neo4j graphsummit

Hands on GraphRAG Workshop

Amsterdam 2025

Erik Bijl | Solution Engineering

Marco de Luca | Sr. Solution Engineer

Stephen Chin | VP Developer Relations

neo4j graphsummit

Who are we?

Erik Bijl

```
(Erik) - [:LIVES_IN] -> (`the Netherlands`)
(Erik) - [:USED_TO_BE] -> (`Data Scientist`)
(Erik) - [:WORKED_IN] -> (`Banking`)
(Erik) - [:IS_PART_OF] -> (`EMEA Field Team`)
(Erik) - [:LOVES] -> (`Football`)
```

Marco de Luca

```
(Marco) - [:LIVES_IN] -> (`Germany`)
(Marco) - [:WORKED_IN] -> (`IT`)
(Marco) - [:IS_PART_OF] -> (`EMEA Field Team`)
(Marco) - [:LOVES] -> (`Running`)
```

Side note: You can read Cypher queries now!

neo4j graphsummit

Agenda

1 **Workshop objectives**
Setting the stage

2 **Groups and setup**
Let’s get set up

3 **Module 1**
Graph basics: loading, queries, vectors

4 **Module 2**
Taming Unstructured Data

5 **Module 3**
GraphRAG and Agents

6 **Wrap up**
Resources and Q&A

neo4j graphsummit

Workshop Rules

- Ask questions straight away, this is an interactive session
- Raise your hand if you are stuck
- Slides & notebooks will be shared
- Have fun!

neo4j graphsummit

Before We Start

Connect to the notebooks:

attendee101 -> 150

https://bit.ly/neo4j-ams-one

attendee151 -> 200

https://bit.ly/neo4j-ams-two

Neo4j Browser : https://browser.neo4j.io/preview/

neo4j graphsummit

Quick Poll (by show of hands)

neo4j graphsummit

Skills, skills, skills…

neo4j graphsummit

Module 1

Graph basics: Queries, Algorithms & Vectors

Module 1

Graph Basics

- Creating a Graph from Structured Data
- Basic Cypher Queries and Pattern Matching
- Graph Algorithms
- Text Embeddings for Semantic Analysis
- Vector Search

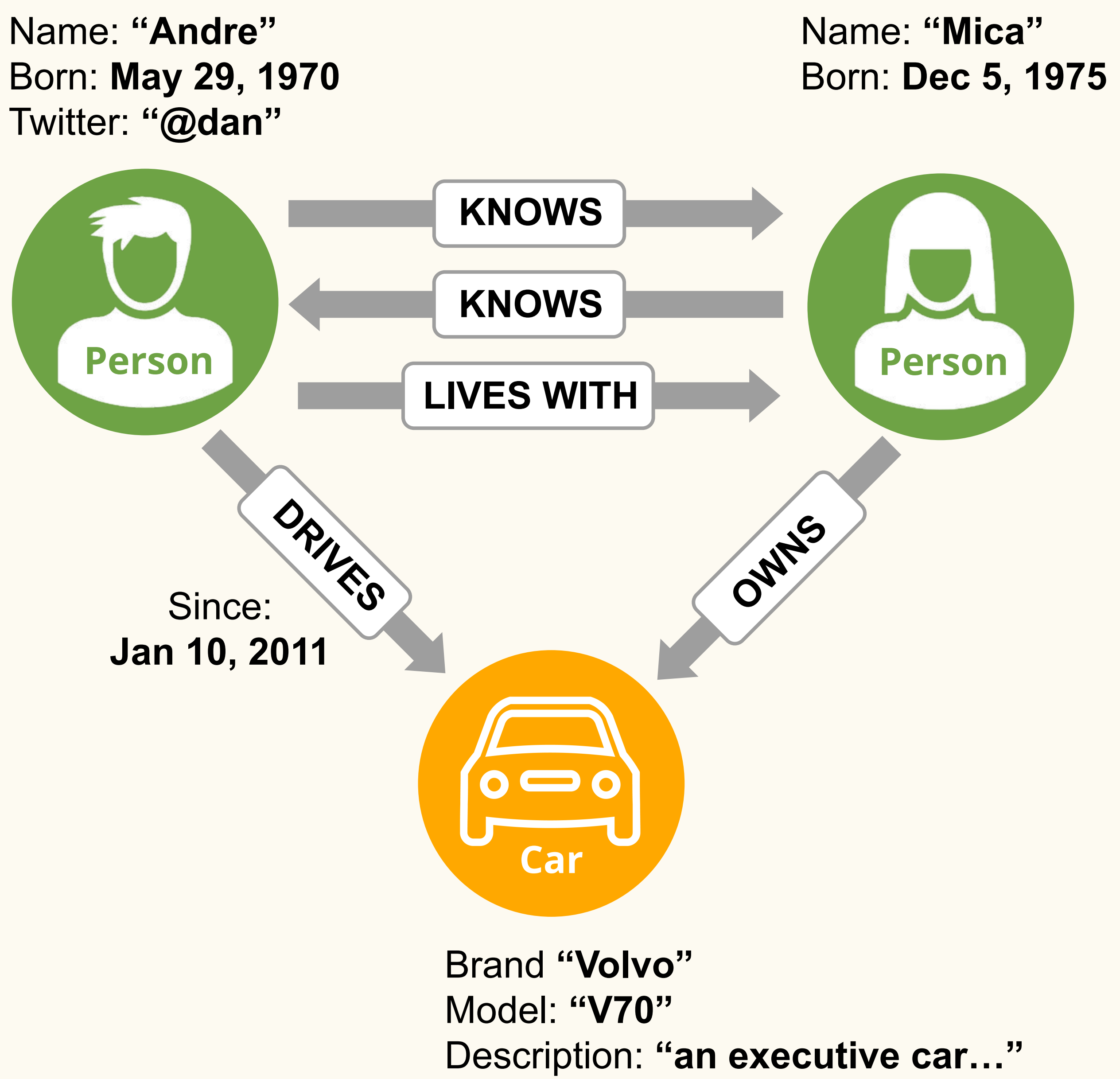
Knowledge Graph = design patterns to organize & access interrelated data

Property Graph Data Model

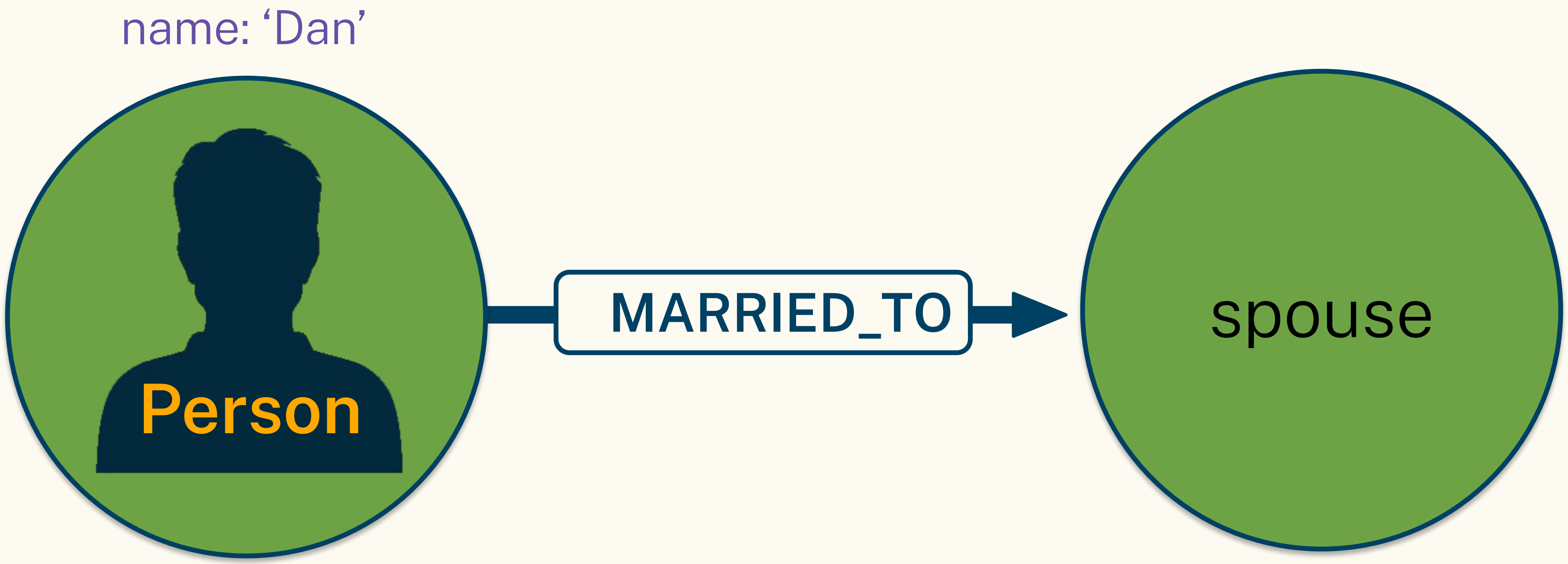
Nodes represent entities in the graph

Relationships represent associations or interactions between nodes

Properties represent attributes of nodes or relationships



Cypher: A Powerful & Expressive Query Language



NODE | RELATIONSHIP TYPE | NODE

```
MATCH (p:Person {name:“Dan”} ) -[:MARRIED_TO]-> (spouse)
```

VARIABLE | LABEL | PROPERTY | VARIABLE

```
RETURN p.name as husband, spouse
```


neo4j graphsummit

Module 1
Go to Notebook

Neo4j Graph Data Science (GDS)

The Largest Catalog of Graph Algorithms

Pathfinding & Search

Centrality

Community Detection

Machine Learning

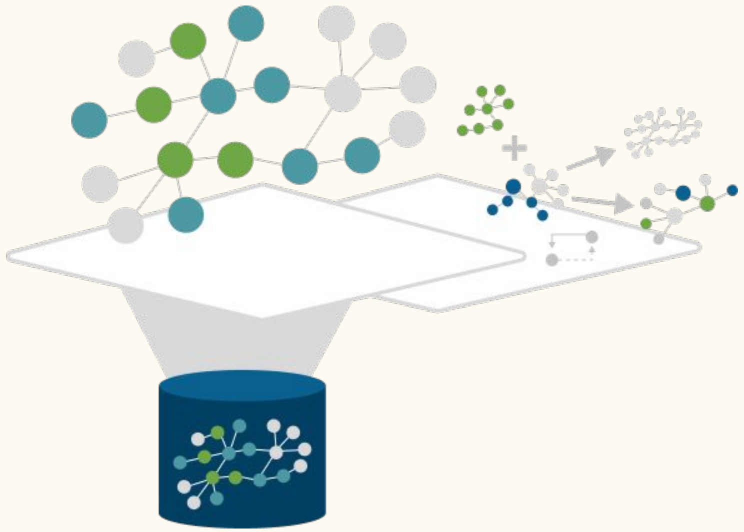
Link Prediction

Similarity

and more ...

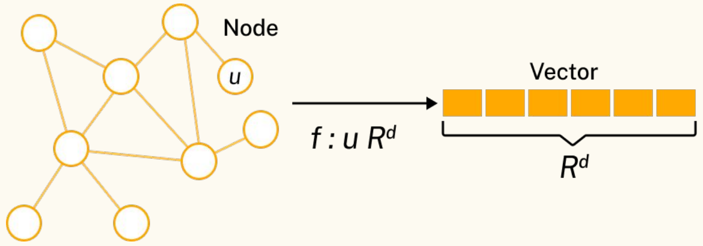
Over 65 pretuned, parallelized algorithms.

Native Graph Catalog and Analytics Workspace



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

Graph Embeddings for ML &AI



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

50+ Graph Algorithms in Neo4j

Pathfinding & Search

- Shortest Path
- Single-Source Shortest Path
- All Pairs Shortest Path
- A* Shortest Path
- Yen’s K Shortest Path
- Minimum Weight Spanning Tree
- K-Spanning Tree (MST)
- Random Walk
- Breadth & Depth First Search

Centrality / Importance

- Degree Centrality
- Closeness Centrality
- Harmonic Centrality
- Betweenness Centrality & Approx.
- PageRank
- Personalized PageRank
- ArticleRank
- Eigenvector Centrality

Community Detection

- Triangle Count
- Local Clustering Coefficient
- Connected Components (Union Find)
- Strongly Connected Components
- Label Propagation
- Louvain Modularity
- K-1 Coloring
- Modularity Optimization

Link Prediction

- Adamic Adar
- Common Neighbors
- Preferential Attachment
- Resource Allocations
- Same Community
- Total Neighbors

Similarity

- Euclidean Distance
- Cosine Similarity
- Node Similarity (Jaccard)
- Overlap Similarity
- Pearson Similarity
- Approximate KNN

Embeddings

- Node2Vec
- Random Projections
- GraphSAGE

… Auxiliary Functions:

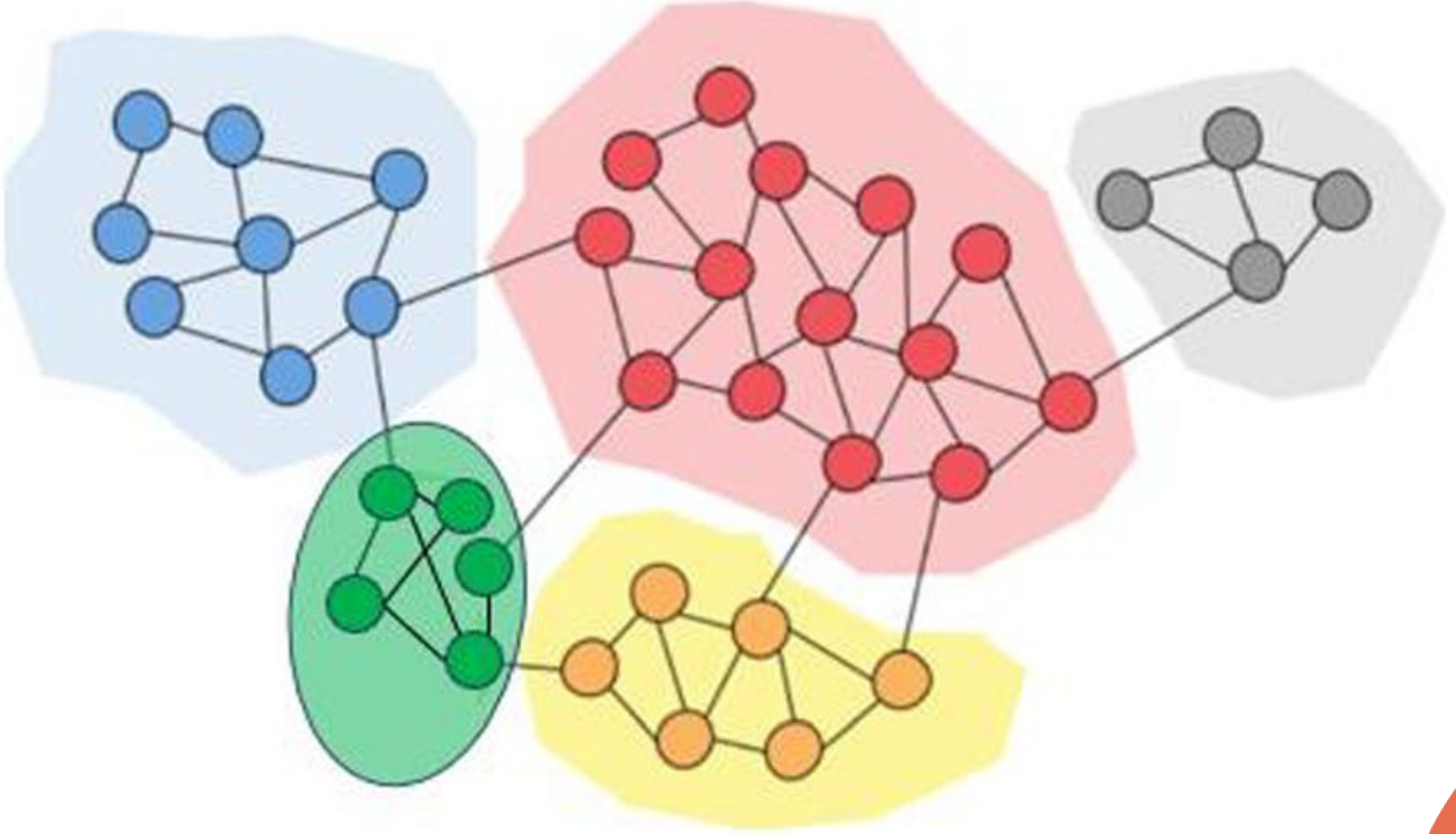
- Random graph generation
- Graph export
- One hot encoding
- Distributions & metrics

Today we will explore Community Detection

Evaluate how groups of nodes may be clustered or partitioned in a graph.

