Unpacking Graph RAG: An overview of history, terminologies and examples

Prashanth Rao

AI Engineer, Kùzu Inc. 🇨🇦

kuzudb.com

The AI Conference

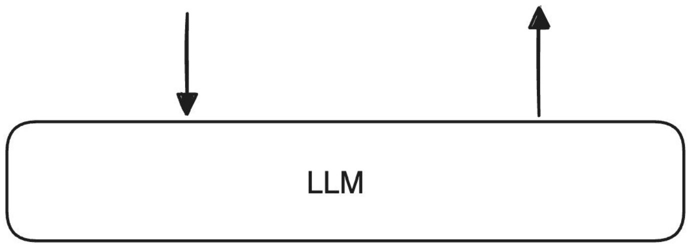
San Francisco | 11 Sep 2024

Overview

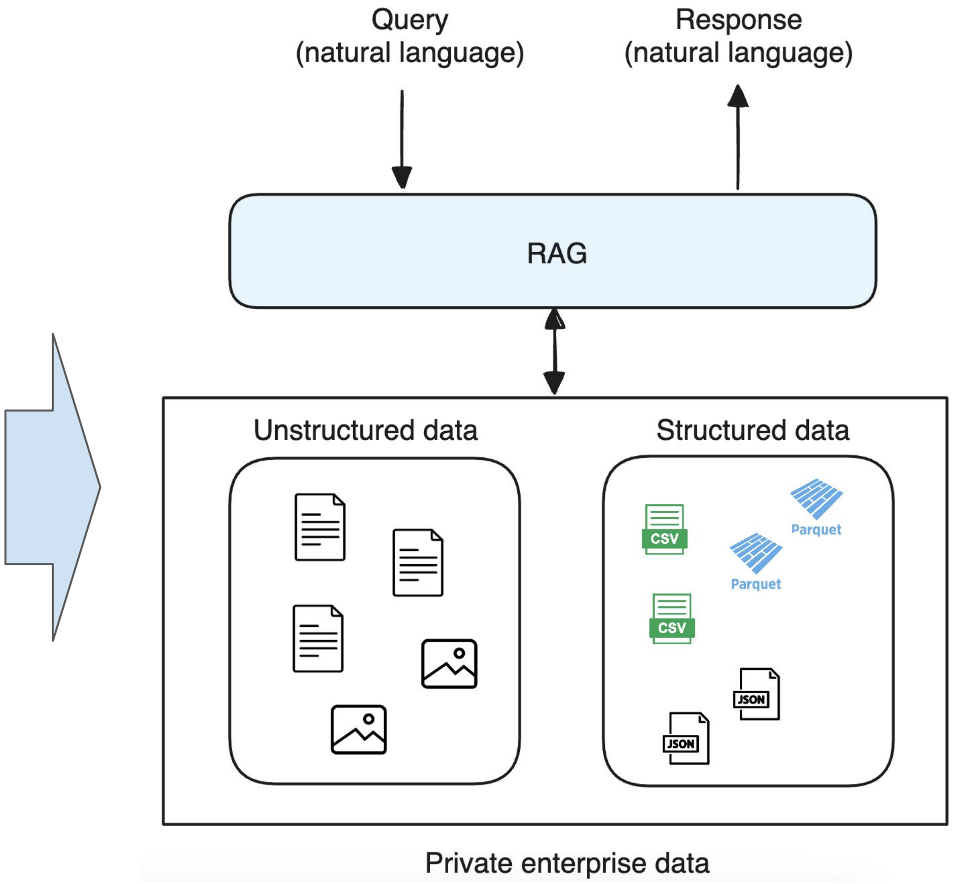
- **Graph RAG** has become an incredible buzz term in recent times
- What is Graph RAG, and what are its **components**?
- Do graphs measurably improve RAG, **in practice**?
- Can we devise a **framework** to better understand Graph RAG?
  - What is the “graph” in Graph RAG? What do the nodes and edges represent?
  - How is the retrieval process different from traditional (vector-only) RAG?
- What role do **databases** play in the pipeline?

Retrieval in the age of LLMs

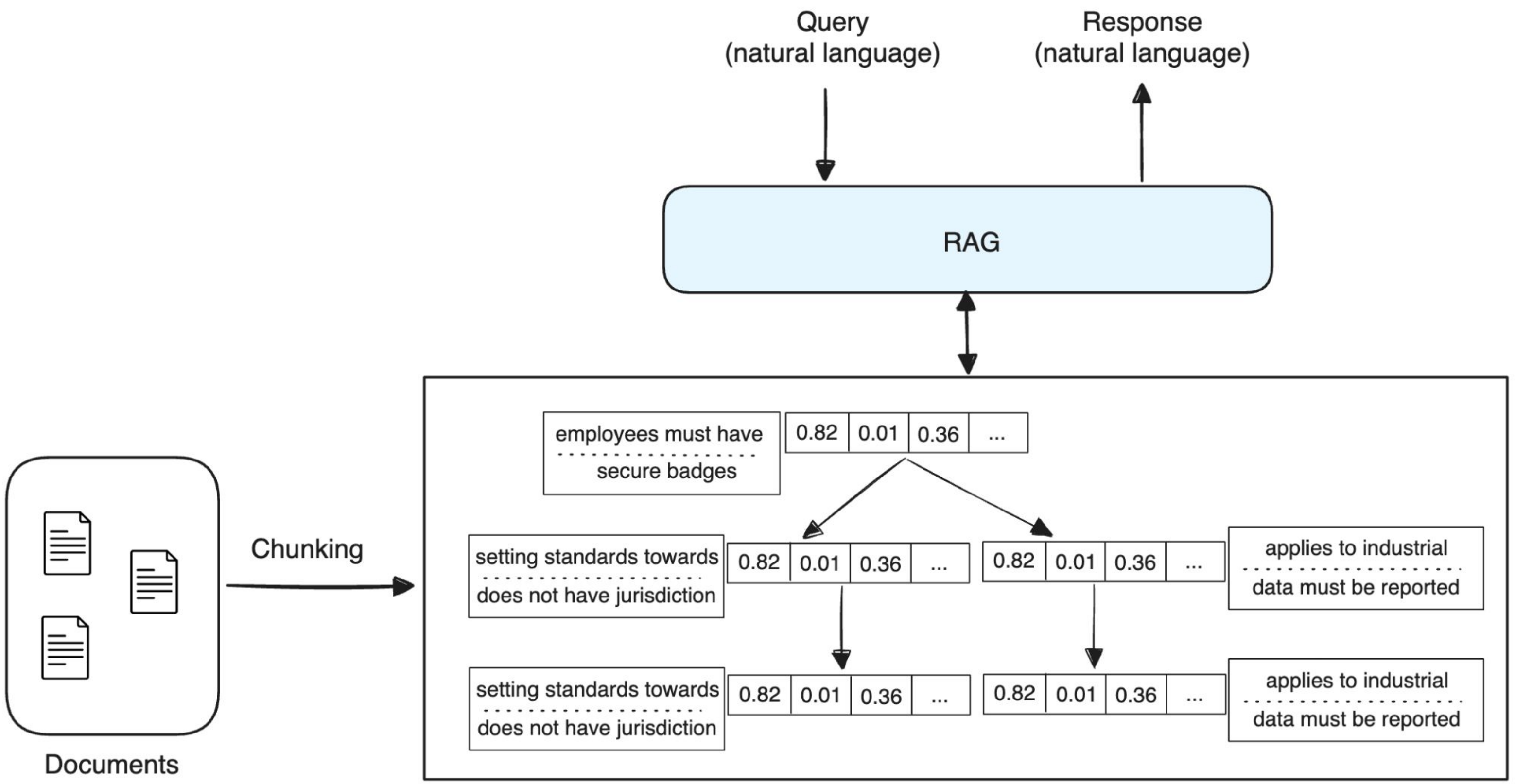
“Chat with an LLM”



