StackAI weaviate

Building Reliable Enterprise AI with Agentic RAG

Design architectures for building grounded, autonomous RAG agents at scale in production with StackAI and Weaviate.

Table of Contents

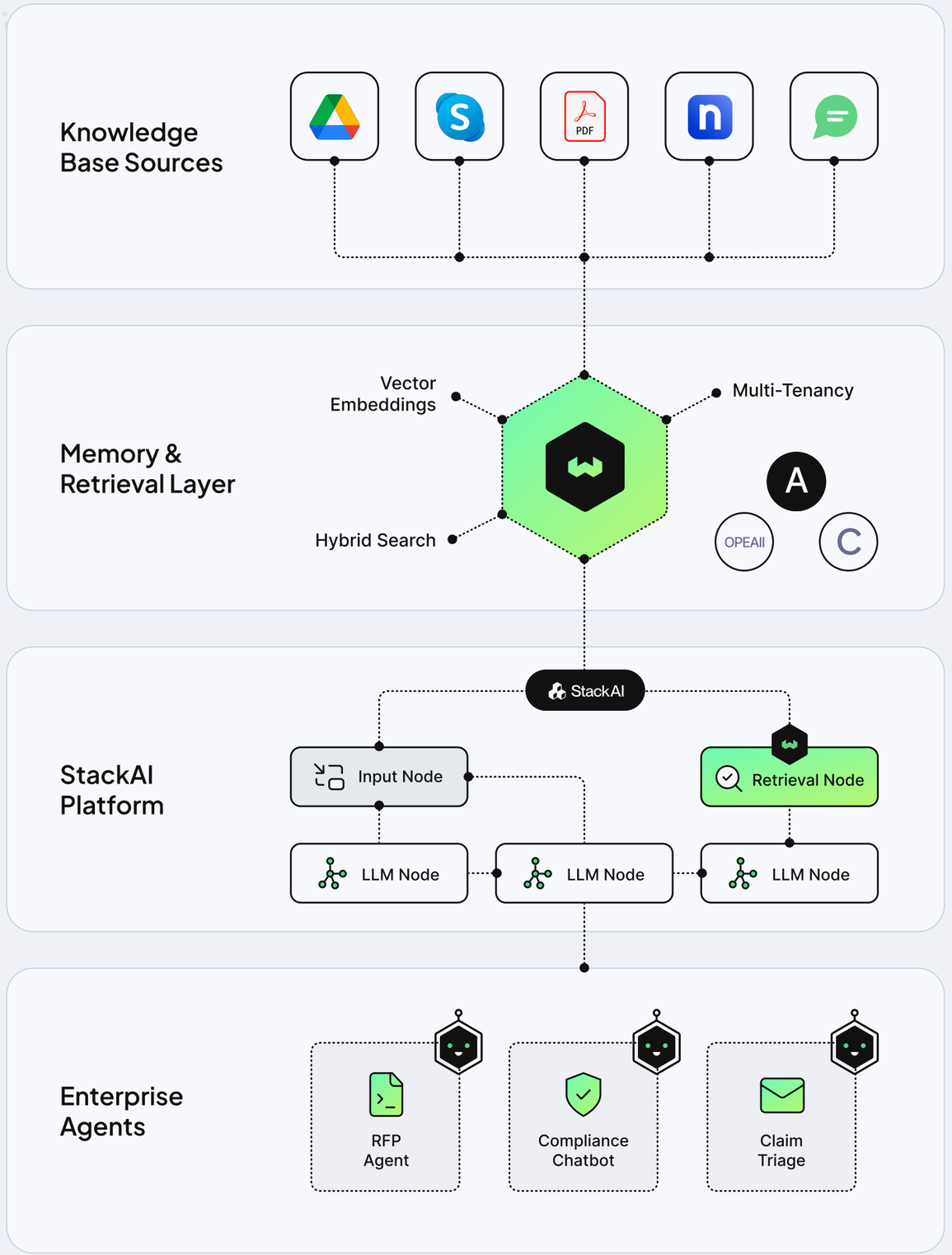
Introduction

Over the past two years, enterprise AI has gone from novelty to inevitability. Many large organizations are increasingly curious about implementing agentic AI throughout their operations—and actually, 79% of these enterprises expect full-scale adoption of agentic AI over the next three years. But hard questions of safety, adoption, and genuine usability means the leap from “interesting” to “operational” remains surprisingly difficult, leading to 95% of enterprise AI pilots failing in practice, according to MIT.

