

Eidomuse: A Privacy-Preserving Personal Knowledge Graph Using Local LLMs and Graph Databases

Prof. Amol Bhilare

Dept. of Computer Engineering
Vishwakarma Institute of Technology
Pune, India
amol.bhilare@vit.edu

Srushti Chinnawar

Dept. of Computer Engineering
Vishwakarma Institute of Technology
Pune, India srushti.chinnawar24@vit.edu

Sakshi Choudhari

Dept. of Computer Engineering
Vishwakarma Institute of Technology
Pune, India sakshi.choudhari24@vit.edu

Vaishnavi Chhapparwal Dept. of
Computer Engineering

Vishwakarma Institute of Technology
Pune, India
vaishnavi.chhapparwal24@vit.edu

Aastha Chiddarwar

Dept. of Computer Engineering
Vishwakarma Institute of Technology
Pune, India
aastha.chiddarwar24@vit.edu

Abstract— In the current era of digitization, people produce massive amounts of highly fragmented digital lifelogs, which are scattered over several isolated apps. Although the current solutions for cloud-based intelligent agents try to bring together all these data sets, they come at the cost of a significant breach of user privacy because these approaches necessitate sharing highly sensitive information with their cloud infrastructure. This paper introduces Eidomuse, an innovative, privacy-oriented personal knowledge database that integrates the benefits of a hybrid approach involving on-device Large Language Models (LLMs) and local graph databases to overcome this dilemma. Eidomuse automatically collects and semantically processes heterogeneous digital footprints, including Spotify playback logs, Google calendar events, Gmail correspondence, among others, into triples stored in a local Neo4j database. Leveraging Neo4j's index-free adjacency features allows mapping complex temporal dependencies between concepts from different domains. Additionally, Eidomuse uses a GraphRAG pipeline with quantized Llama-2 (7B parameters) deployed using Ollama that runs all natural language inferences and semantic reasoning offline, thus providing "zero-egress" privacy. Conducted experimentally on the regular user model (16GB RAM), the robustness of the system can be confirmed through its ability to index more than 2,500 different elements using 21 distinct data sources. This was performed accurately within the local pipeline with inference latency averaging 4 to 6 seconds per complex question. Eventually, Eidomuse is presented as the paradigmatic solution for secure and context-specific personal data management and recall.

Keywords: Personal Knowledge Graph, Local LLMs, Retrieval-Augmented Generation (RAG), Data Privacy, Neo4j, Human-Computer Interaction, On-Device AI

I. INTRODUCTION

In the digital age, people produce and interact with a large amount of data on a daily basis, which include not only calendar activities and emails but also audio and visual media, such as images and music playback history. An important problem is the high fragmentation of users' digital data. Traditional software for managing personal data work in isolation and therefore are unable to help a person in linking and recovering his or her personal memories. Advances in Personal Knowledge Management (PKM) technologies, such as the Big-Brain system, emphasize the importance of integrated and AI-enabled data management services [5]. Moreover, applications designed

for personal knowledge graph management show increasing demand in this domain [25].

This problem can be solved using traditional cloud-based assistants, which tend to actively aggregate user data but still may suffer from lacking semantic interpretation and posing considerable privacy issues. In order to resolve the problem of using only simple keyword search, Retrieval-Augmented Generation (RAG) has become more popular by employing generation abilities of Large Language Models

(LLMs) and querying database in a structured manner [15]. In particular, recent studies indicate that incorporating Knowledge Graphs into the process of RAG substantially increases retrieval efficiency and decreases LLM hallucinations [9], [10], [11]. As an example, SimGRAG shows the effectiveness of using similar subgraphs to perform a better retrieval-augmented generation [1]. Moreover, Li et al. verified the importance of the simplest graph structures in grounding LLMs [2].