Cannot easily retrieve from private enterprise data



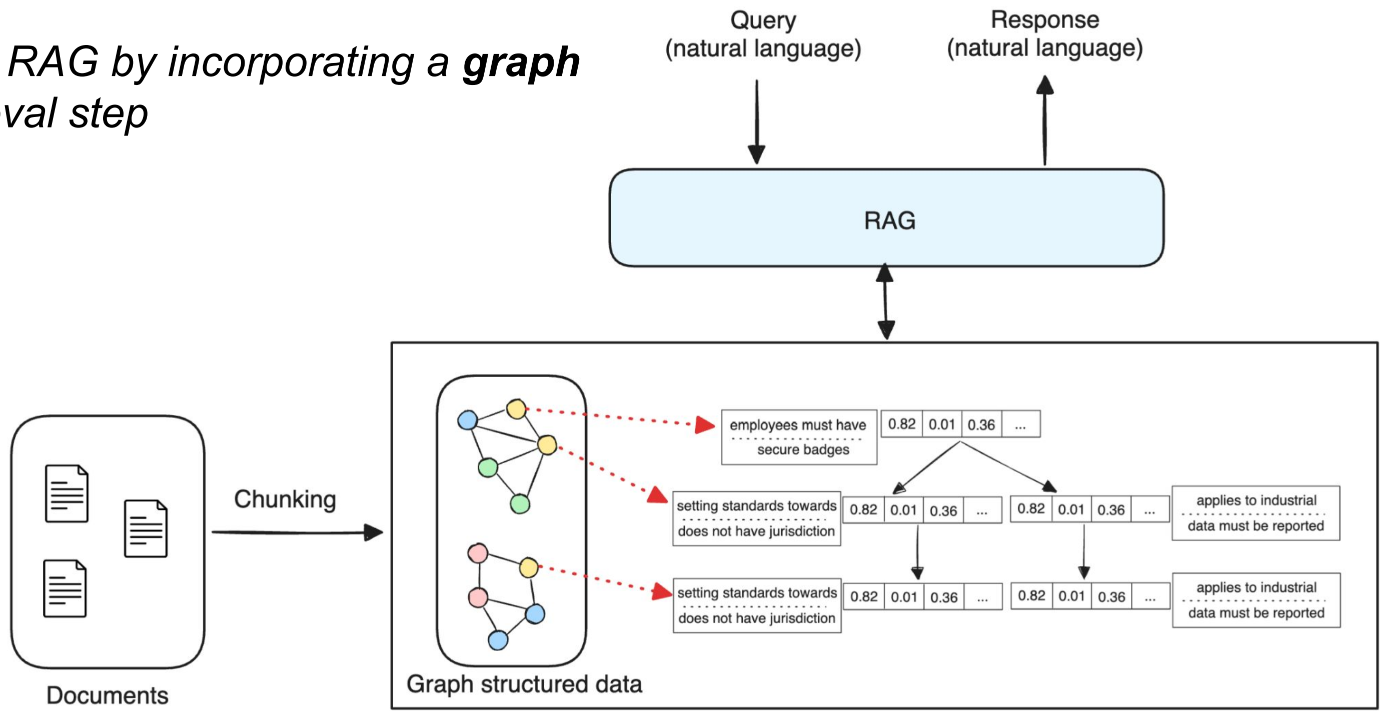
A deeper look at traditional RAG



Vector database (retrieve top-k documents)

What is Graph RAG?

*Extends traditional RAG by incorporating a **graph** as part of the retrieval step*



Query (natural language)
Response (natural language)
RAG
Documents
Chunking
Graph structured data
employees must have secure badges
0.82 0.01 0.36 ...
setting standards towards does not have jurisdiction
applies to industrial data must be reported

In any system that uses this approach:

- Question 1: What is the graph? I.e., what are its nodes and edges?
- Question 2: How is the retrieval process different from traditional RAG?

Why enhance unstructured data with a graph?

