Fluree Report:

Decentralized Knowledge Graphs Enable Most Accurate Generative AI Results

INTRODUCTION

Fluree Report: Decentralized Knowledge Graphs Enable Most Accurate Generative AI Results

It’s been about 18 months since Chat GPT and the power of Generative AI has been unleashed to the world. To date, many organizations have experimented with AI applications, but very few have successfully deployed them into production. Or when rolled out into production, the results (in terms of ROI and productivity benefits) have often been much less impressive than expected.

In fact, Intel’s 2023 ML Insider survey suggests only 10% of organizations put Generative AI solutions into production in 2023.

This paper will describe advances in knowledge graph technology to reduce risk of inaccuracy in LLM applications, enabling their use in the most strategic and differentiating scenarios, and achieving the highest levels of productivity and ROI.

Contents

INTRODUCTION

The Threat of Inaccuracy

While every major organization is experimenting with Generative AI, Production AI has yet to go mainstream, due to a threat of inaccuracy (hallucination, bias, incompleteness, and incorrectness)

The threat of inaccuracy is the result of two significant barriers to enterprise AI: 1) security and privacy concerns limit access to key digital assets, and 2) a lack of grounded truth to prevent hallucinations. Let's explore:

1

Security, Privacy, and Data Access

The absence of fine-grained security and data privacy controls across the data plane prevents the most valuable and unique data assets from being utilized in AI applications.

Since the outputs of AI applications are delivered directly to business end users or customers, bypassing IT safeguards, the consequences of insecure data controls in AI applications can be significant. As a result, most AI applications in production today rely on the generalized knowledge of the LLM or are trained on some of the safest but least strategic data assets.

2

Contextual Grounding in Truth

AI confidently hallucinates answers based on statistical probabilities when the direct answer isn't immediately available. Because it's hard to tell real answers from made-up answers, AI apps today tend to be used for the use cases where an 80% or less accuracy level is tolerable, but those tend to be the less impactful or strategic to the company.

Adding Context Through RAG

RAG adds live sources of authoritative truth, mitigating risk of hallucination.

One of the most predominant methods of reducing risk of AI hallucination is through the introduction of external sources of truth. In this pattern, instead of having the LLMs fully generate a response to a question, the LLMs are trained to find truthful answers in another source and then finish generating the response around the grounded truth it found.

The most common method of enabling real-time interaction with live sources of authoritative truth is through Retrieval Augmented Generation (or “RAG”) training.

With RAG, we teach and prompt LLMs how to retrieve knowledge from external sources outside of its trained corpus of information as part of the generation of the response to a question.

RAG can be used to fetch data from an external source, whether it is a Vector database for unstructured data (where unstructured data has been converted into mathematical vectors to aid indexing and searching), or for structured data – a relational database, data warehouse, data lake or graph database.

Even though RAG limits the ability for the LLM to hallucinate, it can still return incorrect responses if it retrieves the information incorrectly from the external source – i.e. if it fires off an incorrect query to the source. The goal with RAG training is to reduce the errors in query generation through prompting and training.

Further Reading

We’ve previously discussed how Knowledge Graph databases can be used as a powerful external data source when combined with LLMs in this document: LLMs Are Becoming Less Accurate. Here’s Where Knowledge Graphs Can Help.

Research and Findings

LLM Accuracy: Comparing the RAG Approach Across Relational, Graph, and Decentralized Knowledge Graphs

> Fluree conducted research that quantifies how the type of RAG source where the data is saved (i.e., database, data lake, data warehouse, Knowledge Graph database) can have a significant impact in the accuracy and performance of the LLM to retrieve the right answers.

Research Director

Eliud Polanco, President, Fluree

Eliud Polanco is a seasoned data executive with extensive experience in leading global enterprise data transformation and management initiatives.

Previous to his current role as President of Fluree, Eliud was formerly the Head of Analytics at Scotiabank, Global Head of Analytics and Big Data at HSBC, head of Anti-Financial Crime Technology Architecture for U.S.DeutscheBank, and Head of Data Innovation at Citi.

