

**Knowledge graph pre právne
dokumenty**

GraphRAG

Frameworks

LangGraph / LangChain - workflow:

- Modelovanie workflow
- State management pre multi-step processing
- Llm integrácia pre extrakciu entít a vzťahov

Neo4j – grafová databáza:

- Právne vzťahy a hierarchie
- Python implementacia pre LangChain

Spacey – NLP preprocessing

Pyvis – grafy pre python rozhranie

D3.js – grafy pre pre web rozhranie

React – web UI

Architektúra systémy

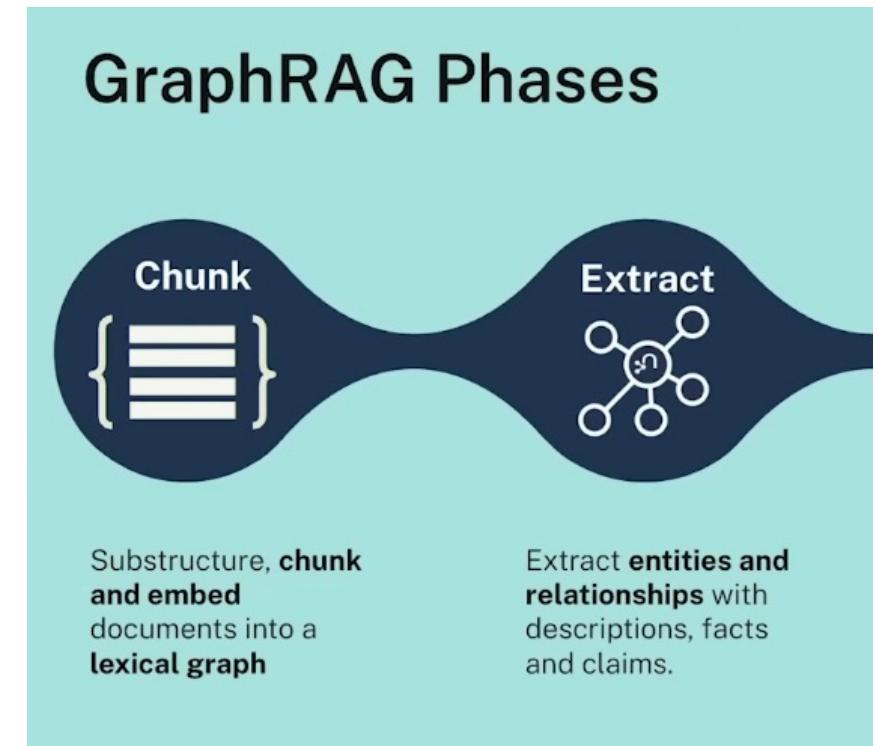
Fáza 1: Spracovanie a extrakcia

Právny text -> Preprocessing -> Relation & NER Extraction -> Štruktúrované dátá

LLMGraphTransformer: Uses an LLM to extract entities and relationships from text.

```
// pseudokód

def preprocess(text):
    // cistenie, segmentacia
    // rozpoznanie §, odsekov, bodov
def extract_entities(text):
    // Named-Entity Recognition: pravne pojmy,
    institucie, § cisla
    // LLM-based extraction
def extract_relations(entities, context):
    // vzťahy medzi entitami
```



Example Pattern

Name: Graph Enhanced Vector Search

Description: The user question is embedded using the same embedder used to create chunk embeddings. A vector similarity search is executed on the chunk embeddings to find k (number previously configured by developer/user) most similar chunks. A traversal of the Domain Graph starting at the found chunks is executed to retrieve more context.

