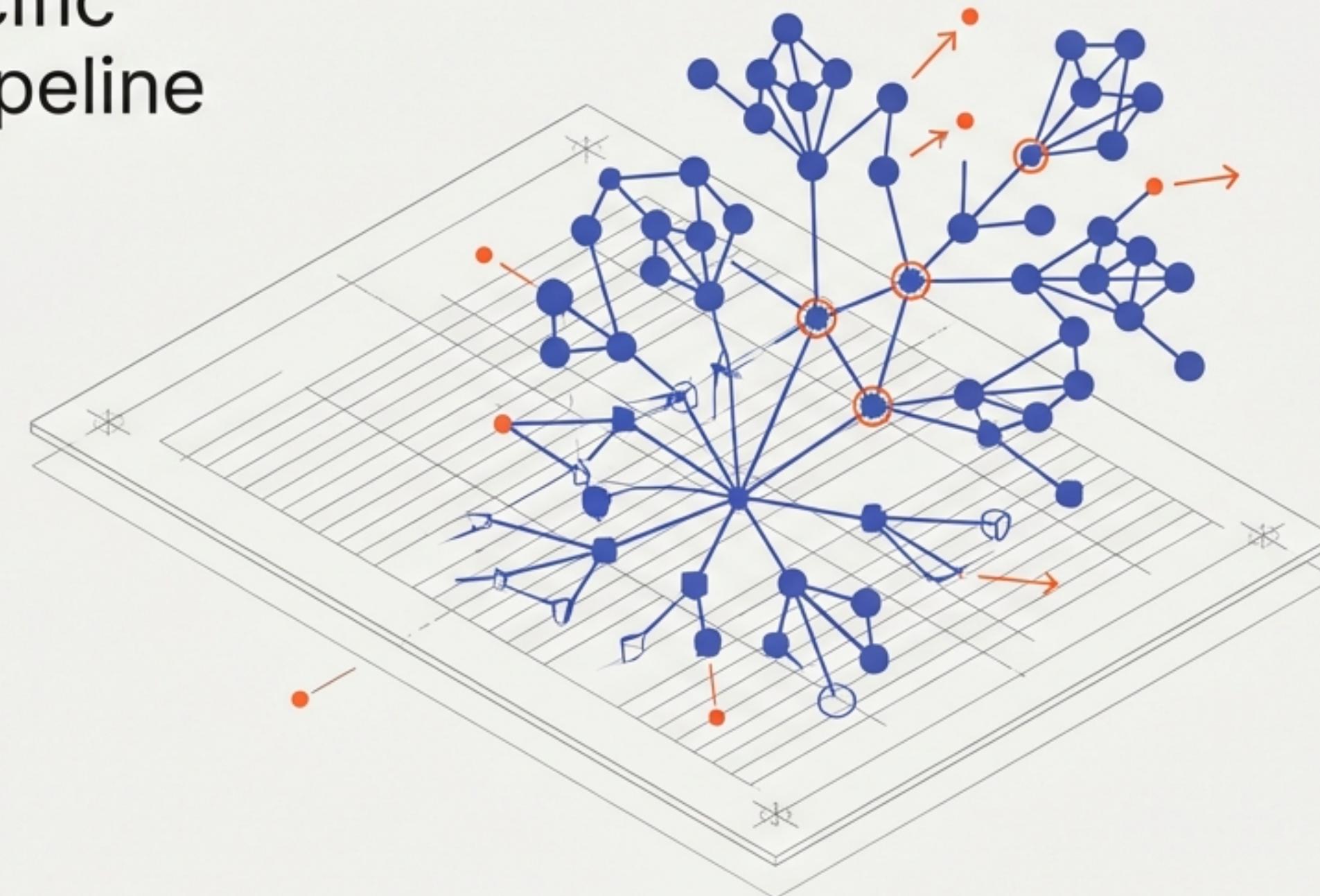


# PROJEKT IMARA

## Domain-specific GraphRAG Pipeline



MODUL: Abschlussarbeit CAS Machine Learning for Software Engineers (ML4SE)  
AUTOREN: Marco Allenspach, Lukas Koller, Emanuel Sovrano  
DATUM: 17.01.2026

Evaluierung von AI-Native GraphRAG Strategien und  
Model Fine-tuning zur Überwindung der Grenzen  
klassischer RAG-Systeme.

# Vom unstrukturierten Text zum vernetzten Wissen

## Management Summary

Traditionelle neuronale Netze kodieren lineare Beziehungen, doch die reale Welt ist komplex und multidimensional. Projekt IMARA implementiert eine Pipeline, die Wissen aus PDF-Dokumenten in graph-basierte RAG-Systeme überführt, um "Multi-Hop"-Reasoning zu ermöglichen.