In his most recent role as Head of Analytics and Data Standards at Scotiabank, Eliud led a full-spectrum data transformation initiative to implement new tools and technology architecture strategies, both on-premises as well as on Cloud, for ingesting, analyzing, cleansing, and creating consumption ready data assets.

LLM Accuracy Compared Across Relational, Graph, and Decentralized Knowledge Graphs

Methodology

- We evaluated the performance of LLMs in completing a RAG task by asking Chat GPT-4 a series of 20 questions.

- The correct answers required retrieving data from multiple database systems at inference-time, using sources external to the LLM's original training corpus.

- The questions start simple, where the answer is a query that is self-contained inside one table from the database. But they get progressively harder, where the correct answer is a query that joins data from up to 9 different sources.

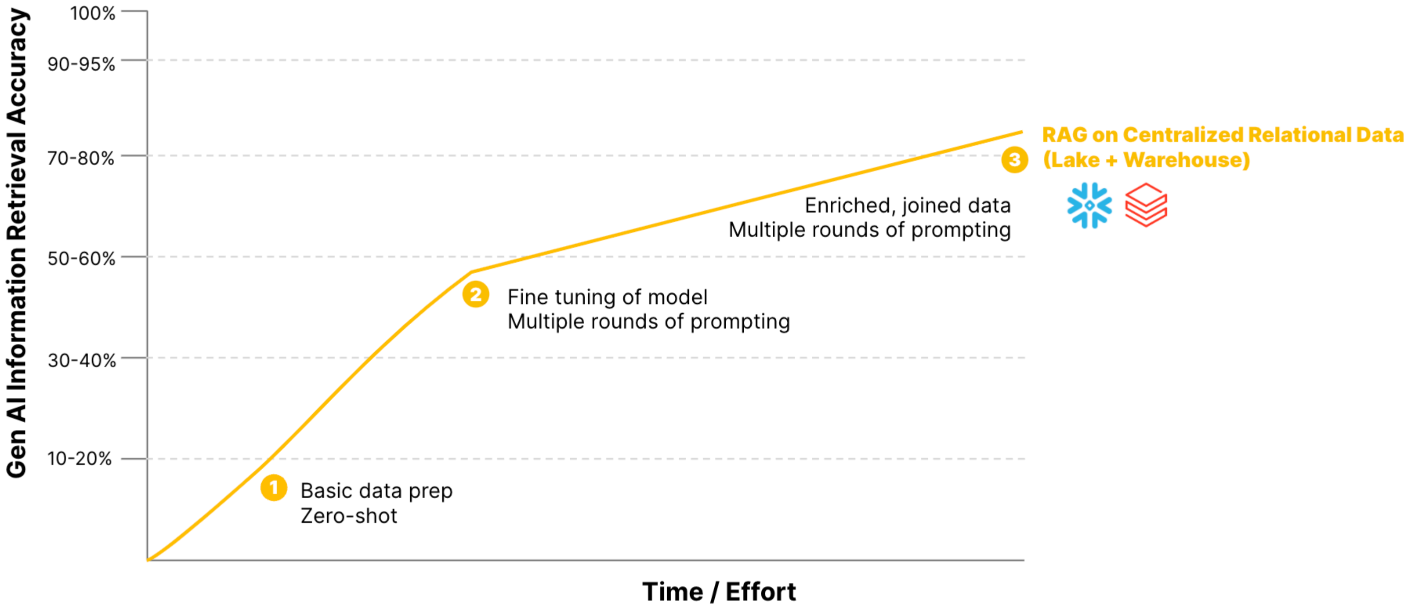
- We evaluated these performances across three levels of data preparation effort: 1) Zero-Shot 2) Fine tuning the model with multiple rounds of prompting and 3) Enriching and joining the data first, as well as multiple rounds of prompting. For rounds 2) and 3), we also introduced 5 new random questions in addition to the baseline 20 to measure post-tuning accuracy on novel queries.

- We performed these evaluations against (1) Centralized Relational Data, (2) Knowledge Graphs and (3) Decentralized Knowledge Graphs.

