Disclaimer

The content set forth herein is Red Hat confidential information and does not constitute in any way a binding or legal agreement or impose any legal obligation or duty on Red Hat. This information is provided for discussion purposes only and is subject to change for any reason without notice.

The execution for this vision is now *mostly* document in Jira:

- **Parent Outcome + Milestones**:
  - Crafting an OpenShift AI-Native End-to-End Agentic/RAG experience
    - DP: https://issues.redhat.com/browse/RHOAISTRAT-477
    - TP: https://issues.redhat.com/browse/RHOAISTRAT-479
    - GA: https://issues.redhat.com/browse/RHOAISTRAT-480

- **Roadmap Dashboard:**
  - **RFEs** created for the milestones: https://issues.redhat.com/secure/Dashboard.jspa?selectPageId=12373691

Document Maintainer: Adel Zaalouk
Last Updated: May 20, 2025

Why RAG?

Customers don't want AI for its own sake; they want tangible outcomes like improved lives, streamlined operations, or new opportunities. AI applications are a means to an end, constrained by factors like speed, accuracy, data sources, and budget. For example, a financial analyst report needs to be comprehensive, highly accurate, and generated within 5 minutes, drawing from diverse data. A news aggregator requires moderately high-speed summaries with moderate accuracy and a very limited budget, handling high volume. Booking a flight demands low latency and zero errors within a specific budget, utilizing real-time APIs.

AI, particularly with the rise of language models and tools like ChatGPT, offers significant advantages over traditional methods in achieving these outcomes due to its superior processing speed, decision-making, and task execution capabilities.

Retrieval-Augmented Generation (RAG) has emerged as a leading AI system implementation, especially in enterprises, with Databricks finding 60% of LLM applications using some form of RAG. RAG addresses key challenges faced by pre-trained language models, such as hallucination and outdated information, by providing access to external knowledge bases.

Why RAG is Essential:

- **Overcomes Context Limitations:** RAG optimizes the context provided to models, ensuring the most relevant information is used within context window constraints, even with large knowledge bases.
- **Reduces Hallucination:** It grounds responses in retrieved external knowledge, minimizing the generation of incorrect or fabricated information.
- **Context Specificity:** RAG enables dynamic context construction, tailoring input to each query.
- **Improves Accuracy and Detail (Grounding):** It provides models with specific details and facts, leading to more accurate and detailed responses.
- **Manages User Data:** Facilitates the secure inclusion of user-specific data only in relevant queries.
- **Accesses Up-to-Date Information:** Unlike static LLMs, RAG systems can access current data, crucial for applications requiring the latest information.
- **Enhances Transparency:** RAG systems can cite sources, increasing user trust.
- **Cost-Effectiveness:** It allows for the use of smaller, more efficient models in conjunction with retrieval mechanisms.

Market Landscape and Economic Impact

The market for RAG is experiencing significant growth. Databricks reports that 60% of LLM applications utilize some form of RAG. The global RAG market, valued at $1,042.7 million in 2023, is projected to reach approximately $13.85 billion by 2030, with a Compound Annual Growth Rate (CAGR) of 44.7%. North America currently dominates the market, but the Asia Pacific region is expected to show the fastest growth.
From an economic perspective, companies deploying RAG solutions report an average ROI of 150% over three years, which can increase to around 200% for agentic RAG. The development and maintenance of these systems are also projected to create an estimated 10,000 new jobs in the AI sector by 2026.

Key Market Use Cases

RAG is being adopted across various industries for diverse applications, including:

- **Knowledge Question Answering:** Providing accurate answers in customer service using product manuals or FAQs.
- **Code Generation:** Retrieving relevant code snippets and documentation to assist in code creation.
- **Recommendation Systems:** Enhancing recommendations by providing relevant context.
- **Customer Service:** Improving support accuracy with access to current product information.
- **Personal Assistants:** Enabling more comprehensive and accurate information from AI assistants.
- **Multi-hop Question Answering:** Handling complex, multi-step questions through iterative retrieval.
- **Legal Applications:** Retrieving legal documents and case law for reliable legal opinions.
- **General Task Assistance:** Aiding users in various tasks requiring information access and decision-making.

The rising demand for hyper-personalized content in areas like marketing and e-commerce is also a significant driver for RAG adoption, allowing for tailored ad copy and product recommendations.