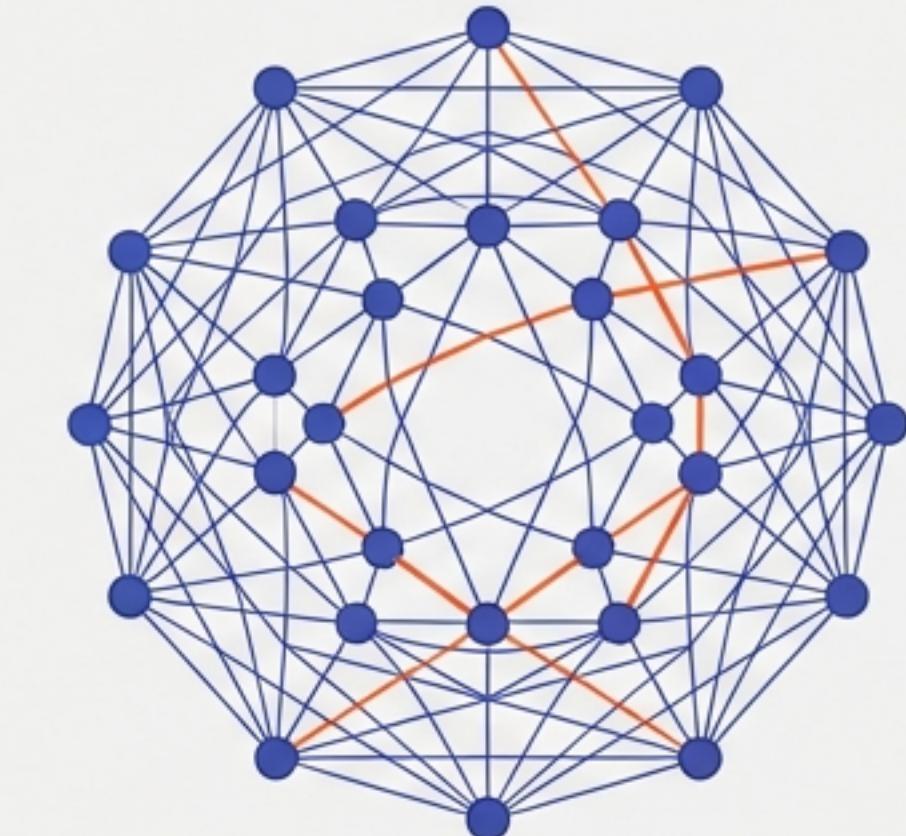
- **Problem:** Konventionelles RAG scheitert an der Synthese verteilter Informationen.
- **Lösung:** Implementierung und Vergleich von drei Architekturen (LeanRAG, LinearRAG, GraphMERT) mit einer vollautomatisierten End-to-End Pipeline.
- **Ergebnis:** Erfolgreicher Nachweis, dass strukturierte Wissensgraphen Vorteile bei komplexen Abfragen bieten, validiert durch OpenRAGBench.

### Linear vs. Complex

Neural / Linear



Real World / Graph



# Die Grenzen von 'Naivem RAG' (**Vector Blindness**)

Warum Vektorschreie allein Kontextfragmente nicht verbinden kann.

## 1. Die Analogy:

Vektorbasierte RAG bietet einem Forscher einen Stapel isolierter Karteikarten. Es fehlt der Überblick.



## 2. Kontextuelle Fragmentierung:

**Chunking** zerstört den natürlichen Informationsfluss.



## 3. Blindheit für Relationen:

Vektoren verstehen Ähnlichkeit, aber keine **Kausalität**, Abhängigkeit oder Hierarchie.

