TURNING SCATTERED INSTITUTIONAL KNOWLEDGE INTO AN ON-DEMAND, CONVERSATIONAL RESOURCE

Generative AI for Enterprise Knowledge Management

Executive briefing — CIO, COO & department heads | ~20 min

The Knowledge Silo Problem

- Institutional knowledge is fragmented across wikis, drives, tickets, Slack threads, and people's inboxes
- Employees spend a meaningful share of the workweek searching for information that already exists somewhere
- Tenure loss compounds the problem — departing experts take undocumented context with them
- Duplicate work and inconsistent answers arise when teams can't find the authoritative source
- Search tools built for the web don't understand internal jargon, org structure, or document context

How RAG Works: The Architecture Behind the Answer

- Retrieval-Augmented Generation (RAG) grounds LLM responses in your own documents, not just training data
- Pipeline: source systems → chunking → embedding model → vector database → retriever → LLM → cited answer
- Embedding models (e.g., OpenAI text-embedding-3, Cohere Embed) convert text into searchable vector representations
- Vector databases (Pinecone, Weaviate, pgvector, Azure AI Search) store and retrieve the most relevant chunks
- Generation layer (Anthropic Claude, Azure OpenAI, or self-hosted models) synthesizes retrieved context into a grounded response with citations

Where It Pays Off: Core Use Cases

- Enterprise search: natural-language queries across policies, contracts, and technical docs instead of keyword hunting
- Employee onboarding: new hires get instant, cited answers instead of waiting on a buddy or a stale wiki page
- SOP and compliance lookup: front-line staff get the current procedure, not last year's PDF version
- Internal help desk deflection: common IT/HR/legal questions answered without a ticket
- Cross-team research: engineers and analysts surface prior work before starting from scratch

Data Governance Is the Foundation, Not an Afterthought

- Establish a system of record: which repository is authoritative when documents conflict
- Define document lifecycle rules — retirement, versioning, and re-indexing when source content changes
- Classify content by sensitivity before ingestion; not everything belongs in a general-purpose index
- Assign content owners accountable for accuracy, freshness, and removal requests
- Governance failures surface as confidently wrong answers — the model will cite outdated policy with the same tone as current policy

Security and Access Control: Non-Negotiable by Design

- Retrieval must respect existing permissions — the system should never surface content a user couldn't already access directly
- Implement document-level or attribute-based access control (ABAC) inside the retrieval layer, not just at the app login
- Log every query and retrieved source for audit and incident response
- Keep sensitive categories (legal hold, HR case files, M&A materials) out of general-purpose indexes entirely
- Evaluate whether data leaves your tenant — prefer deployment models (private cloud, VPC, enterprise API agreements) that keep prompts and documents out of third-party training

Pilot Example: A 200-Person Engineering Organization

- Illustrative scenario, not a verified case study — presented to show realistic pilot scope and structure
- Scope: internal engineering wiki, architecture decision records, and past incident postmortems (roughly 15,000 documents)
- 6-week pilot with a 25-person volunteer group before wider rollout
- Success criteria set in advance: answer accuracy, citation correctness, and time-to-answer versus manual search
- Pilot findings shaped the access-control model and content curation process before broader deployment

Employee Adoption: What to Track and Expect

- Track weekly active users against total licensed seats, not just total logins
- Industry pilots commonly report 30-60% weekly active usage within the first quarter among engaged teams — treat as a directional benchmark, not a guarantee
- Monitor query-to-resolution rate: did the user get a usable answer without escalating to a human
- Watch for adoption plateaus after initial novelty — usually a signal of content gaps or trust issues, not a tooling failure
- Segment adoption by role and tenure to identify where the tool adds the most value first

The Real Cost of LLM API Usage

- Cost scales with tokens processed — both the retrieved context sent in and the answer generated out
- Long context windows improve answer quality but multiply per-query cost; chunk size tuning matters
- Enterprise-tier pricing (Anthropic, OpenAI, Azure OpenAI) typically ranges from fractions of a cent to a few cents per query depending on model and context length
- Caching frequent queries and reusing embeddings reduces recurring cost significantly at scale
- Budget for indexing and re-indexing costs separately from per-query inference costs — both grow with document volume

Build vs. Buy: Copilot, Claude, or Custom

- Microsoft Copilot: fastest path if already standardized on Microsoft 365; limited flexibility outside that ecosystem
- Anthropic Claude / Azure OpenAI via API: strong fit for custom retrieval pipelines and differentiated internal workflows
- Fully custom build: highest control and lowest long-term unit cost at scale, but requires sustained ML/platform engineering investment
- Decision hinges on three factors: data sensitivity requirements, existing tooling ecosystem, and internal engineering capacity
- Hybrid approach is common — off-the-shelf for general productivity, custom RAG for high-value proprietary knowledge domains

Integration With the Tools Employees Already Use

- Meet employees where they work: Slack, Microsoft Teams, or the intranet portal, not a new standalone app
- Connect to source systems via native connectors (SharePoint, Confluence, Google Drive, ServiceNow, Jira) to keep the index current
- Support single sign-on (SSO) and existing identity providers to avoid a parallel credential system
- Expose an API layer so other internal tools can embed knowledge retrieval rather than duplicating it
- Plan for connector maintenance — API changes in source systems can silently break ingestion pipelines

Change Management: The Part That Determines Success

- Technology adoption fails more often from trust and habit gaps than from model quality issues
- Identify and equip department champions who can model usage and answer peer questions early
- Communicate clearly what the tool is good at and where it still requires human judgment
- Retire or de-emphasize the old search habits you're replacing — parallel systems slow adoption
- Set a feedback loop so users can flag wrong or outdated answers, and show that reports get acted on

Managing the Risks: Hallucination and IP Leakage

- Hallucination risk drops significantly with RAG grounding but is never eliminated — require citations on every generated answer
- Establish a human-review threshold for high-stakes outputs (legal, financial, customer-facing)
- IP leakage risk: enforce data residency and confirm vendor contracts exclude your prompts and documents from model training
- Prompt injection from ingested documents is a real attack surface — sanitize and validate content before indexing
- Maintain an incident process specifically for AI-generated misinformation, separate from standard IT incident response

Measuring Success: The Metrics That Matter

- Time-to-answer: average time from question asked to usable answer received, versus prior manual search baseline
- Deflection rate: percentage of queries resolved without escalating to a human expert or support ticket
- Answer accuracy and citation correctness, sampled and reviewed on a recurring cadence
- Weekly active usage and retention across departments, not just total registered users
- Cost per resolved query, tracked against the manual-search baseline it replaces

Roadmap: From Pilot to Enterprise Scale

- Phase 1 (0-3 months): pilot with one department, narrow document scope, tight feedback loop
- Phase 2 (3-6 months): expand access controls, add source connectors, formalize governance ownership
- Phase 3 (6-12 months): scale to additional departments, integrate into existing workplace tools
- Phase 4 (12+ months): evaluate build-vs-buy economics at scale, optimize cost per query, explore agentic workflows
- Review checkpoints at each phase gate — proceed only when adoption and accuracy metrics from the prior phase hold