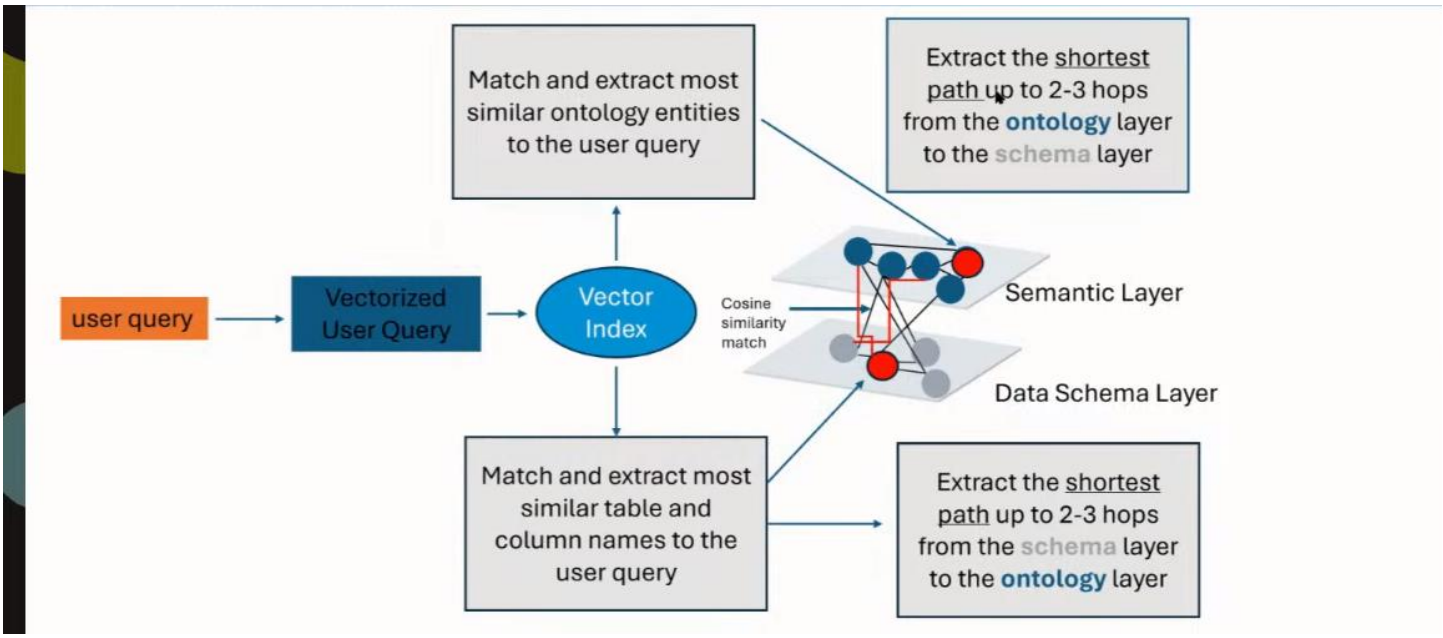
# Example of Fixed Layer Architecture for a semantic layer

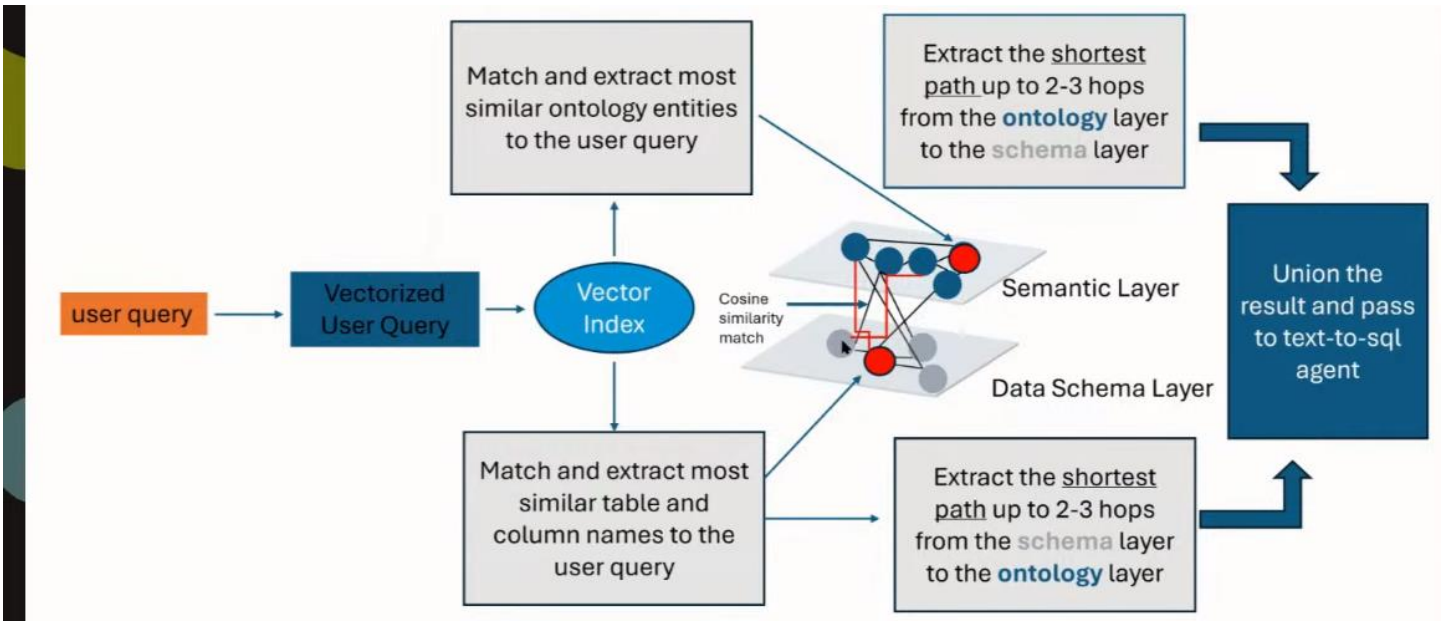



Match and extract most similar ontology entities to the user query
user query
Vectorized User Query
Vector Index
Cosine similarity match
Semantic Layer
Data Schema Layer
Match and extract most similar table and column names to the user query
Extract the shortest path up to 2-3 hops from the schema layer to the ontology layer


Match and extract most similar ontology entities to the user query
Extract the shortest path up to 2-3 hops from the ontology layer to the schema layer
user query
Vectorized User Query
Vector Index
Cosine similarity match
Semantic Layer
Data Schema Layer
Match and extract most similar table and column names to the user query
Extract the shortest path up to 2-3 hops from the schema layer to the ontology layer


Match and extract most similar ontology entities to the user query
Extract the shortest path up to 2-3 hops from the ontology layer to the schema layer
user query
Vectorized User Query
Vector Index
Cosine similarity match
Semantic Layer
Data Schema Layer
Union the result and pass to text-to-sql agent
Match and extract most similar table and column names to the user query
Extract the shortest path up to 2-3 hops from the schema layer to the ontology layer

# Fixed entity approach

- Two- or Three layer KG avoids duplication of entities by-design
- Documents and named entities are added and connected using cosine similarity
- Possibility for (kind-of) reinforcement learning
- Very felixible hybrid search
- Ability to connect different knowledge domains