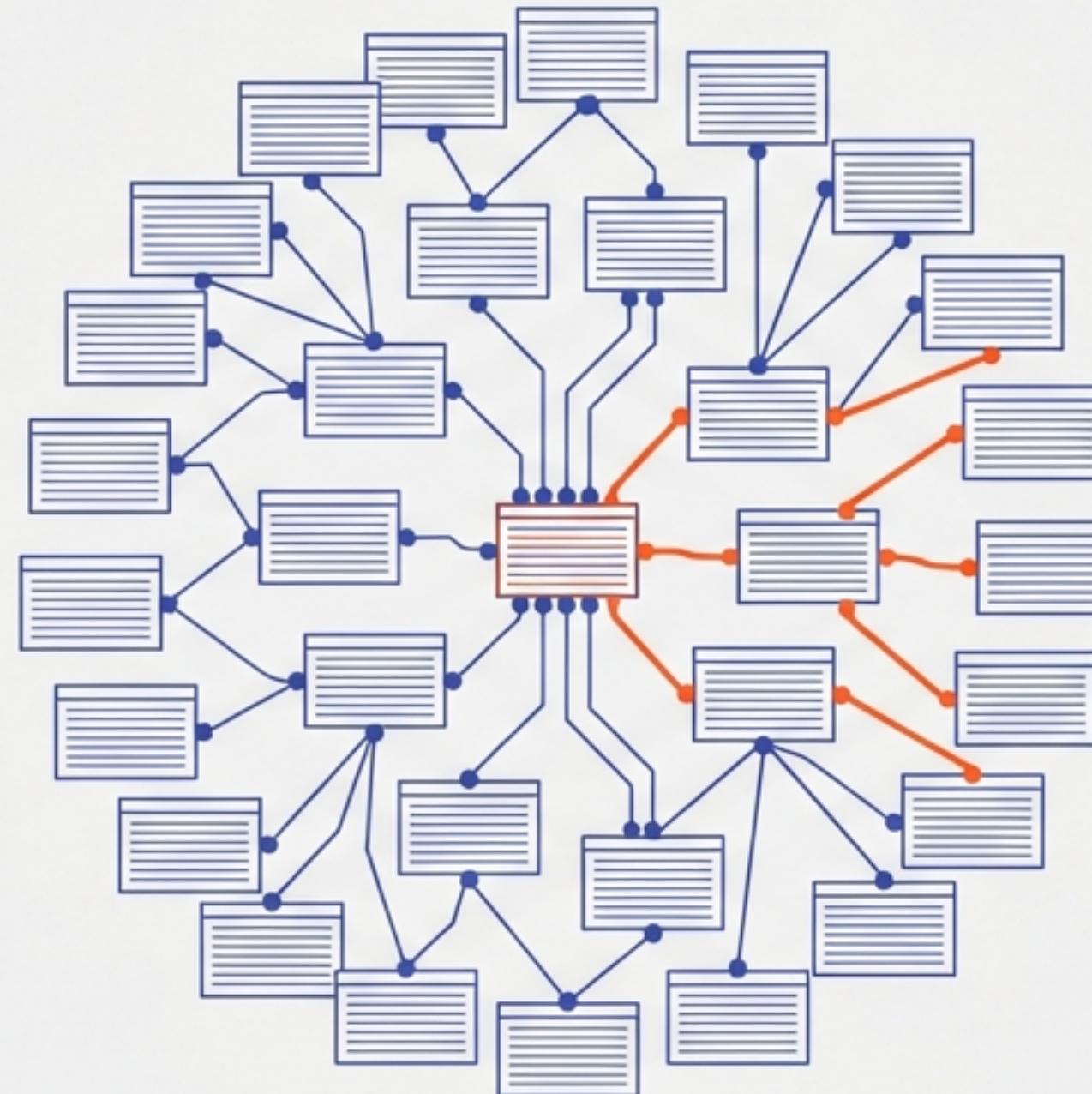
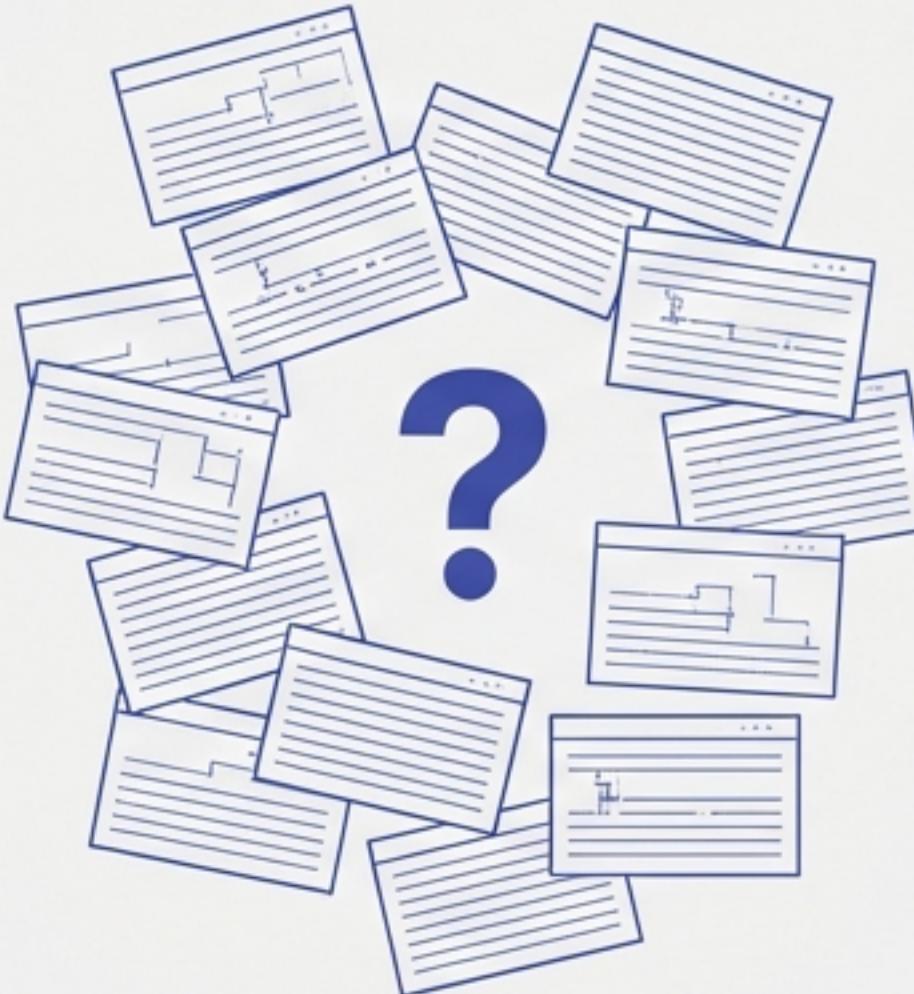


## 4. Multi-Hop Failure:

Unfähigkeit, Informationssprünge zu überbrücken (z.B. Störung → Produkt → Kampagne).