- Graphs are **object-oriented** in nature – they represent entities or objects in the real world via nodes, and how they are connected via edges

- Graphs capture relationships between entities **explicitly**
  - Traversing the vicinity of an entity to get added context is *natural and easy*

- A graph data model is a good fit to **add structure** to related entities extracted from unstructured data

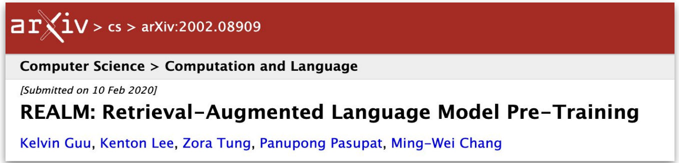
- Importantly, graph triples/edges `<subject, predicate, object>`, can be represented as **simple sentences** (useful to generate context)

Some history…

Early primary sources for "RAG"

Feb 2020 [Google]

REALM: Retrieval-Augmented Language Model Pre-Training



arXiv > cs > arXiv:2002.08909

Computer Science > Computation and Language

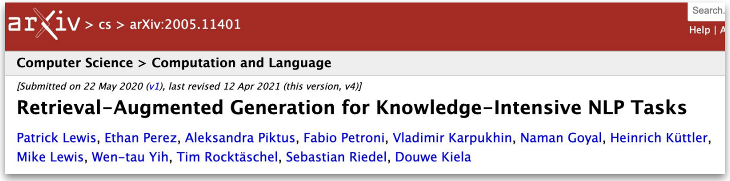
[Submitted on 10 Feb 2020]

REALM: Retrieval-Augmented Language Model Pre-Training

Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, Ming-Wei Chang

Apr 2021 [Facebook AI Research]

Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks



arXiv > cs > arXiv:2005.11401

Search

Help | A

Computer Science > Computation and Language

[Submitted on 22 May 2020 (v1), last revised 12 Apr 2021 (this version, v4)]

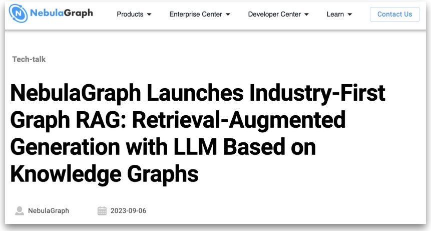
Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, Douwe Kiela

Early primary source for "Graph RAG"

Sep 2023 [NebulaGraph]

NebulaGraph Launches Industry-First Graph RAG: Retrieval-Augmented Generation with LLM Based on Knowledge Graphs



Tech-talk

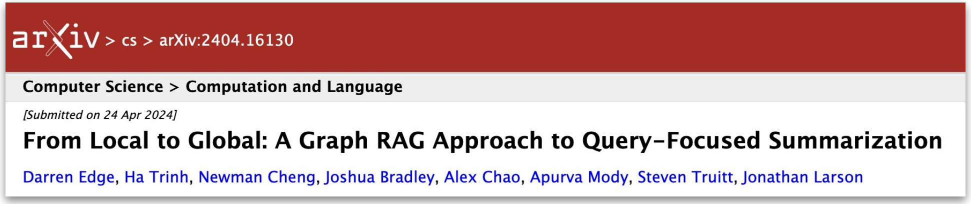
NebulaGraph Launches Industry-First Graph RAG: Retrieval-Augmented Generation with LLM Based on Knowledge Graphs

NebulaGraph 2023-09-06

Ongoing mindshare: Microsoft's "Graph RAG"

Apr 2024 [Microsoft Research]

From Local to Global: A Graph RAG Approach to Query-Focused Summarization



arXiv > cs > arXiv:2404.16130

Computer Science > Computation and Language

[Submitted on 24 Apr 2024]

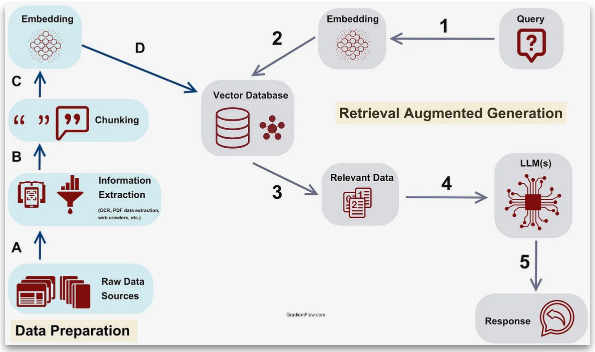
From Local to Global: A Graph RAG Approach to Query-Focused Summarization

Darren Edge, Ha Trinh, Newman Cheng, Joshua Bradley, Alex Chao, Apurva Mody, Steven Truitt, Jonathan Larson

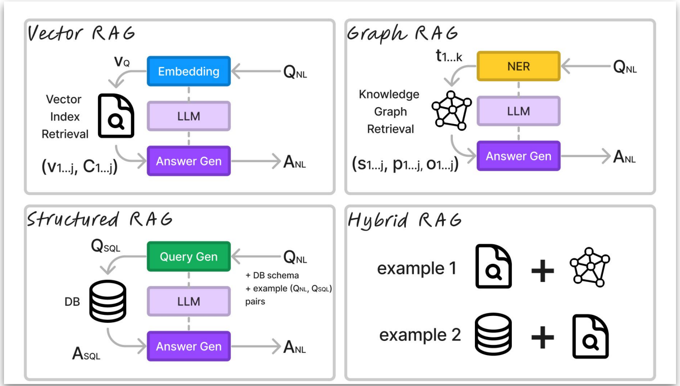
The emergence of “Hybrid RAG”

Not to be confused with “hybrid *search*”, **Hybrid RAG** is what you call RAG when you combine multiple retrieval methods

Jan 2024 [WhyHow.ai]
“Injecting Knowledge Graphs in different RAG stages”
Chia Jeng Yang

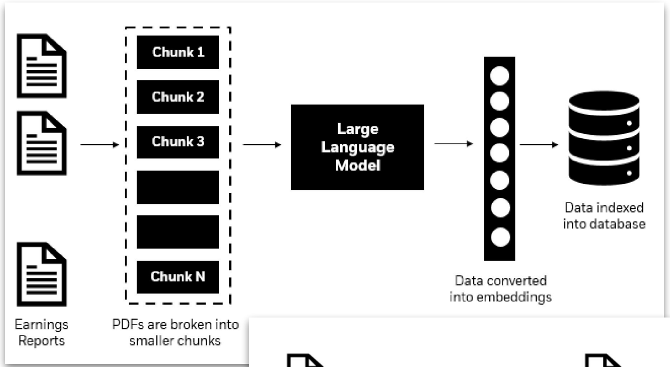


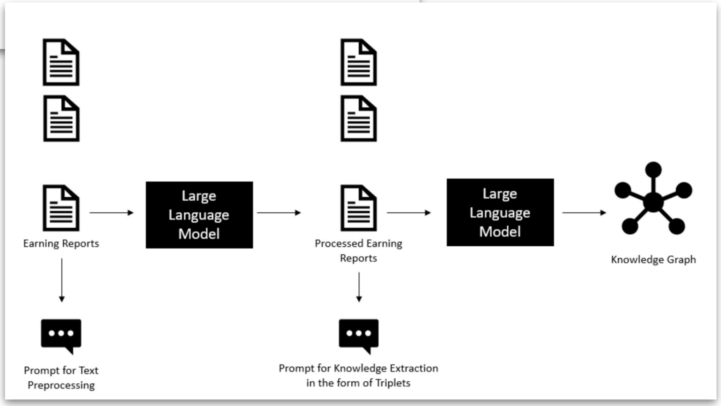
Feb 2024 [guitton.co]
“Graphs and Language”
Louis Guitton



Do graphs measurably improve RAG in practice?

