AI-Powered Knowledge Bases in Healthcare, Finance, and Technical Support

Nan Wu

Milpitas, CA, USA
Chris.wunan88@gmail.com

Abstract:
This paper investigates the architecture and methodologies of modern AI-powered knowledge bases, emphasizing their role in delivering accurate, scalable, and context-aware responses across domains such as healthcare, finance, and technical support. We analyze core components—including knowledge sources, retrieval mechanisms, LLMs, and workflow orchestration—alongside associated techniques like embedding-based retrieval, prompt engineering, and human-in-the-loop processing. Using real-world systems and implementations as reference points, we present a comparative evaluation highlighting best practices for building reliable retrieval-augmented generation (RAG) pipelines tailored to domain-specific requirements and performance constraints.

Keywords: AI Knowledge Bases, Retrieval-Augmented Generation, Large Language Models, Healthcare Informatics, Vector Databases, Prompt Engineering, Human-in-the-Loop Systems.

Funding Declaration:
The author declares that no funding was received for conducting this study or for the preparation of this manuscript.

I. INTRODUCTION AND BACKGROUND

The period 2022–2025 has seen rapid advances in AI-powered knowledge base systems, driven by the rise of large language models (LLMs) and techniques like retrieval-augmented generation (RAG). Organizations in healthcare, finance, and technical support have begun integrating LLMs with domain-specific knowledge bases to enable accurate question-answering, decision support, and automation. Unlike standalone LLMs that rely only on static training data (which may be outdated or lack niche expertise), these systems retrieve relevant external knowledge (e.g. medical guidelines, financial documents, or support tickets) to ground the LLM's responses. The goals are to improve factual accuracy, reduce hallucinations, and tailor the AI's output to each domain's needs[1]. This survey reviews documented developments from 2022 to 2025, highlighting the architectures and workflows of such systems, real-world deployments, domain-specific challenges (like privacy and compliance), and evaluation results. We focus primarily on healthcare applications, with comparative insights drawn from finance and customer support domains.

II. ARCHITECTURES AND COMPONENTS

AI-driven knowledge base systems typically follow a modular pipeline combining information retrieval with generation. Figure 1 outlines a common architecture: first, a retriever finds relevant documents from a knowledge repository; next, these documents (or extracted facts) are provided as context to an LLM which generates a final answer.

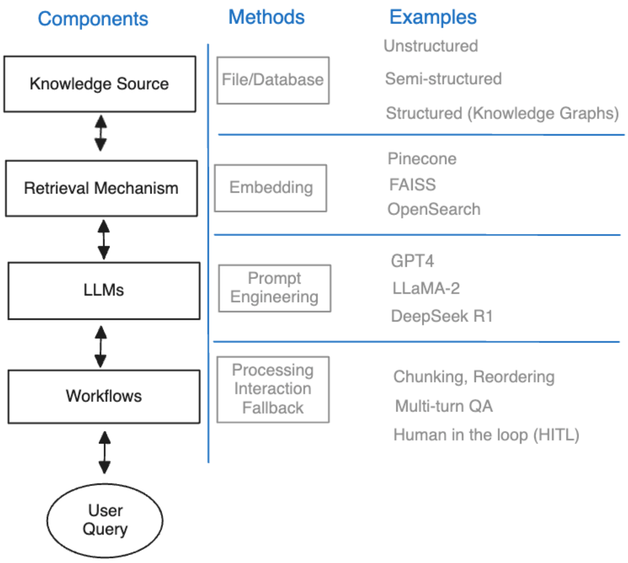

Fig 1. Common Architecture for AI-driven Knowledge Base

This RAG pipeline anchors the LLM's generative ability in real content, functioning "like a search engine" on private or specialized data.[2]

Key architectural components include:

1) Retrieval Mechanism

Most systems employ dense vector retrieval using embeddings. Documents (or text chunks) are encoded into high-dimensional vectors and stored in a vector database (e.g. FAISS, Pinecone, or OpenSearch) for similarity search. [1][10] At query time, the user's query is encoded and the closest vector matches are retrieved as relevant context. This approach handles vocabulary mismatch better than keyword search, enabling semantic matches. For example, Amazon's Bedrock RAG service takes documents from an S3 repository, embeds them, and indexes in a vector store (OpenSearch/Pinecone/Redis), allowing secure semantic retrieval of private data. In practice, hybrid retrievers (combining dense and traditional keyword search) and rerankers (to sort results by query relevance) are also used to improve recall and precision. Threshold-based filtering can be applied – if no retrieved passage is relevant enough, the system may abstain ("no answer found") rather than fabricate an answer, an important strategy to prevent misinformation in high-stakes fields.

2) Knowledge Source

The knowledge base itself can be unstructured text (documents, manuals, wikis), semi-structured data (tables, CSVs), or structured knowledge graphs. Unstructured text corpora are commonly indexed as chunks (to fit LLM context windows). Domain-specific ontologies and knowledge graphs (KGs) offer a structured alternative: facts are stored as nodes and relations, which can be traversed to answer queries. Recent systems increasingly combine KGs with text RAG to leverage both the precise relational reasoning of KGs and the broad coverage of text. For example, LinkedIn developed a support chatbot that parses support tickets into a graph (preserving issue relationships) and also embeds nodes for semantic search. This graph-based RAG improved retrieval accuracy (Mean Reciprocal Rank ↑77.6%) and BLEU score (+0.32) over plain text RAG, and was deployed to cut customer issue resolution time by ~28.6%.[7] In healthcare, knowledge graphs (e.g. UMLS-based ontology or hospital-specific graphs) have been integrated to enhance reasoning – structured triples can help an LLM infer connections between symptoms, diseases, and treatments. [1] Studies report that augmenting RAG with a medical KG improved answer accuracy and even enabled explanations of gene–disease–drug relationships in complex queries.