# Das Paradigma 'AI-Native GraphRAG'

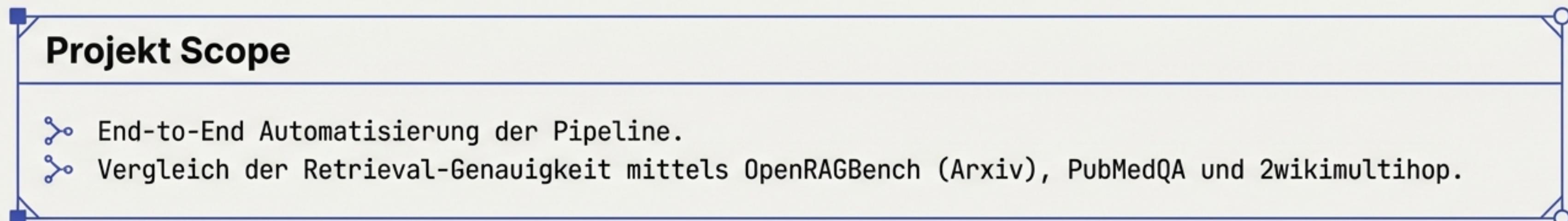
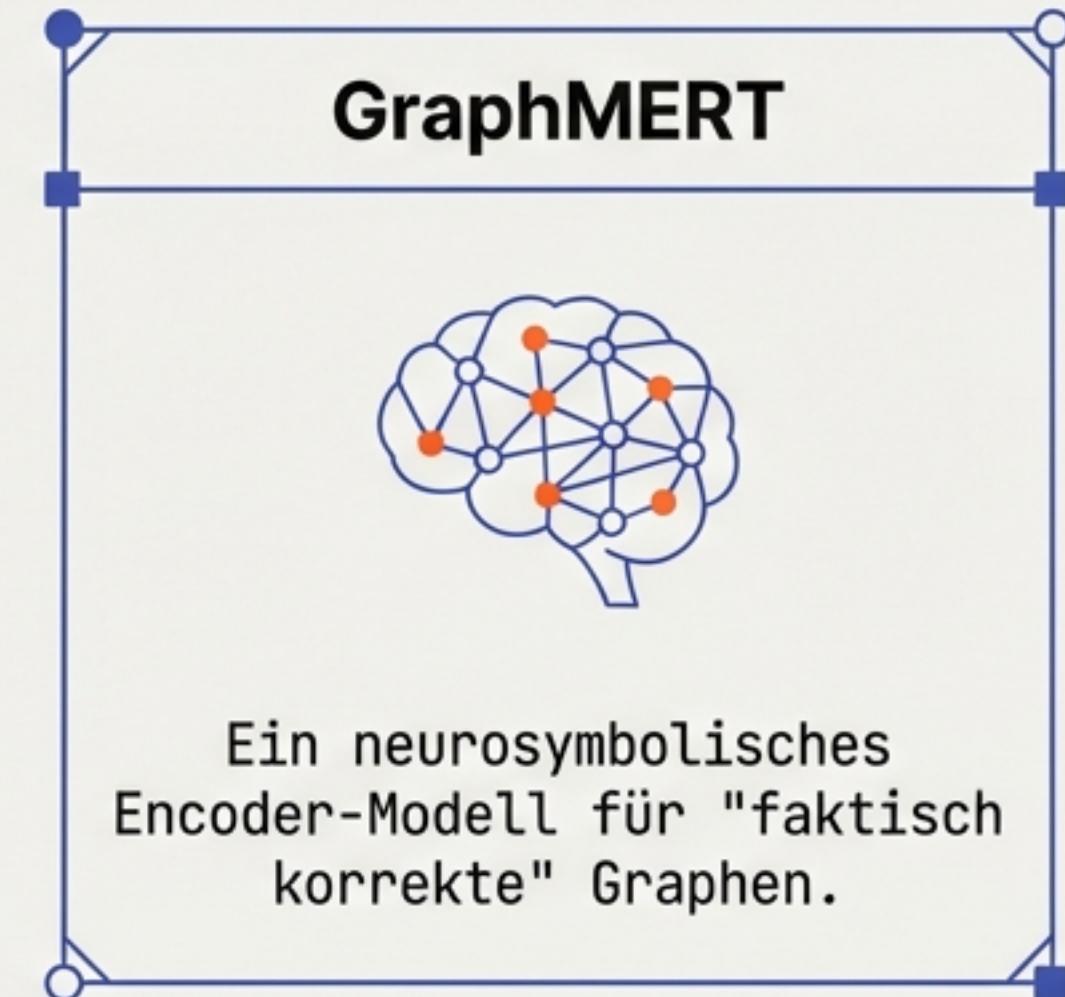
Struktur als Kontext: Transformation passiver Daten in ein aktives Modell.



- **Definition:** Eine KI benötigt für effektives Denken ein Modell des Anwendungsbereichs, nicht nur eine Fakten-Sammlung.
- **Das Ziel:** Automatisierung des gesamten Workflows von unstrukturierten Daten bis zur Antwort, unter Abstraktion der Graphentheorie-Komplexität.
- **Die Lösung:** Im Gegensatz zu den Karteikarten erstellt GraphRAG eine 'umfassende Mindmap', die entscheidende Verbindungen aufdeckt.

# Projektziele & Evaluierte Systeme

Drei Strategien zur Erstellung dichter Wissensgraphen.



# Methodik: PDF-Extraktion mit Docling

Herausforderungen bei der Umwandlung unstrukturierter Dokumente.

Herausforderung	Detail	Lösung
VRAM Limit (16GB)	"Endless-loops" bei vollen Features.	Ausführung direkt in Python auf CPU statt Container.
Formel-Parsing	Extrem rechenintensiv.	Einsatz eines dedizierten zweiten Rechners für Massenextraktion.
Parameter-Optimierung	Initiale Qualität ungenügend.	Feinjustierung führte zu massiver Geschwindigkeits- und Genauigkeitssteigerung.