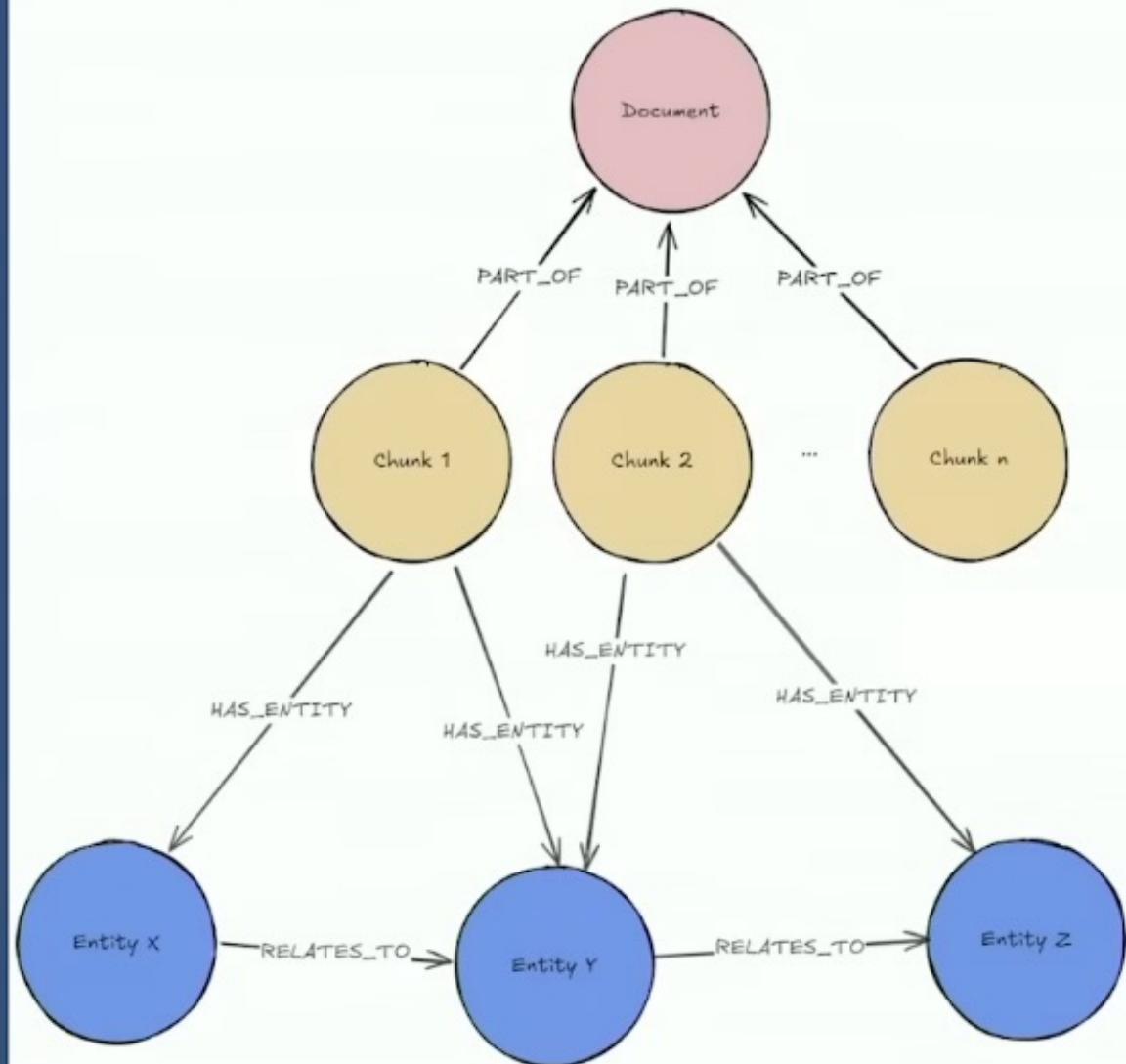
Context: The biggest problem with basic GraphRAG patterns is finding all relevant context necessary to answer a question. The context can be spread across many chunks not being found by the search. Relating the real-world entities from the chunks to each other and retrieving these relationships together with a vector search provides additional context about these entities that the chunks refer to. They can also be used to relate chunks to each other through the entity network.

Required pre-processing: Use an LLM to execute entity and relationship extraction on the chunks. Import the retrieved triples into the graph.

Variations: Entity disambiguation, Question-guided/Schema-defined extraction, Entity embeddings, Ontology-driven traversal

```
MATCH (node)-[:PART_OF]->(d:Document)
MATCH (node)-[:HAS_ENTITY]->(e)
MATCH path=(e)((()-[rels:!HAS_ENTITY&!PART_OF]-()){0,2}(:!Chunk&!Document))
...
RETURN ...
```

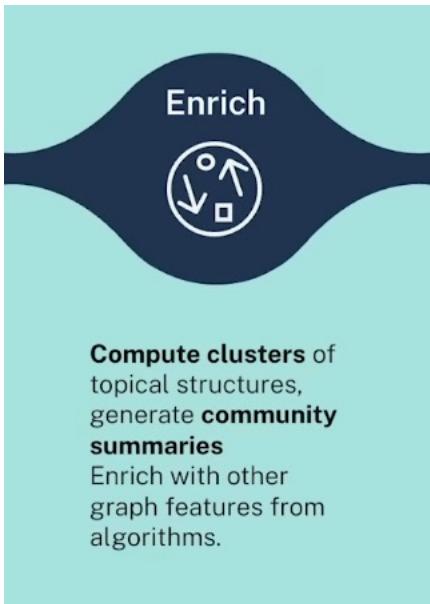
Knowledge graph shape: Lexical Graph with Extracted Entities