HybridRAG: Integrating Knowledge Graphs and Vector Retrieval Augmented Generation for Efficient Information Extraction (BlackRock & Nvidia), Aug 2024


Chunk 1
Chunk 2
Chunk 3
Chunk N
Large Language Model
Data converted into embeddings
Data indexed into database
Earnings Reports
PDFs are broken into smaller chunks

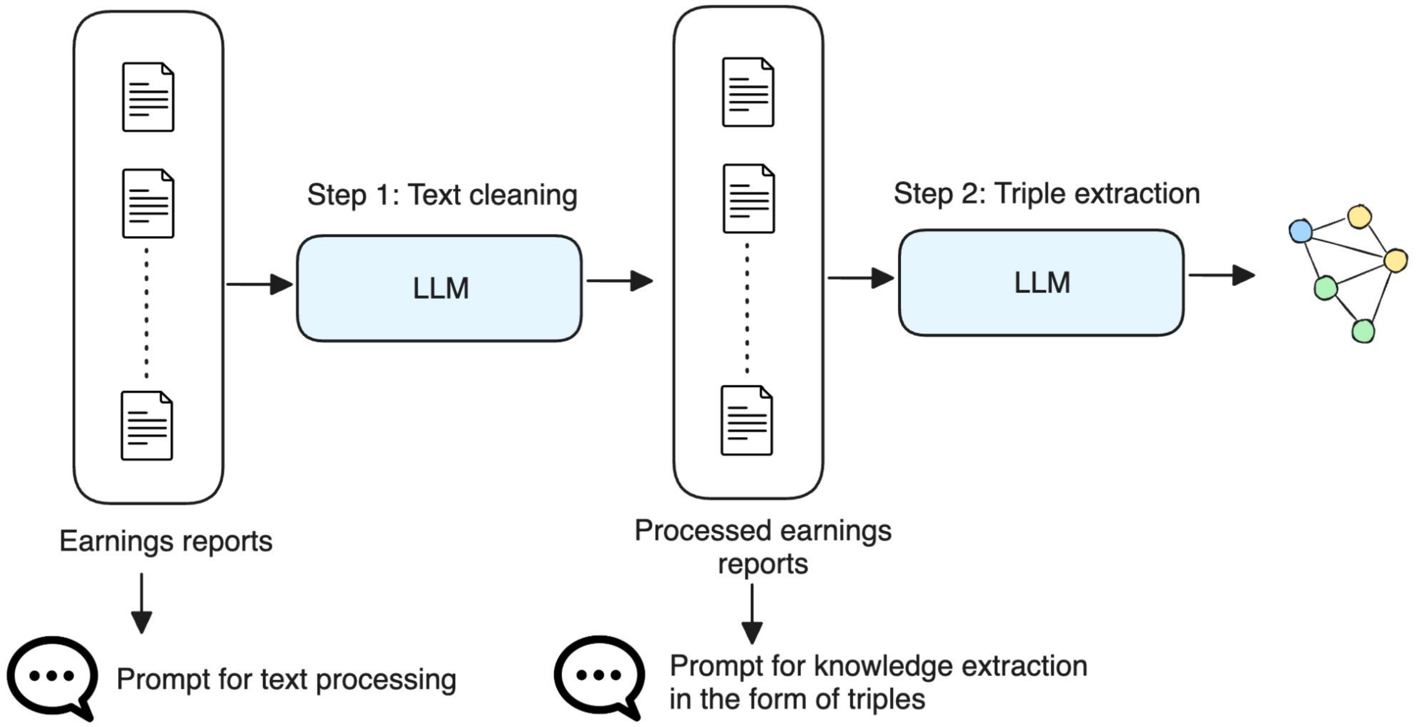

Earning Reports
Large Language Model
Processed Earning Reports
Large Language Model
Knowledge Graph
Prompt for Text Preprocessing
Prompt for Knowledge Extraction in the form of Triplets

Evaluation: Hybrid RAG system **does better overall** than systems that were based on vector retrievals or graph retrievals alone

Source: https://arxiv.org/pdf/2408.04948v1

Unpacking BlackRock's Hybrid RAG (1)

Question 1: What is the graph? What do its nodes and edges represent?


Step 1: Text cleaning
LLM
Step 2: Triple extraction
LLM
Earnings reports
Processed earnings reports
Prompt for text processing
Prompt for knowledge extraction in the form of triples

Unpacking BlackRock's Hybrid RAG (2)

Example of summarization and triple extraction

Chunk 1

Larry Fink is the CEO and co-founder of BlackRock, the world's largest asset management firm, established in 1988 …

Chunk 2

Born in Los Angeles, California, in 1952, Fink grew up in Van Nuys and later earned his MBA from UCLA's Anderson School of Management …

⋮

Chunk n

…
10.0 trillions of dollars in asset management …
…

Step 1: Text processing →

Processed chunk 1

Larry Fink is the CEO and co-founder of BlackRock.
BlackRock was established in 1988.