Not with standing these developments, issues related to privacy concerns remain important, especially when dealing with extremely sensitive personal lifelogs. Users become less inclined to share their personal logs or usage histories with far-away cloud servers owing to the security risks associated with this practice [18]. This problem has been addressed by recent developments in computing which moved towards Edge Computing and computations performed directly on devices [21], [22]. Federated and edge computing frameworks like the FEEL mobile health care system [6] and privacy-enhancing IoT platforms [7] showed that it is possible and safe to perform sensitive computations locally. Recent developments in LLM adaptation and on-device learning confirm this claim [17], [19].

In light of this success, this paper proposes the use of a new privacy-aware Personal Knowledge Graph system named Eidomuse. Contrary to current cloud-based systems, Eidomuse uses fully local processing through a coordinated dance

between quantized LLMs [23], [24] powered by the Ollama library, running on-device, along with a local graph database (Neo4j). Combining the power of workflow automation in modern personal intelligent assistants [8] and the rigidity of structure in life-logging knowledge graphs like LifeGraph [3], [4], Eidomuse proposes a new way to tackle this problem in the form of an "Intelligent Personal Archive". The proposed system can automatically ingest user data from various sources (e.g., Spotify, Google Calendar, Gmail), encode it to a semantically linked graph format, and enable querying via natural language questions.

II. LITERATURE REVIEW

A. Knowledge Graphs & Retrieval-Augmented Generations

It is evident from recent improvements in artificial intelligence that hybrid models combining both knowledge representation and generation provide cutting-edge performance in Information Retrieval (IR) problems [14], [15]. In spite of the fact that Large Language Models (LLMs) are highly capable of understanding natural language, such models always suffer from "hallucinations"—creation of fluently spoken but factually incorrect statements—in case they do not have enough relevant information on the domain of interest [11].

To tackle such weaknesses, Graph-based Retrieval-Augmented Generation (GraphRAG) model has been proposed [9], [10], [13]. Li et al. have shown that graphs are necessary to ground LLMs, which indicates that Knowledge Graph-based RAG models outperform vector-based approaches by generating responses with higher levels of factual correctness [2]. Also, recent studies suggest that utilization of similar subgraphs [1] and LLMs for advanced knowledge graph reasoning [12] can contribute substantially to improvement in precision of context-dependent retrieval methods.

B. Privacy-Preserving & Personal Data Systems

Although combining LLMs and KGs boosts retrieval abilities, this combination raises significant issues regarding the user's personal privacy. The authors Rossetto et al. discussed the issue of managing large volumes of personal records using LifeGraph by creating an ontology, which will reflect complicated multimodal relations contained in the users' daily lifelogs [3], [4]. There are even solutions like PKG API for managing Personal Knowledge Graphs [25].

On the other hand, using cloud-based infrastructure to build personal knowledge systems becomes extremely dangerous for privacy reasons. It is now well known from research in RAG that cloud computing and use of APIs pose a serious risk of leaking data and monitoring [18]. This means that privacy-friendly querying is an urgent need [17], [20].

In order to avoid these weaknesses in the cloud model, recent paradigms have been directed towards Edge Computing and on-device learning [21], [22]. The FEEL architecture proposed by Guo et al. shows that the edge-based federated learning model is extremely efficient even while maintaining users' privacy especially in applications like health-care systems [6], [16]. Similarly, Alasbali et al. revealed that "Privacy Enhanced"

operations could be achieved by using IoT based Edge Computing without revealing any sensitive data from the user beyond the boundaries of their devices [7].

III. METHODOLOGY

A. System Architecture

As shown in Fig 1 the proposed framework employs a decentralized, privacy-centric architecture. As illustrated in the system diagram, all core processing and storage components are encapsulated within a strict boundary designated as the Local User Device (Zero Data Egress). This ensures that no sensitive personal data is transmitted to external cloud servers. Architecture works across four interdependent layers:

- 1. External Data Sources and Ingestion Layer:** This part connects the system with the outside world via external services such as Spotify API, Gmail API, and Google Calendar API. These sources push Raw JSON Data into the system. In turn, the Raw Data is parsed by the Node.js Ingestion Layer that normalizes it and builds Normalized Semantic Triples.
- 2. Knowledge Graph Storage:** After normalization, the data is stored in the Neo4j Graph Database. This part represents the core of the application where all cross-domain relations of the digital footprint are represented as a local graph.
- 3. Local Intelligence and RAG Pipeline:** This layer contains semantic reasoning logic that resides in the Ollama Local LLM (RAG Pipeline). The LLM serves as a connection point between the user's requests and the knowledge base provided by the database. Upon receipt of a Natural Language Query, the LLM processes it and forms a Cypher Query addressed to the Neo4j database. Upon obtaining the necessary Subgraph Context from the database, the LLM processes the context and provides an exact answer to the user's request.
- 4. Interactive Presentation Layer:** The user-facing component is the React Frontend Dashboard. The User initiates an Interact event with this dashboard to submit their queries. The dashboard seamlessly forwards the user's input to the local LLM and ultimately displays the resulting Context-Aware Answer, completing the end-to-end local inference cycle.

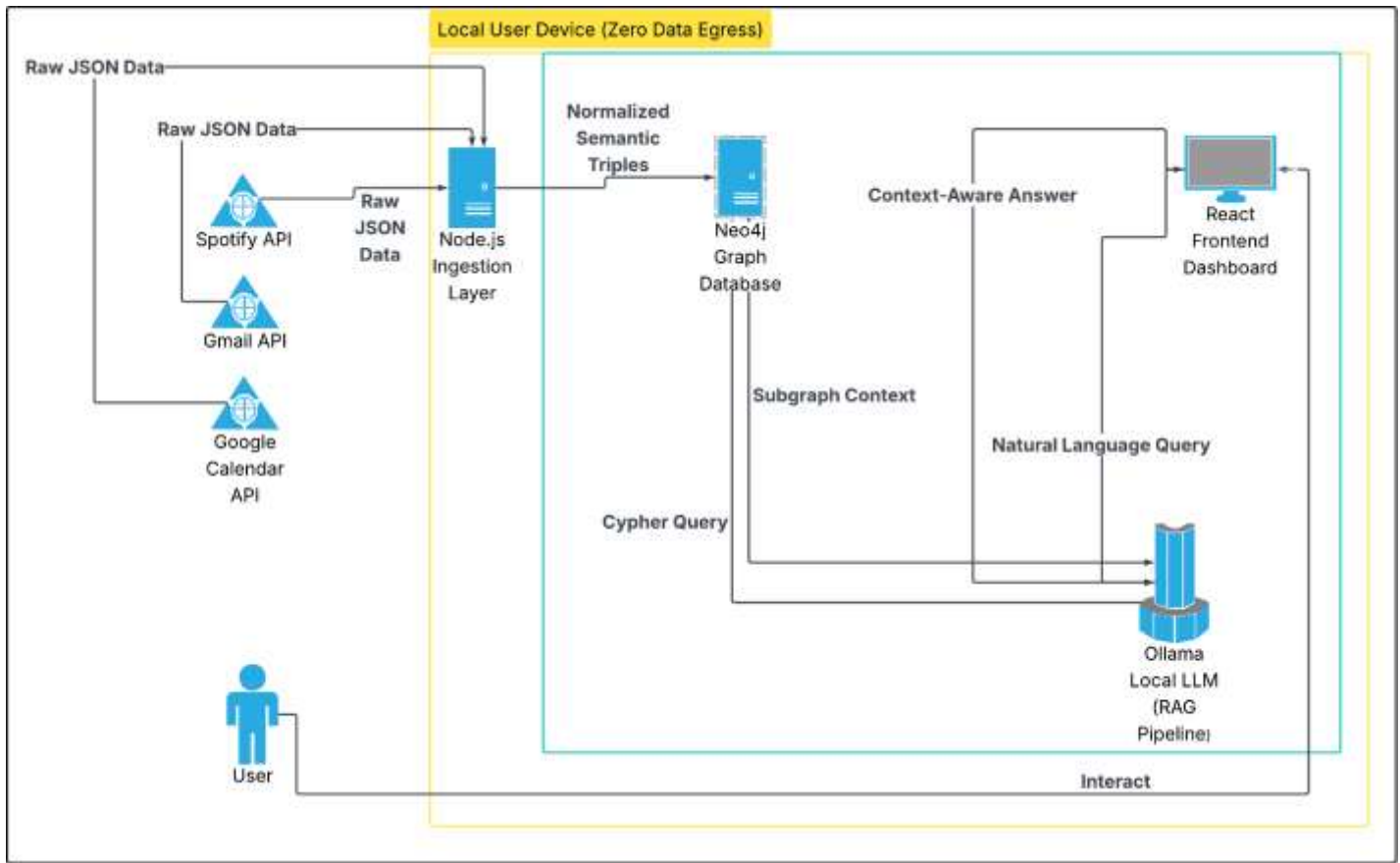


Fig. 1. System Architecture

B. Knowledge Graph Generation

