Retrieval-Augmented Generation (RAG) – Deep Dive

- For Fiserv Internal AI Fluency Series – Sessions 1–4
- Covers Vector RAG, Hybrid RAG, Graph RAG, Agentic RAG
- Includes use cases, when to use which, FAQs

What is RAG?

- RAG = Retrieval-Augmented Generation
- LLM response is enhanced by retrieving external documents or facts
- Why? Because LLMs have knowledge cutoff, risk hallucination, and lack context

RAG Workflow (General)

- 1. User asks question
- 2. Retrieve relevant docs from knowledge base (vector DB, SQL, graph, etc.)
- 3. Inject docs into LLM prompt
- 4. LLM generates grounded answer with citations

Types of RAG (Covered in Sessions)

- Vector RAG: Embed text chunks, semantic similarity search
- Hybrid RAG: Combine unstructured (vector) + structured (SQL/metadata) data
- Graph RAG: Build entity/relation graph, improve reasoning/navigation
- Agentic RAG: LLM acts as a planner, calls multiple tools (vector, SQL, synthesis)

When to Use Which RAG?

- Vector RAG → Quick wins, FAQs, policy search
- Hybrid RAG → When rules/metadata exist in structured DBs (KYC, UPI, compliance)
- Graph RAG → When relationships/entities matter (merchant → product → rules)
- Agentic RAG → Complex, multi-step questions needing reasoning across sources

Demo Modes in UI

- In all sessions, UI lets you pick mode: Vector / Hybrid / Graph / Agentic
- Switch live to show differences in retrieved context & answers
- Transcript download = audit trail

Enterprise Considerations

- Accuracy: Add reranking, quality evals
- Safety: PII redaction, prompt guardrails, hallucination filters
- Scale: Chroma/Weaviate + SQL + Knowledge Graph
- Audit: Logging, prompt versioning, OpenTelemetry traces

Likely Attendee Questions (and Answers)

- Q: Why not fine-tune the LLM instead of RAG?
- A: RAG is cheaper, more flexible, avoids retraining for every update
- Q: How large can the knowledge base be?
- A: Vector DBs like Chroma scale to millions of docs; retrieval limited by embedding/search efficiency
- Q: Can RAG answer if context is missing?
- A: A good design makes model say 'Not found in context'
- Q: Which embeddings are best?
- A: Start with all-MiniLM (fast); upgrade to domain-specific models if needed
- Q: What about latency?
- A: Vector search adds ms; Agentic flow adds seconds due to multiple tool calls

Key Takeaways

- RAG = bridge between enterprise knowledge and LLM reasoning
- Choose RAG type based on data & question complexity
- UI demo highlights differences: Vector / Hybrid / Graph / Agentic
- Next: scaling RAG with monitoring, guardrails, and agent orchestration