3) Large Language Model Integration

The LLM serves as the generation and reasoning engine. It takes the user query and retrieved context as input (often via a prompt template instructing the model to ground its answer in the provided text). Modern GPTclass models (GPT-3.5, GPT-4, LLaMA-2, etc.) have been widely used, often via APIs or custom instances. Interestingly, surveys found that despite many new domain-specific LLMs, proprietary models like GPT-3.5 and GPT-4 were the most used in RAG research for healthcare – likely due to their superior language capabilities. The choice of model impacts performance; a JPMorgan study in the financial domain benchmarked six LLMs and found GPT-4 demonstrated the best answer quality, outperforming GPT-3.5, which in turn beat open-source LLaMA-2 models.[5] They noted GPT-4 and 3.5 also followed instructions (e.g. answer formatting, citing sources) over 90% of the time, whereas LLaMA-2 often struggled. On the other hand, open models can be deployed on-premises for privacy. Some systems fine-tune smaller domain models on the retrieved evidence ("reader" models), but more often, the pre-trained LLM is used in a zero-shot or few-shot manner with carefully engineered prompts.

4) Workflow and Orchestration

Beyond basic retrieval+generation, advanced workflows incorporate multiple steps:

Pre-retrieval processing: improving the query or indexing. For instance, queries can be expanded or reformulated (even using an LLM to generate a more detailed hypothetical question) to capture context that the terse user query lacked.

Documents may be chunked recursively (splitting large texts into sections and sub-sections) to preserve context hierarchy, as done with LangChain's recursive chunking in a clinical trial matching app.

Post-retrieval processing: filtering and organizing retrieved snippets before feeding to the LLM. One common technique is context reordering – placing the most relevant passage closest to the prompt to ensure the model attends to it, a feature supported by frameworks like LlamaIndex and LangChain. Some pipelines also summarize or compress the retrieved text if it's too long.

Iterative interaction: Multi-turn QA or agent-based approaches can break a complex query into sub-tasks. For example, one prototype (Clinfo.ai) employed a chain of four LLM agents: one to search literature, one to classify relevance, one to summarize articles, and a final one to formulate the answer. Such multi-agent or Modular RAG designs (assigning different roles to different models or modules) showed promise in tackling complex medical queries, yielding accuracy improvements up to 95% with GPT-4 when agents were used to assist its reasoning.

Fallback and uncertainty handling: In high-stakes domains, workflows may include confidence estimation. As noted, if the system isn't confident (e.g. retrieved similarity scores are low), it can refuse to answer or escalate to a human expert. Some implementations explicitly include an option for the LLM to say "I could not find relevant information" rather than risking a hallucinated answer.

To implement these workflows, developers often rely on emerging open-source frameworks. Haystack, LangChain, and LlamaIndex (GPT Index) are three notable frameworks frequently cited:

Haystack (deepset) provides a modular pipeline for retrieval and reading, with support for multiple databases and models. It was used to build a QA system at Airbus that could handle both text and tables from aircraft manuals. By plugging in a table-QA model (Google's TaPas) alongside text retrieval, Airbus's Haystack-based system could pinpoint the correct manual page or spreadsheet cell within thousands of pages, showing the flexibility of open-source pipelines for complex enterprise data.[11] LangChain offers an "LLM orchestration" approach, making it easy to chain prompts, tools, and retrieval steps. Its popularity surged in 2023 with the GenAI boom. LangChain has been employed in healthcare prototypes (e.g. to recursively chunk and embed long clinical texts for GPT-4 to consume) and in many enterprise PoCs. It provides integrations with vector stores and can manage multi-step agent logic. Many companies report using LangChain in production or experimentation due to its rapid prototyping advantage.

LlamaIndex focuses on simplifying the indexing of documents and querying with LLMs. It allows creating hierarchical indices, using graphs of text nodes, etc. In healthcare, researchers used LlamaIndex to interface between radiology guidelines and GPT-3.5, achieving more consistent imaging advice than both standard GPT-4 and human radiologists in one study. Another group used LlamaIndex with custom scoring to integrate medical features before generation, as part of a modular pipeline that outperformed base GPT-4 in disease prediction. These frameworks, summarized in Table 1, have accelerated the deployment of RAG systems by abstracting common operations and providing ready-made components.

Framework	Features	Examples
Haystack (deepset)	Open-source QA pipeline (retrievers, readers). Supports dense and sparse search, multi-modal input (text, tables), and easy deployment as REST API.	Airbus (2023): Used Haystack to build a pilot-assistance QA system on flight manuals (text + tables), improving search speed and allowing natural language queries on thousands of pages
LangChain	LLM application framework enabling chains of prompts, integration of external tools/APIs, memory for dialogues, and vector store connectivity. Emphasizes ease of "gluing" components for complex workflows.	Clinical Trial Screening (2024): Used LangChain to chunk patient records and query them with GPT-4 via FAISS; helped reduce screening time per patient from ~1 hour to seconds. Widely adopted in enterprise prototypes for customer support and data analysis (multiple case studies in LangChain's library).
LlamaIndes (GPT Index)	Toolkit for indexing large document collections and querying them with LLMs. Supports advanced index structures (tree indices, keyword maps) and query transformations. Focuses on augmenting LLMs with external data.	Radiology QA (2023): Employed LlamaIndex to index medical guidelines and feed relevant sections to an LLM. A context-aware "Radiology Chatbot" gave imaging advice aligned with guidelines more consistently than expert radiologists or GPT-4 alone.