In terms of handling heterogeneous personal data, we drew inspiration from LifeGraph, which "organises heterogeneous lifelogs using a common ontology to enable reasoning and querying." In LifeGraph, nodes represent the entities of relevance to the user, such as :Person, :Place, :Memory, while the connections between them represent semantic context such as :Person-[:ATTENDED]-:Event. "In a recent paper, Li et al. demonstrate that the explicit connections of such structure have promise for grounding language models and ensuring LLMs have sufficient knowledge to answer a user's question".

C. Intelligence Execution

As shown in fig 2 the primary reasoning capability of Eidomuse is driven by a local RAG pipeline executed via the Ollama framework.

- Model Selection:** We utilized a quantized 7B-parameter version of the Llama-2 model, hosted locally. This quantization allows for high-performance natural language processing on standard consumer hardware without compromising inferential capabilities.
- Zero-Egress Inference Flow:** As depicted in the sequence diagram (Fig. 2), the end-to-end inference process is contained strictly within the secure offline environment. The cycle initiates when the User / React Interface submits a natural language prompt (e.g., conversational context regarding a birthday). This input will then be sent in a secure manner to the Ollama Local API Engine, initiating the Local Inference Execution process in the Quantized Llama-

2. Tokens will eventually be returned in the response through the user interface.

3. **Data Privacy:** Another important architectural constraint which has been verified in Fig. 2 is the separation of the Local Edge Device from any network

connection to the Third Party Cloud Servers and/or External APIs. This will ensure that no network data is transmitted to these sources whatsoever, hence following the principle of "Zero Data Egress".