In many ways, this is because large language models (LLMs) like ChatGPT still cannot safely work with the data that actually runs an enterprise—uploading proprietary or sensitive client data to an external LLM presents many obvious noncompliance risks. Further, LLMs do not have the built-in access controls, permissions, or end-user connection check capabilities to meet enterprise security demands. But without context and grounding in internal knowledge bases, policies, regulatory guidance, or historical documents, enterprise AI is severely limited in usefulness and operationally untrustworthy for internal teams and stakeholders alike.

In this e-book, we’ll introduce Retrieval-Augmented Generation (RAG), which has emerged as the industry standard for building reliable and scalable AI systems. RAG grounds AI agents within real-world data, allowing them to safely interact with internal documents, retrieve key information, and cite important regulations and policies, allowing enterprise AI to provide and act on trustworthy and up-to-date information in the real world.

StackAI Platform with Weaviate Architecture

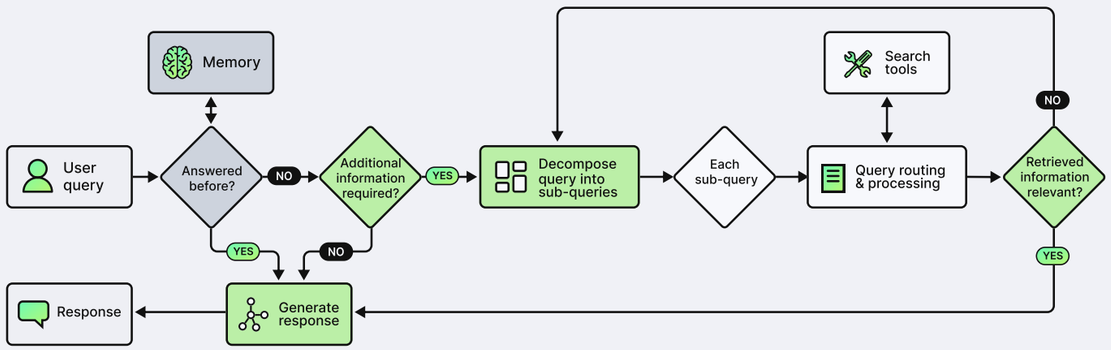


What Are AI Agents?

AI agents are the next step beyond basic generative AI. Unlike traditional chatbots that only respond to prompts, agents carry out real workflows across enterprise systems by making a series of autonomous decisions in order to reach a goal. An AI agent is generally made up of these four components: LLMs, access to tools, memory, and reasoning.

1 **Make dynamic decisions about information flow.** Rather than following a predetermined path, agents decide what to do next based on what they've learned.

2 **Maintain state across multiple interactions.** Unlike simple Q&A systems, agents remember what they've done and use that history to inform future decisions.

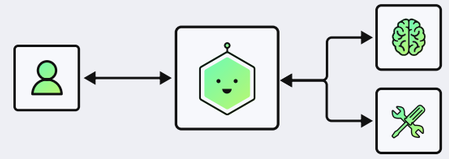


4 **Modify their approach based on results.** When one strategy isn't working, they can try different approaches.

3 **Use tools adaptively.** They can select from available tools and combine them in ways that weren't explicitly programmed.

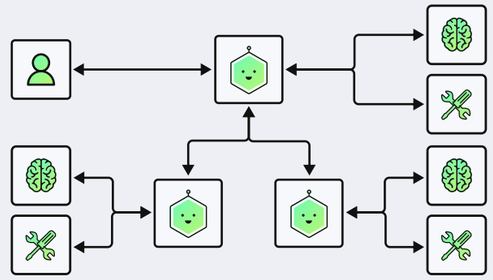
Single-Agent Architecture

Attempt to handle all tasks themselves, which works well for moderately complex workflows.