	Accuracy Averages		
RAG Data Source	**Basic, Zero-Shot**	**Fine-Tuned Model**	**Enriched, Joined Data**
Centralized Relational	8%-15%	50%-60%	80%
Knowledge Graph	60%-65%	70%-80%	80%-90%
Decentralized Knowledge Graph	60%-65%	70%-80%	95%+

Centralized Databases

The below chart measures performance and accuracy of the LLM in completing a series of progressively difficult RAG tasks, versus the level of investment of time, effort, manpower and compute (particularly GPU costs) to train the LLM to complete the task. The further to the right, the more time/effort and resource is required. The higher up, the more accurate the performance is.



1) Zero-Shot

The initial results that can be achieved just through meta-data alone, with zero training or tuning of the model.

2) Fine-Tuning

The results after fine tuning the model and engaging in several rounds of prompting

3) Enriched

The results after doing data integration work to further prep and refine data, pre-join it with other data in views or materialized enriched tables, and then further fine tuning the model with additional prompts on the views or enriched data sets

Centralized Databases

Just by providing the LLM the technical schema (or DDL) that describes the names of tables, columns, constraints or triggers to columns, and join key relationships (e.g., primary and foreign keys), with no further training, on a zero-shot basis the LLMs were able to generate the correct query for 5 of the 20 questions asked (~20% accuracy). This is not that bad! With no effort and just using the LLM's generalized knowledge on how to write SQL it was able to answer some of the easy questions right away.

When we provide proper training and fine tune the model, the accuracy increases by 3x (to ~60% accuracy). For the simpler questions where the answers require a SQL query that joins four or less tables, the answers were typically correct. But if the correct answer required more thoughtful reasoning or joining five or more tables, the responses were still incorrect.

When we pre-join tables, or create denormalized data views that reduce the need for the LLMs to construct complex joins, and then tune the models how to retrieve the data from this new data asset, the results get even better (to 80%), but this has come at a considerable cost and expense.

In essence, this returns to the traditional approach of creating predefined business views or data marts designed to address a specific set of questions. However, this approach is limited to the questions we initially considered. As end users come up with new questions that go beyond the boundaries of the pre-fabricated data, the LLMs will need to go out and join more and more data in real-time, reducing its overall accuracy over the long haul.