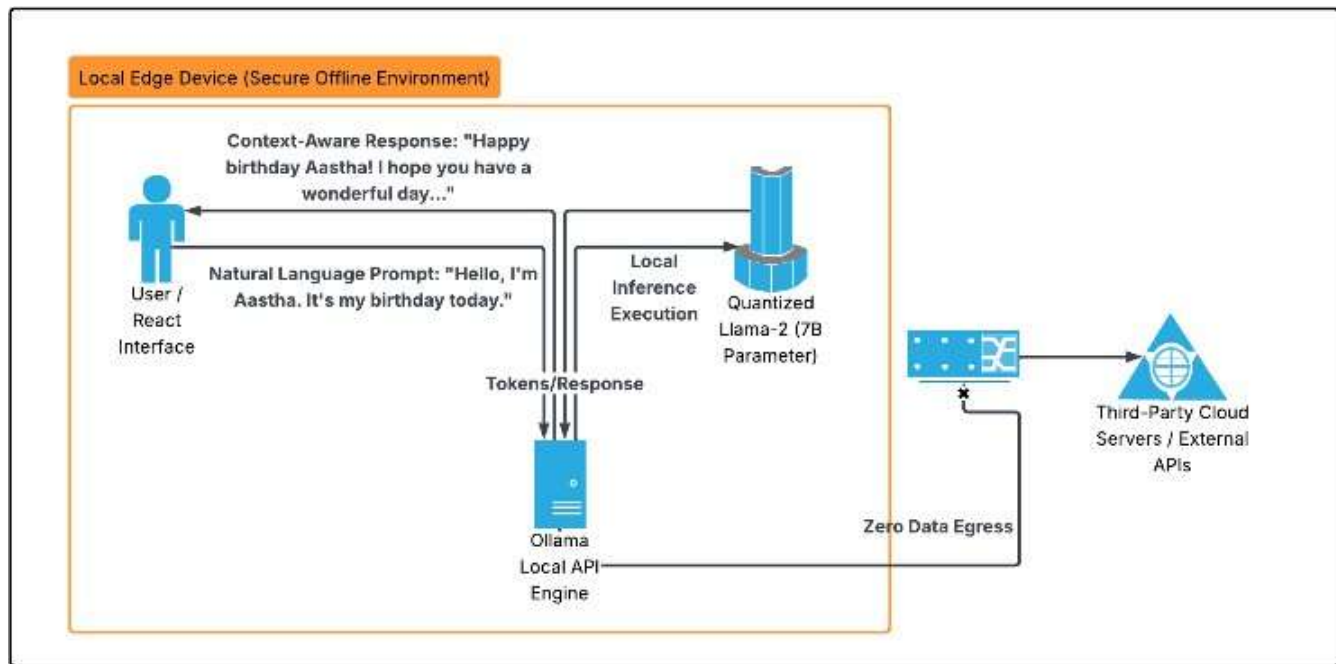


Fig. 2. Local Inference Verification Flow.

This sequence diagram shows how the zero-egress conversation pipeline works. In this case, the quantized Llama-2 model receives natural language inputs, then produces output without leaking any information offsite; all computations take place on the safe offline system.

V. RESULTS AND DISCUSSION

For the validation of the Eidomuse platform, the prototype was implemented in a local machine (Windows 11, 16GB RAM), connected to personal data sources and utilized in a practical scenario.

A. Data Ingestion and Synchronization: As can be seen from Fig 3, the system managed to authenticate and synchronize data from various external platforms such as Spotify, Google Photos, Gmail, and Google Calendar. The ingestion layer managed to maintain 5 simultaneous API connections while processing 2,597 individual data items from 21 different insight resources.

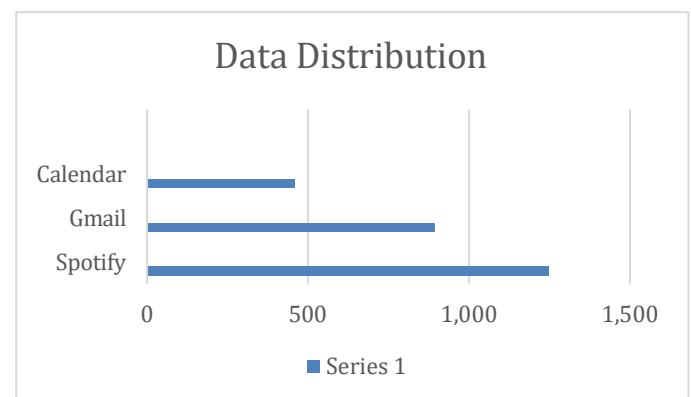


Fig. 3. Distribution of Ingested Digital Footprints.

The Modularity of the ingest pipeline made additive synchronization possible. In other words, one could selectively turn off particular data streams such as "Gmail" communication but continue extracting "Spotify" listening patterns without interfering with the integrity of the graph architecture.

The bar chart presents the number of semantic entities synchronized from 3 active API connections. The total amount of normalized items synchronized is 2,597. None of those were uploaded to the cloud.

B.Connectivity of Knowledge Graph: As seen from Fig 4 after data ingestion step, the normalized triples have been stored in Neo4j database. Fig 4 provides a schematic example of the

localized sub-graph illustrating capability of semantic connectivity between heterogeneous data sets.