Processed chunk 2

Larry Fink was born in Los Angeles, California.
Larry Fink earned his MBA from UCLA

⋮

Processed chunk n

…
BlackRock manages 10.5 trillion dollars in assets.
…

Step 2: Triple extraction →

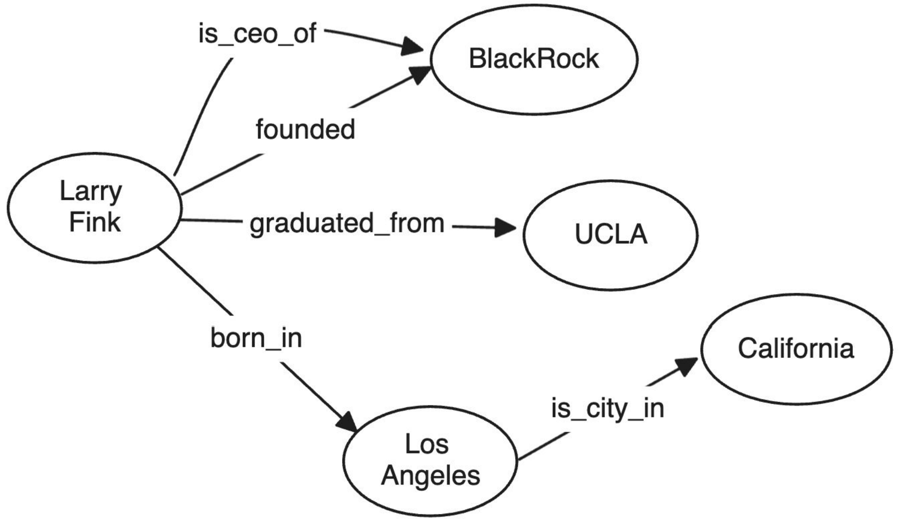
<Larry Fink, is_ceo_of, BlackRock >
<Larry Fink, founded, BlackRock >
<BlackRock, founded_in, 1988 >

<Larry Fink, born_in, Los Angeles >
<Los Angeles, is_city_in, California >
<Larry Fink, graduated_from, UCLA >

<BlackRock, asset_value, 10.5 trillion >

Unpacking BlackRock's Hybrid RAG (3)

Recall: Graphs can model simple sentences



Chunk 1

<Larry Fink, is_ceo_of, BlackRock >
<Larry Fink, founded, BlackRock >

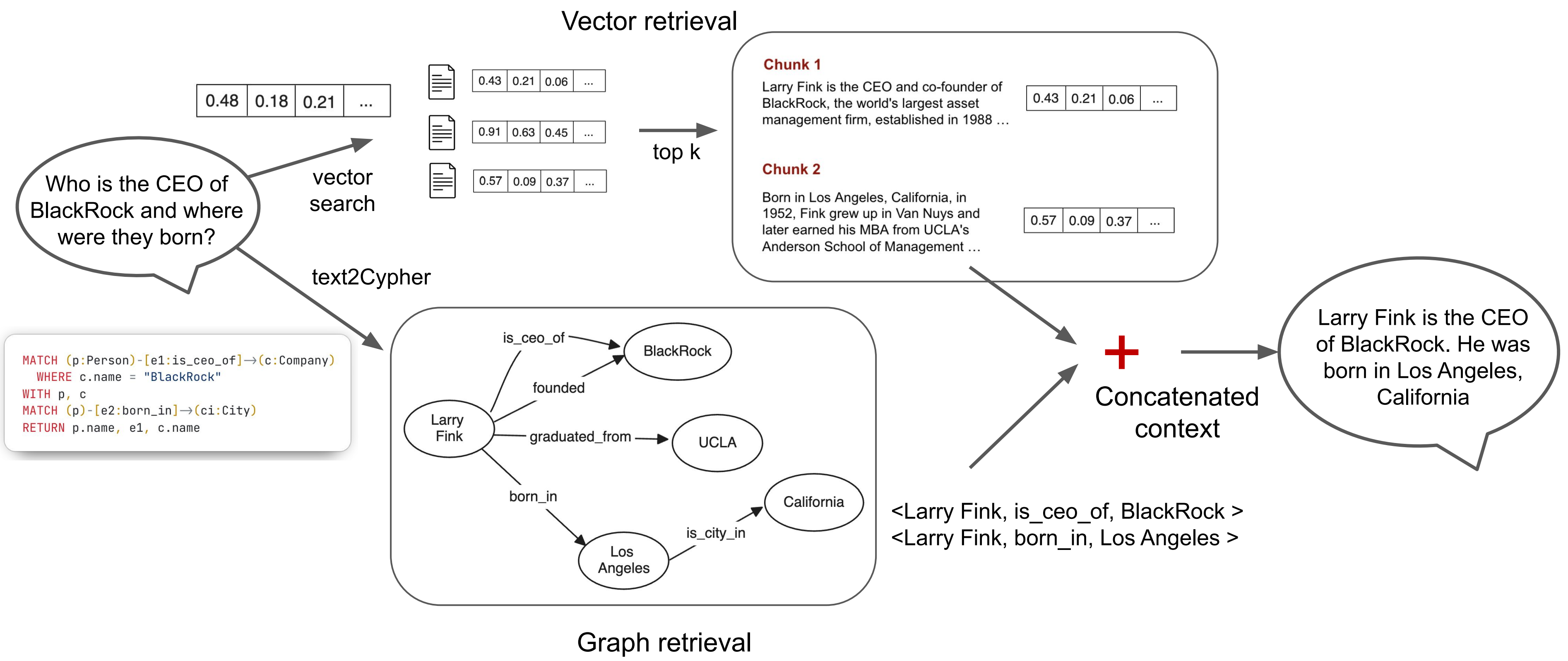
Chunk 2

<Larry Fink, born_in, Los Angeles >
<Los Angeles, is_city_in, California >
<Larry Fink, graduated_from, UCLA >

- Benefit 1: Information in disparate chunks are now **directly connected**
- Benefit 2: Triples are a form of capturing the **essence** of text chunks in very simple sentences
- Benefit 3: Can now put the triples into a graph DB where you can query it using a **query language**

Unpacking BlackRock’s Hybrid RAG (4)

Question 2: How is retrieval different from traditional RAG?



Chunk 1

Larry Fink is the CEO and co-founder of BlackRock, the world's largest asset management firm, established in 1988 …

0.43 | 0.21 | 0.06 | ...