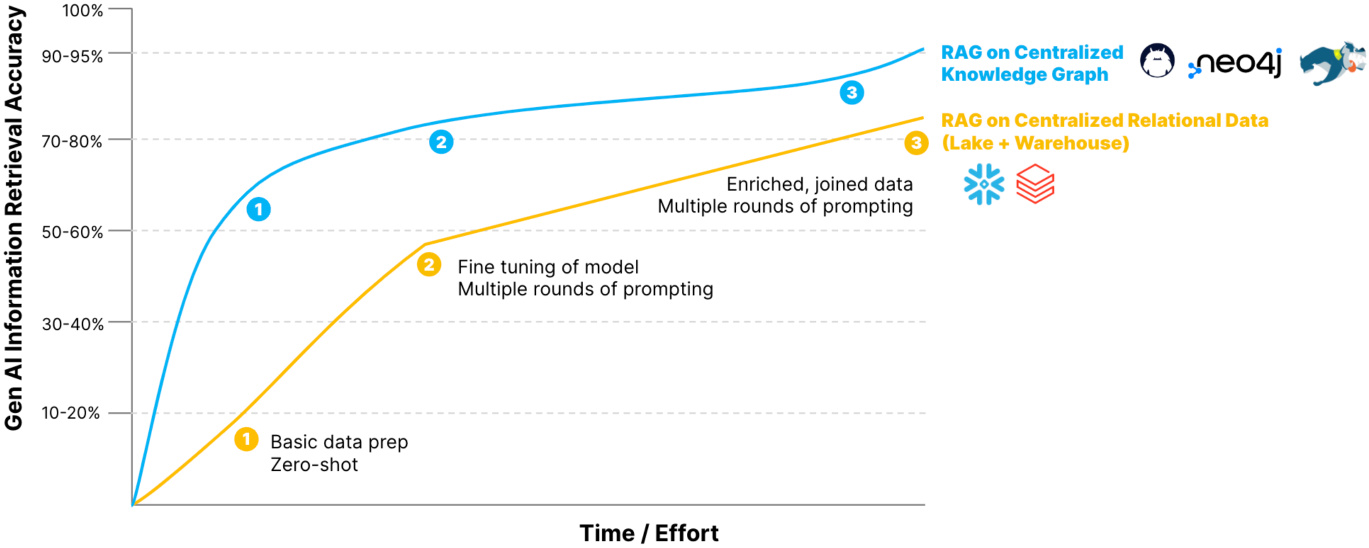
Key Finding

We believe that traditional relational data structures, which require data to be pre-integrated in a central location and joined with other relational tables in real-time to answer end-user queries, are inherently limited and not scalable for real-time AI apps. As a result, they are unlikely to achieve greater than 80% accuracy and utility in production-grade AI applications.

Centralized Knowledge Graph

Moving to centralized knowledge graphs as the external RAG data source, we observed an overall increase in accuracy over relational databases - with a significant improvement starting with zero-shot.

Foundational Reading: What is a Knowledge Graph? (Fluree Blog)



1) Zero-Shot

This time, if we simply represented the technical schema as a semantic ontology (RDF or OWL instead of DDL) and prompted Chat GPT to return back a semantic query (SPARQL instead of SQL), on a zero-shot basis the LLMs were able to generate the correct query for 15 of the 20 questions asked, a 3x performance boost compared to relational!

Furthermore, Chat GPT itself contained the knowledge to convert the DDL into an ontology file. This means that with almost no effort, just by expressing meta-data in a semantic way LLM performance is improved.

Why? LLMs are themselves massive networks of statistical correlations and therefore naturally understand how to deal with data represented as nodes and relationships. There's a hidden synergy between graph & vectors: both are about relationships, but different types of relationships. Graph could lend more precision so results from those vector similarity searches won't be as bizarre.

Baseline RAG, without knowledge graph technology, seems to struggle with connecting semantic concepts.

Centralized Knowledge Graph

Moving to 2) fine-tuning and 3) enriched, semantically-linked data as next stages in data quality, we continued to observe a positive accuracy trend.

2) Fine-Tuning

If we re-model the data from normalized SQL data tables into a proper Knowledge Graph triple store (i.e., we converted the data from relational tables into a format like RDF) and then perform several rounds of training, we then achieved a peak 80% accuracy level from the LLMs.

This was with far lesser effort compared to the relational model. However, we run into similar issues where complex questions that require higher levels of reasoning cannot fully be answered by the data in the Graph, unless we load and convert more and more data into Graph form.

3) Enriched

At this stage we have converted and brought into our Enterprise Knowledge Graph a significant number of datasets that have been semantically linked together and where duplicate entities were resolved. LLM performance peaks at around 95%, which may be good enough for many use cases (but perhaps not the ones that have a critical regulatory compliance impact).

However, that last 5% often requires integrating data that is tough to access or move into the centralized graph due to data privacy or regulatory compliance (cross-border data travel) issues. Or this may require pre-integrating lots of external or market data into the Graph which would take a lot of time and be very expensive.