Architektúra systémy

Fáza 2: Konštrukcia grafu

Štruktúrované dátá -> Graph Building -> Knowledge Graph (Neo4j)



```
// pseudokod pre zstrojenie grafu so Cypher queries

def create_entity(entity):
    // vytvori node v grafe
def create_relation(relation):
    // vytvori hranu v grafe
def query_related_entities(entity_text, depth)
    // najde prepojene entity
```

Architektúra systémy

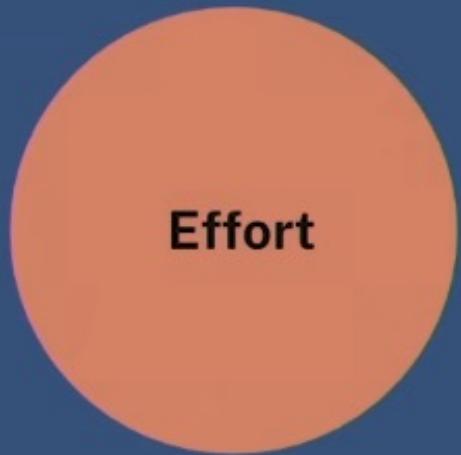
Fáza 3: Vizualizácia a interakcia

Knowledge Graph (Neo4j) -> Query & Visualization -> UI



1. **GraphCypherQAChain:** Handles question-answering over the graph database.
2. **Query Translation:** Converts a natural language question into a graph query like Cypher for Neo4j.
3. **Execution:** Runs the query against the graph to retrieve structured facts and relationships.
4. **Response Generation:** The retrieved context is combined with the original question and passed to an LLM which produces a clear natural language answer.

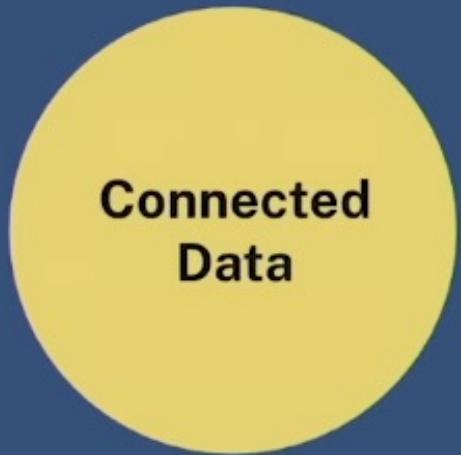
Effort & Impact



Data Engineering

- Model
- Extract
- Index
- Enrich

Neo4j



Rich connected network

- unstructured text,
- structured entities
- derived insights



Contextual retrieval

- multi-hop queries
- higher relevance
- explainability
- guidance for agents and tools