Chunk 2

Born in Los Angeles, California, in 1952, Fink grew up in Van Nuys and later earned his MBA from UCLA's Anderson School of Management …

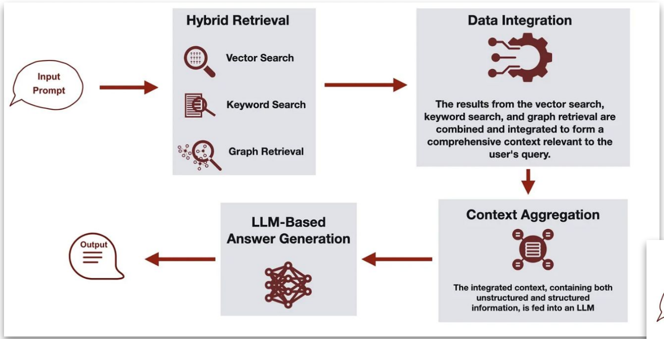
0.57 | 0.09 | 0.37 | ...

```
MATCH (p:Person)-[e1:is_ceo_of]→(c:Company)
  WHERE c.name = "BlackRock"
WITH p, c
MATCH (p)-[e2:born_in]→(ci:City)
RETURN p.name, e1, c.name
```

<Larry Fink, is_ceo_of, BlackRock >
<Larry Fink, born_in, Los Angeles >

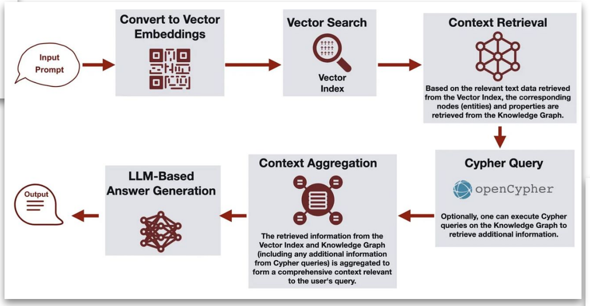
Retrieval strategies in Graph RAG

Concatenate context from a vector retrieval + graph retrieval (Hybrid RAG)

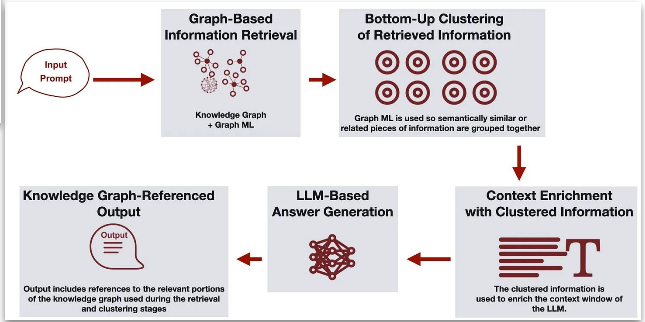

Input Prompt
Hybrid Retrieval
Vector Search
Keyword Search
Graph Retrieval
Data Integration
The results from the vector search, keyword search, and graph retrieval are combined and integrated to form a comprehensive context relevant to the user's query.
Context Aggregation
The integrated context, containing both unstructured and structured information, is fed into an LLM
LLM-Based Answer Generation
Output

+ Agents, prompt tuning, query expansion, and more…

Graph-enhanced QA:
Perform graph traversals downstream of vector/hybrid search


Input Prompt
Convert to Vector Embeddings
Vector Search
Vector Index
Context Retrieval
Based on the relevant text data retrieved from the Vector Index, the corresponding nodes (entities) and properties are retrieved from the Knowledge Graph.
Cypher Query
openCypher
Optionally, one can execute Cypher queries on the Knowledge Graph to retrieve additional information.
Context Aggregation
The retrieved information from the Vector Index and Knowledge Graph (including any additional information from Cypher queries) is aggregated to form a comprehensive context relevant to the user's query.
LLM-Based Answer Generation
Output

Semantic clustering
(Microsoft's local to global Graph RAG)


Input Prompt
Graph-Based Information Retrieval
Knowledge Graph + Graph ML
Bottom-Up Clustering of Retrieved Information
Graph ML is used so semantically similar or related pieces of information are grouped together
Context Enrichment with Clustered Information
The clustered information is used to enrich the context window of the LLM.
LLM-Based Answer Generation
Knowledge Graph-Referenced Output
Output
Output includes references to the relevant portions of the knowledge graph used during the retrieval and clustering stages

GraphRAG: Design Patterns, Challenges, Recommendations

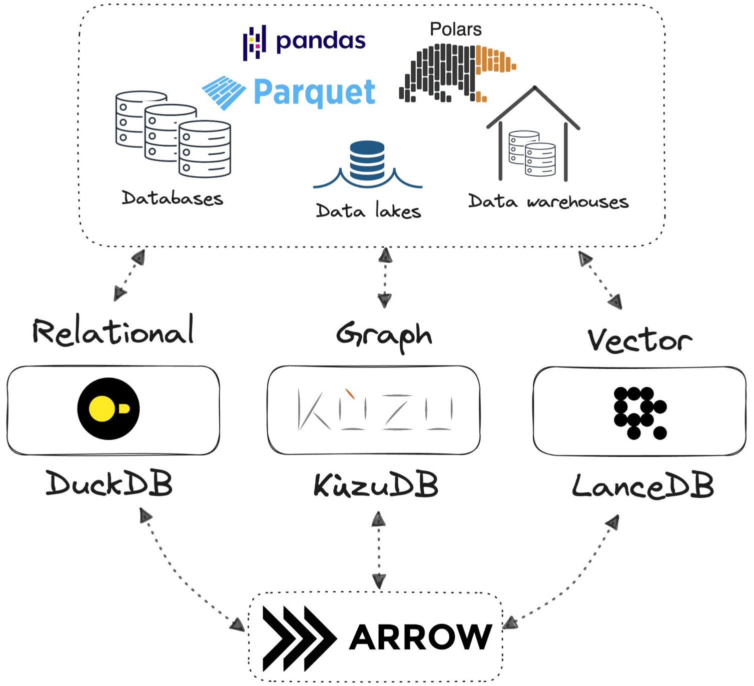
[Gradient Flow newsletter]

The role of databases

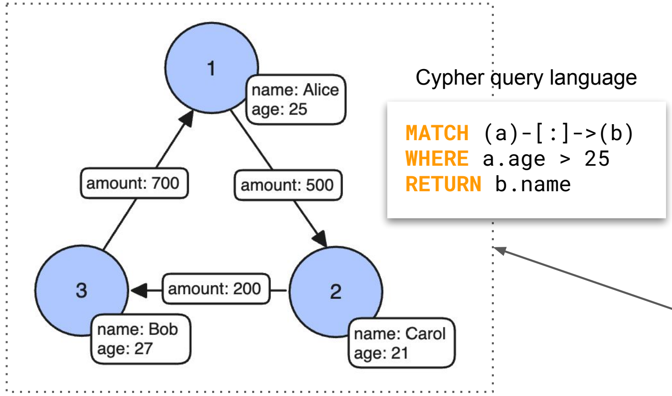
- In practice, graph construction is an **iterative** process – a graph is rarely “complete”, and needs to be built upon over time, as more data arrives
- Data **preprocessing** and **exploration** are key in early stages, and many graph databases offer **visual** tools to aid in this process
- Strong **persistence** guarantees within a database can help with reproducibility and sharing data across the organization
- **Scalability** is baked into a database’s design, enabling developers to more easily move from a PoC to a production-ready scenario