Community id properties assigned to node based on relationship structure.

Useful for:
- Segmentation
- Clustering
- Entity resolution
- Summarization (for AI)



Knowledge Graphs – New & Improved!

NOW WITH VECTORS!

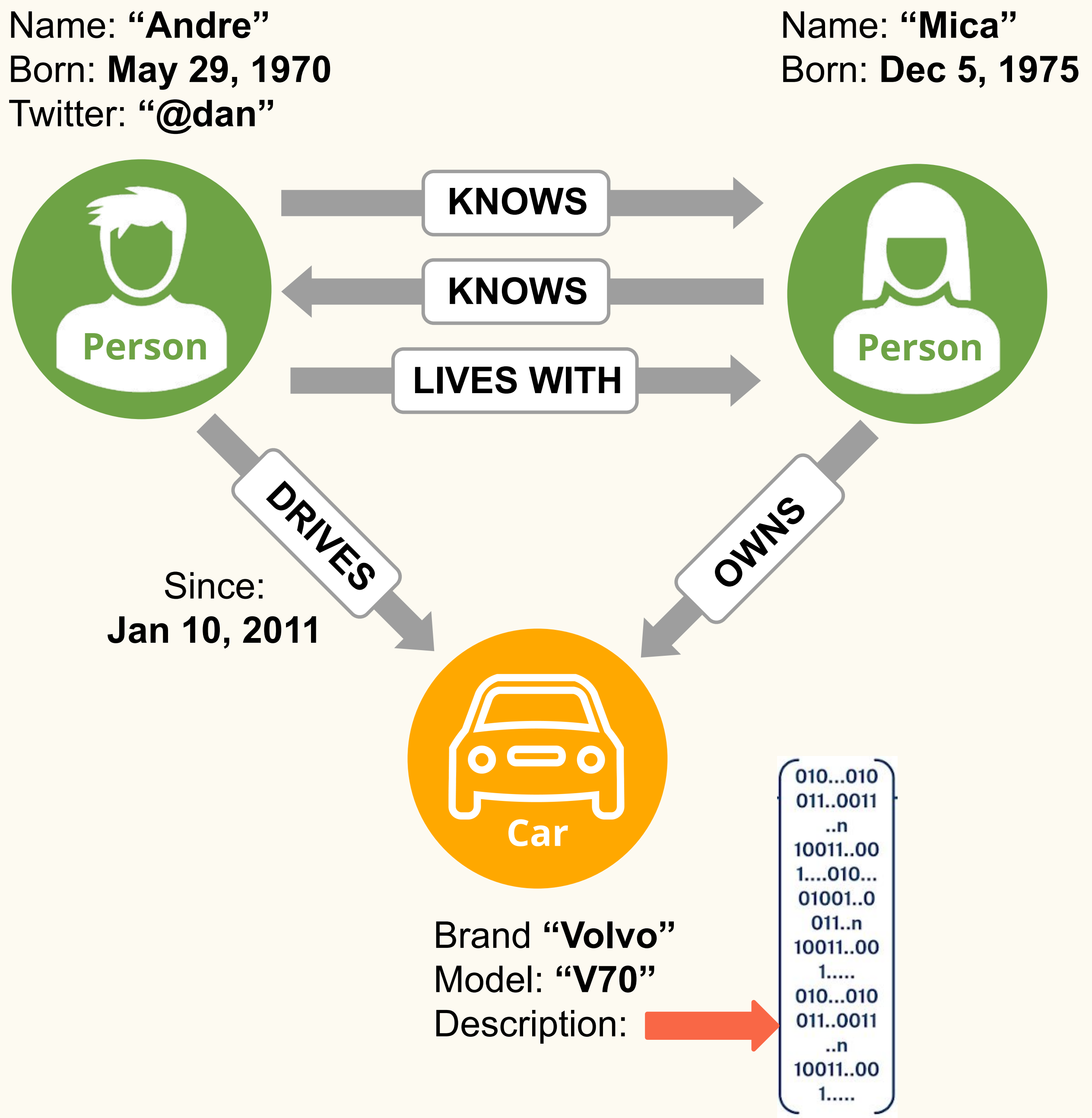
Knowledge Graph = design patterns to organize & access interrelated data

Property Graph Data Model

Nodes represent entities in the graph

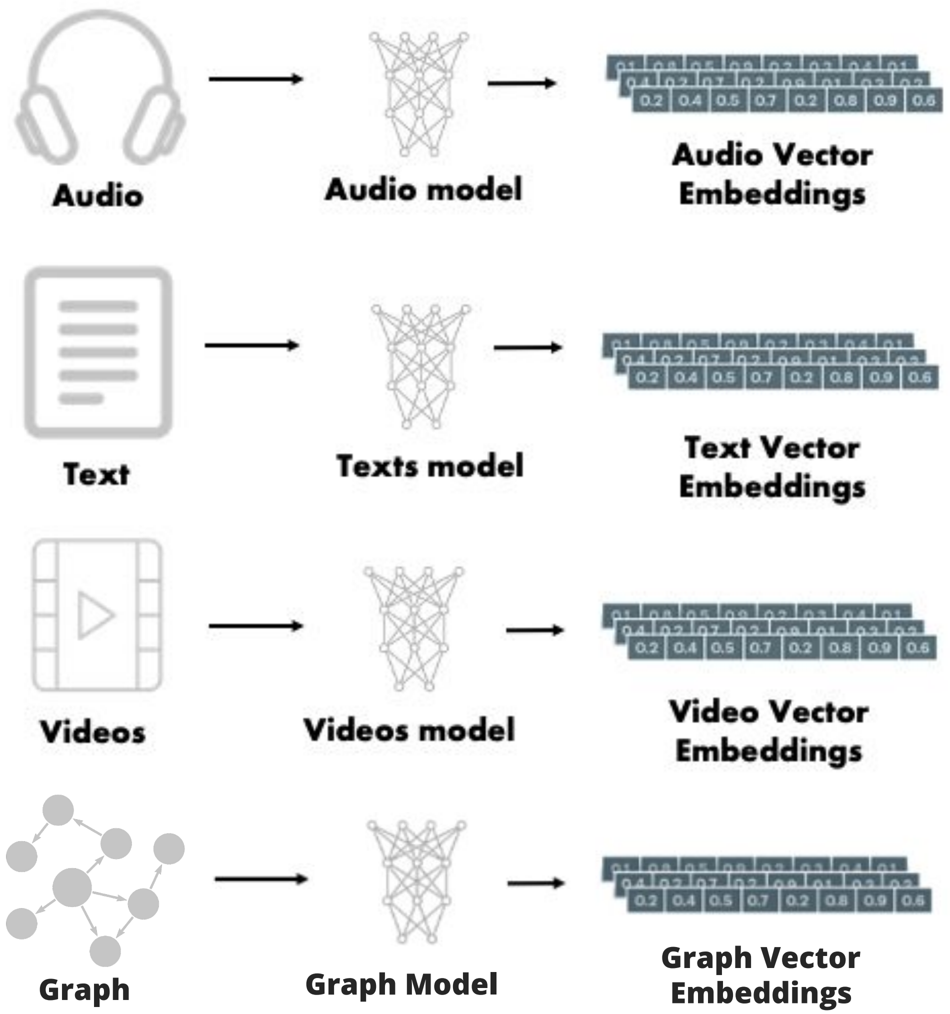
Relationships represent associations or interactions between nodes

Properties represent attributes of nodes or relationships



What are Embeddings?

- A type of data compression
- Transform messy data (text, images, audio, etc.) into a compact format for ML algorithms
- Most often numeric **vectors** (a.k.a arrays) with 100s or 1000s of elements
- Preserve information such that ***"similar" items have proportionally "similar" embedding vectors***
- Similarity is measured with vector algorithms (cosine, euclidean, etc.)



Search & Vectors in Neo4j

Neo4j makes search efficient through a variety of indexes including

- **Range:** General index for predicates based on equality and range.
- **Point:** Predicts on geospatial points like distance bounding boxes, etc.
- **Text:** Predicates on strings like contains, ends with, etc.
- **FullText:** Text search based on tokenization and analyzers
- ***Vector: ANN (Approximate Nearest Neighbor) search on vectors***

How is similarity measured

What would you use for openai embedding models?

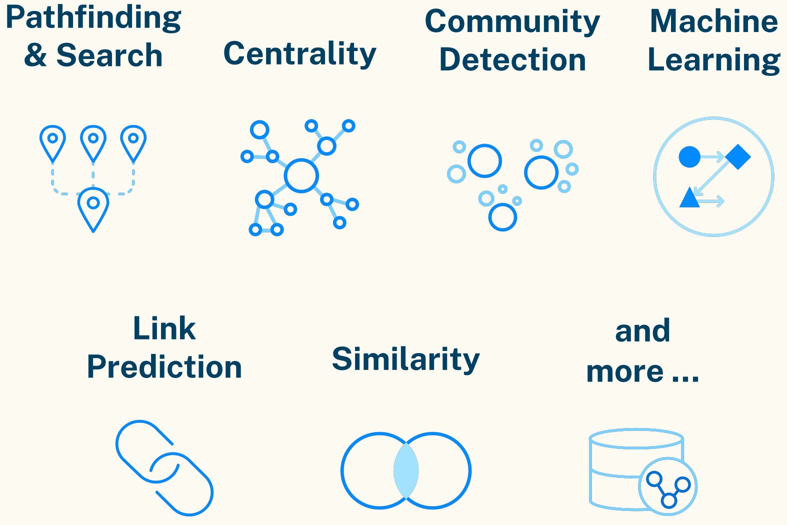
For OpenAI embedding models, **cosine similarity** is the preferred choice. Here's why:

Why Cosine Similarity?

1. **OpenAI Embeddings Are Normalized**
   - OpenAI embedding models (like `text-embedding-ada-002`) generate **normalized vectors**, meaning each vector has a unit length (i.e., a magnitude of 1).
   - In such cases, **cosine similarity and Euclidean distance are proportional**, but cosine similarity is computationally more efficient.
2. **Focus on Semantic Meaning**
   - Since cosine similarity measures the **angle between vectors**, it captures semantic similarity between words, sentences, or documents effectively.
   - Euclidean distance, on the other hand, is sensitive to magnitude differences, which aren't meaningful in normalized embeddings.
3. **Industry Best Practice**
   - Most applications using OpenAI embeddings (e.g., search ranking, recommendation systems, and semantic clustering) rely on cosine similarity.
   - OpenAI's own documentation recommends using **cosine similarity or dot product** over Euclidean distance.

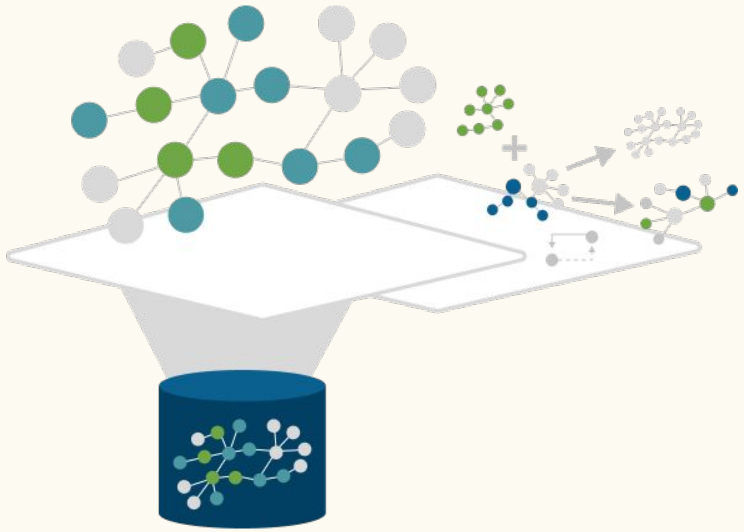
Neo4j Graph Data Science (GDS)

The Largest Catalog of Graph Algorithms



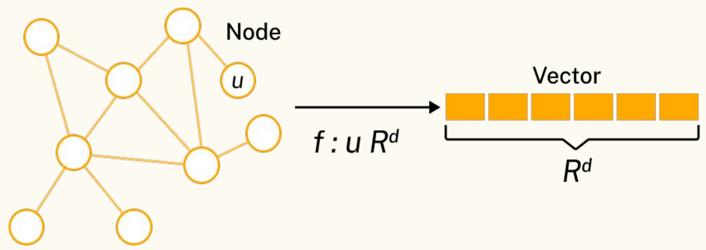
Over 65 pretuned, parallelized algorithms.