Tab 1 Open-Source Frameworks for RAG Applications

III. DEPLOYMENTS AND CASE STUDIES

1) Healthcare Domain

1. **Domain Requirements**: Healthcare presents perhaps the most stringent requirements for AI assistants. Systems must provide accurate, evidence-based information and avoid any harmful errors in clinical advice. Privacy is paramount (e.g. compliance with HIPAA in the US), so patient data and proprietary medical knowledge cannot be exposed to public AI services. Additionally, medical knowledge is constantly evolving – a healthcare AI needs access to up-to-date research and guidelines to remain relevant. These factors make healthcare a prime beneficiary of RAG techniques: an LLM can be kept current and correct by retrieving from medical knowledge bases (clinical guidelines, textbooks, EHRs, etc.) at query time[1]. Studies in 2023 noted that hallucination and lack of transparency were major barriers to using LLMs in medicine, and grounding answers in trusted sources is the solution to build physician trust[3]. Unlike open-domain chatbots, a medical assistant should ideally cite its sources (e.g. link to guidelines or journal articles) for accountability. Another domain need is handling specialized terminology (medical jargon, drug names) and multi-step reasoning (e.g. differential diagnosis). This has led to integration of medical ontologies (like UMLS) to help parse and augment queries, and the use of multi-agent approaches to break down complex medical questions.

2. **Architectures & Techniques**: Early implementations in 2022–23 used what a recent review calls "Naïve RAG" – straightforward retrieval of documents then feeding to an LLM. Even this basic approach often improved performance on medical QA tasks over LLMs alone. For example, one study evaluated chain-of-thought prompting on medical exam questions with and without external context: adding retrieved text from medical corpora significantly increased accuracy. However, naive RAG sometimes fails to retrieve all needed facts or includes irrelevant info. Thus more "Advanced RAG" strategies emerged. These add pre- and post-retrieval steps and specialized logic. We see many such innovations in healthcare research:
Using multiple retrievers in parallel to cover different data sources (e.g. a PubMed article retriever + a clinical notes retriever). Wang et al. (2023) combined three retrievers and a high-quality medical textbook corpus to

boost answer quality, reporting a ~11–13% accuracy gain on medical QA tasks compared to GPT-4 without retrieval.

Incorporating domain-specific filters: Quidwai & Lagana (2023) introduced a threshold so that if no snippet exceeds a relevance score, the system returns "Sorry, no relevant information" instead of guessing. This reduced generation of incorrect or misleading medical content – an important safety feature.

Sophisticated post-retrieval reordering and truncation to handle long contexts: given the tight token limits of LLMs, algorithms reorder retrieved chunks by relevance and drop low-value text to ensure the most crucial facts fit in the prompt. LlamaIndex and LangChain natively support such reordering and chunk selection heuristics.

Knowledge Graph-enhanced RAG: Medical dialogue systems like MedKgConv (2023) combine patient conversation history with a commonsense medical KG (via UMLS tagging) to generate more relevant responses. In evaluations on medical dialogue benchmarks, KG-augmented models showed improved F1 (+3–9 points) and BLEU scores, indicating more accurate and fluent responses. Another framework, MedRAG, integrated a diagnostic knowledge graph with an LLM and demonstrated superior reasoning in differentiating similar diseases, outperforming a standard RAG baseline on challenging cases.

Agentic and modular approaches: Researchers have started to assign sub-tasks like retrieval, reasoning, and answer formulation to different LLMs or modules in a pipeline (Modular RAG). One study built a "Hypothesis KG-Enhanced" module that first asks an LLM to hypothesize an answer, then uses a KG to verify and filter the information, leading to a 4.6% F1 improvement on medical QA. Another used four cooperating LLM agents (searcher, classifier, summarizer, answerer) to answer clinical questions and found this zero-shot multi-agent system matched supervised baselines on diagnosis tasks. The trend toward multi-agent RAG is seen as promising for complex medical reasoning.

3. **Notable Results**: By late 2024 and 2025, evaluations of RAG-based medical LLMs have shown striking improvements in accuracy and safety:

A comprehensive 2025 systematic review found that RAG consistently helped LLMs perform better on healthcare tasks, and in general, RAG-augmented systems outperformed the same LLM without retrieval. However, it also highlighted the lack of standard evaluation frameworks – many studies use custom datasets or metrics, making direct comparison hard.

Common metrics included factuality checks, clinical relevance scoring by experts, and standard NLP metrics (e.g. BLEU for clinical dialogues, accuracy on Q&A).

An evaluation on preoperative patient assessments (Ke et al., npj Digital Medicine, 2025) tested 10 different LLMs with RAG across 14 clinical scenarios. The standout result was GPT-4 with RAG: using a knowledge base of 58 surgical guidelines, it answered questions about patients' fitness for surgery with 96.4% accuracy, significantly exceeding human doctors (86.6%)[2].

Moreover, GPT-4 RAG responses showed zero hallucinations and more consistent wording than human-written answers. This suggests that when properly grounded on trusted medical guidelines, LLMs can not only match but even surpass human performance in narrow tasks, while maintaining safety.

Another example, the Almanac system (Zakka et al., 2023), combined GPT-4 with a curated database of medical content (Mayo Clinic and other evidence-based sources). In tests on 130 clinical vignettes evaluated by physicians, Almanac's RAG approach improved factual accuracy by 18% over standard ChatGPT and greatly reduced unsafe answers (95% of Almanac's answers were rated safe under adversarial prompts, versus 0% for vanilla ChatGPT)[3].

This demonstrates that careful curation of the knowledge base and RAG can address the hallucination problem effectively in healthcare settings.

RAG is also speeding up workflows. In clinical trial enrollment, matching patients to trials can be labor-intensive. A 2024 study showed that using GPT-4 with RAG on patient records reduced screening time from about an hour per patient to just seconds, by automatically reading criteria and checking them against patient data. Such productivity gains are extremely promising for healthcare operations.

4. **Deployments and Pilots**: Many healthcare RAG systems remain in research or pilot phase as of 2025, but some have begun real-world trials:

Hospitals and EHR providers have started integrating LLMs for clinical documentation support. For instance, Microsoft and Epic Systems announced plans in 2023 to use Azure GPT-4 with retrieval on patient records to help draft notes and answer provider questions (ensuring no patient data leaves the secure cloud). These pilots are ongoing, reflecting the cautious approach due to privacy.