## Console Log

**ERROR:** VRAM Limit Exceeded

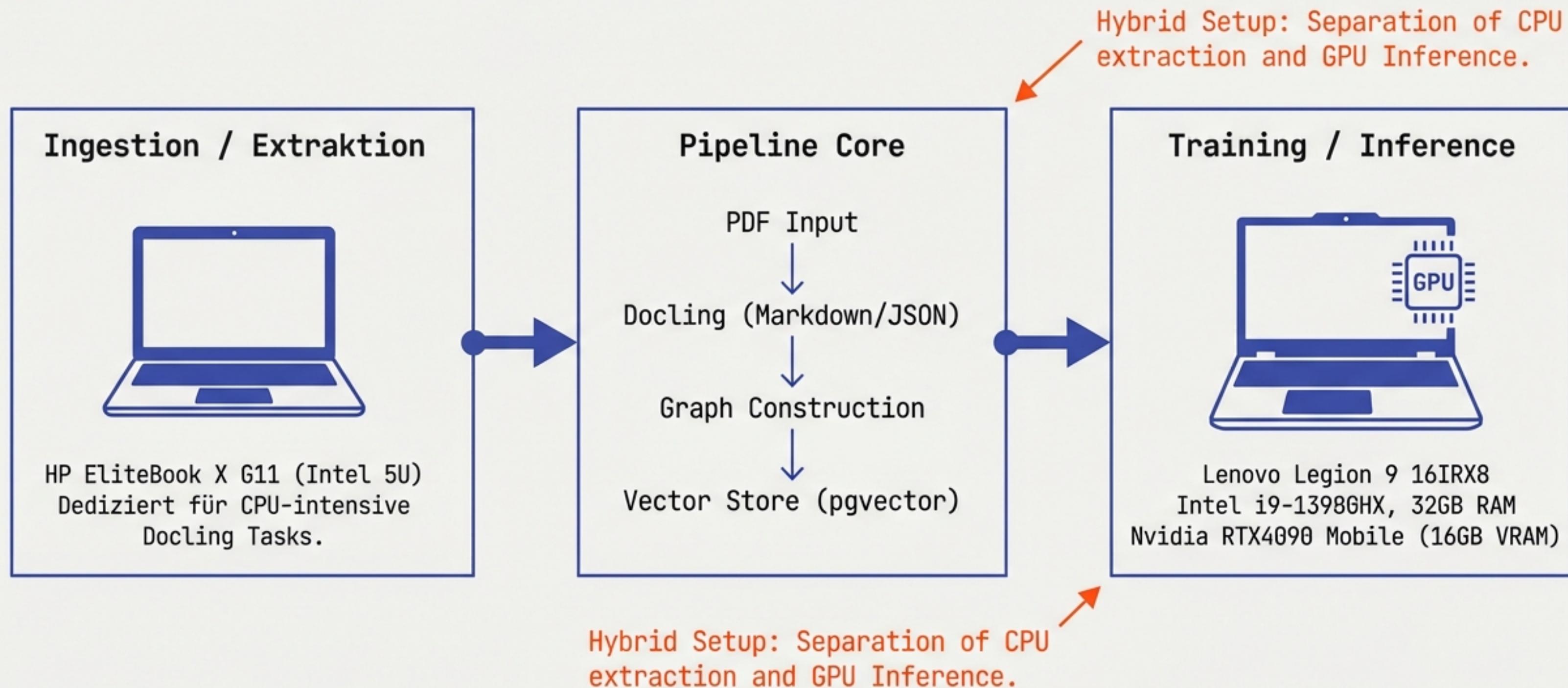
**WARNING:** Endless loop detected

OPTIMIZING PARAMETERS...

SUCCESS: Processing time reduced.

# Systemarchitektur & Hardware Setup

Hybride Infrastruktur für Ingestion und Inferenz.



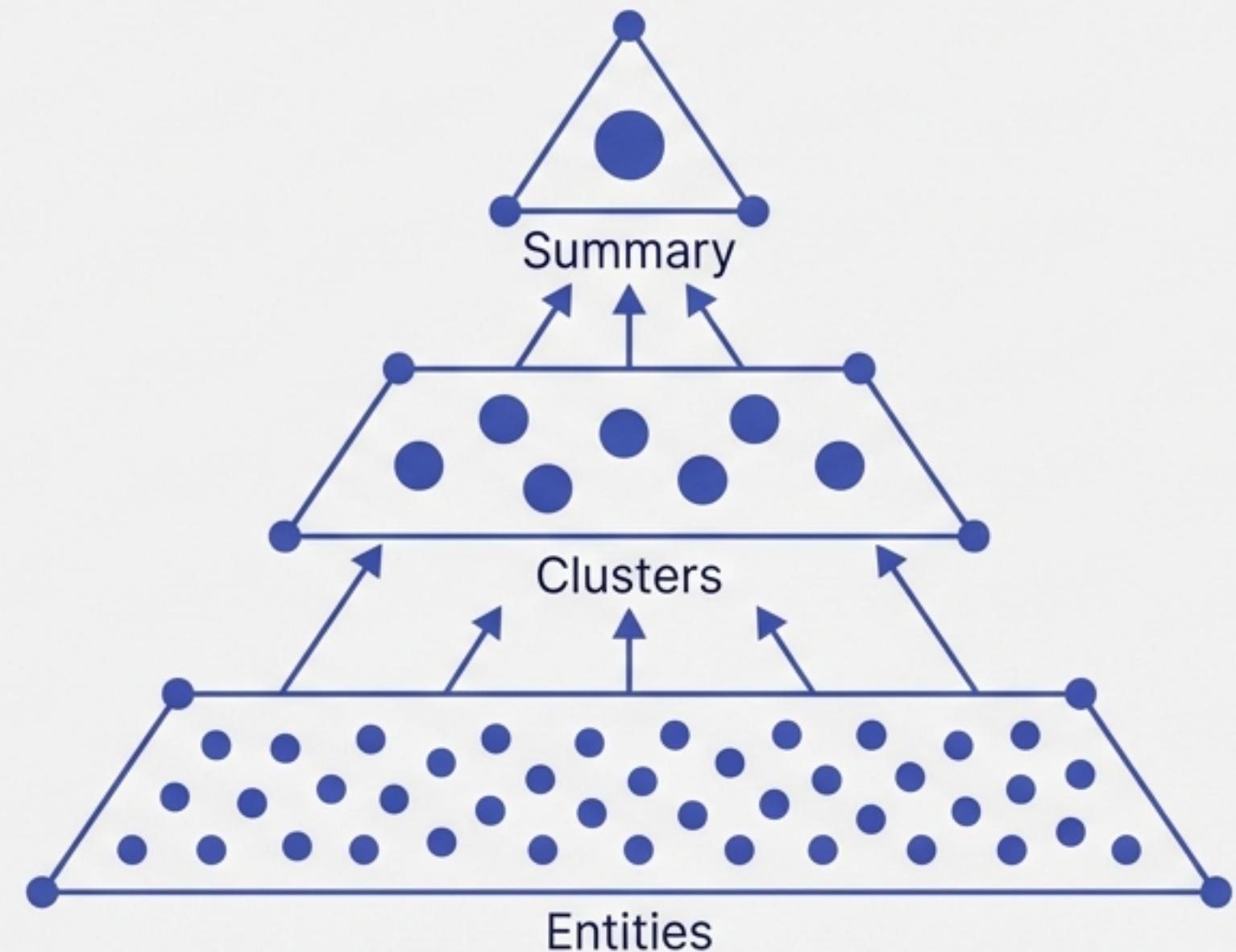
# Ansatz 1: LeanRAG (Hierarchisch)

Reduktion von Redundanz durch semantische Aggregation.