Competitive Landscape

The document highlights Microsoft Corporation and Google LLC as frontrunners in the RAG market, with companies like Amazon Web Services, NVIDIA Corporation, and IBM Corporation as key innovators. Google's Vertex AI Search and its "Build your own RAG" architecture are presented as representative examples, emphasizing key components such as:

- **Document Processing:** Handling diverse formats, structural element extraction, and smart chunking.
- **Text Embeddings:** High-quality embeddings for effective retrieval, with tunable size and domain-specific tuning.
- **Vector Search:** Scalable, low-latency, and cost-effective storage and search for embeddings, supporting query-time filtering and hybrid search.
- **Semantic Ranking:** Re-ranking retrieved results for higher relevance using specialized models.
- **Grounded Generation:** Using LLMs (e.g., fine-tuned Gemini) to produce responses with citations based on retrieved context, reducing hallucinations.
- **Check Grounding:** Verifying factuality and identifying contradictions in generated statements.

Red Hat's Opportunity: A Modular AI System

The document identifies a clear "AI system" gap for Red Hat, particularly in the RAG implementation space. To address this, Red Hat plans to leverage its strengths to build a modular AI system architecture, with a focus on solid, reusable APIs for consumption by internal and external users.

Target Personas

Red Hat aims to build for both AI Platform Engineers (AIOps) and AI Engineers:

- **AI Platform Engineers (AIOps):** Responsible for configuring the platform and exposing primitives/APIs for AI Engineers.
- **AI Engineers:** Involved in building applications on top of foundation models, optimizing AI systems to deliver specific business outcomes.
- AI Users: The ultimate beneficiaries, who use the AI systems/applications built by AI Engineers.

See Who Are we Building For? The RAG Edition for more details/.

Differentiated Product Strategy

Red Hat's differentiation strategy centers on:

- **Open Source, Customizable, and Flexible:** Embracing its open-source heritage, Red Hat will offer opinionated defaults (like the Granite model and InstructLab for community-built LLMs) while ensuring extensibility and pluggability for other models, orchestrators, and document processors (e.g., Docling). The focus is on contracts, not rigid implementations.
- **Open Standards, Interoperability, and Specification (Desired):** Red Hat aims to drive or adopt open standards to ensure a seamless plug-and-play experience for modular RAG components, recognizing the high value of such standards despite their cost.
- **API-First Approach for AI Engineers:** Red Hat will leverage its deep expertise in API machinery (like Kubernetes) to expose granular API controls, prioritizing the AI engineer's experience and allowing customization of every aspect of the RAG pipeline.
- **Hybrid AI (Built on Hybrid Platforms) (Desired):** A core strength, offering consistent experiences across public/private clouds (via OpenShift), runtimes, and hardware. This provides customers with platform choice and helps them avoid vendor lock-in, a key differentiator from single-cloud providers.
- **Best-in-Class Module Customization (Model Layer):** Red Hat excels in "skill building" with models through InstructLab, which focuses on model distillation and synthetic data generation. This approach allows for fine-tuning models (including embedding and reranker models) to encapsulate domain-specific knowledge and lingua, complementing retrieval.

- **Enterprise-Grade & Responsible (Desired):** Prioritizing on-premise deployment, data control, compliance (HIPAA, GDPR), and building on secure, responsible technologies like Granite and OpenShift, with a focus on TrustyAI for explainability and safety.
- **Good Interfaces & Developer-centricity (Desired):** Providing robust UXes, SDKs, developer hubs, and a strong community to empower AI engineers.
- **Deep Value Chain:** Customers benefit from Red Hat's comprehensive support, tested products, training, consulting, packaging, distribution, operational knowledge, security, and brand reputation.

Vision and Principles

The overarching vision is to create a flexible AI ecosystem with modular components, interconnected through well-defined interfaces and APIs, aiming for standardization where possible. This will empower AI engineers to select, combine, and customize the best tools for their specific use cases. RAG serves as an initial focus, but the system will be designed to accommodate advancements beyond current RAG limitations, such as agentic RAG and broader orchestration/planning.
Key Principles for this long-term vision include:

- **Extensibility/Modularity:** Every component is a distinct, interchangeable module.
- **Interoperability**: Modules communicate through standardized interfaces.
- **User-Centric:** Empowering AI engineers and platform engineers to deliver great AI user experiences.
- **Strong Opinions, Loosely Held:** Providing opinionated defaults while allowing for extensive customization.

Red Hat's strategy for building an AI system, particularly a modular RAG platform, seems well-aligned with market needs and its core strengths as an open-source enterprise software provider.