Components

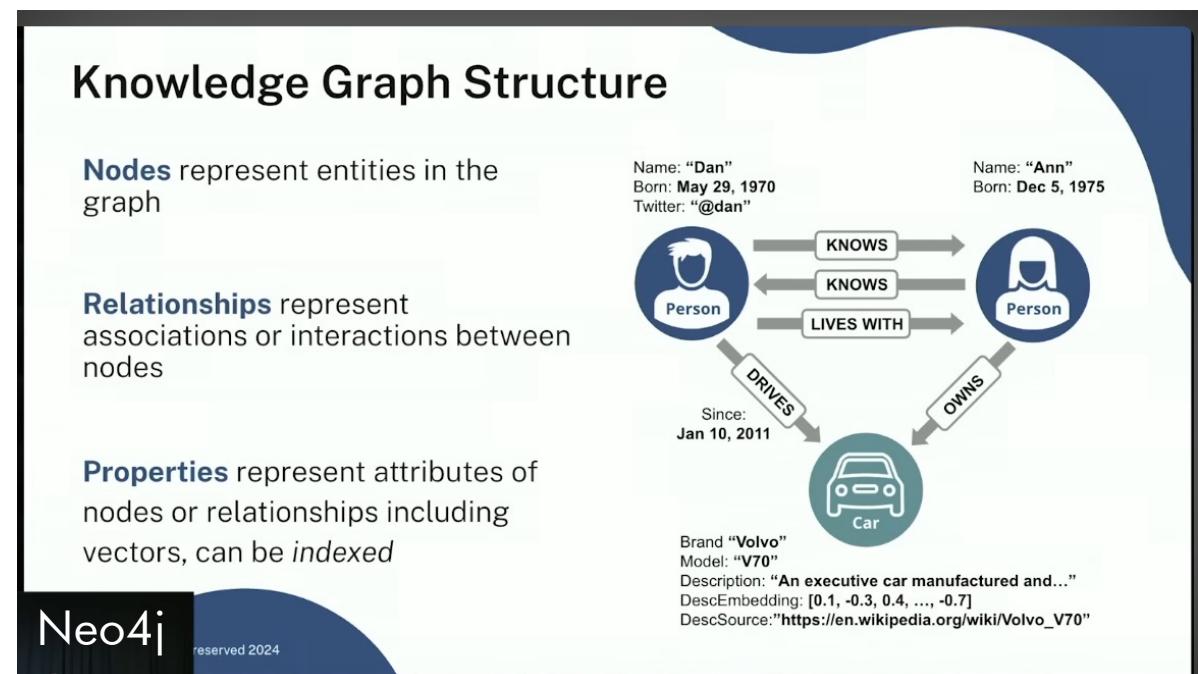
Entity Recognition:

- **Právne pojmy:** zmluva, zodpovednosť, pravo, povinnosť
- **Paragrafy:** § 124, § 45 ods. 2 pism. a)
- **Subjekty:** fyzická osoba, právnická osoba, súd
- **Dokumenty:** zákon, vyhláška, nariadenie

// zhľukovanie do komunit //

Relation Types:

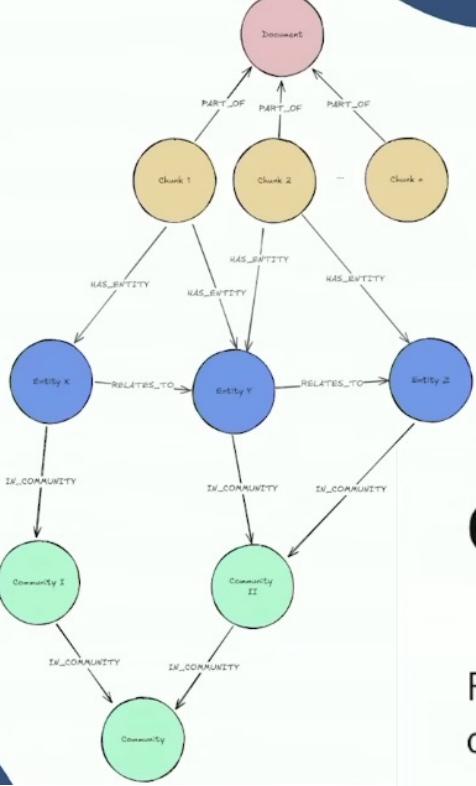
- ODKAZUJE_NA
- DEFINUJE
- UPRAVUJE
- DOPLNUJE
- PODMIENUJE
- RUSI



Meta - High Level Community Summaries

Neo4j

reserved 2024



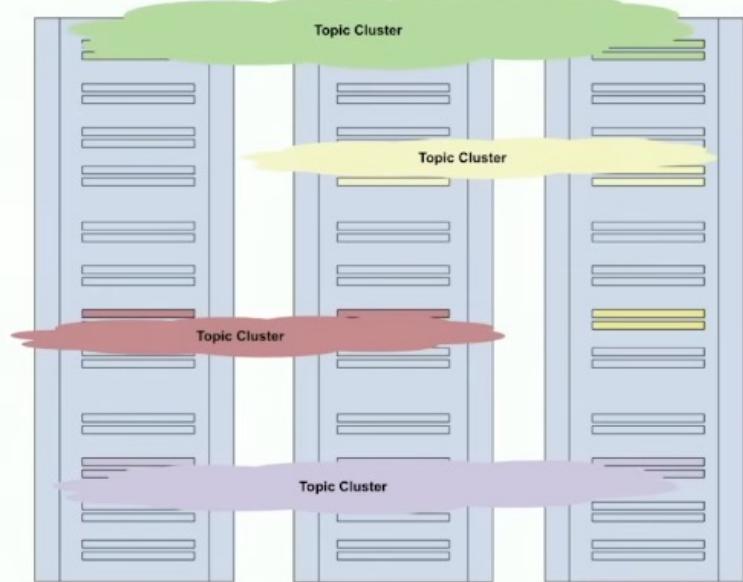
Global Search - Finding High Level Answers

For questions that cover the entirety
of a Corpus.