Native Graph Catalog and Analytics Workspace



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

Graph Embeddings for ML &AI



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

50+ Graph Algorithms in Neo4j

Pathfinding & Search

- Shortest Path
- Single-Source Shortest Path
- All Pairs Shortest Path
- A* Shortest Path
- Yen's K Shortest Path
- Minimum Weight Spanning Tree
- K-Spanning Tree (MST)
- Random Walk
- Breadth & Depth First Search

Centrality / Importance

- Degree Centrality
- Closeness Centrality
- Harmonic Centrality
- Betweenness Centrality & Approx.
- PageRank
- Personalized PageRank
- ArticleRank
- Eigenvector Centrality

Community Detection

- Triangle Count
- Local Clustering Coefficient
- Connected Components (Union Find)
- Strongly Connected Components
- Label Propagation
- Louvain Modularity
- K-1 Coloring
- Modularity Optimization

Link Prediction

- Adamic Adar
- Common Neighbors
- Preferential Attachment
- Resource Allocations
- Same Community
- Total Neighbors

Similarity

- Euclidean Distance
- Cosine Similarity
- Node Similarity (Jaccard)
- Overlap Similarity
- Pearson Similarity
- Approximate KNN

Embeddings

- Node2Vec
- Random Projections
- GraphSAGE

... Auxiliary Functions:

- Random graph generation
- Graph export
- One hot encoding
- Distributions & metrics

Today we will explore

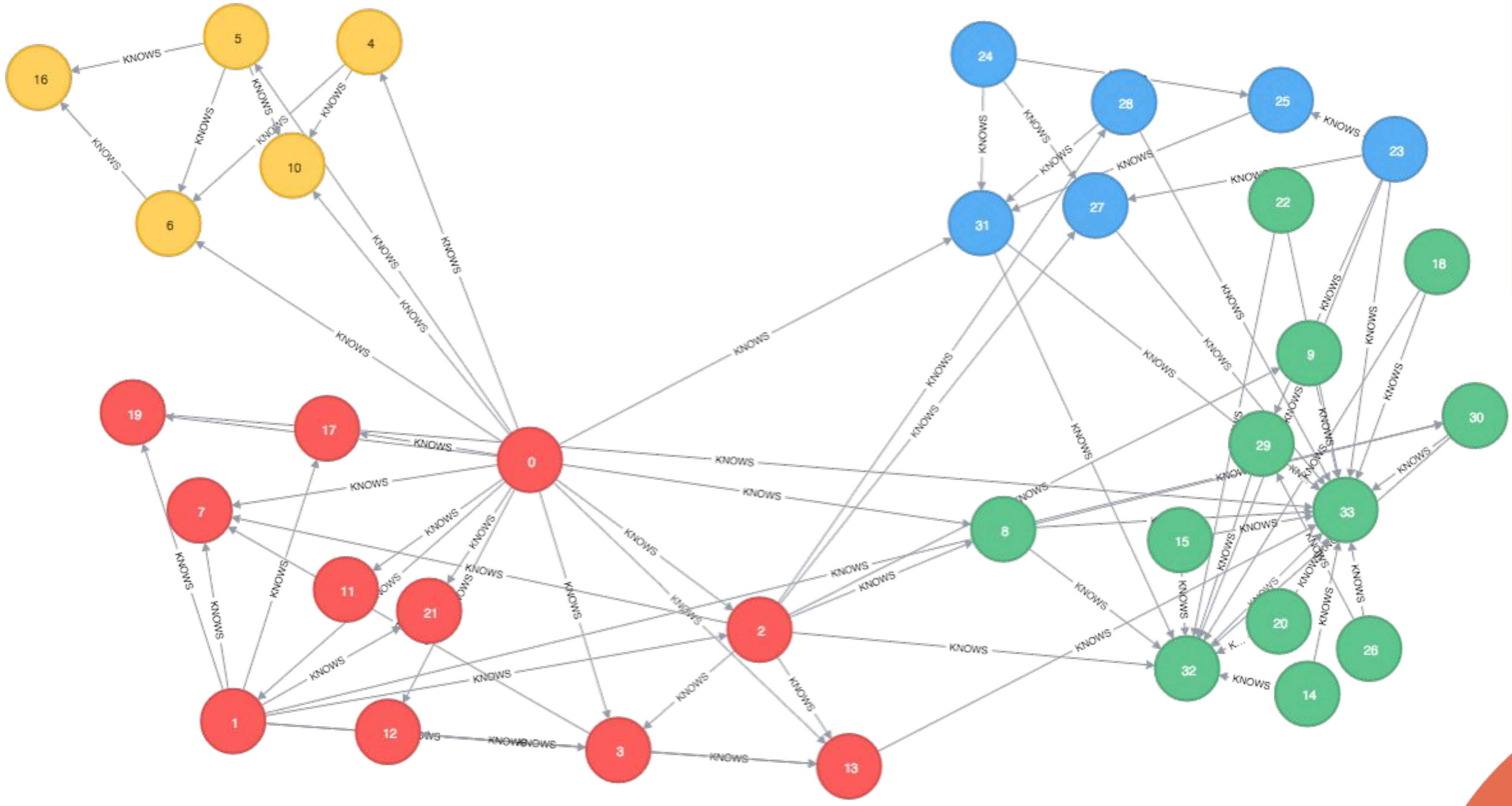
Community Detection

Evaluate how groups of nodes may be clustered or partitioned in a graph.

Community id properties assigned to node based on relationship structure.

Useful for:

- Segmentation
- Clustering
- Entity resolution
- Summarization (for AI)



neo4j graphsummit

Module 2

Taming Unstructured Data

Module 2

Unstructured Data

- **Creating a graph from Unstructured Data**
- **Entity Extraction using Domain Model + LLM**

neo4j graphsummit

Module 2
Go to Notebook

graphsummit

From Unstructured to Structured: An example...

RFP Generation GenAI App

Why a KG Matters in RFP GenAI App?

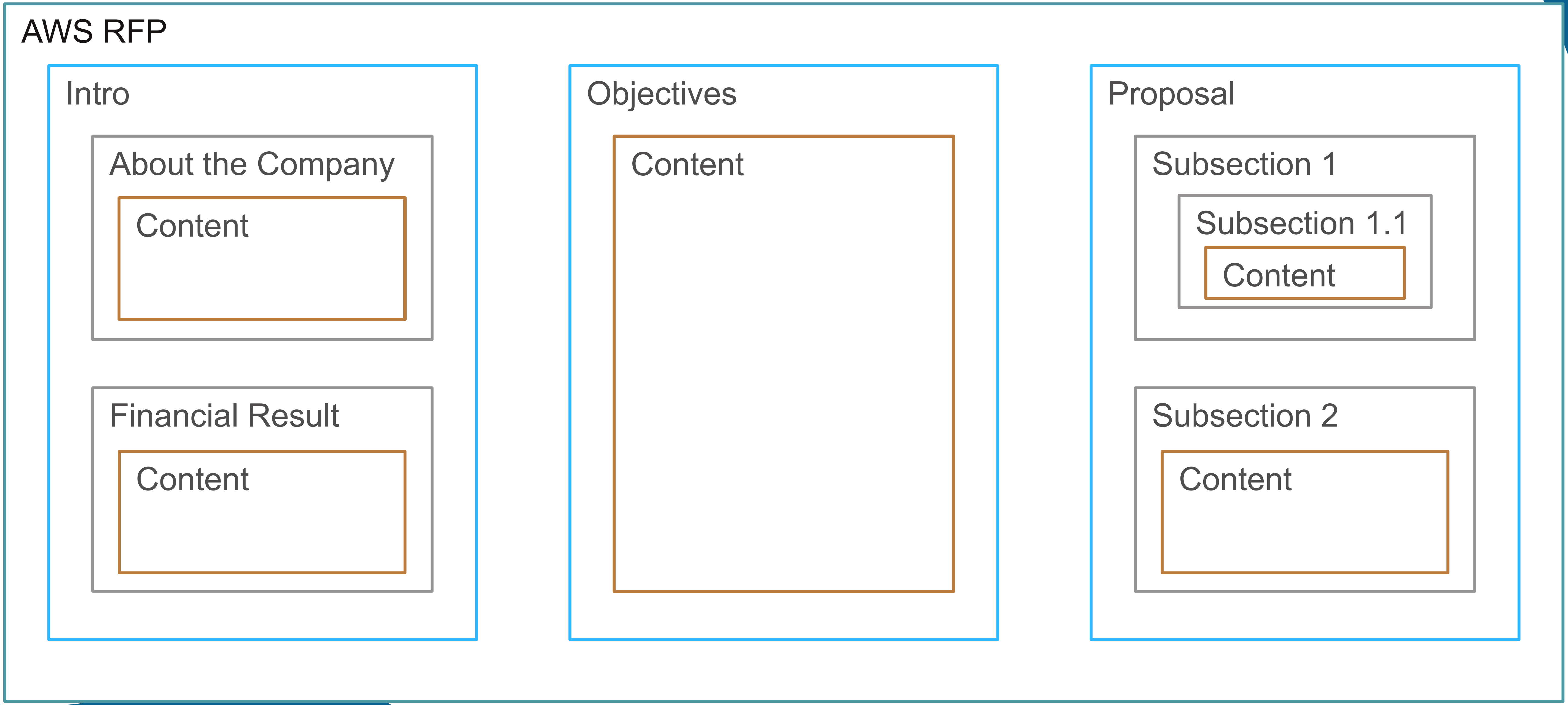
Challenges

- **Time consuming** to read previous RFP across **multiple repositories**
- **Repetitive** and **manual tasks** to synthesise the content
- **Non-standard structure** of RFP making it difficult to do data modelling

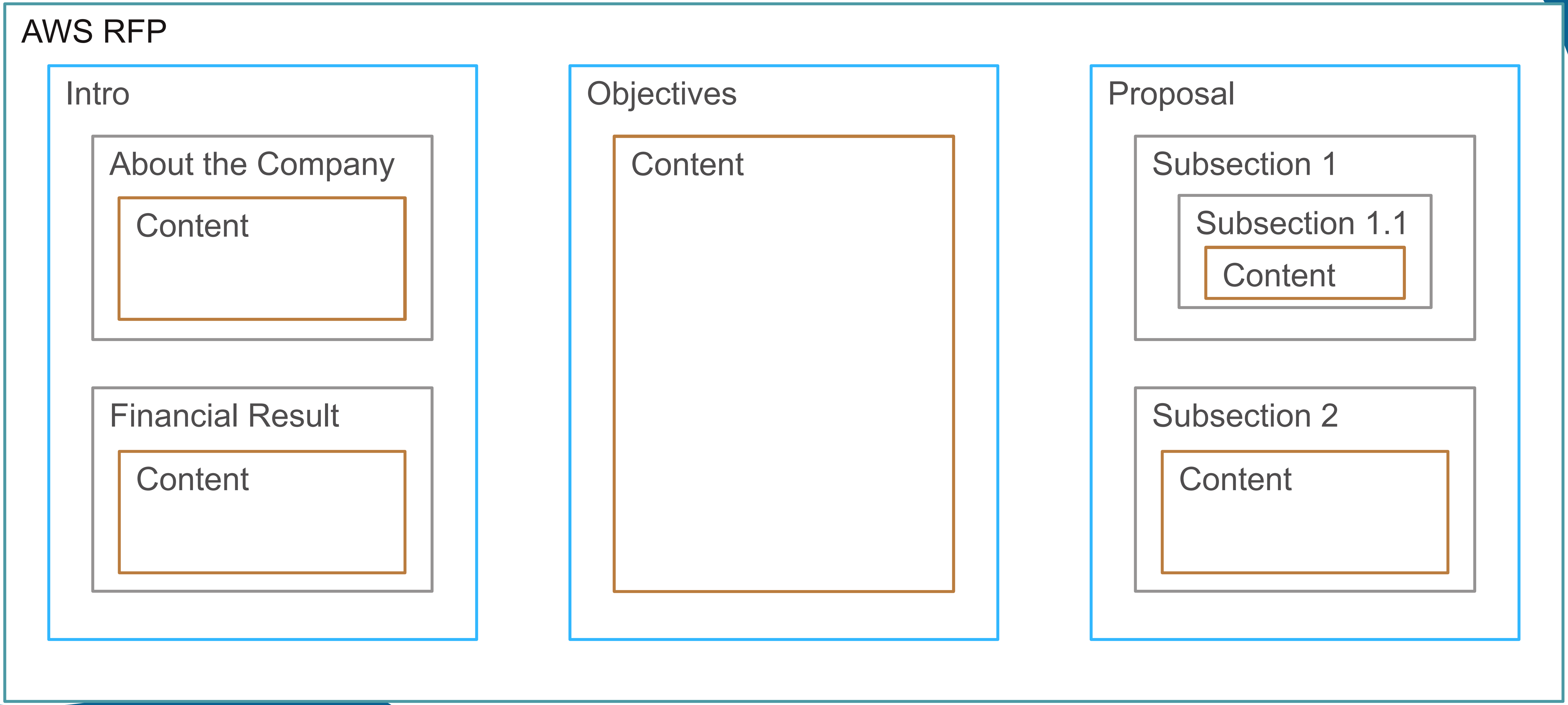
Outcomes

- **Knowledge base** to collect, store and retrieve **domain-specific** information
- **Drive efficient, accurate, contextual** and **explainable** way to streamline RFP responses
- **Flexible storage** that's adoptable to the varying structure of an RFP

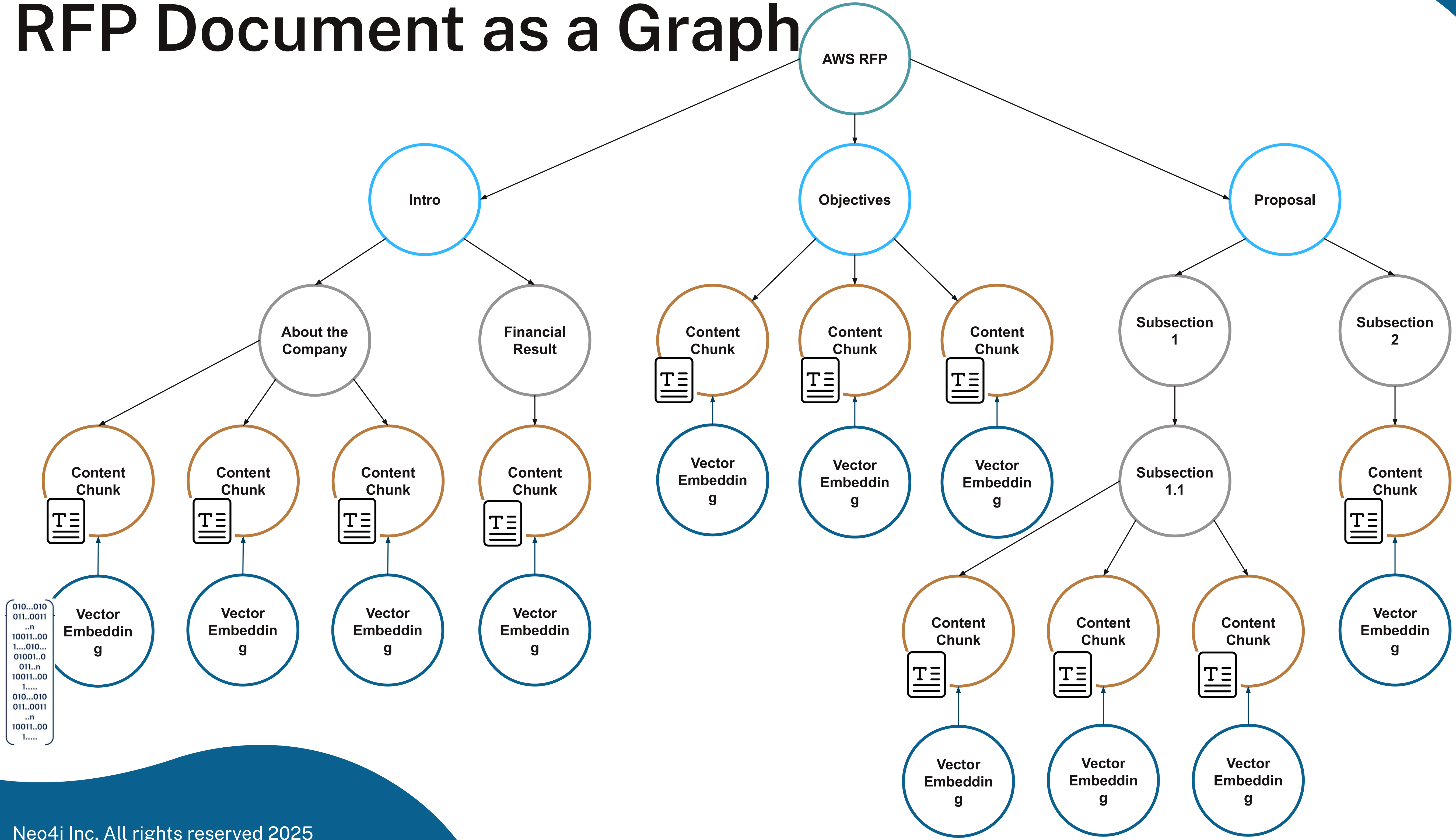
Anatomy of an RFP Document



Anatomy of a Document



RFP Document as a Graph



Why Neo4j KG Matters in RFP GenAI App?