**Semantic Aggregation:** Gruppierung von Low-Level-Entitäten in semantisch kohärente Cluster/Zusammenfassungen.

**Traversal Strategy:** Hierarchisches, strukturgeleitetes Retrieval ('Bottom-up' von der Entität zum Cluster).

**Impact:** Reduziert Retrieval-Redundanz um ca. 46% im Vergleich zu flachen Baselines.



# Ansatz 2: LinearRAG (Relation-Free)

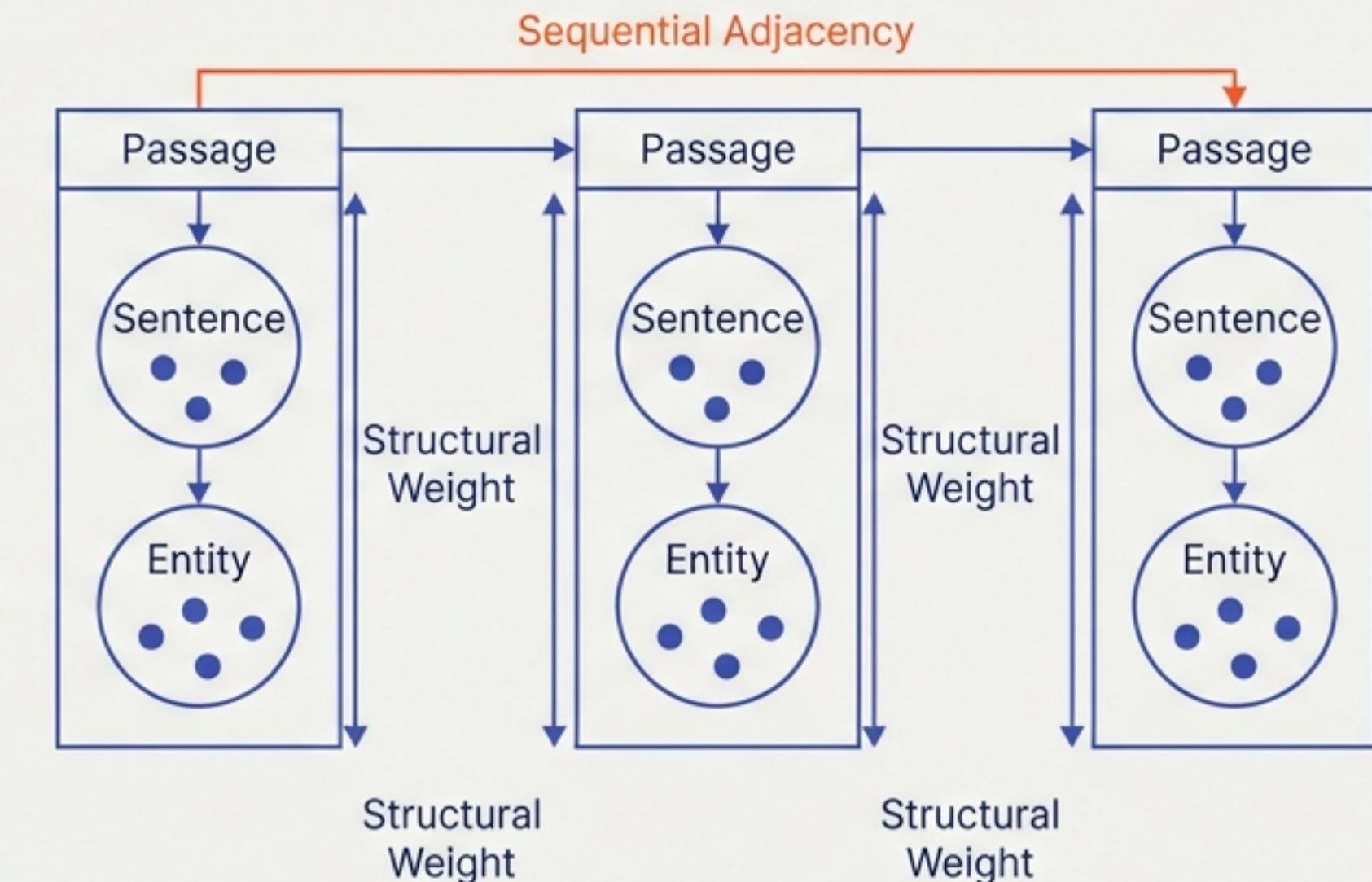
Effizienz durch implizite Relationen.