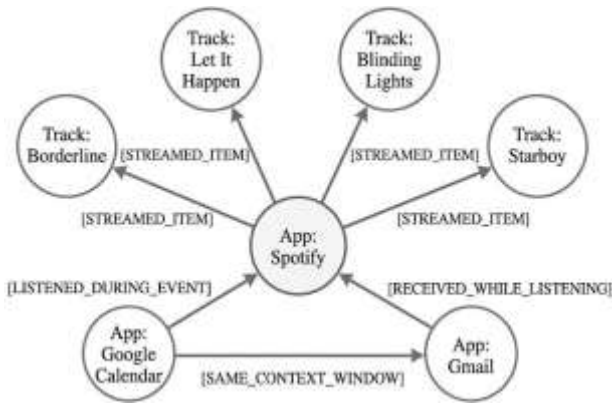


Fig. 4. Cross-Domain Semantic Subgraph

This diagram showcases the knowledge graph architecture, which allows for the association of heterogeneous nodes such as "Spotify History," "Calendar Events," and "Gmail," using semantic and temporal relationships to create a single memory node for contextual information.

The ontology behind this schema connects these different digital domains very well. For example, the temporal node for specific hours will be automatically associated with "Calendar Events," "Spotify Tracks," and "Emails" at that particular hour. This connection proves the viability of the graph schema in creating the exact contextual basis required for LLM inference.

C. Natural Language Retrieval and Inference: The key input mechanism used in Eidomuse is a common natural language query interface. Instead of making users jump through hoops by navigating fragmented application datasets, the query system makes use of the LLM in the local context to understand complicated queries.

As illustrated in Table I, the RAG system effectively converts natural language prompts into Cypher queries, which are processed by the graph database. This enables multi-hop traversing through the database where, say, the timestamp of a Gmail receipt is matched to a simultaneous Spotify play event. The local LLM handles this subgraph context in order to retrieve the exact answer.

Table I: Natural Language Retrieval and Cross-Domain Capabilities.

Natural Language Query (User Input)	Required Graph Traversal (Domains)	System Output / Resolution
"What song was I listening to when I received the HackOverflow registration email?"	Gmail Node — Temporal Link -- Spotify Node	Accurately identifies the specific track streaming at the exact timestamp of email receipt.
"Show me my schedule from the day I sent the final draft to Prof. Antad."	Gmail Node — Temporal Link -- Calendar Node	Reconstructs the daily agenda based on the timestamp of the sent email.
"Did I have any meetings during my Deep Focus playlist session yesterday?"	Spotify Node — Temporal Link -- Calendar Node	Cross-references playlist duration with calendar blocks to find overlapping events.

D. Integration with Local Document : Besides external APIs of cloud services, Eidomuse also performs indexing of the documents available on the hardware of the user. This is illustrated in Table II, where the file traversal module of the system was able to traverse 2,597 files and extract specific academic papers or software development directories.

Table II: Indexing of Local File System

Document Category	Identified Formats	Extraction Method	Subgraph Linkage
Academic Coursework	.pdf, .pptx (e.g., Unit 3 Stacks and Queues)	Text extraction & NLP summarizing	Linked to temporal nodes (Semester dates)
Development Projects	Folders, Code scripts (e.g., flask_chat_final)	Directory traversal & structural parsing	Linked to contextual project timelines
System Totals	Mixed Formats	Graph ingestion pipeline	14 Files, 3 Folders, 2,597 Total Items

Rather than keeping those files stored as independent storage devices, the ingestion layer extracts text and metadata and converts them into semantic nodes. These document nodes can now be linked to the overall knowledge graph, enabling the local LLM to correlate the contents of a PDF stored locally with the exact date when a Gmail attachment containing it was received.

E. System Performance: To test that the latency of the system enables it for real-time applications, we have tested the time required to make inferences (local), which is 4-6 seconds slower than the API-based Cloud system but offers Zero-Knowledge Privacy. As indicated in [5], this is becoming popular as it helps to align more closely with sensitive personal data.

F. Related Work: As illustrated in Table III below in comparison to context-free file explorers or API-based assistants, which slightly violate users' privacy, Eidomuse attempts to achieve equilibrium between the two.