Challenges

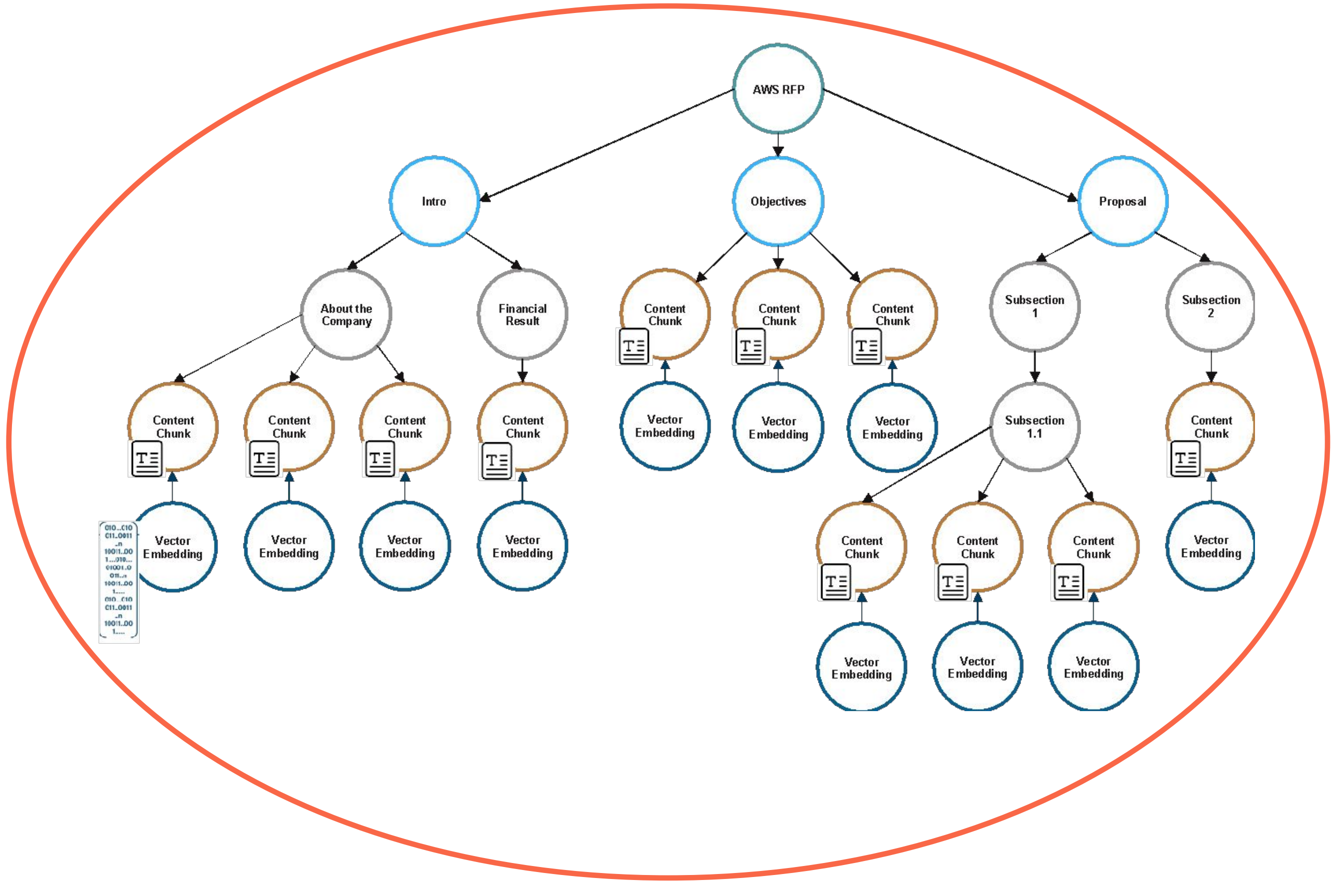
- **Time consuming** to read previous RFP across **multiple repositories**
- **Repetitive** and **manual tasks** to synthesise the content
- **Non-standard structure** of RFP making it difficult to do data modelling

Outcomes

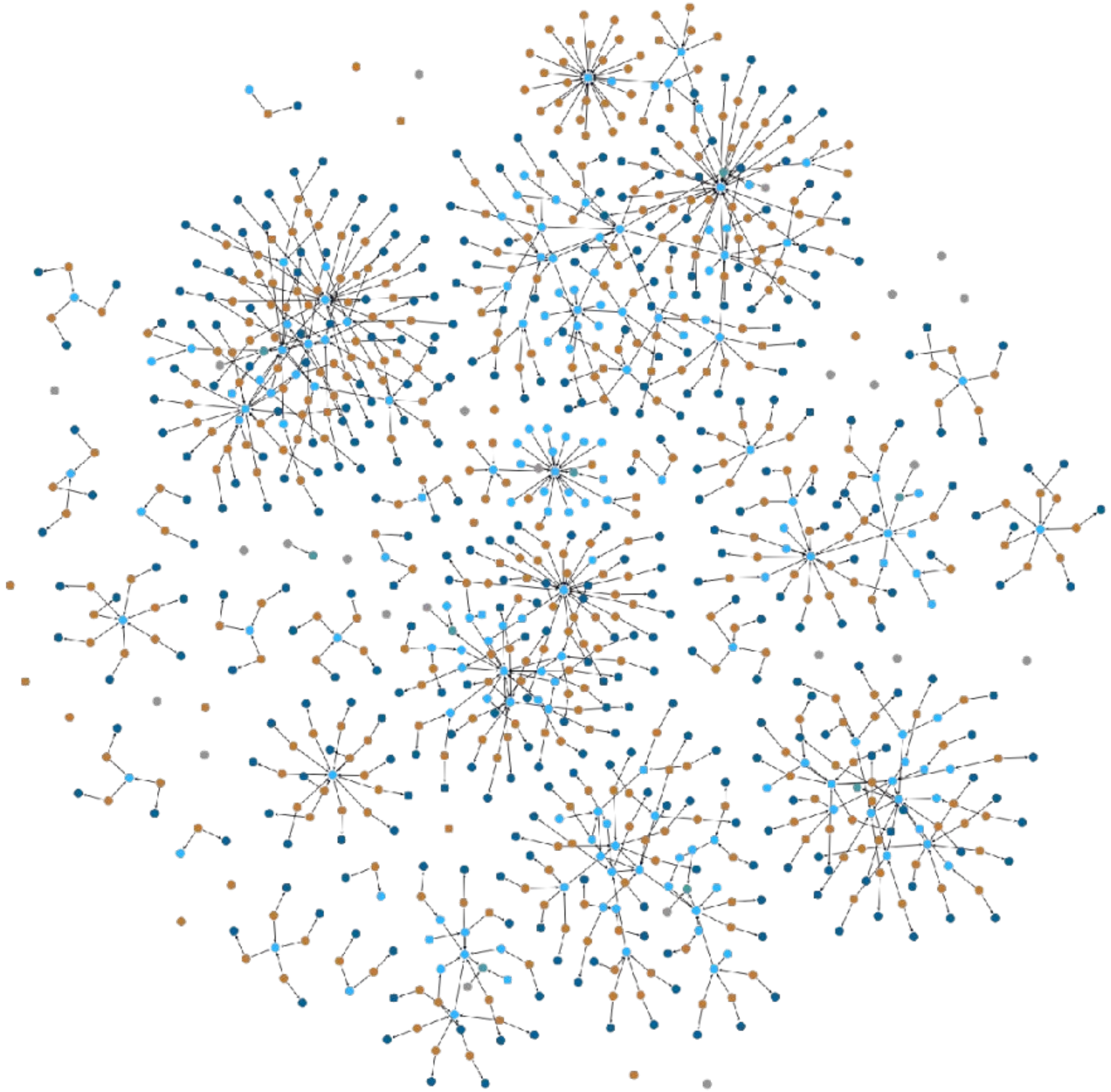
- **Knowledge base** to collect, store and retrieve **domain-specific** information
- **Drive efficient, accurate, contextual** and **explainable** way to streamline RFP responses
- **Flexible storage** that's adoptable to the varying structure of an RFP