Key Finding

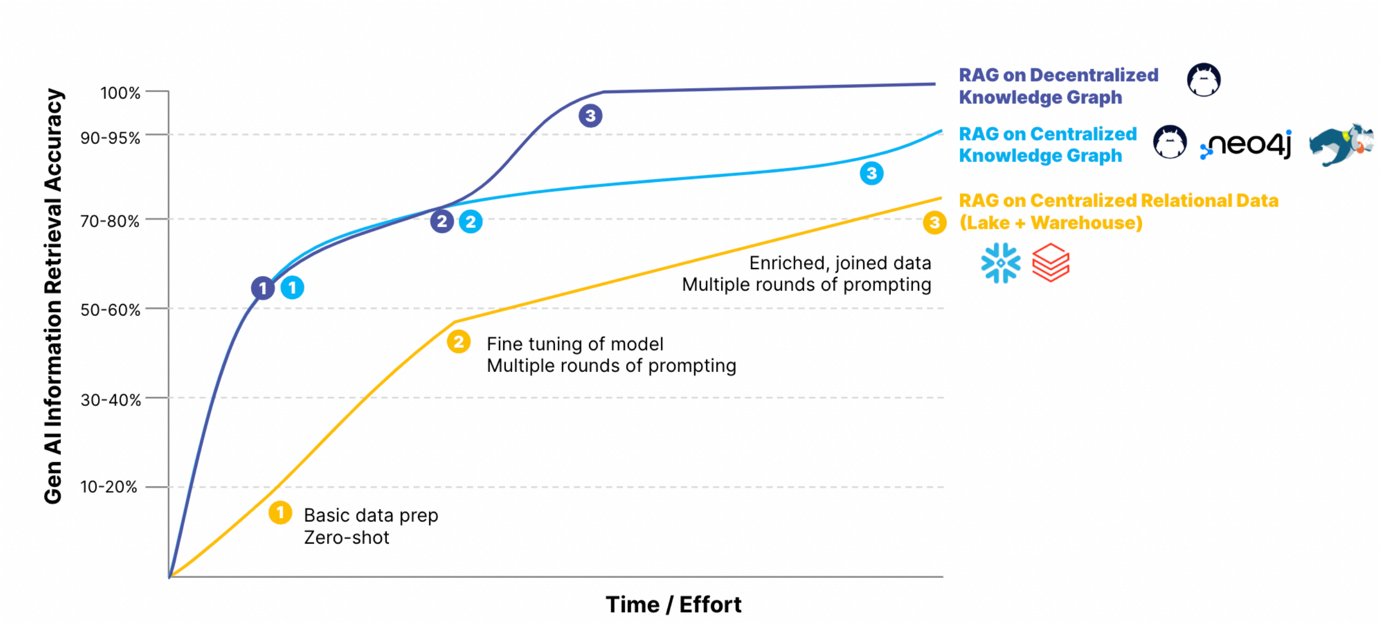
Knowledge Graphs unlock context and meaning of data to LLMs in a very powerful way. However, centralized Knowledge Graphs will also be inherently limited by the issues that have traditionally plagued centralized data lakehouse efforts: it's just not easy or viable to move all the data required to solve enterprise scale problems into a single enterprise graph, whether on premises or on the Cloud.

Further Reading

Numerous reports indicate that graph technology enhances baseline accuracy for RAG approaches, both with zero-shot and fine-tuned data.

- *Microsoft - GraphRAG: Unlocking LLM discovery on narrative private data*
- *VectorHub - Improving RAG performance with Knowledge Graphs*
- *Neo4J - Enhancing the Accuracy of RAG Applications With Knowledge Graphs*
- *Deloitte - Responsible Enterprise Decisions with Knowledge-enriched Generative AI*

Decentralized Knowledge Graph



Analysis

Our Zero Shot and basic fine tuning tests operated very similarly to any semantic Knowledge Graph. The LLMs inherently understand the semantic structure of the data, and the questions can be answered at 80% accuracy for the data loaded into the graph.

However, when asking complex questions that required information from various disparate data sources, we found that decentralized knowledge graphs significantly out-performed centralized sources.