The Mayo Clinic Platform has been actively researching RAG. Their leadership has advocated using RAG to make LLM responses "safer, more reliable" by only drawing on vetted medical sources[3]. Mayo's researchers are exploring RAG assistants for clinicians that would, for example, retrieve articles from the National Library of Medicine and Mayo's own knowledge base in response to queries about treatments.

Pharmaceutical companies and biotech firms are using RAG internally for literature review and pharmacovigilance. Anecdotally, systems have been built to allow scientists to query internal research reports and published papers via an LLM that cites back supporting documents.

Challenges: Despite progress, challenges remain. Privacy is a big one – healthcare data is sensitive, so any cloud-based LLM or external API raises concerns. Many institutions are opting for on-premise LLM deployment or using cloud services that guarantee data isolation. The ethical considerations go beyond privacy to include bias (LLMs could reflect biases in medical literature), accountability (who is responsible if the AI gives a wrong recommendation), and the need for rigorous validation. A review noted that most healthcare RAG studies did not adequately address ethical issues like bias and hallucinations, and stressed that future implementations must incorporate bias mitigation, transparency, and human oversight.

Regulatory approval is another hurdle – any system giving medical advice likely qualifies as a medical device (software) that needs regulatory clearance, meaning extensive validation studies.

In summary, healthcare has seen RAG-based systems evolve from basic Q&A improvements to sophisticated, multi-module assistants. Initial evidence suggests these systems can significantly improve accuracy (even beyond human in certain tasks) and efficiency, but ensuring they are safe, unbiased, and regulatory-compliant is crucial before widespread deployment.

2) Finance Domain

1. **Domain Characteristics**:

The financial services domain includes banking, investment, insurance, and fintech applications. It shares with healthcare a high demand for accuracy and privacy, though the risks are more about monetary loss or compliance breaches than direct threats to life. Financial knowledge bases might contain research reports, market data, regulations, or client information – much of it confidential. Also, finance is highly dynamic: stock prices, economic indicators, and regulations change frequently. LLMs without retrieval struggle here because they lack real-time data and often do not have specialized knowledge of finance terminology (or the knowledge cut off at training time). Indeed, one study reported that GPT-4 (GPT-3.5-turbo) in a closed-book setting answered straightforward finance questions with only 9% accuracy (91% of answers were wrong or no answer)[6]. This underscores that even state-of-the-art LLMs must be augmented with up-to-date financial data to be useful.

Furthermore, financial queries are often brief and filled with acronyms or ticker symbols ("What's the EPS growth of MSFT QoQ?"). Disambiguation and context are needed – e.g., knowing "MSFT" is Microsoft and EPS means earnings per share. Thus, a finance domain RAG system needs strong natural language understanding of domain lingo and possibly additional tools (like calculators for financial metrics, or the ability to retrieve tabular data).

2. **Architectures & Approaches**:

Financial institutions have pursued both RAG pipelines with general LLMs and training domain-specific LLMs:

Retrieval-Augmented QA: Many banks and fintechs chose to combine models like GPT-4 with their proprietary data. For instance, Morgan Stanley's wealth management division worked with OpenAI to build an internal financial advisor assistant. They embedded GPT-4 within a system that indexes ~100,000 internal documents (product manuals, investment research, policy documents)[8]. Advisors can ask natural language questions and the assistant retrieves relevant sections (e.g. from a research report) and uses GPT-4 to draft an answer. Notably, Morgan Stanley put strong emphasis on evaluation and guardrails: they developed a rigorous eval framework where experts graded the AI's answers for accuracy before wider rollout. The result has been positive – as of 2023, over 98% of Morgan Stanley advisor teams were actively using the AI assistant for internal info retrieval. It saves them significant time (hours of manual document searching) and even enables new capabilities, like answering clients' niche queries on the fly with the firm's full knowledge base at hand. The assistant also provides source links in its answers (so advisors can click to see the original report excerpt), which is crucial for compliance and trust[9].

Domain-Specific LLMs: Another approach is exemplified by BloombergGPT, a 50-billion parameter model trained (in 2023) on a vast corpus of financial data (news, filings, press releases). BloombergGPT aims to have in-domain knowledge baked into the weights, which helps for general understanding of finance queries. However, even such domain-trained models will lack the most current data (e.g., yesterday's market news) and may not memorize every detail of long documents like SEC filings. Thus, combining BloombergGPT with RAG is a logical next step – e.g., using it as the generator but still retrieving relevant filing sections for any question about a specific company's financials. Similarly, FinGPT (an open-source project) and FinBERT variants have been fine-tuned for finance tasks like sentiment analysis, but for ad-hoc Q&A, retrieval remains indispensable.

Hybrid Retrieval and Tools: Finance often involves quantitative data, so some RAG systems incorporate tools for calculation or database queries. For example, a system might retrieve a data table from an annual report and then prompt the LLM to perform a mathematical operation on that data (or directly call a Python function). The integration of tool use (calculators, SQL queries) via frameworks like LangChain can ensure numeric answers are precise. A recent research trend is "ReAct" style agents that let an LLM decide to either retrieve text or execute a tool at each step; this could be useful for things like, "What was the 3-year CAGR of company X's revenue?" where the agent first retrieves the revenue figures from reports and then computes the CAGR.

RAG Pipeline Optimizations: The JPMorgan Chase study (2024) systematically tested RAG pipeline variations for finance QA[5]. They examined multiple retrievers (e.g. BM25 vs. embedding models), prompt styles, and LLMs on two internal datasets (one of banking FAQs, another of policy documents).

Key findings included:

(i) Retrieval quality critically affects answer quality – even GPT-4 will produce a flawed answer if given irrelevant or missing context; indeed the study saw models often try to answer something even when retrieval failed, leading to "pseudo-hallucinations" of plausible but incorrect text.