Knowledge Graph as the Knowledge Base

Document in a KG

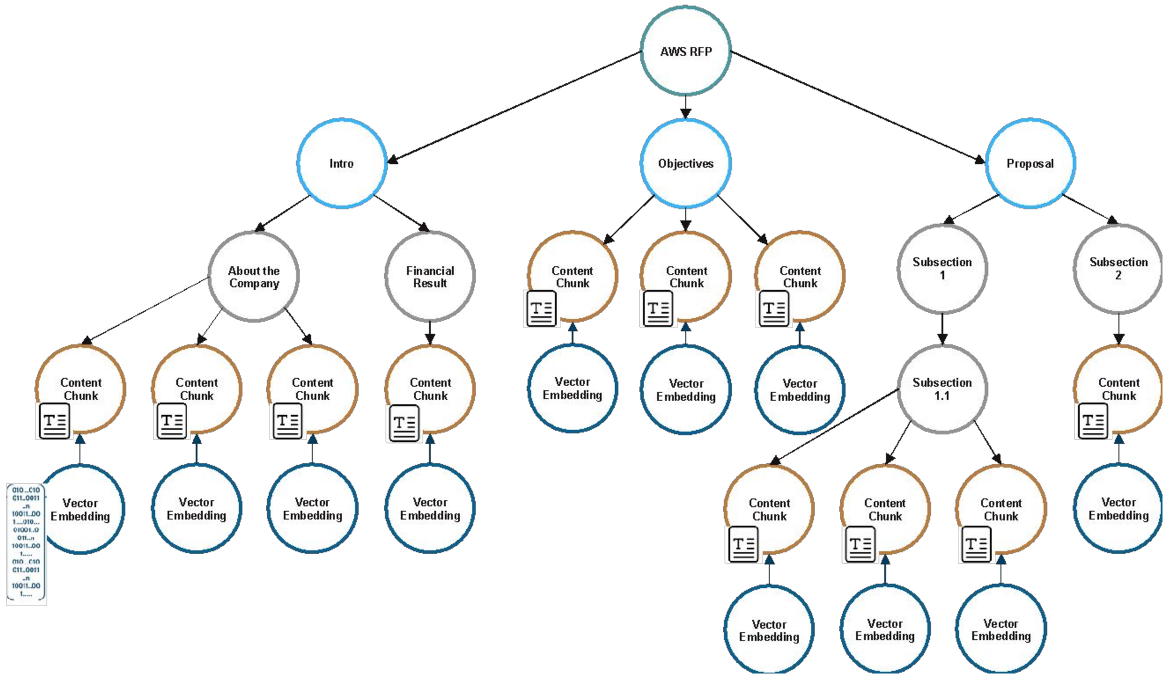


Knowledge Graph

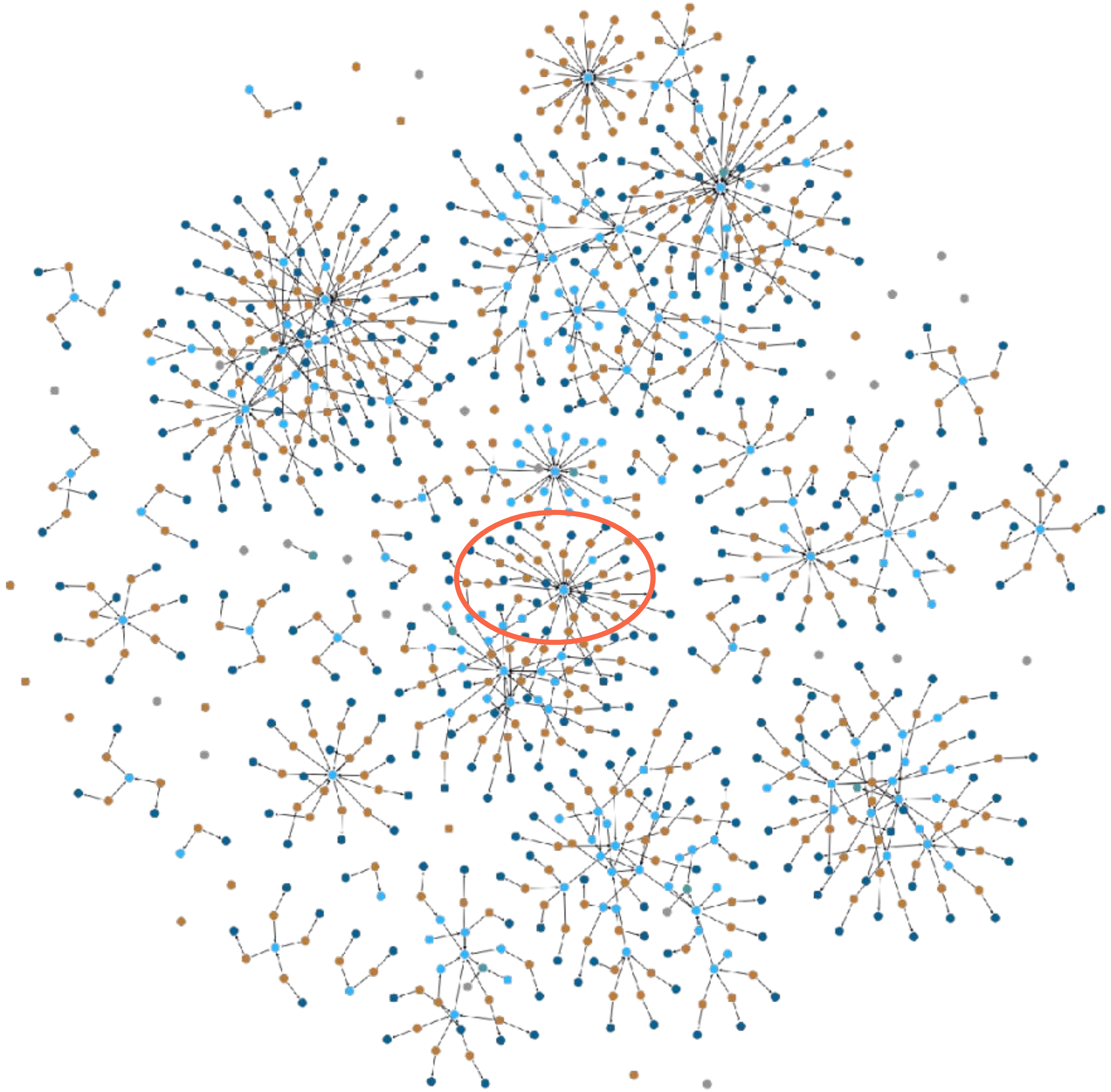


Knowledge Graph as the Knowledge Base

Document in a KG



Knowledge Graph

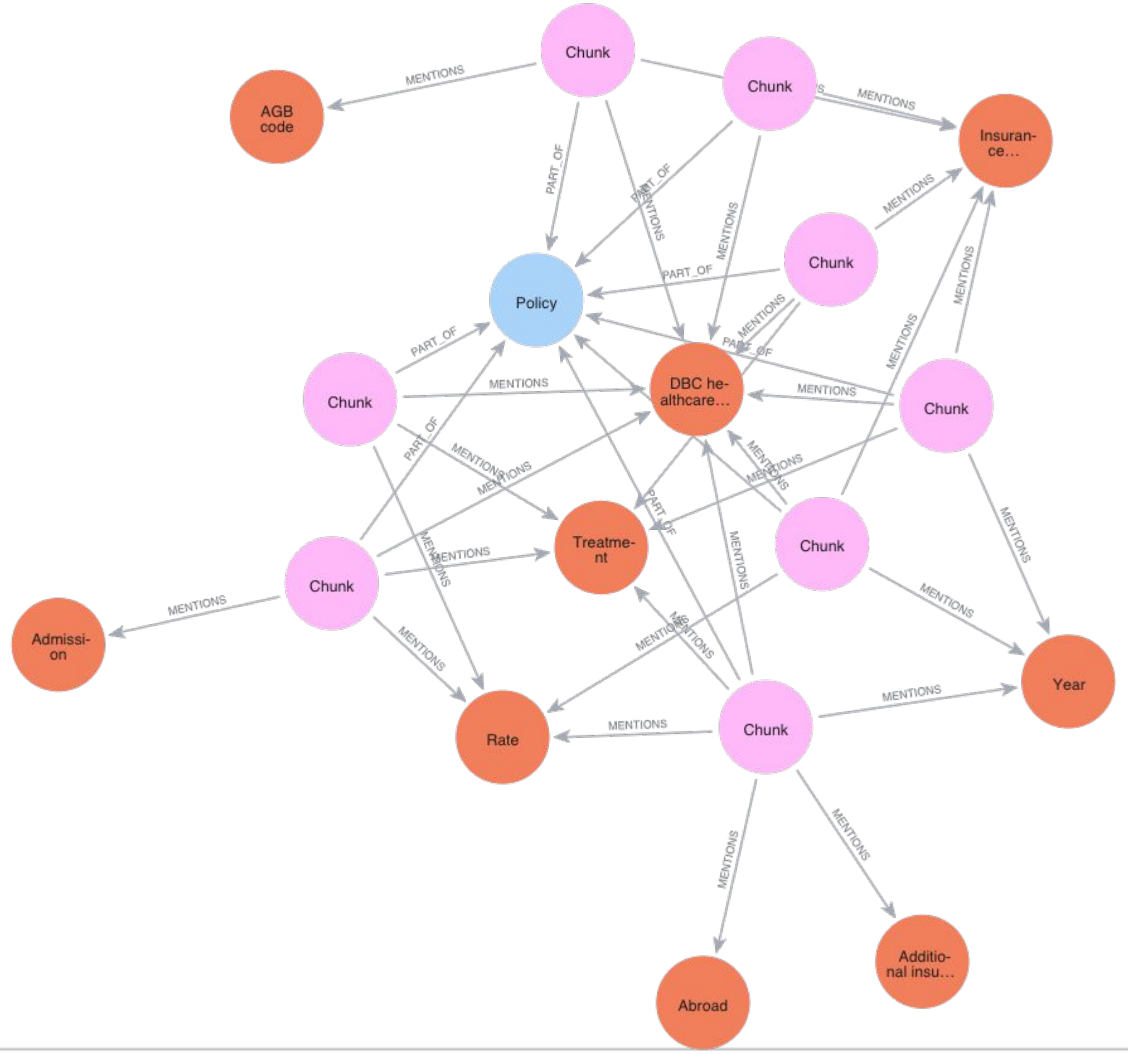


Add Domain Knowledge

- There are references in the document that give more context
- These add more context to the chunks

See also:

- Abroad (definitions)
- DBC healthcare product (definitions)
- Rate (definitions)
- Treatment (definitions)
- Year (definitions)




```
MATCH (c:Chunk)-[:MENTIONS]-(d:Definition)
WHERE c.id in $chunk_ids
WITH DISTINCT d as d
RETURN d.definition as definition, d.description as description
```


neo4j graphsummit

Module 3

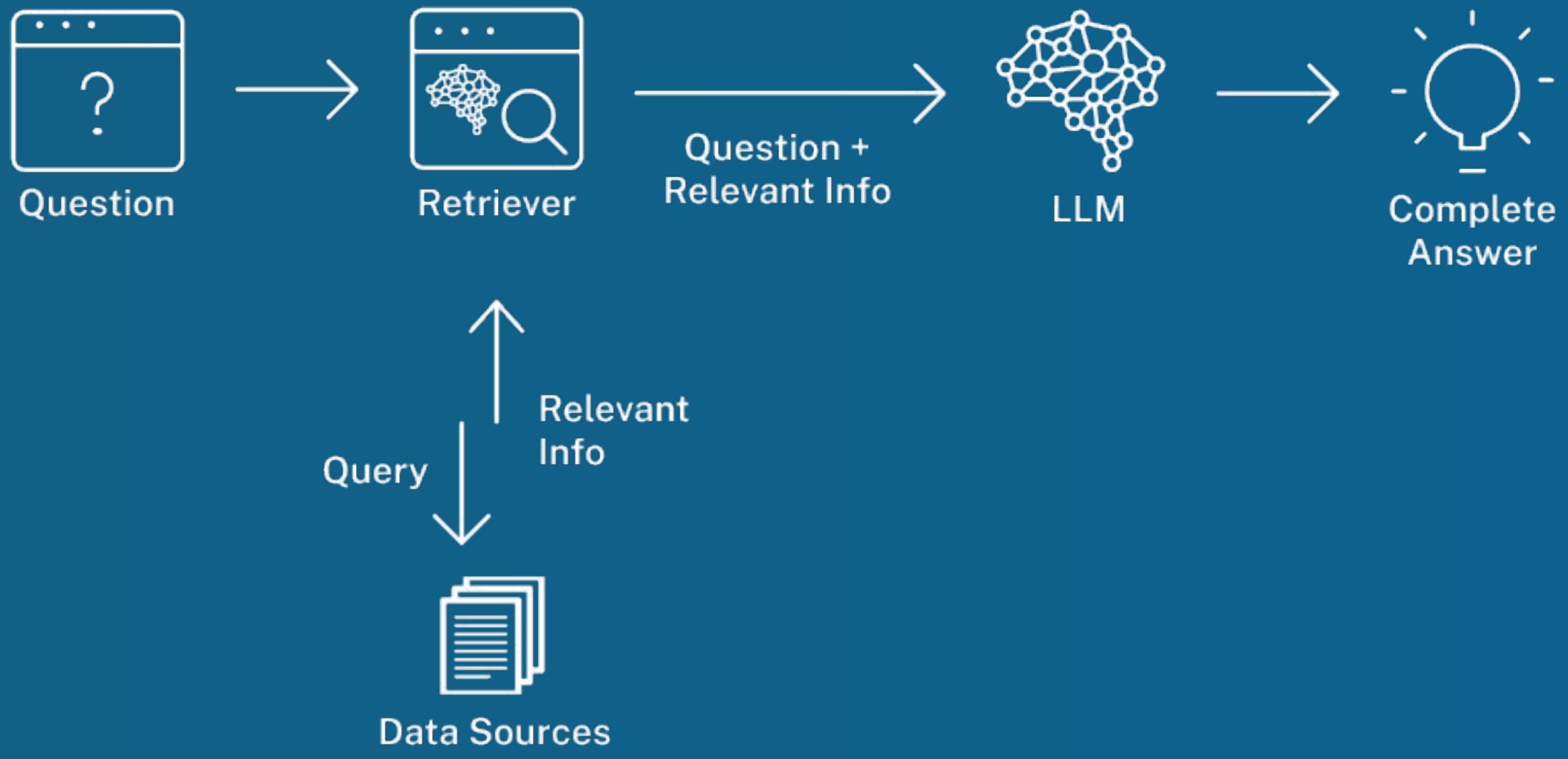
GraphRAG and Agents

Module 3

GraphRAG & Agents

- Experiment with queries for an Agent
- Define Tooling
- Create an agents with the available tools
- Chatbot for an Agent
- Text2Cypher (if we got time)