**Zero LLM Token Consumption:** Graph-Konstruktion ohne generative KI, nur mittels NLP (scispaCy).

**Context-Preserving:** Nutzung von Lightweight Entity Recognition und semantischem Linking.

**Kantentypen:** Strukturelle Verbindungen (Satz → Passage) und sequenzielle Adjazenz statt halluzinierter Relationen.

**Benefit:** Linear Time Complexity.



# Ansatz 3: GraphMERT (Neurosymbolisch)

Skalierbare Destillation zuverlässiger Wissensgraphen

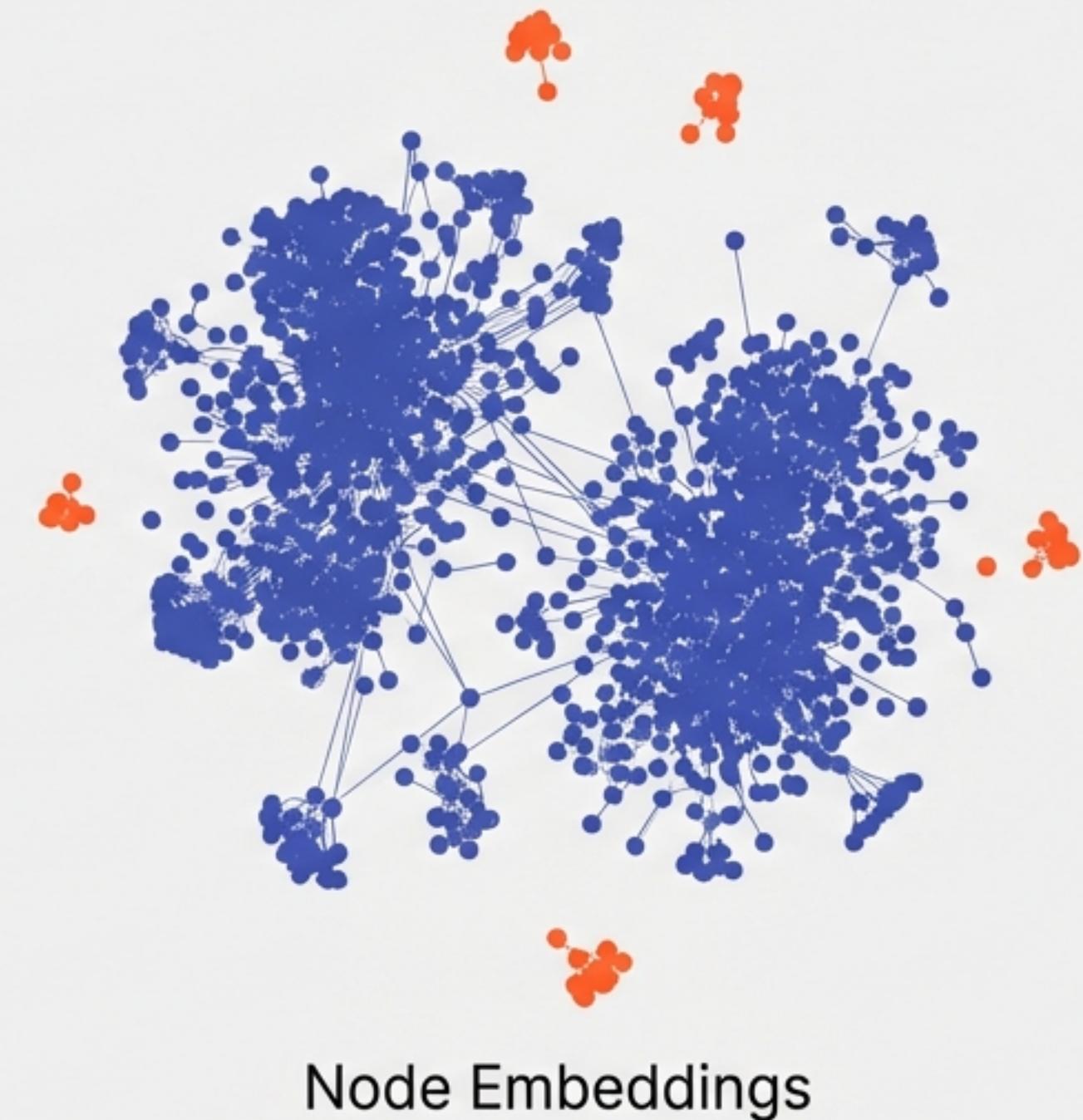
## Konzept:

Neurosymbolischer Stack – Neuronales Lernen von Abstraktionen + Symbolische Graphen für verifizierbares Schliessen.

## Performance:

Erreichte einen **FActScore** von **69.8%** bei medizinischen Texten (vs. 40.2% bei Basis-LLMs).

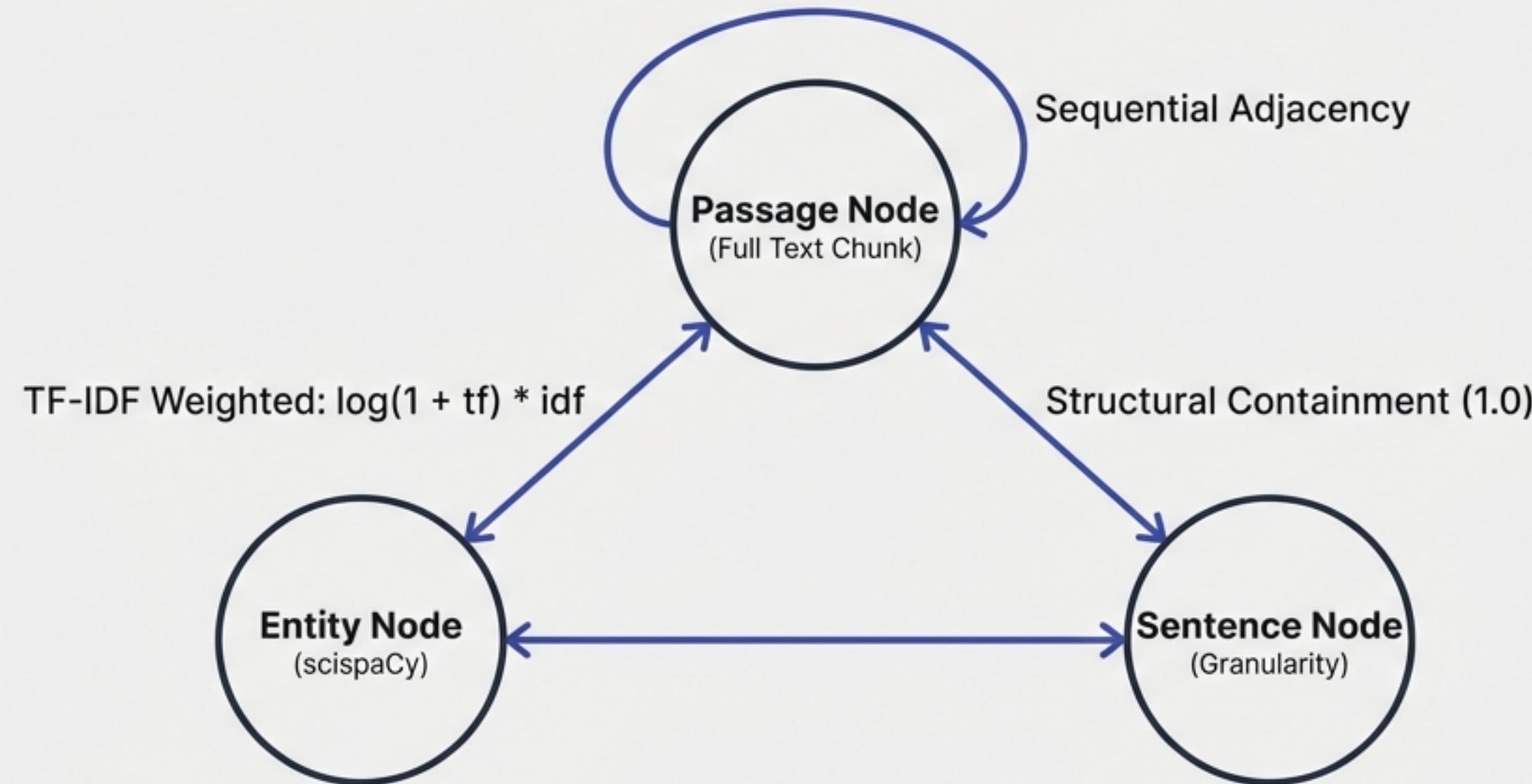
**Ziel:** Erstellung “faktisch korrekter” (Provenance) und “valider” (Ontologie-konsistenter) Graphen.