Cross document topic clusters.

Can use all or most relevant
community summaries (vector or
hybrid search)

Choose level of granularity.

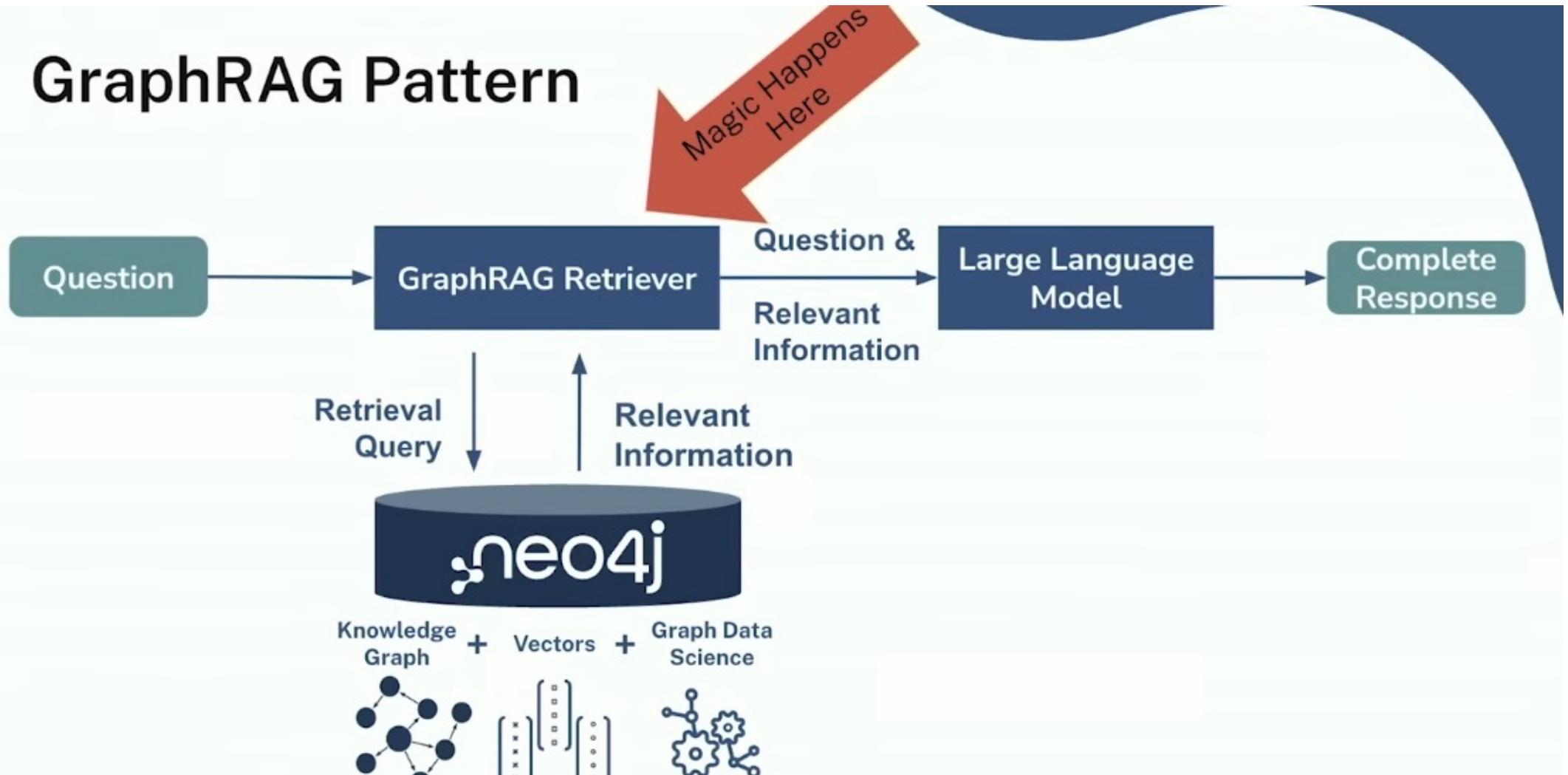


Neo4j

reserved 2024

graphrag.com/reference/graphrag/global-community-summary-retriever/

GraphRAG Pattern

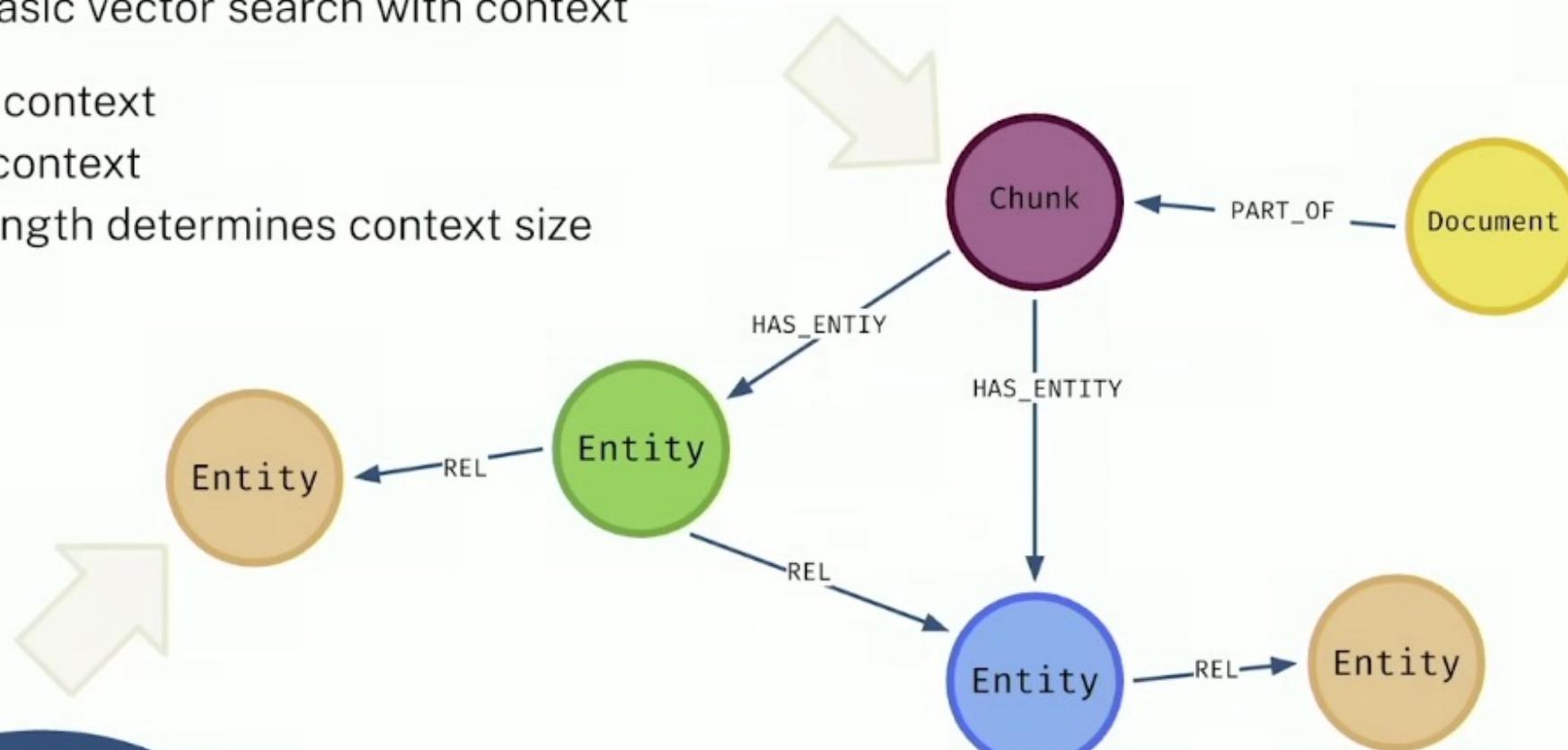


Neo4j

GraphRAG - Graph Enhanced Vector Search

Enhances basic vector search with context

- lexical context
- entity context
- path length determines context size



BlackStone – Spacy model pre UK pravne dokumenty

LexNLP library – pravny NLP toolkit

Slov-Lex – slovenska legislativa dataset

Obrazky z prezentacie su z:

<https://www.youtube.com/watch?v=XNneh6-eyPg>

<https://www.geeksforgeeks.org/artificial-intelligence/knowledge-graphs-using-langchain/>