Ordinary RAG

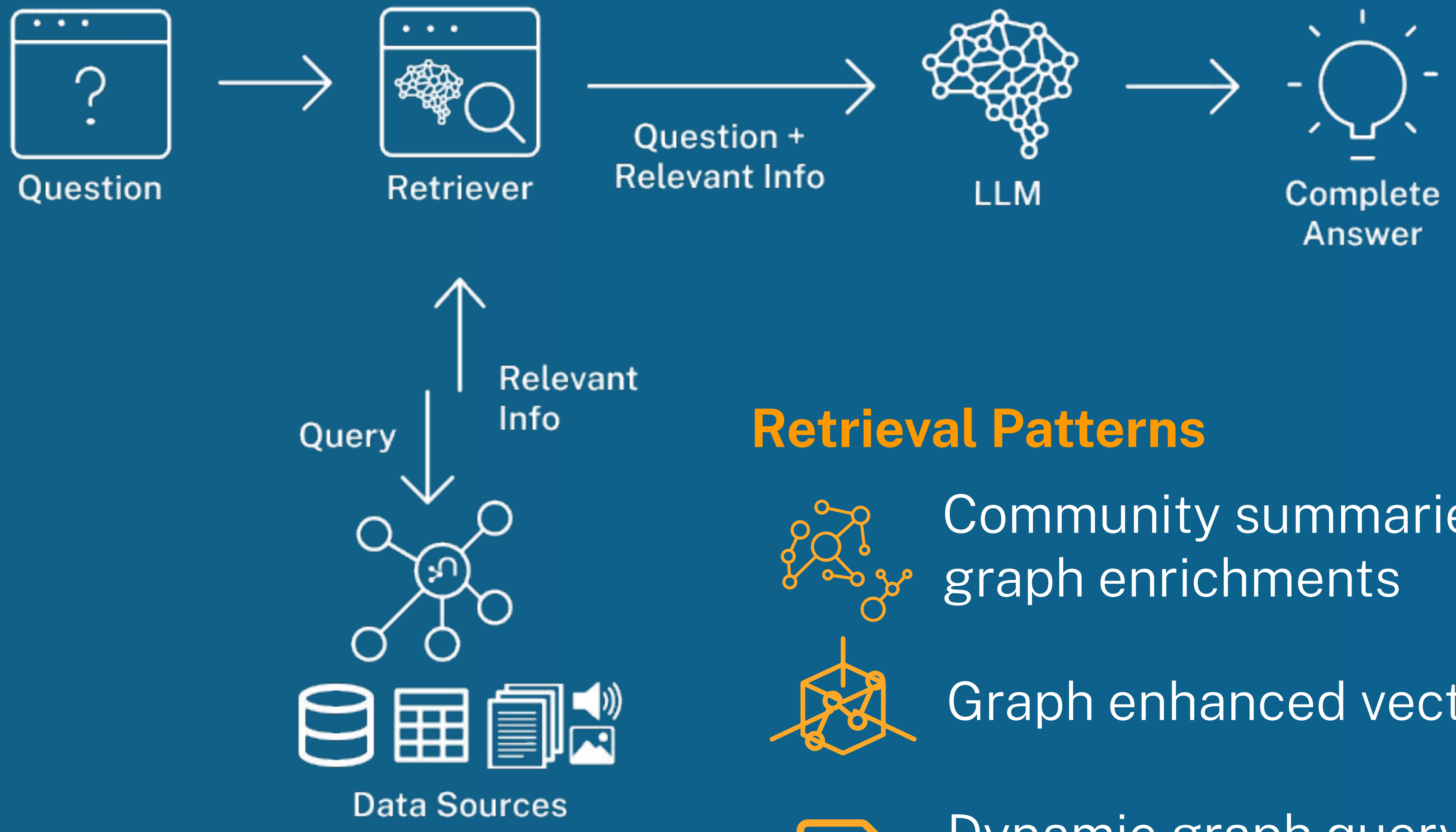


GraphRAG: RAG with graph

It's actually quite broad…

GraphRAG is Retrieval Augmented Generation (RAG) using Knowledge Graphs

Improves GenAI by taking advantage of rich graph data structures



Retrieval Patterns

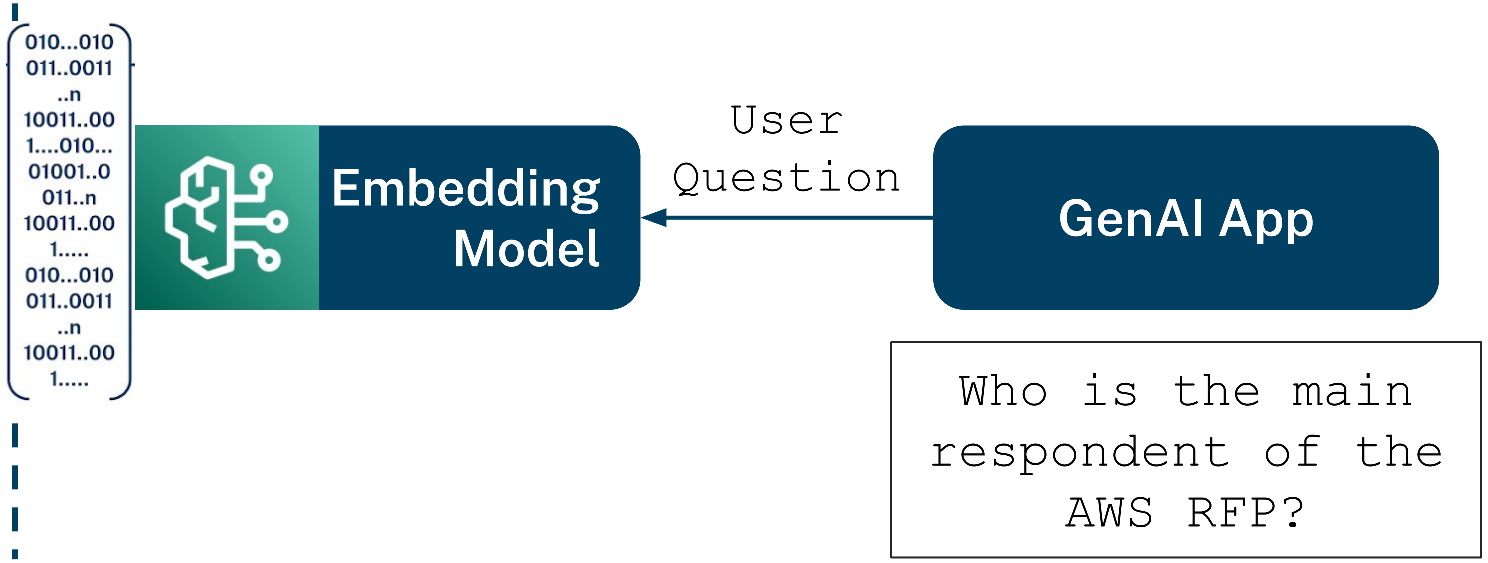
- Community summaries & graph enrichments
- Graph enhanced vector search
- Dynamic graph query generation
- \+ more: graph vectors, parent-child retrievers, etc.

Accurate, Contextual and Explainable

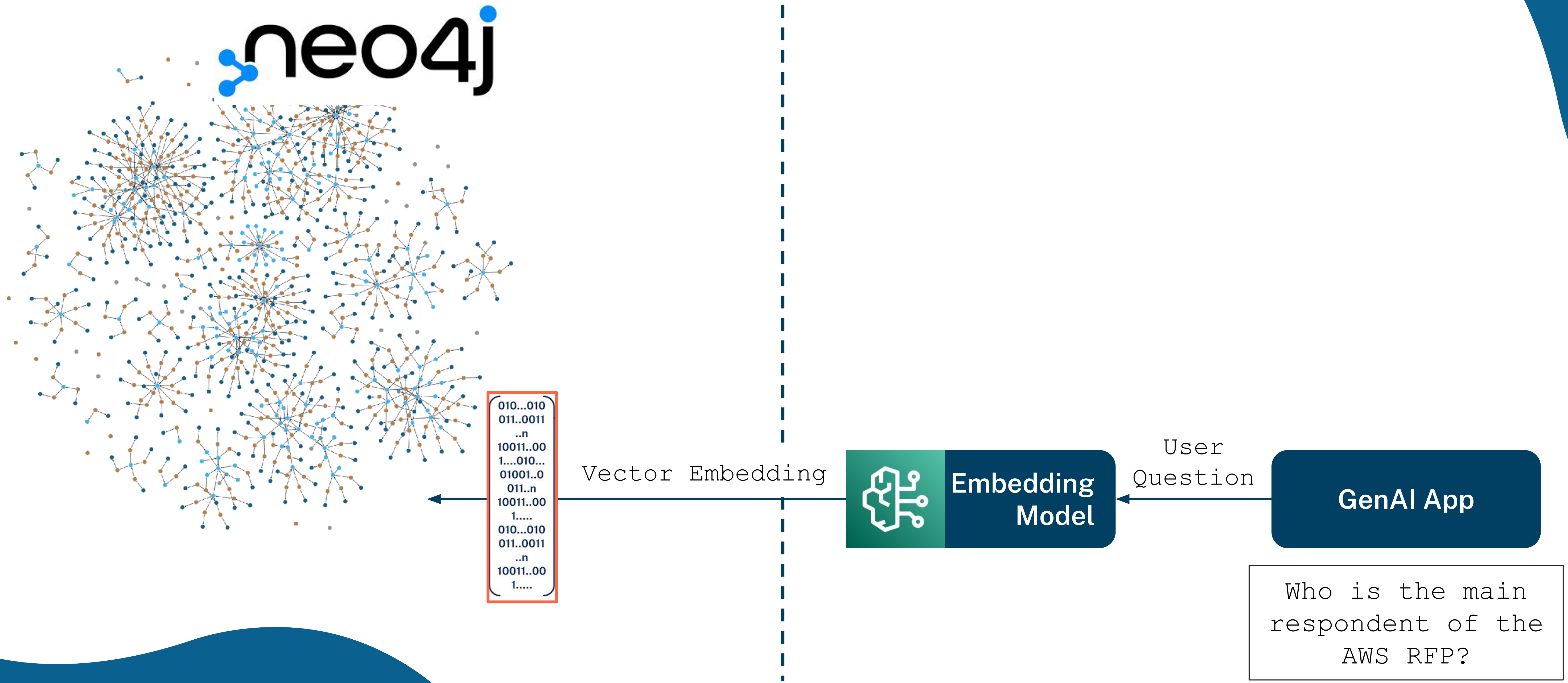
Who is the main respondent of the AWS RFP?