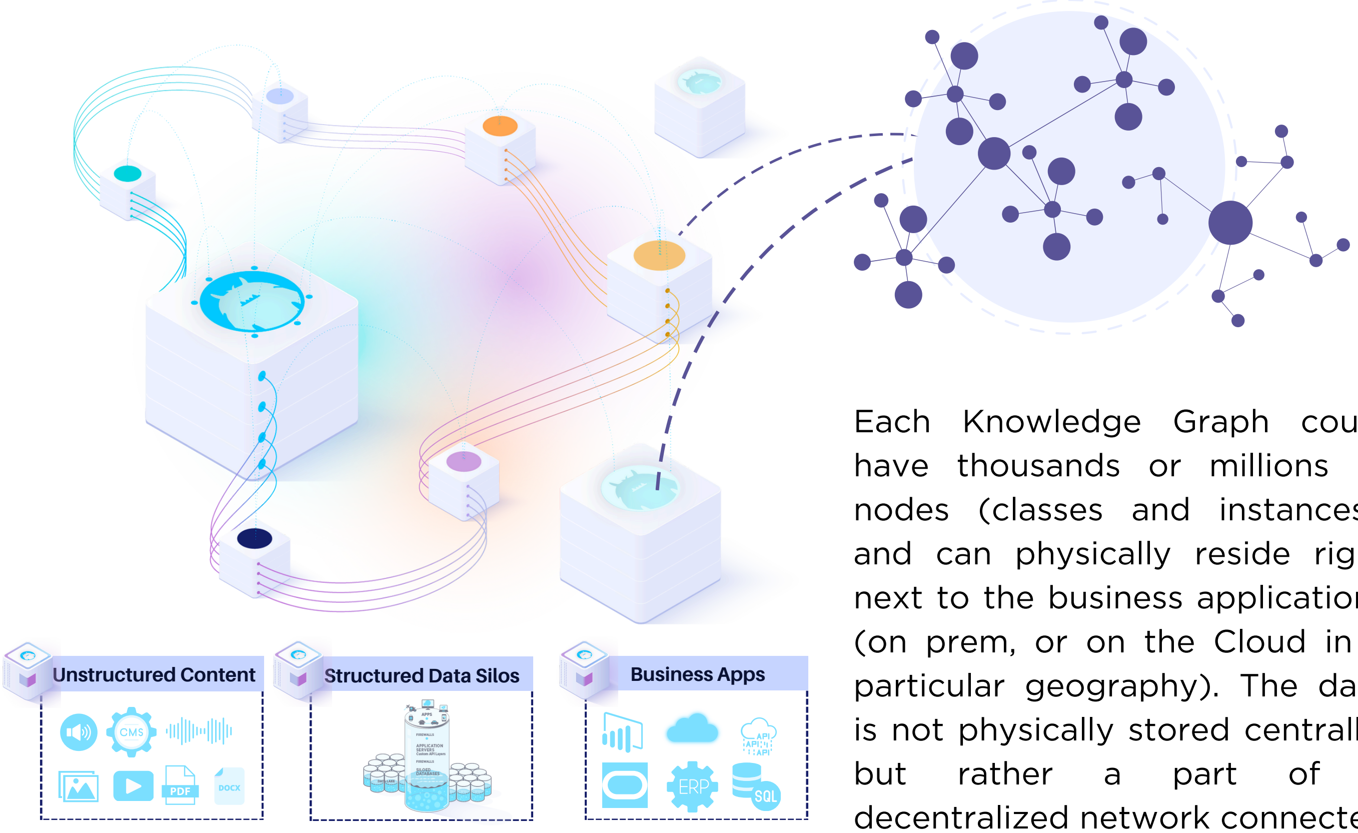
We can use a decentralized fabric of Knowledge Graphs to securely connect to data sources located anywhere, from any cloud, in any geography as long as permitted by policy. With a semantic fabric infrastructure in place that enables data to not have to be moved in order to be linked, we can now achieve a consistent 99-100% accuracy rate on RAG tasks

Key Finding

The mixture of semantic Knowledge Graphs with decentralized technology is the key to achieving enterprise grade AI applications that can be safely deployed into production for even the most rigorous use cases with extremely low-risk thresholds.

What is a Decentralized Knowledge Graph?

A decentralized Knowledge Graph is a network of independently managed Knowledge Graphs that can be connected at query time, based on rights and permissions. Integrated access and usage policies ensure that queries can span across any possible data source and safely and securely access, link, and return decentralized data to the user.



Each Knowledge Graph could have thousands or millions of nodes (classes and instances), and can physically reside right next to the business applications (on prem, or on the Cloud in a particular geography). The data is not physically stored centrally, but rather a part of a decentralized network connected through semantic web standards.