Databases are evolving alongside RAG

- Embeddability + ease of setup + interoperability + permissive licensing
- These characteristics do **not** preclude scalability or performance

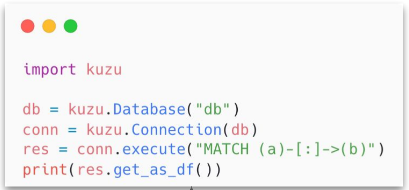


Usability features of Kùzu

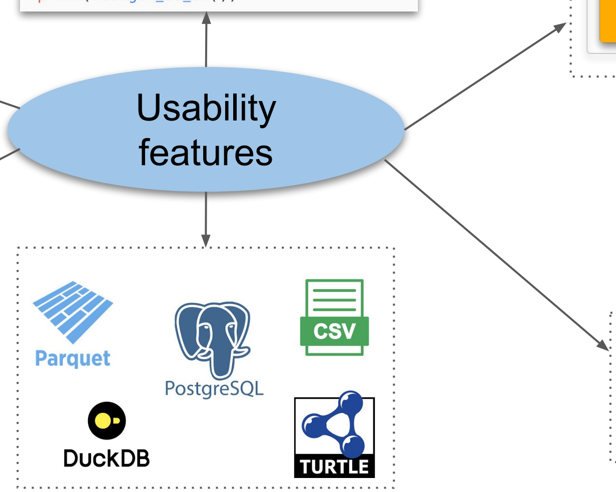


```
MATCH (a)-[:]->(b)
WHERE a.age > 25
RETURN b.name
```

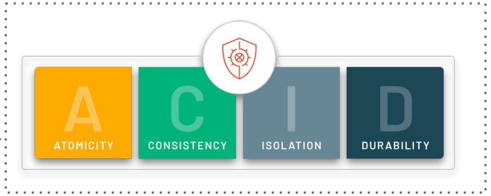
Embedded
(similar philosophy to DuckDB, LanceDB)



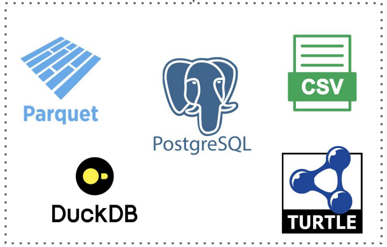
```
import kuzu

db = kuzu.Database("db")
conn = kuzu.Connection(db)
res = conn.execute("MATCH (a)-[:]->(b)")
print(res.get_as_df())
```



ACID transactions



Integrations with ML/AI frameworks



Interoperable with many formats

Permissively licensed

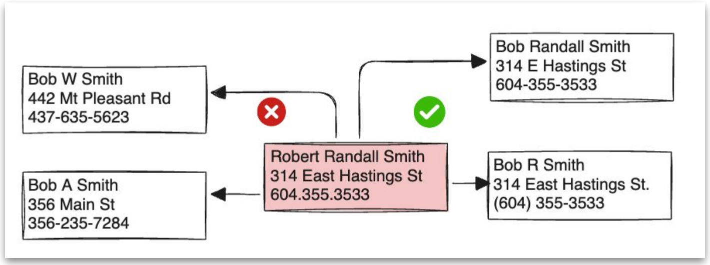
Learn more at https://kuzudb.com

Why use Kùzu as part of a Graph RAG system?

- Interoperability & scalability: Graphs are typically constructed from a variety of structured & unstructured sources
- Model data as property graphs, with the imposition of structure (strict schema)
- Combine your existing property graphs with domain-specific RDF graphs while *still querying them in Cypher*
- Add a persistent graph layer to advanced Graph RAG methods that utilize GNNs, node embeddings and/or graph algorithms (e.g., clustering)
    - Seamless interoperability with NetworkX (+ native graph algorithm support coming soon)
    - Serves as a PyTorch Geometric backend

Note on graph construction: Quality is paramount

- A significant bottleneck in implementing Graph RAG remains **constructing high quality graphs**
- Retrieval performance, which can disproportionately affect the generation outcome
- Entity resolution is a key (and often necessary) step when combining data from structured and unstructured sources
  - **Senzing** API, SDK and Desktop tools
  - **WhyHow.ai** Knowledge Graph Studio Platform



Takeaways

- Graph RAG is **not a monolith** – graphs and vector search can be combined in various ways, using many different components in the indexing/serving stages
- Each stage can be built and tuned independently, so it's important to design concrete **evaluation** strategies (which are also continually evolving)
- Qualify **how** and **where** the graph is being used, and whether or not there exists a persistent graph storage layer
  - What is the "graph" in Graph RAG? What do the nodes and edges represent?
  - How is the retrieval process different from traditional RAG?
- As the tooling improves, it's likely that graphs will become **core components** of many information retrieval systems

Contribution ideas: Let’s get building!

The best way to learn how to use Graph RAG is by **building** and **evaluating**

AdalFlow	https://github.com/SylphAI-Inc/AdalFlow	(Issue #122)
Cognee	https://github.com/topoteretes/cognee	(Issue #54)
Kotaemon	https://github.com/Cinnamon/kotaemon	(Issue #134)
Strwythura	https://github.com/DerwenAI/strwythura	(Issue #3)
nano-graphrag	https://github.com/gusye1234/nano-graphrag	(Issue #2)

These, and many other such interesting open source projects are ongoing!

Thank you!