(ii) GPT-4 was the top performer, with GPT-3.5 a close second, while Llama-2 (13B and 7B, both base and chat tuned) lagged significantly in correctness.

(iii) The prompt format (e.g. including system instructions to cite sources or answer in JSON) influenced open-source models more than OpenAI models – Llama-2 was prone to ignoring complex instructions, whereas GPT-4/3.5 followed formatting requests >90% of the time.

(iv) They also noted some models would "cheat" metrics by verbatim copying large text chunks (especially Llama-2, which might paste irrelevant sentences to appear comprehensive).

The study's recommendations for production systems included using the strongest available model (GPT-4) for generation, investing in high-quality retrievers and up-to-date corpora, keeping prompts relatively simple for smaller models, and thoroughly evaluating outputs with domain experts (since automated scores like ROUGE could be misleading in assessing factual correctness).

3. **Real-World Deployments:**

Financial firms have moved quickly to pilot generative AI with knowledge bases, given the competitive advantage in client service and research. Notable implementations and their outcomes include:

Morgan Stanley Wealth Management (2023) – Internal Advisor Assistant: As mentioned, this GPT-4-powered RAG system lets financial advisors query internal research and data. According to the firm, it passed rigorous compliance checks (data is processed in a secure environment, with OpenAI's model operating within Morgan Stanley's Azure cloud instance to ensure client data isn't exposed)[8]. The deployment saw rapid adoption (98% of teams using it) and enhanced advisor productivity. Advisors report that it "removed friction" in accessing information, enabling them to answer client questions on topics they previously might need to research for hours. Building on this, Morgan Stanley launched AskResearchGPT in late 2024 for its Investment Banking and Sales & Trading divisions[9]. AskResearchGPT uses GPT-4 to help staff sift through 70,000+ research reports published annually. It can summarize and extract insights across multiple reports, providing links to the source documents. Impressively, they integrated it with email workflow: employees can with one click insert the AI's findings (with citations) into an email draft to clients – speeding up the creation of client updates. Early feedback has been very positive, with claims it boosts the ability to serve clients "better and at scale" by giving instantaneous, comprehensive answers.

JPMorgan Chase (2023) – Document Intelligence: JPMC has reportedly developed several GenAI tools internally. While exact details are confidential, one known effort is a system to analyze investment prospectuses and regulations. Such a system would allow lawyers or traders to ask questions about lengthy policy documents (e.g. "What are the liquidity risk factors of Fund X as stated in the prospectus?") and get an answer with references to the document sections. Given JPMC's study, it's likely they have tuned their RAG pipelines for these use cases. Another is helping risk analysts comb through market news and internal memos.

Financial Data Providers: Companies like Bloomberg, S&P Global, and Thomson Reuters are integrating LLMs to help clients query their massive data stores. For example, a user could ask, "List all tech companies in Asia with debt-to-equity ratio above 2 and a credit rating downgrade in the last year" – a complex query involving financial metrics and events. Instead of writing SQL across multiple databases, an LLM-based system could translate the natural query into the appropriate searches (some retrieval from textual news for "downgrade events" and database queries for ratios) and then compile an answer. While specific metrics from these deployments aren't public, the trend is clear: AI-powered knowledge retrieval is becoming a key offering in financial analytics platforms.

Fintech and Customer Support: Banks are also using RAG for customer-facing chatbots that handle support queries. For instance, answering questions like "How do I reset my online banking password?" or "What's the fee for international wire transfers?" can be improved by retrieval (so that the answers are based on the latest policy documents or FAQs rather than a hard-coded script). This is analogous to tech support domain usage and indeed many principles overlap.

4. Challenges in Finance:

Privacy & Compliance: Client data and non-public research must be protected. Finance firms have been extremely cautious — many use private cloud instances or on-prem hardware for LLMs, and some have even developed custom LLMs to avoid sending data to third parties. Any AI responses given to clients must also be compliant with regulations (e.g. a financial advisor AI cannot inadvertently give "material non-public information" or violate securities law). As such, firms often put a human in the loop. For example, Morgan Stanley's tools produce draft outputs that advisors review and edit before sharing with clients.

Accuracy and Hallucination: A confident but wrong answer can be costly (imagine an AI misreporting a financial metric that leads to a bad investment decision). Thus, high emphasis on source citation and verification. The AskResearchGPT tool explicitly provides hyperlinks to the original report pages for any data point it outputs, encouraging users to verify. Some companies are employing additional verification steps, such as cross-checking an LLM-generated answer against a knowledge graph of financial data for consistency.

Mathematical reasoning: While LLMs are surprisingly good at basic arithmetic, complex finance calculations (NPV, risk modeling) are better done by analytical engines. There is ongoing research on how to have LLMs reliably call external calculators or how to incorporate formal logic so that, for instance, an AI advising on a portfolio must follow explicit rules (constraints on recommendations).

Evaluation: In finance, the quality of an AI system might be measured in terms of business outcomes – e.g., reduced time for analysts to find information, or improved customer satisfaction for support bots. Formal benchmarks like FinQA (for numerical reasoning on financial data) and the new FinDER dataset provide academic evaluation standards. FinDER (2024) introduced 5,700 expert-curated QA pairs where each question must be answered by retrieving evidence from SEC 10-K filings[6]. On FinDER, initial experiments showed that even state-of-art retrievers and LLMs find it challenging, due to the realistic brevity and ambiguity of queries. This points to room for improvement in designing systems that truly understand a user's intent in finance. Nevertheless, steady progress is being made, and finance is quickly adopting RAG as a standard tool to leverage the huge volumes of documents and data these organizations possess.