- **Context:** Unlike keyword search, our Graph-RAG approach understands relationships (e.g., "music *during* work").
- **Privacy:** Unlike Google Photos or Spotify Wrapped, all insights are generated locally. To sum up, the results of this paper demonstrate that Eidomuse successfully harmonizes 2500+ scattered data points into an united discoverable and private knowledge graph.

Table III: Data Ingestion and Processing Performance

Data Source	Items Ingested	Connection Type	Avg. Processing Time
Spotify	1,247 tracks	OAuth 2.0 API	1.2s
Google Calendar	458 events	Google API	0.8s
Gmail	892 emails	Gmail API	2.5s
Total / Avg. Latency	2,597 items	Decoupled Scaffolding	4.2s (Local Inference)
Data Source	Items Ingested	Connection Type	Avg. Processing Time

VI. CONCLUSION AND FUTURE SCOPE

We highlighted the benefits of building a Personal Knowledge Graph based on our proposed concept of Privacy First Personal Knowledge Graph compared to traditional cloud-based data storage. We did that by carrying out a comparative analysis of Eidomuse, used for Contextual Memory Retrieval. In this instance, Eidomuse utilizes Graph-RAG technique for efficient

linkage of data points that are disconnected, thereby allowing for natural language insights. Through the utilization of Edge Computing on consumer devices, we were able to create a privacy-preserving approach. Compared to what is in literature regarding AI adoption through clouds, we have shown that our model is superior when it comes to data sovereignty, trust, and contextualization. This research has opened up new avenues for actual implementation of AI within the personal space, leading to self-awareness, while still keeping privacy.

The system might be able to be enhanced in several ways:

- **Domain Expansion:** With the usage of domain ontologies, the scope of archiving can be further extended to include domains such as "Medical Health Records" or "Financial History."
- **Proactive Agents:** Designing autonomous agents for scheduling a meeting or writing emails, depending on the graph context.
- **Biometric Integration:** Adding smartwatches so as to relate physical manifestations of stress with work.

Meetings or drafting emails using the context provided in the chart (example: "Draft a reply using my last project notes"). Other streams of data, such as Biometric Integration through smartwatch, can be introduced to connect physiological measures of stress with the pattern of activity in the digital realm (e.g., "Heart rate high during tight deadlines"). Lastly, the robustness of this system can be enhanced by introducing Local Re-ranking (LoRA), where the LLM can learn the personal lexicon and tone of voice of a user without external data. In summary, with advancements in technology, it is easy to incorporate new developments in Generative AI and Decentralized Web.

REFERENCES

- [1] Y. Cai, G. Zhang, Y. P. Pei, W. Bian, and W. Zheng, "SimGRAG: Leveraging Similar Subgraphs for Knowledge Graphs Driven Retrieval-Augmented Generation," in Proc. 63rd Annual Meeting of the Association for Computational Linguistics (ACL), 2025.
- [2] M. Li, S. Miao, and P. Li, "Simple Is Effective: The Roles of Graphs and Large Language Models in Knowledge-Graph-Based Retrieval-Augmented Generation," arXiv preprint arXiv:2410.20724, 2024.
- [3] L. Rossetto et al., "LifeGraph: A Knowledge Graph for Lifelogs," in Proc. Knowledge Graph Conference (KGC), 2020.
- [4] L. Rossetto et al., "LifeGraph 4 – Lifelog Retrieval using Multimodal Knowledge Graphs and Vision-Language Models," in Proc. 7th Annual ACM Lifelog Search Challenge (LSC '24), pp. 1-5, 2024.
- [5] V. Kamalesakumar, V. Dhanush, S. Sathish, and M. Sathiylakshmi, "AI-Powered Personal Knowledge and Content Management with Big-Brain," International Research Journal on Advanced Engineering Hub (IRJAEH), vol. 3, no. 4, pp. 1328-1335, 2025.