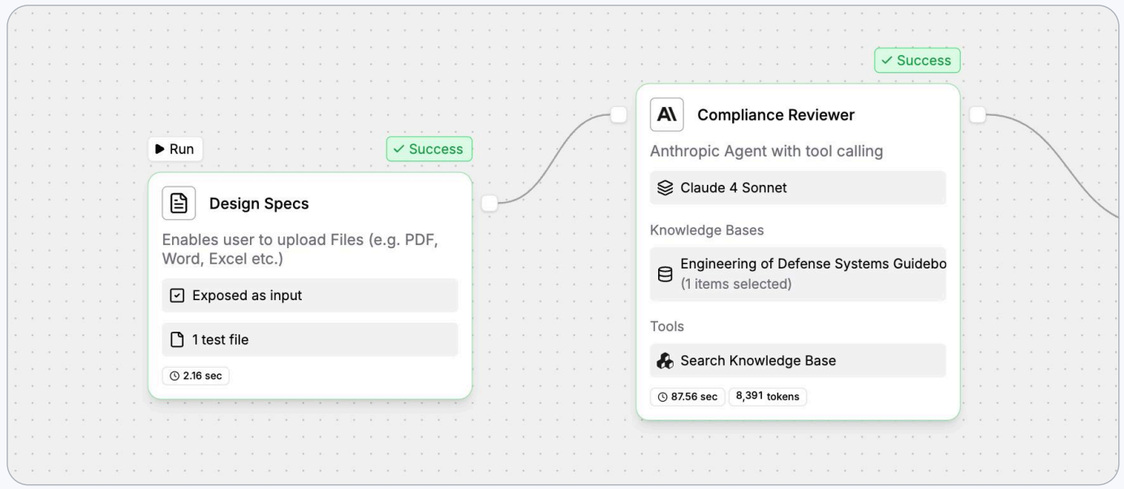
Multi-Agent Architecture

Distribute work across specialized agents per task. Allows for complex workflows but introduces coordination challenges.



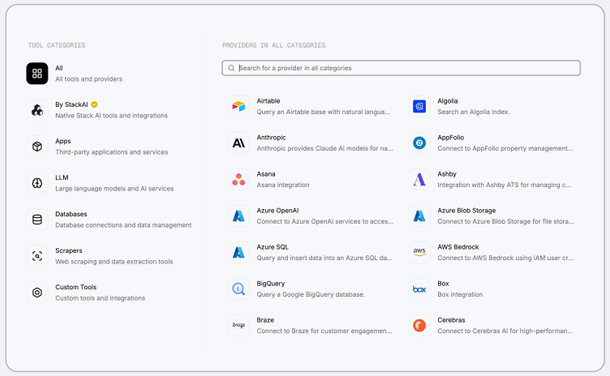
LLMs and Reasoning

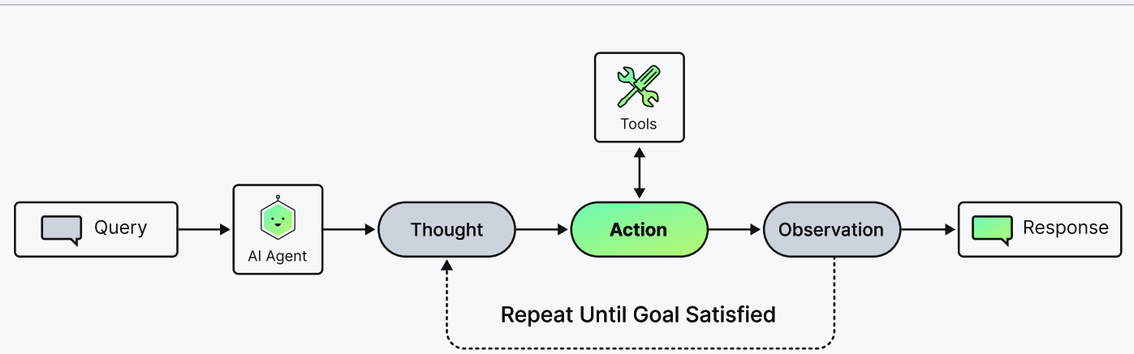
LLMs and Reasoning: LLMs/generative AI are the "brain" of the AI agent, having been prompted and instructed to ensure the completion of a certain goal. These models are able to reason over certain tasks: breaking problems down into manageable steps, deciding what tools to use and what material to reference, and determining when a response is complete. StackAI offers native access to all leading LLMs, so you can seamlessly drag-and-drop to pick the model that works best for a given task. From there, it's easy to add specific prompts and instructions, connect LLMs to inputs and data, and more.