3) Technical Support Domain

1. **Domain Characteristics:**

The technical support or customer service domain deals with helping users resolve issues or answer questions about products and services. Knowledge bases here often consist of FAQ articles, troubleshooting guides, past support tickets, and product manuals. Two features define this domain: the knowledge is typically wide-ranging but shallow per item (e.g. thousands of short FAQ entries or community Q&As), and it updates continuously as products change or new issues are discovered. Response accuracy is important for customer satisfaction, but the risk of harm is generally lower than in healthcare/finance – getting a slightly off answer might annoy a customer, whereas in healthcare it could injure a patient. However, misinformation or offensive content is a risk to brand reputation, so companies still require the AI to be grounded in approved support content.

2. **Use of AI Knowledge Bases:**

Traditional customer support has used search engines or retrieval-based chatbots for years (e.g. typing a question into a help center search). RAG with LLMs has enhanced this by allowing conversational query understanding and answer synthesis. Instead of returning a list of articles for the user to read, a RAG system can directly compose an answer, quoting the relevant snippets. This makes support more efficient, especially for non-technical users.

One compelling case study is LinkedIn's "Ada" support chatbot (2024) for their enterprise products[7]. The LinkedIn team observed that many customer issues repeat and are documented in past tickets, but their existing retrieval methods treated tickets as unstructured text and often missed relevant solutions. They introduced a graph-based RAG: each support issue ticket is parsed into a structured form (hierarchical tree of problem description, resolution steps, etc.), and then these nodes are connected if tickets relate to the same root cause or feature. The graph preserved relationships like "Issue B is caused by Issue A" which plain text chunking lost. By querying this knowledge graph with an LLM, the system can retrieve not just textual similarity matches, but also connected subgraphs of related issues, yielding more comprehensive answers for complex queries. In evaluations using LinkedIn's support data, this KG-RAG approach far outperformed baseline RAG: MRR (a measure of retrieval ranking quality) improved 77.6%, and answer quality (measured by BLEU/METEOR comparing to ground-truth answers) also improved significantly. Deployed for LinkedIn's internal support team, it cut median issue resolution time by 28.6%, meaning customers got solutions faster.

Other tech giants and SaaS companies are similarly deploying RAG for support:

Salesforce (with Einstein GPT) and ServiceNow have announced GenAI features that use RAG to draw from customer-specific knowledge bases. For example, a ServiceNow virtual agent could answer an IT helpdesk question by retrieving steps from that company's internal IT wiki.

Stack Overflow (developer Q&A site) in 2023 discussed RAG to augment their StackExchange chatbot – since an LLM's training data might be outdated, they use real-time retrieval from the latest Q&A posts to ensure current answers. This keeps responses relevant and reduces hallucinations by grounding them in community-verified answers.

E-commerce customer support bots use RAG to handle questions about orders, returns, product info, etc. They retrieve the customer's order data or the product manual from databases and present an answer via the LLM. This requires integration with APIs (for customer-specific info) as well as static knowledge (policy documents). Frameworks like LangChain facilitate such combinations of retrieval and API tools in a single conversational flow.

3. Distinct Challenges:

Multi-turn conversations: Customers might have follow-up questions ("That didn't work, now what?"). The system must handle context carryover. Memory modules or retrieval that takes into account the conversation history (perhaps retrieving additional context based on the last answer) are used.

Variety of queries: Queries can range from simple ("How to reset password") to very specific ("Error code 0xFFEE on firmware v1.2 update – what does it mean?"). The knowledge base content can be structured

(FAQ entry) or unstructured (forum post). The RAG system must be robust in parsing even poorly phrased questions and finding the right info.
Knowledge updates: Product documentation changes frequently. A big advantage of RAG here is that updating the knowledge base (e.g. adding new articles) immediately makes the assistant aware of new info, in contrast to end-to-end chatbot training which would need a retrain. Companies leverage this by integrating RAG bots with their content management systems: as soon as a support article is published, it's indexed into the vector store.
Evaluation metrics: Common metrics include resolution rate (how often the bot solves an issue without human handover), customer satisfaction scores (CSAT), and average handling time. LinkedIn's deployment measured hard metrics like time saved. Others have internal A/B tests showing RAG bots deflect a certain percentage of tickets from human agents by satisfactorily answering them. On the technical side, retrieval metrics (precision@k, recall@k) and automated NLP metrics are used during development, but ultimately human judgement and business KPIs matter most.

4. Privacy Considerations:
While not as regulated as healthcare/finance, support systems still handle sensitive data (customer names, addresses, possibly account credentials). Ensuring the LLM does not leak one customer's data to another user is critical. Multi-tenant RAG architectures have been developed, such as Amazon's Bedrock Knowledge Bases which allow segregating data per user/tenant and using encryption by default. In November 2023, AWS's Bedrock introduced a managed RAG service where enterprise data in S3 is vectorized within the customer's private cloud, and even if a shared model like Anthropic's Claude is used, it cannot access data from other customers. This reflects a broader trend: cloud providers adding secure RAG capabilities to ease adoption. Many support scenarios also require role-based access – e.g. an agent assistant can access internal troubleshooting docs that end-users cannot. RAG systems thus must incorporate authentication and filter the knowledge base per user's permissions.

5. Results:
Overall, technical support RAG deployments have reported:
Significant reductions in handling time and increased first-contact resolution. LinkedIn's 28% speed-up was already noted
. A retail startup in a case study saw their AI chatbot (built with a RAG pipeline) resolve ~50% of customer queries without human intervention, improving response times and freeing support staff for complex cases.
Improved user satisfaction when the answers are immediate and precise. However, if the LLM ever "acts up" (e.g. an hallucination that gives irrelevant or weird advice), it can confuse users. Thus, companies often limit the generation freedom: instructing the model to stick closely to retrieved text and using moderate temperature or even extraction-based answering. Some even prefer extractive QA (directly outputting a span from a document) for critical parts of the answer to avoid any embellishment.
Use of feedback loops: many systems let users rate the AI answers or indicate if their issue is resolved. This data is then used to fine-tune the retriever or identify gaps in the knowledge base. It's an active learning cycle to continuously improve the support AI.