Node Embeddings

# Deep Dive: Graph-Konstruktion (LinearRAG Impl.)

Deterministische "On-the-Fly" Erstellung ohne LLM-Overhead.



**Loading:** Idempotenz via **MD5 Hashes**.

**Retrieval:** Hybrid Approach combining **Vector Search** (Candidates) with **Personalized PageRank** (Graph Expansion).

# Resultate: Graph Topologie (OpenRAGBench)

Metriken des aus 1'001 wissenschaftlichen Publikationen erstellten Graphen.

JetBrains Mono

Input: 1'001 Papers (Arxiv)

Inter

Efficiency: ~3.6 Mio. Entitäten  
rein CPU-basiert extrahiert.

**1,751,262**

Total Nodes

**7,370,454**

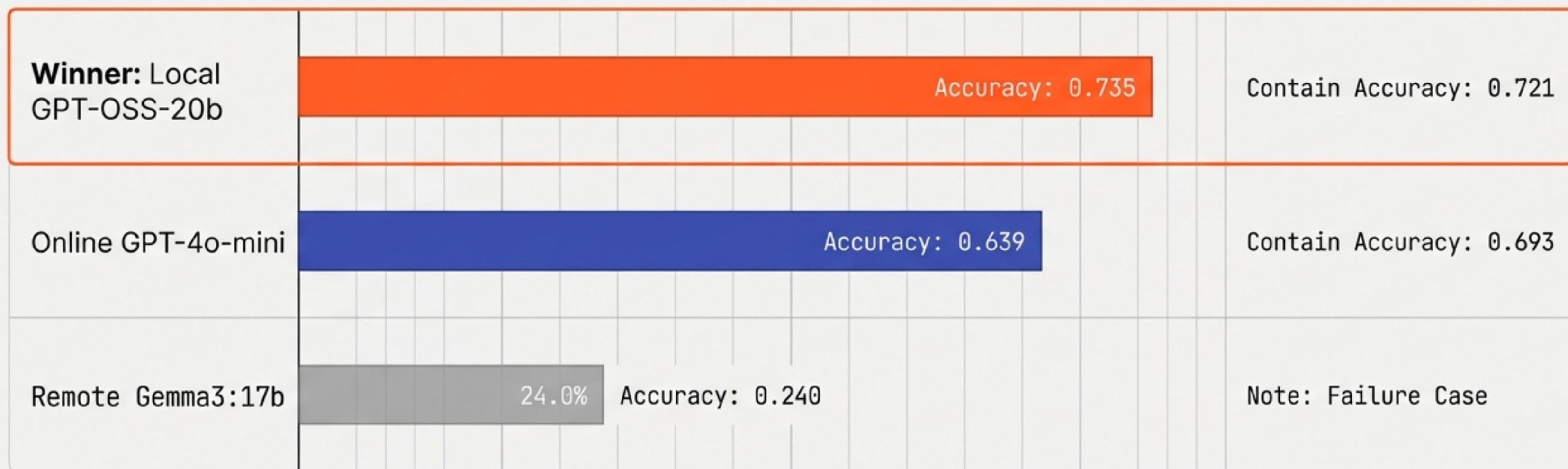
Total Edges

JetBrains Mono

JetBrains Mono

# Benchmarking: Retrieval Performance

Evaluierung auf komplexen Datensätzen (2wikimultihop)



Observation: High 'Contain Accuracy' proves the graph successfully retrieves relevant context even if the LLM sometimes fails the answer generation.

# Lessons Learned & Risikomanagement

Erkenntnisse aus der Implementierung.

## Platform Matters

Windows/MLFlow Inkompatibilitäten und erzwungene Reboots waren massive Zeitfresser.

## Entity Quality

Qualität hängt an Entity Extraction. Problem der Mehrdeutigkeit (z.B. "Müller" als Beruf vs. Name).

## Optimization

ScispaCy proved superior for academic texts due to "Shared Academic Discourse".

## Idempotenz

MD5-Hashing war essenziell für performantes Re-Loading ohne Neuindizierung.

# Fazit & Ausblick

## Die Zukunft ist strukturiert.

GraphRAG überbrückt erfolgreich die Lücke für komplexes Reasoning, wo Vektor-RAG versagt. Ein 80M domänenspezifisches Modell kann grössere Basismodelle schlagen.

## Future Work

- **Hypergraphen:** Clustering identischer Relationen.
- **Raum & Zeit:** Integration von Dimensionen (z.B. Aktienkurse über Zeit).
- **Konzept-Graphen:** Abstraktionsschicht für mehrsprachige Knowledge Bases.

**Die Transformation einer passiven Dokumentensammlung in ein navigierbares Modell der Welt ist der nächste Schritt der KI-Evolution.**