Tools

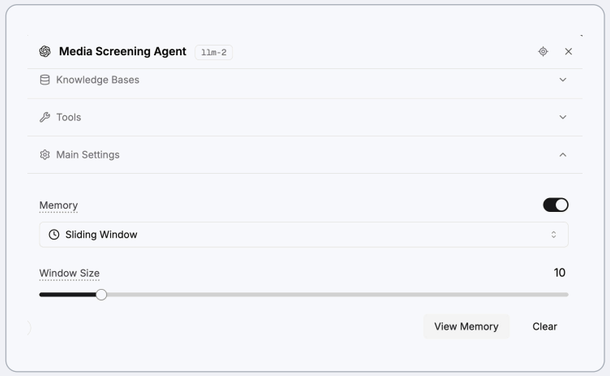
To take action (write to spreadsheets, reference a knowledge base, notify a channel in Slack, etc.), LLMs must be connected to tools or functions. In StackAI, giving agents tools is as simple as selecting actions in AppFolio, SharePoint, Smartsheet, and more, from one unified dropdown menu.





Memory

Agents often have some form of memory (both short-term and long-term), which allows them to store logs of their reasoning process, conversation histories, or information collected during different execution steps. In StackAI, you can adjust the memory for specific LLM nodes with a sliding scale and dropdown menu.



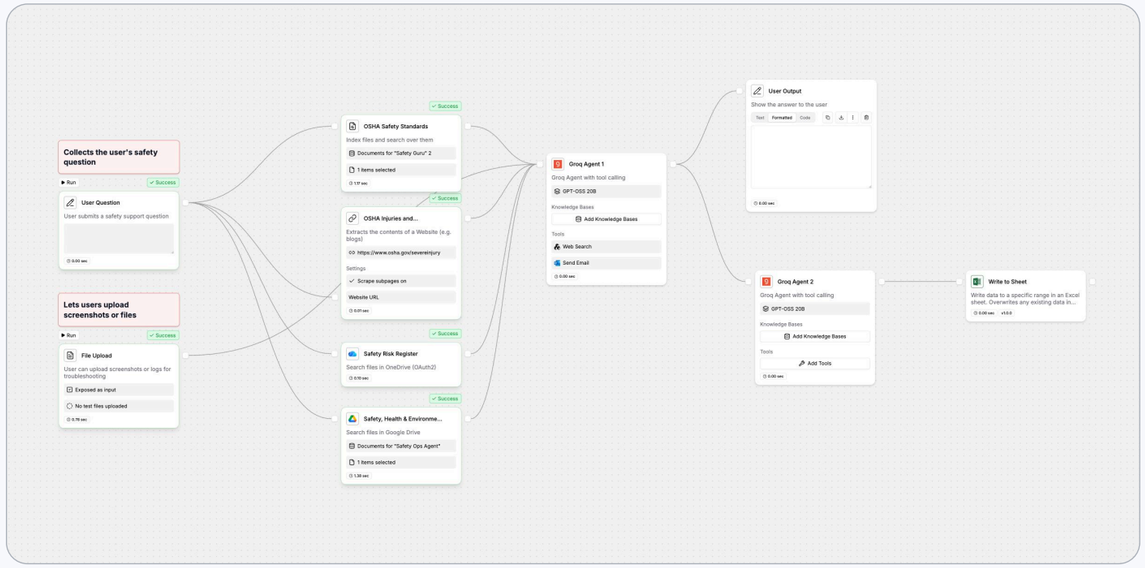
StackAI offers the option to switch between types of memory for even more customization.

This agent architecture allows an LLM to interact with real-time information to achieve a goal, which may be anything from retrieving information from internal databases, generating documents, routing decisions for approval, triggering next steps, and updating systems of record automatically.