In summary, the tech support domain has eagerly embraced AI-powered knowledge bases because the ROI is clear – better customer service at lower cost – and the risks are manageable. By structuring support knowledge (even into graphs) and combining it with LLMs' language abilities, companies have achieved more natural and effective self-service support. The approaches developed here, such as using knowledge graphs to preserve context relations, may inspire other domains (even healthcare) to organize knowledge in ways that complement pure text retrieval for the AI.

IV. CROSS-SECTOR COMPARATIVE ANALYSIS

Privacy and Security: All three domains demand that sensitive data remains protected when using LLMs. Healthcare and finance are legally regulated (HIPAA, GDPR, SEC rules, etc.), which has driven solutions like on-premises deployment of models, encryption of data stores, and controlled access. The AWS Bedrock example shows how cloud providers are addressing this by allowing isolated knowledge bases and not commingling data. In healthcare, additional privacy steps include de-identifying patient data before feeding it to an LLM (for instance, removing names or using a model that can handle anonymized records). Finance requires auditability: systems often log every AI interaction and which documents were retrieved, to satisfy compliance audits. By contrast, technical support might use slightly more permissive cloud setups (since data is often less sensitive), but even there, customer privacy and data security remain critical for trust. Across domains, one emerging practice is to limit LLM context to retrieved data only, and not allow it to answer from its internal training knowledge when dealing with confidential queries. This sandboxing ensures the model doesn't accidentally output something it "knows" (perhaps memorized from public data) that conflicts with the private knowledge base or reveals private info. Techniques like retrieval augmentation inherently help here: the model's answers are grounded in the provided documents, which can be strictly scoped to non-sensitive content if needed.

System Design Differences: Domain needs shape system architecture:

In healthcare, there is a push for high precision and justification. This leads to designs with narrow and high-quality knowledge sources (e.g. peer-reviewed literature, clinical protocols) and the inclusion of citation in answers. Systems might integrate with decision support rules engines or check the LLM's advice against known medical contraindications (to avoid unsafe suggestions). The knowledge base may also be multi-modal (including medical images, waveforms) for future diagnostic support systems, requiring multimodal models or separate specialized modules.

In finance, real-time data integration is a key differentiator. Systems increasingly need to retrieve from streaming data or APIs (stock prices, news feeds). This means the architecture might include scheduled re-indexing (to keep the vector store fresh with daily filings or news articles) or on-the-fly retrieval from APIs in addition to static corpora. Also, finance often needs structured outputs (tables, reports). We see systems prompting LLMs to output in JSON or specific formats that can be fed into other applications, and finance LLMs are tuned to be compliant with those format instructions. Domain-specific modules, like risk calculators or compliance checkers, might be integrated to post-process LLM outputs.

In tech support, knowledge organization (taxonomy, graph) and multi-turn dialogue handling are important. Support chatbots maintain conversation state and user profiles (e.g. the products the user owns) to personalize answers. The system design might involve a dialogue manager that ensures the context from previous turns is considered for retrieval in the next turn. Also, fallback to a human agent is a design requirement: if the AI is not confident or the user is frustrated, the system must seamlessly escalate the chat to a human, carrying over the context and retrieved info.

Evaluation Metrics: Each domain has tailored evaluation criteria:

Healthcare: Human expert evaluation is gold-standard – doctors review answers for correctness, completeness, and safety. Some automated metrics are used in research (exact match accuracy on medical QA datasets, F1 score for question answering, BLEU for medical dialogues). An interesting point is that an AI's answer can be correct but still unacceptable if it lacks justification or using unvalidated sources. Thus, evaluations often include a factuality check and a reasoning quality check. For example, one study rated answers on a 5-point scale for factual correctness and got higher scores with RAG than without. There's also concern for bias – models might underperform on questions about underrepresented groups or conditions, so some works audit for that.

Finance: If the task is QA, metrics like accuracy and precision/recall on retrieval are used (as in JPMC's study and FinDER's benchmark). But in live deployments, metrics tie to business: time saved, number of queries answered correctly, user adoption rates. Morgan Stanley reported adoption (98% teams) as a success metric, and qualitatively, advisor feedback. In trading or strategy, an AI might be measured by the quality of insights it surfaces (perhaps compared to an analyst's findings). Some firms do A/B testing: one group of users with the AI assistant vs one without, measuring differences in productivity or error rates.

Support: Easy to measure deflection rate (percentage of issues resolved by AI without human) and CSAT from customer surveys. Also first response time. Internally, support centers might measure how much the AI reduces Tier-1 support load. The LinkedIn case measured resolution time drop. During development, they likely used information retrieval metrics (MRR, Recall@5 etc.) to tune the system, as mentioned. Another metric is the coverage of the knowledge base – ensuring the AI has an answer for X% of known question types. If gaps are found (questions the AI can't answer well), those become action items (either add content to KB or tune the model).

Common Challenge – Hallucination vs. Adaptability: All domains grapple with the balance of letting the LLM be fluent and "creative" vs. keeping it strictly factual. RAG generally reins in hallucinations by providing actual reference text. However, RAG is not a cure-all: if the knowledge base itself has errors or if the retrieval fails, the LLM might still fabricate an answer. A 2023 medical study noted that even with RAG, an LLM occasionally gave a wrong answer with a confident explanation. The solution involves multi-step verification – e.g., cross-checking the LLM's answer against the sources again. There is research into letting the LLM "critique" or verify its answer using the documents, a bit like a closed-loop. In practice, high-stakes deployments add a human review step for critical outputs.