Based on specific questions being asked, if a node in one Graph has a relationship to nodes in any of the other Graphs, if they have the permissions to access specific nodes in the other graphs, then they could be included in the answer to the question.

Think of a decentralized Knowledge Graph like a peer-to-peer network of graphs. The peers could all be inside your company (e.g., the Product Management unit in a company manages its own Graph, the Risk business unit manages its own Graph, the Compliance business unit manages its own Graph, the Finance business unit manages its own Graph, etc.).

Or they could be a hybrid of internal and external peers, such as when you want to mix your own proprietary data with industry or market data.

What are the Requirements to Deploy A Decentralized Knowledge Graph?

Decentralized Graphs are made possible based on the following three requirements / enablers:

Embedded Security Data Policy Contracts

Because data could be anywhere, we need to enable Information Security and Risk teams to be able to apply policy to any piece of data, anywhere on the network, no matter who created it. In a decentralized Graph, control over access to the node is based on Smart Data Contracts, or programmable and executable logic, directly inside the data written using the semantic model itself.

The Contract could be written by someone at the peer node level (e.g., "no one other than members in this specific Deal team should see this") or at the network level (e.g., "no person who is not a citizen from Italy should be able to see personal information about Italian citizens"). In the latter instance, what is meant by an Italian citizen, or what constitutes personal information would all be defined in the semantic model in the Knowledge Graph that would be used by all members of the Knowledge Graph decentralized network.

Immutable, Verifiable Data

Because any party who is a member of the network could potentially provide statements of truth in our network of trusted sources, we need to know who everyone is, and we need to log anytime anyone touches or tampers with data. This means having verifiable credentials, immutable data with a unique ID so that we can trace that piece of data if it gets used anywhere in the network, and a persistent change log with the ability to see how any piece of data has changed over time.

Think about how we can always go back and audit any Wikipedia entry, see who changed any entry, roll-back or roll-forward changes. Now think about the ability of this being applied to any cell of data in the network.

Lineage and Attribution

Because the answer could come from any node in the network, lineage and attribution of data is especially critical and important. For us to trust the answer from a question, we need to know exactly who in the network asserted the statement of truth, and when they asserted it.

CONCLUSION

Decentralized Knowledge Graphs Are Key Enablers for Complex, Disparate Data Ecosystems.

Decentralized Knowledge Graphs are the key to creating the most comprehensive corpus of information required to address the most sophisticated Generative AI use cases that will return the highest productivity increases and net ROI.

This is because, contrary to intuition, decentralized Knowledge Graphs enable more security and control over data compared to roles or attribute-based access control present in traditional databases or centralized graph technologies.

Decentralized Knowledge Graphs are also the sustainable choice for ROI: as data volumes grow in size and complexity across distinct domains, Decentralized Graphs are well equipped to support emerging LLM business cases in stride.

Fluree – The Only Platform that Enables Access to Trusted Decentralized Data with Semantics and Policy

Why Fluree?

The Fluree data protocol and management platform was designed from the ground up to enable companies to safely and securely connect and integrate cleansed, trusted data from any source, whether internal or external.

Instant Semantic Interoperability

Fluree stores data using JSON-Linked Data (JSON-LD), a lightweight format for saving application data that facilitates semantic relationships and linkage. With around 40% of the World Wide Web using JSON-LD, it is both a format that developers find intuitive and that LLMs can readily interpret due to its prevalence in training data.

Embedded Trust and Security

Additionally, the Fluree data protocol captures and stores lineage metadata (e.g., creator, creation time, change history) and secure Data Contracts using open-source decentralization standards like Verifiable Credentials and Decentralized Identifiers.

Ready to Learn More?

Production enterprise AI requires accuracy, reduced errors, and compliance with global privacy and copyright laws. Fluree uniquely offers trusted, verifiable data management and built-in programmable policy control.

With a decentralized knowledge graph approach, we enable you to build an Enterprise Corpus of data that can easily link all of your proprietary data anywhere in the world, dynamically connect it with industry and market data, and safely expose it to real-time LLM applications.

Visit us at: http://flur.ee

Contact us at: sales@flur.ee