Strategies and Tasks for Agents

Orchestration platforms like StackAI make agentic AI practical for enterprise teams. Instead of wrangling custom code, teams can design agents visually with a drag-and-drop interface and connect LLM nodes directly to tools and actions integrated with the software they already run: Salesforce, ServiceNow, SAP, NetSuite, Jira, Confluence, OneDrive, Teams, and more.

This enables fully operational workflows: an agent can pull historical context from a repository of thousands of documents, reason over it with several LLMs, draft a response or report, route it for human review in Teams, and write the final output back into a CRM or ticketing system from end to end. Critically, this all happens within a governed environment, with permissions, access controls, and auditability built in from the start.



As agents take on real business responsibility, every action, retrieval, decision, and update, must be grounded in real data and fully traceable end to end. The grounding layer that turns AI agents into trustworthy infrastructure is called Retrieval-Augmented Generation (RAG). Ultimately, RAG enables AI agents to act on official internal knowledge rather than guesswork, and to consistently, correctly reference source documents and policies every time.

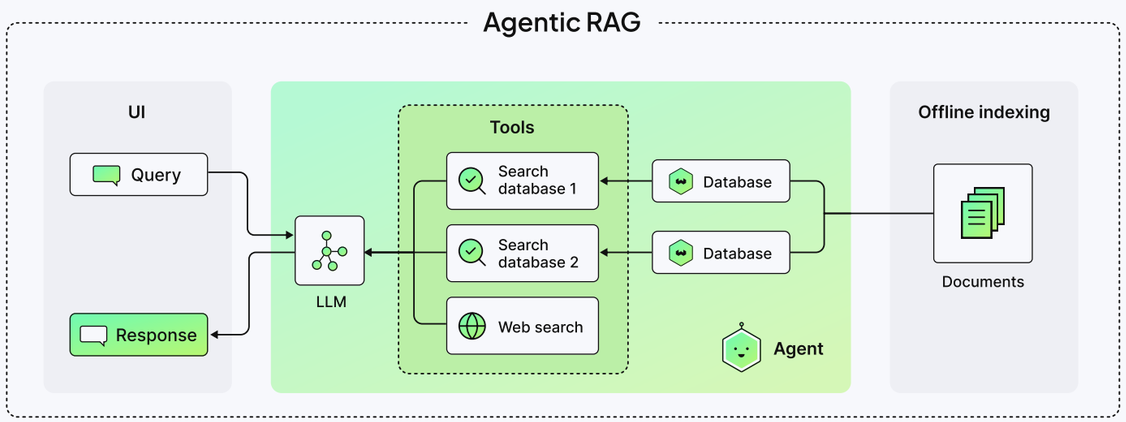
BUILDING RELIABLE ENTERPRISE AI WITH AGENTIC RAG

What Is Retrieval-Augmented Generation (RAG)?