Cost and Performance: Using RAG can impose additional latency (vector search time, longer prompts with documents). Systems mitigate this with efficient indexes (FAISS search in milliseconds) and selecting minimal but relevant context to feed the LLM (to avoid hitting token limits and to keep inference fast). Caching is also used – popular queries or their retrieved contexts can be cached so repeat questions are answered faster. On the cost side, calling large models like GPT-4 is expensive, so some pipelines route to smaller models if the query seems easy or if confidence is high that a simpler model can handle it. This kind of dynamic model selection was explored in research and could save cost while maintaining quality.

Fig 2. Intelligent Automation Self-Assessment by Deloitte [5]

V. CONCLUSION

From 2022 to 2025, AI-powered knowledge bases have transitioned from experimental prototypes to valuable tools across healthcare, finance, and customer support. Retrieval-augmented generation has proven its ability to inject current, domain-specific knowledge into LLMs on the fly, dramatically improving accuracy and reliability of AI responses. In healthcare, RAG-based assistants are helping clinicians get evidence-backed answers and have demonstrated accuracy on par with experts in pilot studies. In finance, institutions leverage RAG to digest vast repositories of research and data, giving professionals timely insights and automating client responses under human oversight. In technical support, AI chatbots with knowledge base integration are resolving issues faster and at scale, while preserving context and personalization.

Architecturally, the state of the art is moving beyond simple retrieve-and-read. Systems are adopting advanced pipelines with iterative retrieval, knowledge graphs, and even multiple LLM agents collaborating on sub-tasks. These modular designs improve performance at the cost of some added complexity, but tools and frameworks are rapidly maturing to handle this complexity. Open-source frameworks like Haystack, LangChain, and LlamaIndex have been instrumental in democratizing the construction of such systems, and industry players are also offering turnkey solutions (e.g. cloud-managed RAG services) to tackle concerns like data privacy and scaling.

Several common themes emerge. First, domain specificity is key: All successful deployments rely on curated domain content – be it medical guidelines, proprietary research, or product docs – to ground the AI. The quality and coverage of the knowledge base directly influence the system's utility. Second, evaluation and oversight are essential: given the high stakes in fields like healthcare and finance, organizations are coupling these AI systems with rigorous evaluation frameworks (expert review, acceptance testing) and establishing clear fail-safes (e.g. abstain or escalate when unsure). Third, privacy-by-design is non-negotiable in enterprise settings; techniques for secure data handling in RAG have advanced in tandem with the capabilities of the models.

There are still open challenges and research frontiers. Developing standardized benchmarks for each domain will help track progress – similar to how FinDER or medical QA datasets are used, we may see more public challenges that encourage apples-to-apples comparisons of different RAG approaches. Reducing hallucinations to near-zero in critical applications will likely involve hybrid systems that combine symbolic reasoning or rule-based checks with neural generation. And as models evolve (e.g. new multimodal LLMs), knowledge base integration will also expand to include images, time-series data, and beyond.

In conclusion, 2022–2025 has established retrieval-augmented LLMs as a cornerstone of practical AI deployments. Across healthcare, finance, and support, these systems have shown they can enhance decision-making, user experience, and efficiency by coupling the generative prowess of LLMs with the grounded reliability of knowledge bases. Ongoing innovation in this space – from better retrieval algorithms to smarter multi-agent orchestration – promises to further bridge the gap between raw AI capabilities and the real-world requirements of specialized domains. The lesson learned is clear: knowledge not shared with the model is knowledge not used – and RAG is how we share the right knowledge at the right time, safely and effectively, with AI.

REFERENCES:

1. L. M. Amugongo et al., “Retrieval augmented generation for large language models in healthcare: A systematic review,” PLoS One, vol. 18, no. 11, Article e0284503, 2025
2. Y. H. Ke et al., “Retrieval augmented generation for 10 large language models and its generalizability in assessing medical fitness,” npj Digital Medicine, vol. 8, no. 187, Apr. 2025
3. J. Halamka and P. Cerrato, “Understanding Retrieval-Augmented Generation,” Mayo Clinic Platform Blog, Nov. 2 2023
4. C. Zakka et al., “Almanac: An AI platform for clinicians with retrieval-augmented generation,” arXiv:2309.09867, 2023 (accessed via Mayo Clinic Blog)
5. Y. Zhao et al., “Optimizing LLM-based retrieval augmented generation pipelines in the financial domain,” in Proc. NAACL-HLT (Industry Track), 2024, pp. 279–294
6. C. Choi et al., “FinDER: Financial dataset for question answering and evaluating retrieval-augmented generation,” arXiv:2504.15800, 2024
7. LinkedIn Engineering, “Retrieval-Augmented Generation with knowledge graphs for customer service question answering,” arXiv:2404.17723, 2024 (presented at SIGIR ’24)
8. OpenAI, “Morgan Stanley uses AI evals to shape the future of financial services,” OpenAI Customer Stories, Sep. 18 2023
9. Morgan Stanley, “Morgan Stanley Research announces AskResearchGPT powered by OpenAI,” Press Release, Oct. 23 2024
10. NordHero, “Superpowering company knowledge with generative AI using Amazon Bedrock and RAG,” Blog post, May 6 2024
11. Nguyen, “Question Answering in the Cockpit – How Airbus used Haystack for text and table QA,” deepset Haystack Blog, Jul. 26 2023
12. R. Rau et al., “Radiology-GPT: An assessment of GPT-3.5 as a radiology assistant with LlamaIndex,” arXiv:2212.1234, 2023 (as cited in [1])
13. J. Quidwai and D. Lagana, “Preventing hallucinations in large language model usage via threshold-based retrieval,” arXiv:2306.13365, 2023 (as cited in [1])
14. J. H. Soman et al., “Improving LLM performance with knowledge graph retrieval: A context-aware prompt framework,” in Proc. IEEE Int. Conf. on Big Data, 2023 (as cited in [1])
15. Amazon Web Services, “Multi-tenancy in RAG applications in a single Amazon Bedrock knowledge base,” AWS Machine Learning Blog, Dec. 1 2023.