GenAI App

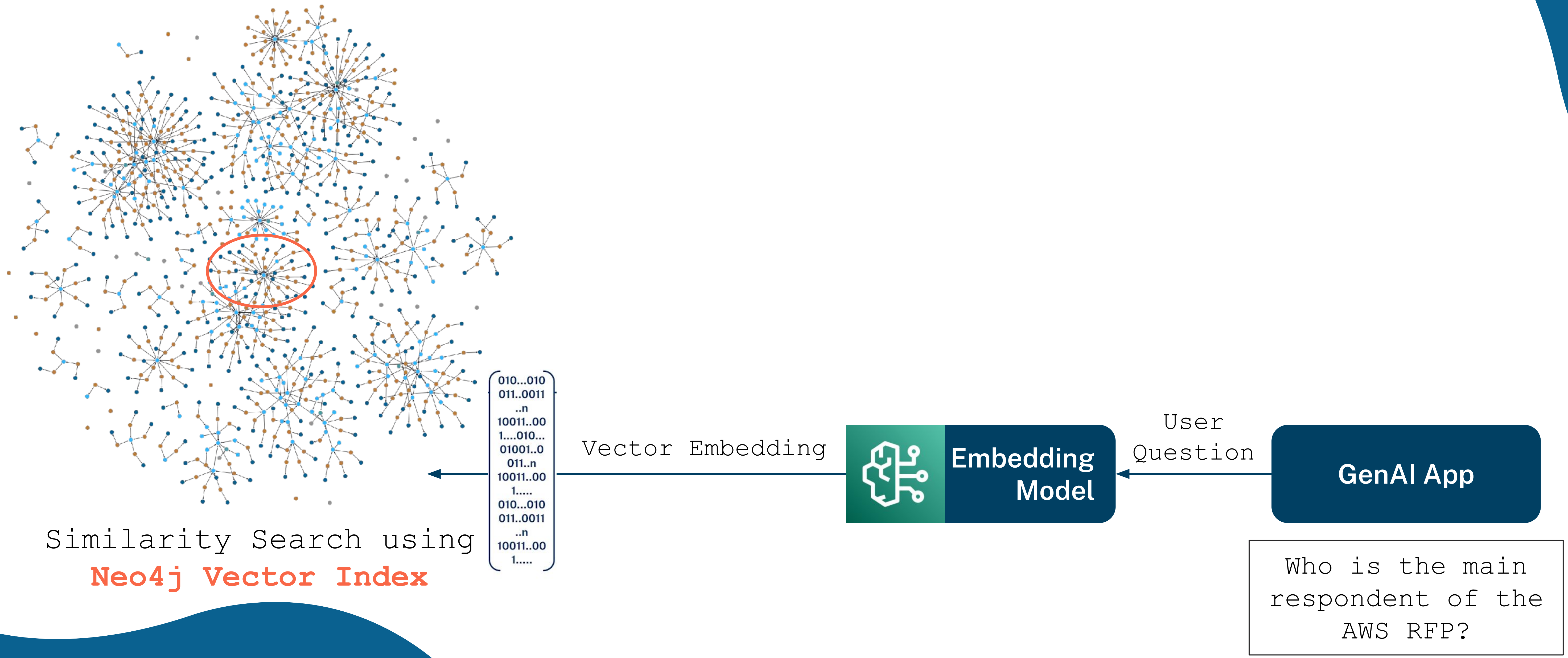
Accurate, Contextual and Explainable



Accurate, Contextual and Explainable

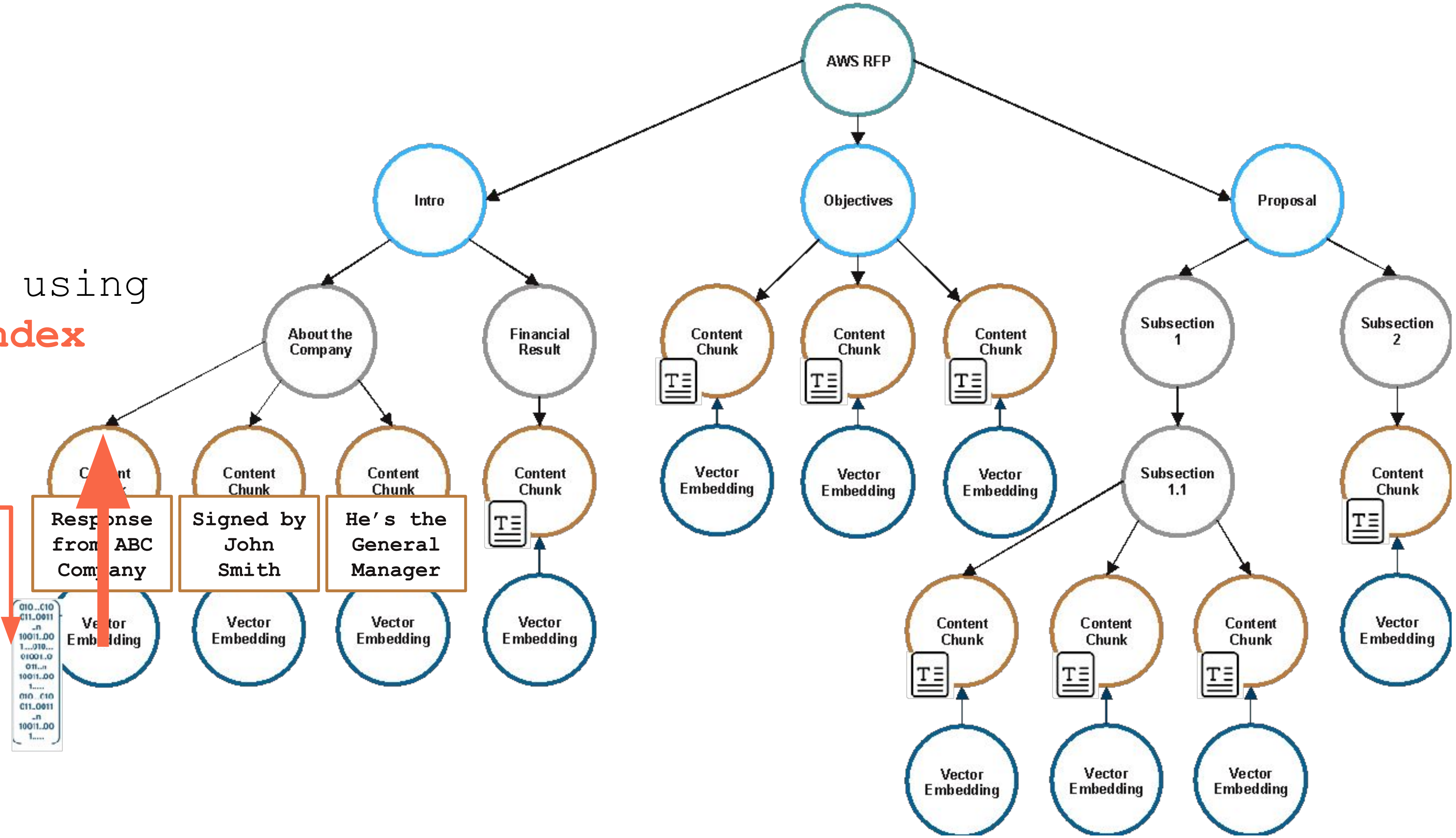


Accurate, Contextual and Explainable



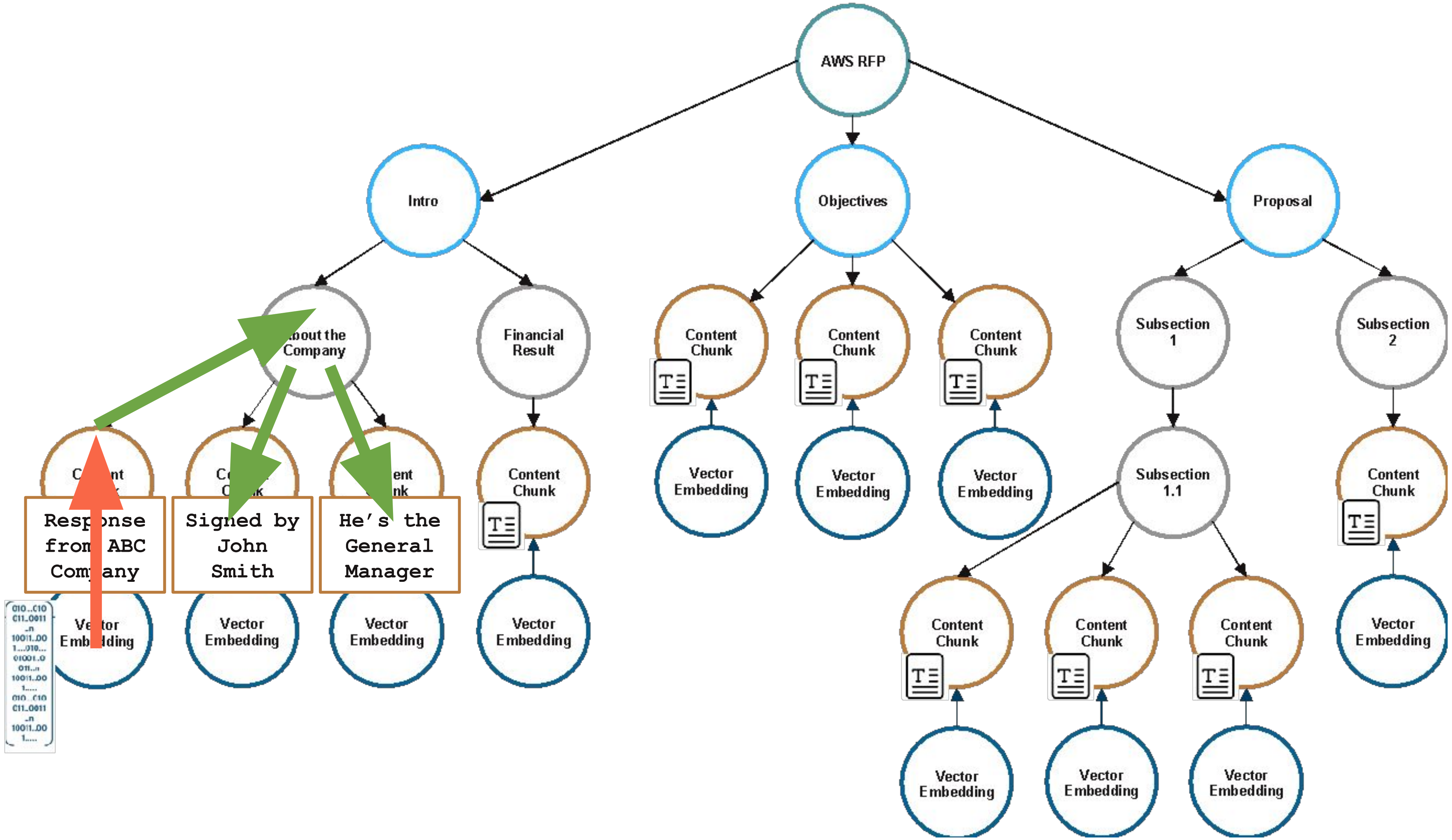
Accurate, Contextual and Explainable

Similarity Search using
Neo4j Vector Index



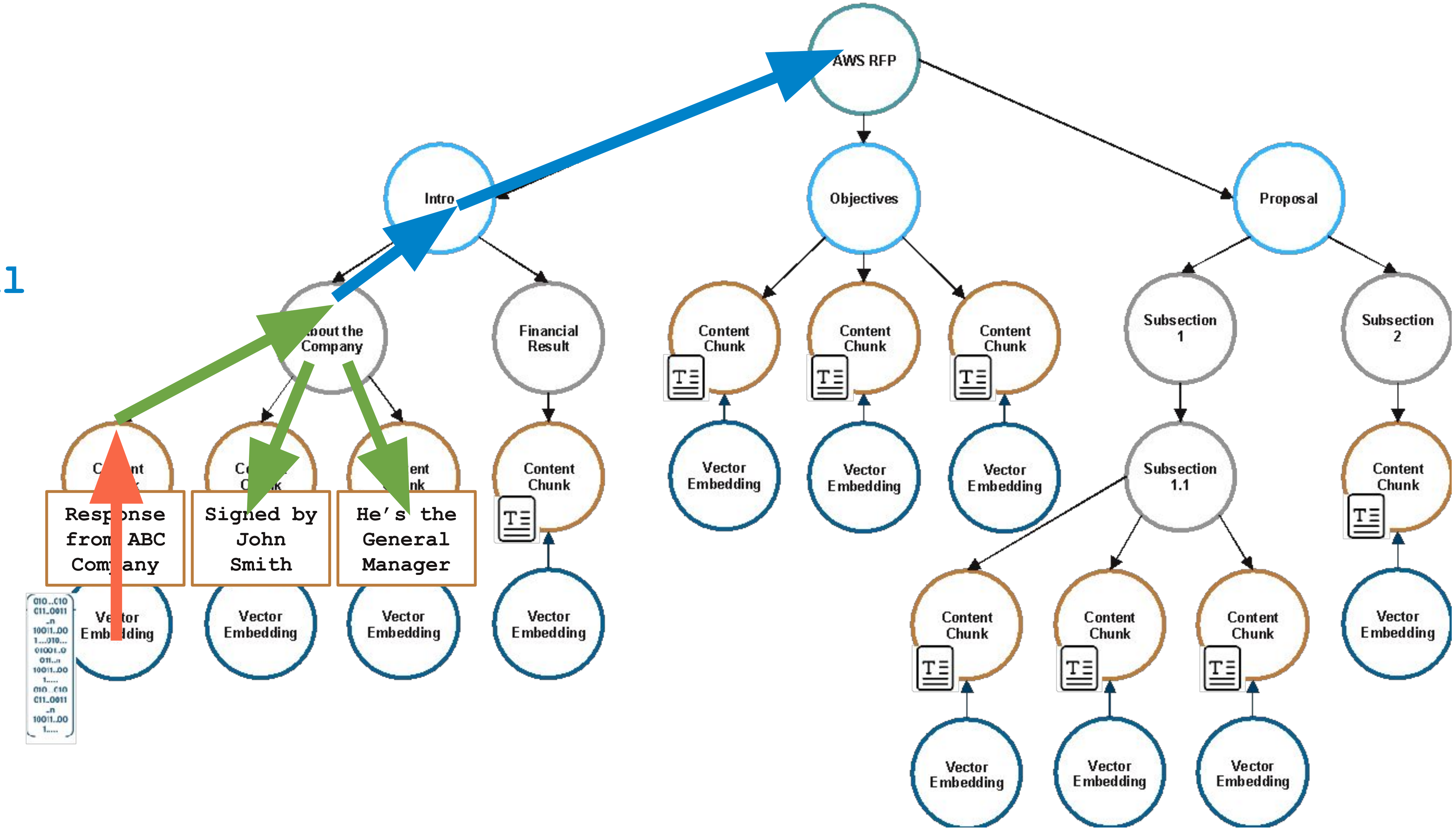
Accurate, Contextual and Explainable

Contextual Knowledge Retrieval within Neo4j KG



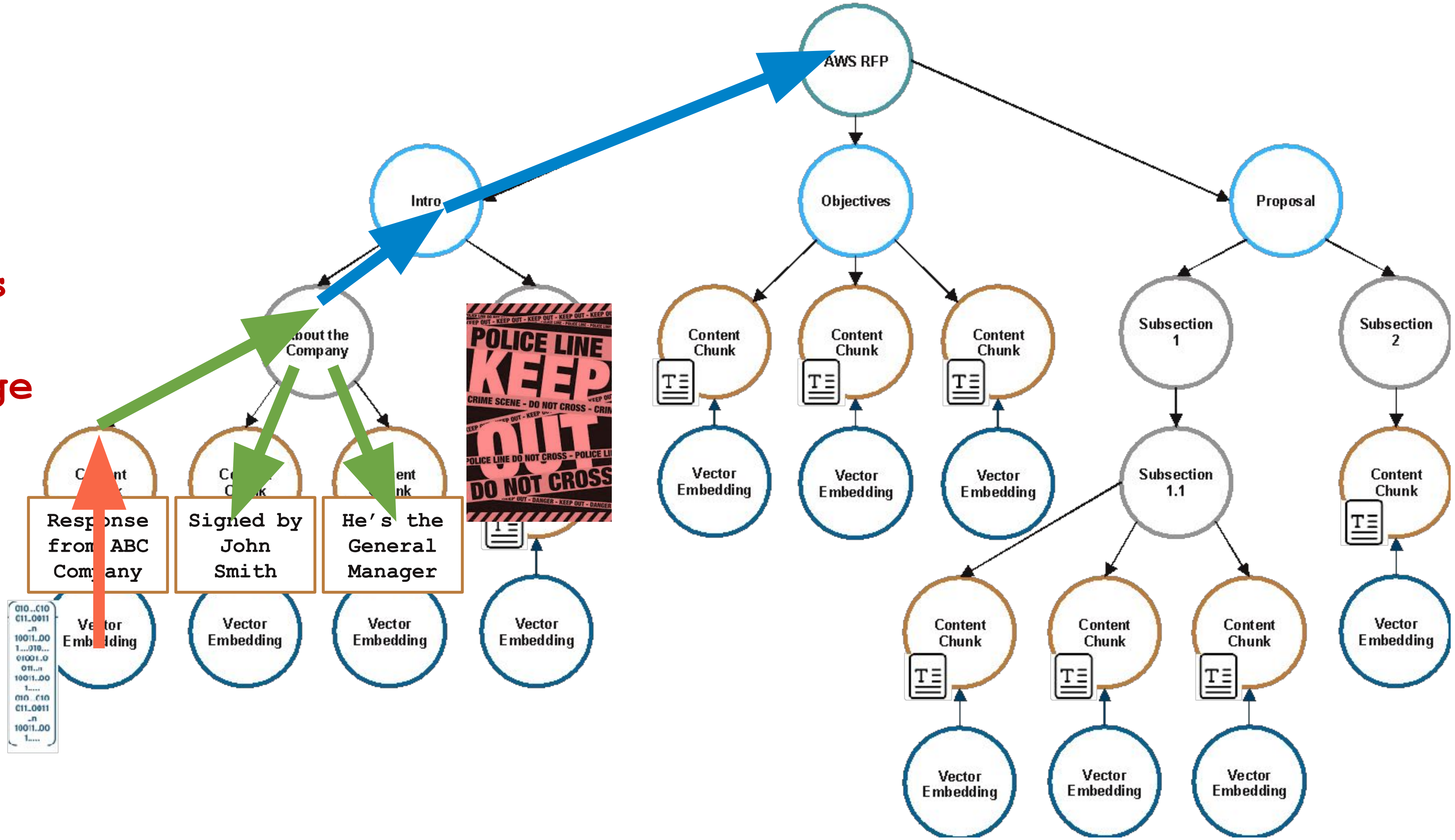
Accurate, Contextual and Explainable

Knowledge Retrieval
to aid in
Explainability

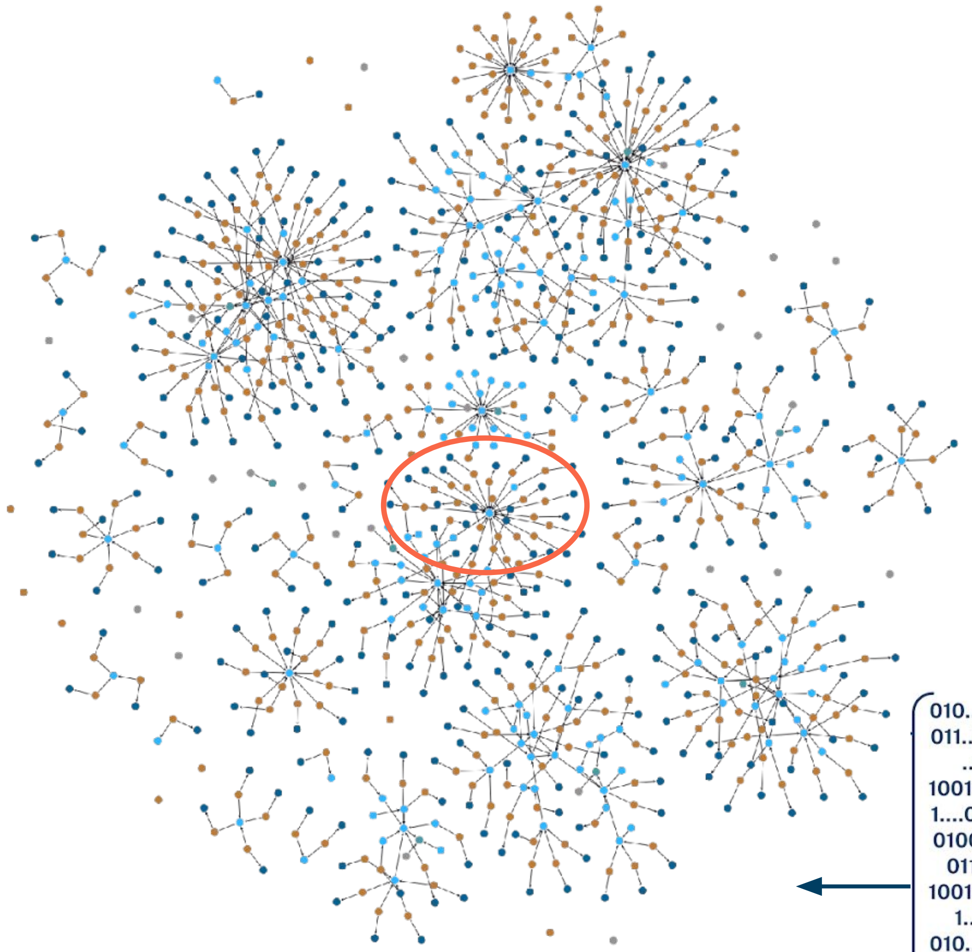


Accurate, Contextual and Explainable

Fine Grained Access Control to prevent **unwarranted Knowledge Retrieval**



Accurate, Contextual and Explainable



Similarity Search using
Neo4j Vector Index

010...010
011..0011
..n
10011..00
1....010...
01001..0
011..n
10011..00
1.....
010...010
011..0011
..n
10011..00
1.....