Kùzu is an **open source** graph database (MIT license)
Check us out on GitHub and give us a ⭐

Join our Discord to discuss more with the community about your graph use case

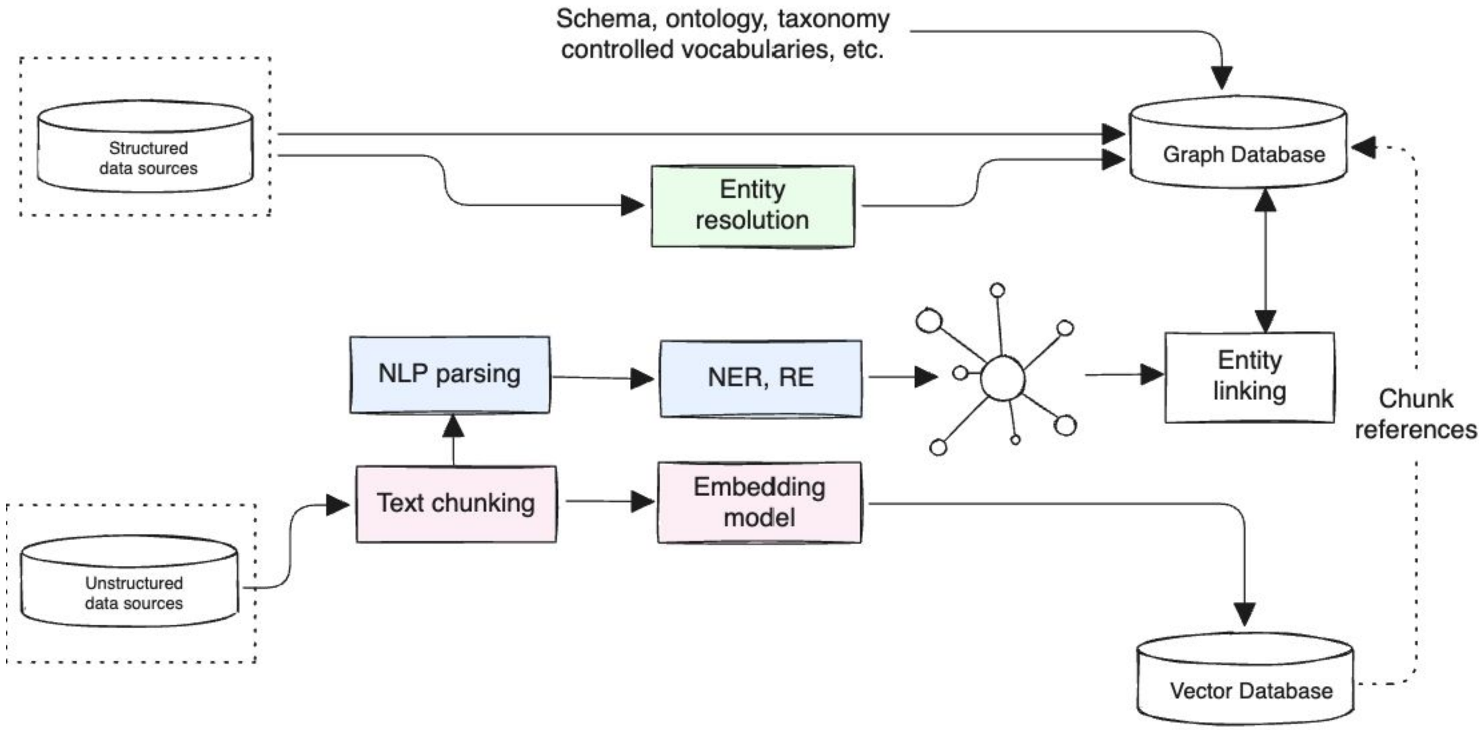
github.com/kuzudb/kuzu

@kuzudb

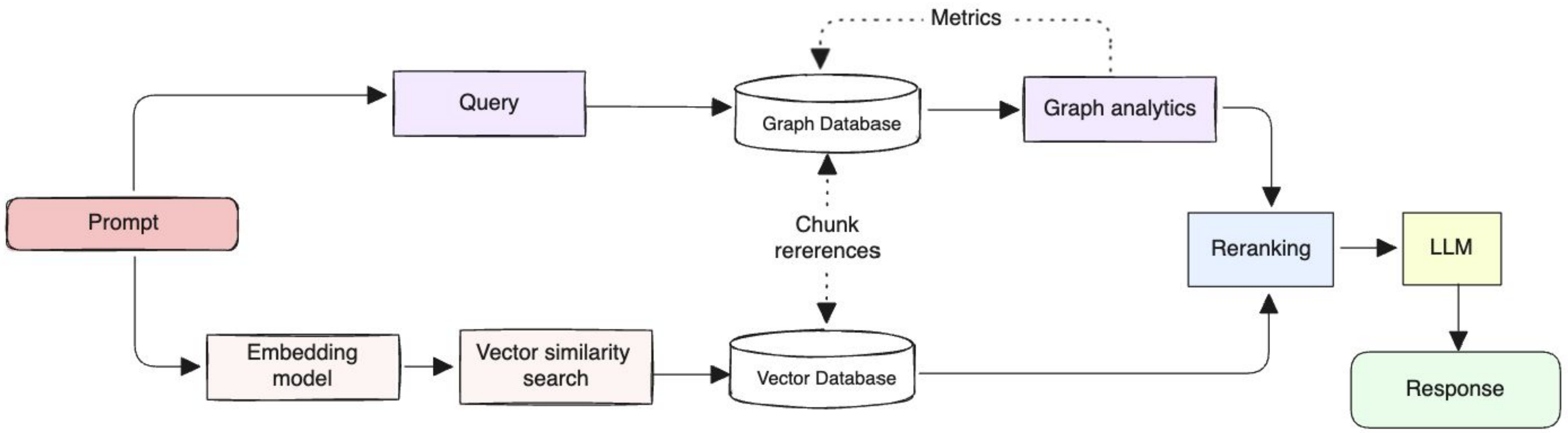
We're **Kùzu Inc.** on LinkedIn!

Additional slides

Stages of Graph RAG (1): Indexing



Stages of Graph RAG (2): Serving



Graph construction: Methods

- NER: Named entity recognition
  - Provide **labels for token spans**, parsed from **unstructured** data
- RE: Relationship extraction
  - Infer semantic **relationships** (labelled edges) between co-occurring entities
- Entity resolution
  - Disambiguate consistent **entities** across datasets from **structured** data
- Entity linking
  - Bridge **structured**/ER and **unstructured**/NER data together in a graph
- Chunk linking
  - Create **explicit links between chunks** via hierarchical structures