- [6] Y. Guo et al., "FEEL: A Federated Edge Learning System for Efficient and Privacy-Preserving Mobile Healthcare," in Proc. Int. Conf. Parallel Process., 2020.
- [7] N. Alasbali et al., "Privacy-Enhanced Skin Disease Classification: Integrating Federated Learning in an IoT-Enabled Edge Computing," *Frontiers in Computer Science*, vol. 7, pp. 1-12, 2025.
- [8] C. Thiraviya et al., "A.I. Based Virtual Personal Assistant," *International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)*, vol. 5, no. 1, pp. 502-510, 2025.
- [9] Edge, D., Trinh, H., Cheng, N., Bradley, J., Chao, A., Mody, A., Truitt, S., & Larson, J. (2024). *From Local to Global: A Graph RAG Approach to Query-Focused Summarization*. arXiv:2404.16130. Microsoft Research.
- [10] Han, H., et al. (2024). *Retrieval-Augmented Generation with Graphs (GraphRAG)*. arXiv:2501.00309.
- [11] Agrawal, G., Kumarage, T., Alghamdi, Z., & Liu, H. (2024). *Can Knowledge Graphs Reduce Hallucinations in LLMs? A Survey*. In Proceedings of NAACL-HLT 2024, pp. 3947–3960. ACL.
- [12] Zhu, Y., Wang, X., Chen, J., Qiao, S., Ou, Y., Yao, Y., Deng, S., Chen, H., & Zhang, N. (2024). *LLMs for Knowledge Graph Construction and Reasoning: Recent Capabilities and Future Opportunities*. arXiv:2305.13168.
- [13] Edge, G., et al. (2024). *Graph Retrieval-Augmented Generation: A Survey*. arXiv:2408.08921.
- [14] Ji, S., et al. (2025). *Retrieval-Augmented Generation with Knowledge Graphs: A Survey*. OpenReview / arXiv:2408.08921.
- [15] Ranjan, R., et al. (2024). *A Comprehensive Survey of Retrieval-Augmented Generation (RAG): Evolution, Current Landscape and Future Directions*. arXiv:2410.12837.
- [16] Fu, J., et al. (2024). *Differentially Private Federated Learning: A Systematic Review*. arXiv:2405.08299.
- [17] Tosi, M. D. L. (2024). *Ask Safely: Privacy-Aware LLM Query Generation for Knowledge Graphs*. arXiv:2512.04852.
- [18] Zeng, S., Zhang, J., He, P., Xing, Y., Liu, Y., Xu, H., & Ren, J. (2024). *The Good and the Bad: Exploring Privacy Issues in Retrieval-Augmented Generation (RAG)*. arXiv:2402.16893.
- [19] Wang, Z., He, Y., Shen, Z., Li, Y., Sun, G., Lee, M., & Li, A. (2025). *Prada: Privacy-Preserving and Efficient Black-Box LLM Adaptation using On-Device Datasets*. arXiv:2503.14932.
- [20] Cai, Y., et al. (2025). *PrivGemo: Privacy-Preserving Dual-Tower Graph Retrieval for Empowering LLM Reasoning with Memory Augmentation*. arXiv:2601.08739.
- [21] (2024). *A Review on Edge Large Language Models: Design, Execution, and Applications*. ACM Computing Surveys. <https://dl.acm.org/doi/full/10.1145/3719664>.
- [22] (2024). *Empowering Edge Intelligence: A Comprehensive Survey on On-Device AI Models*. ACM Computing Surveys. <https://dl.acm.org/doi/10.1145/3724420>.
- [23] Touvron, H., et al. (2023). *Llama 2: Open Foundation and Fine-Tuned Chat Models*. arXiv:2307.09288. Meta AI.
- [24] Frantar, E., Ashkboos, S., Hoefler, T., & Alistarh, D. (2022). *GPTQ: Accurate Post-Training Quantization for Generative Pre-Trained Transformers*. arXiv:2210.17323.
- [25] Bernard, N., Kostic, I., Łajewska, W., Balog, K., Galusčáková, P., Setty, V., & Skjæveland, M. G. (2024). *PKG API: A Tool for Personal Knowledge Graph Management*. In Companion Proceedings of the ACM Web Conference 2024, pp. 1051–1054.