Vector Embedding

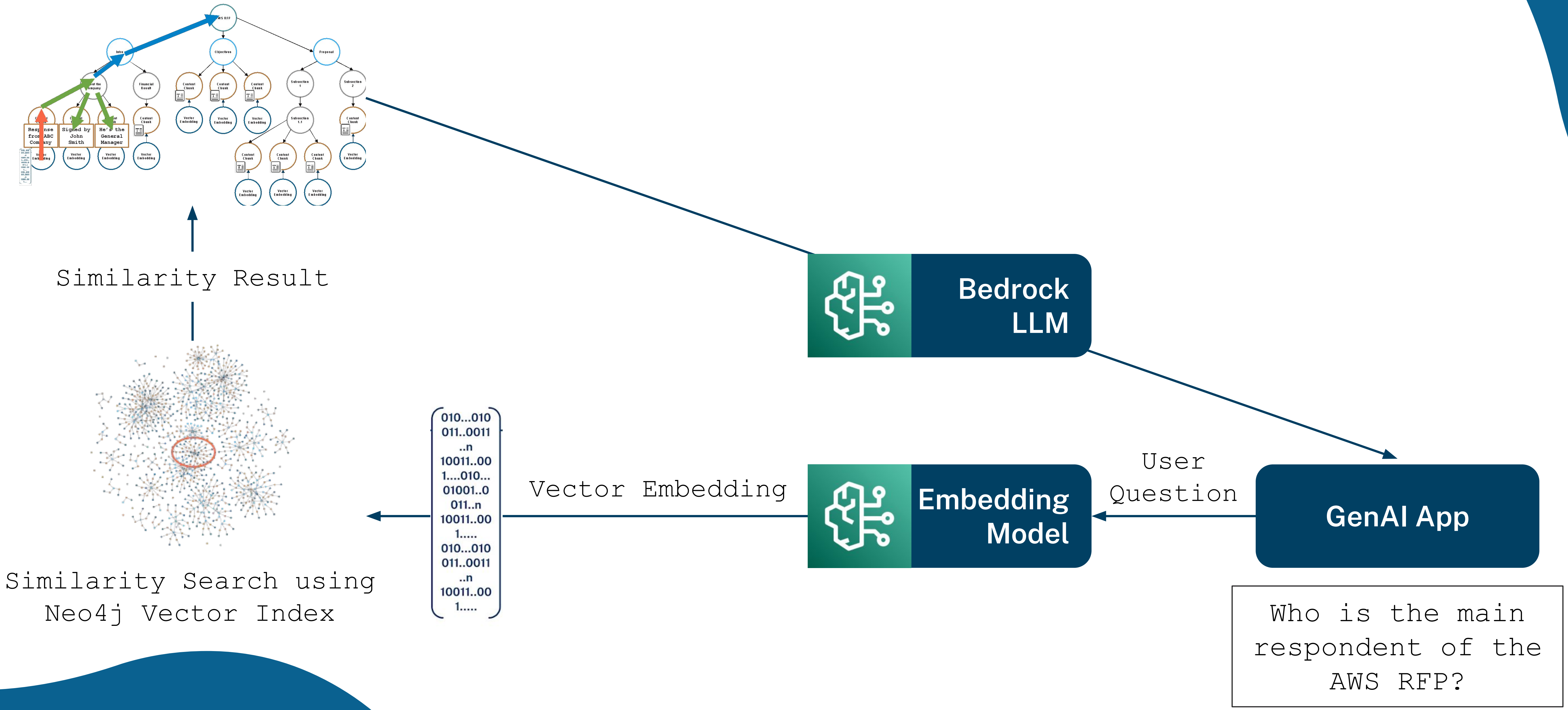
Embedding Model

User Question

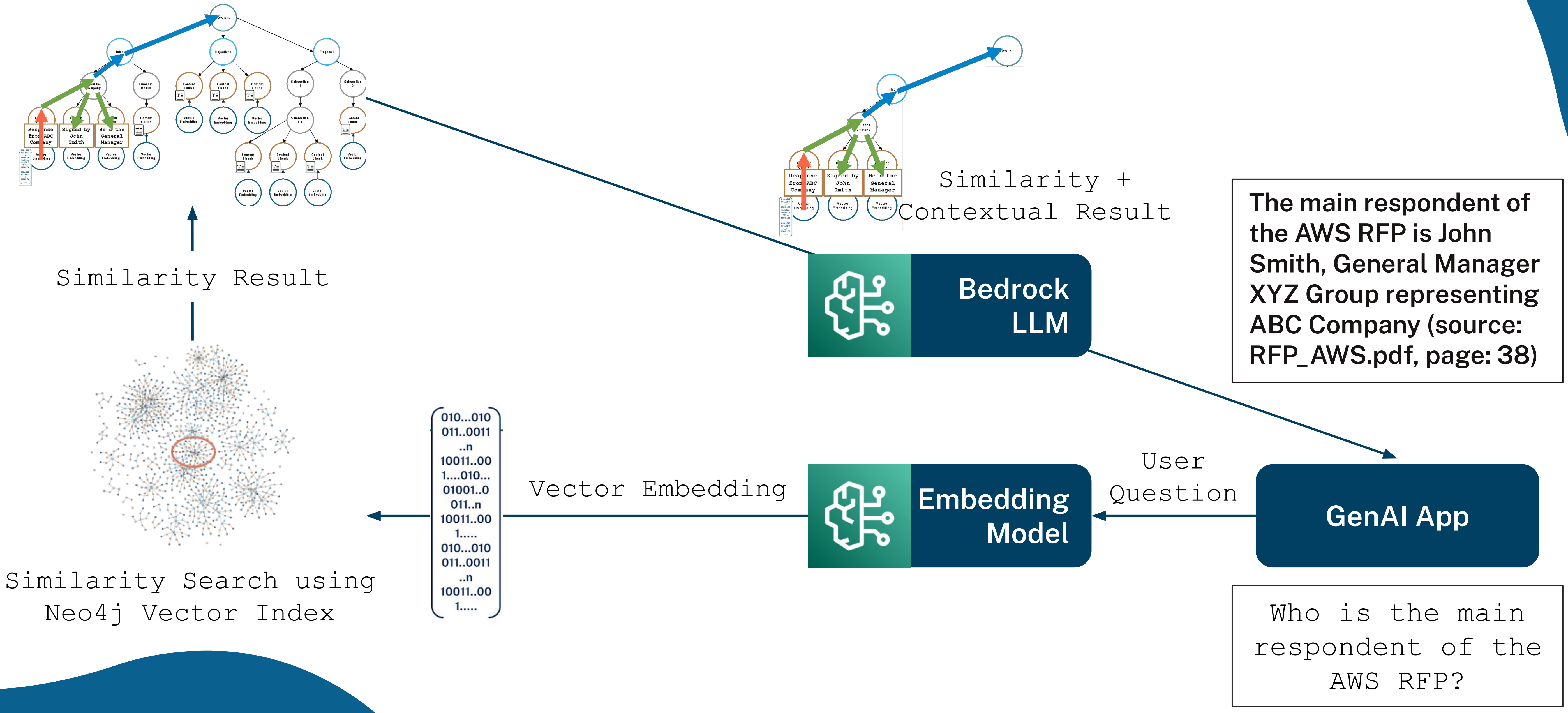
GenAI App

Who is the main respondent of the AWS RFP?

Accurate, Contextual and Explainable



Accurate, Contextual and Explainable



neo4j graphsummit

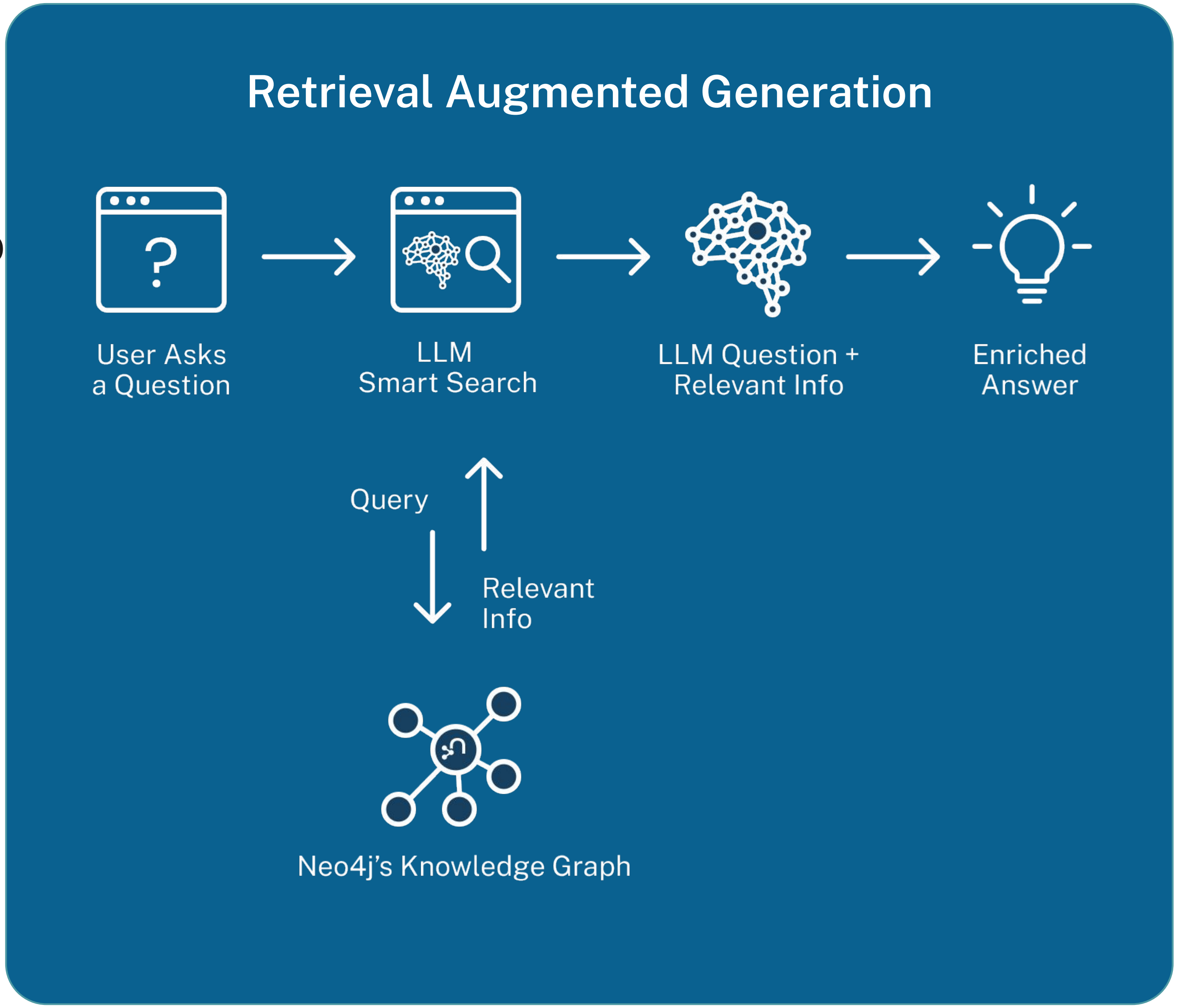
Module 3
Go to Notebook

Wrap Up!

Neo4j Knowledge Graph as Database of Truth

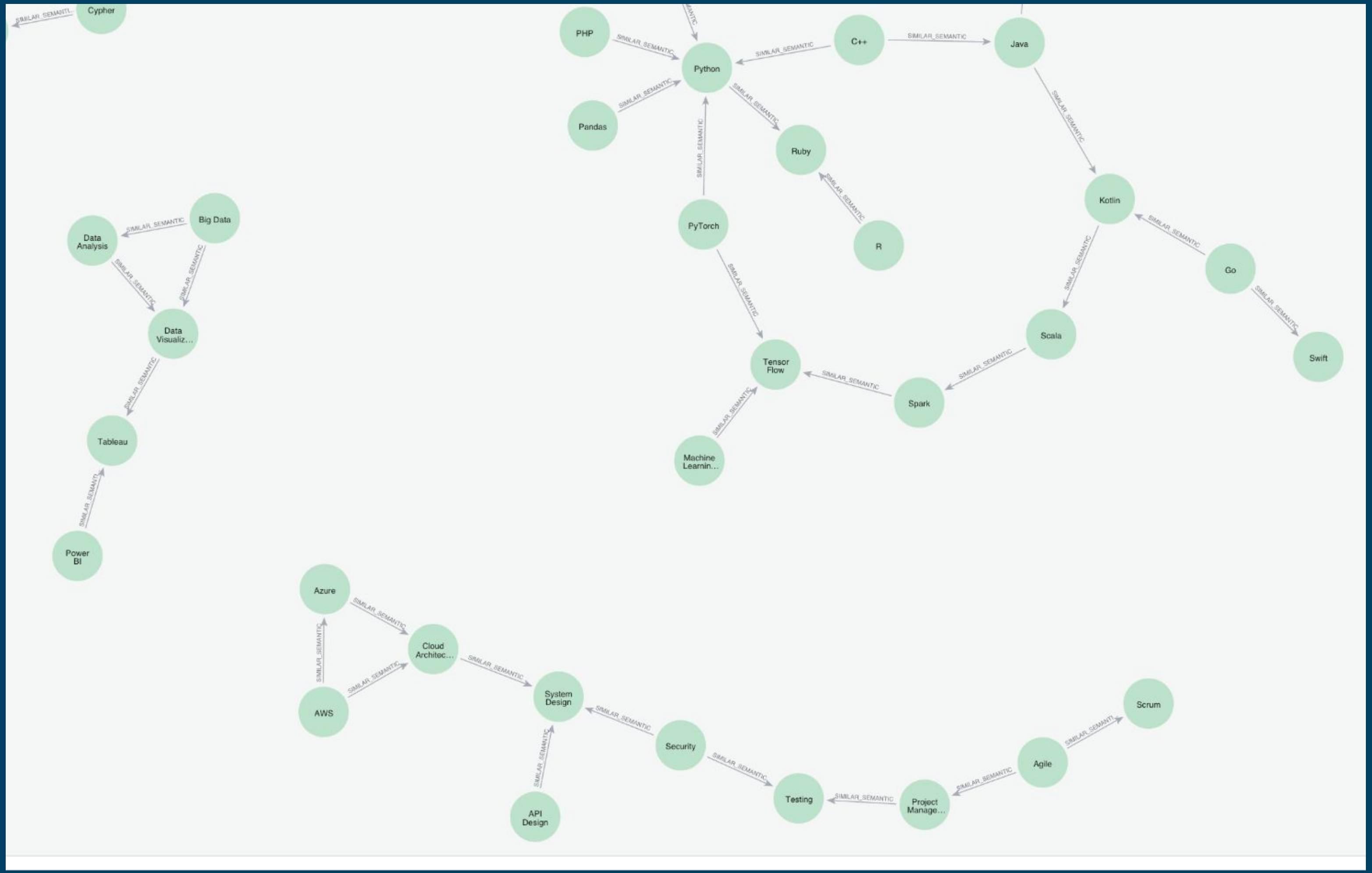
A Neo4j Knowledge Graph combined with LLM's obtain improvements:

- **Accuracy** - Obtain better answers compared to plain vector searches
- **Explainability**: Provide the user with more reasoning on how the results were obtained.
- **Acceleration**: Having all the capabilities in one platform increases understanding and decreases time to value.



Easier Development

Feedback from an AI Engineer



Here is the PR with the changes
I kinda replicated the same action-based cache already in place for Pinecone but thanks to the graph nature of Neo4J most of the operations yield better results:

- thanks to the `Neo4j graph data science` plugin we can store embeddings and calculate cosine similarity at the database level
- getting related actions is as simple as following the relationships between nodes
- **the cache can be visualised**. This is an extremely valuable debugging tool for us to understand if/when and how the cache might be broken/misbehaving (I actually already fixed a couple of bugs just thanks to this)

neo4j graphsummit

GraphRAG Resources

Github Repository with this workshop

Free online graph academy courses, videos & webinars

Developer guides and coded examples

Thank you!