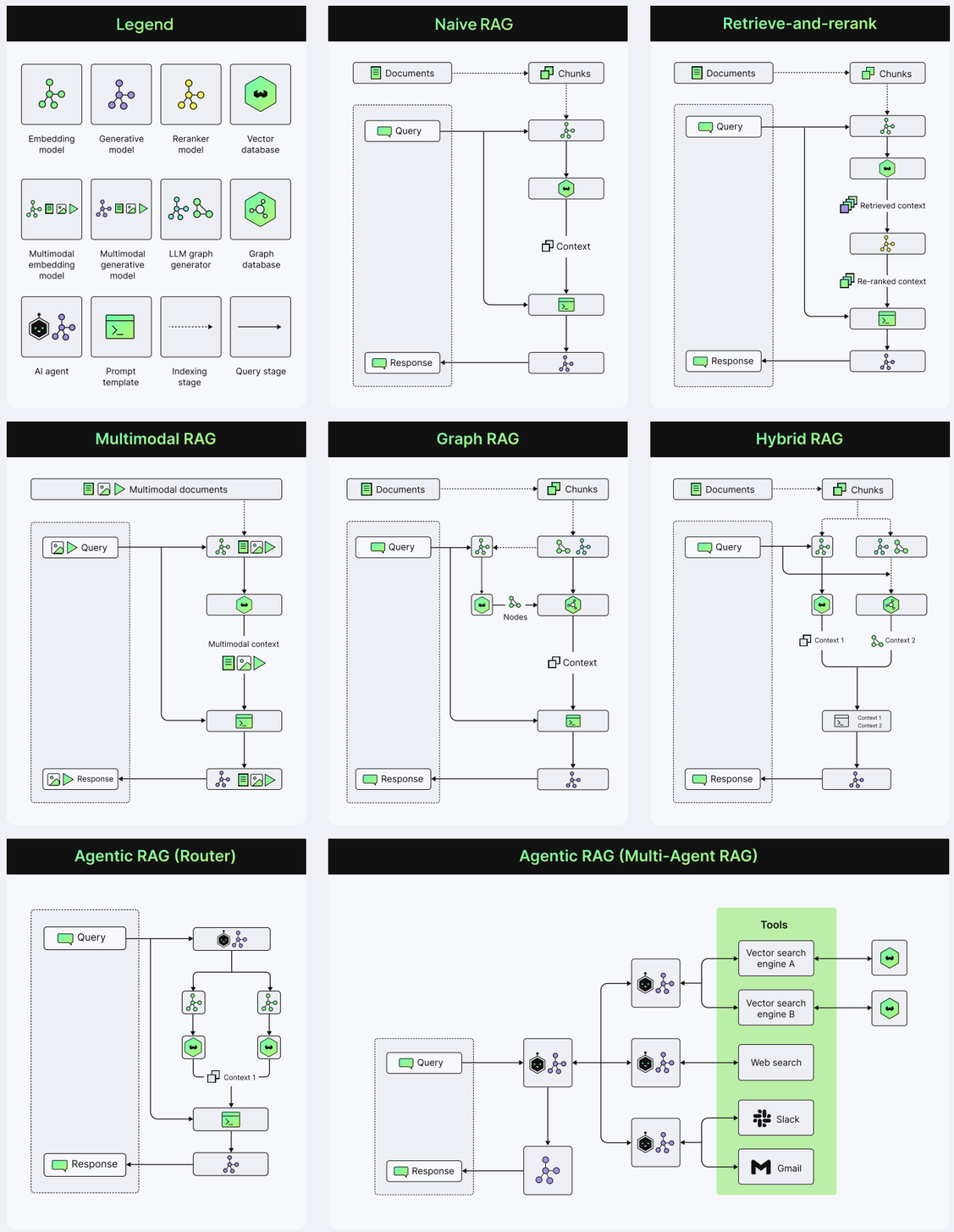
Retrieval-Augmented Generation, or RAG, is what allows AI systems to operate on an organization's real data, not just what an LLM learned during training. Weaviate serves as the core retrieval layer that makes this possible at enterprise scale.

Instead of asking an LLM to guess based on general knowledge, a RAG system built on Weaviate semantically indexes thousands of internal documents, policies, regulations, and repositories, retrieves the most relevant context in milliseconds, and passes that grounded information into the model for real-time reasoning and generation. The output is no longer just a fluent response, but a verifiable, auditable, and context-aware decision with traceability back to source material.

Powered by Weaviate, AI agents built on StackAI are true operational systems. Weaviate enables agents to search, rank, and reference massive internal knowledge bases in milliseconds using hybrid search, metadata filtering, and purpose-built multi-tenant architecture. The result is fast, accurate retrieval at scale, without sacrificing cost efficiency or security. Together, StackAI and Weaviate turn RAG from a concept into reliable, production-grade AI memory.



Retrieval-Augmented Generation Architectures



Check out Weaviate's e-book on Advanced RAG Techniques for more!

Best Practices for Building AI Agents with RAG

RAG agents only deliver enterprise value when they are implemented with discipline across retrieval, prompting, guardrails, and monitoring. The difference between a compelling demo and a reliable production system is almost always in these details.

The Security Layer

When security is built into the architecture and platform, AI agents can safely act on real enterprise data while staying compliant with enterprise needs.

The Retrieval Layer

Finding the right information requires retrieving the correct, authorized version at the right moment in a well-structured, governed knowledge base.

The Instruction Layer

Treating agent instructions as strict operating rules prevents hallucinations and ensures predictable, reliable outputs for enterprise workflows.

The Guardrail Layer

Guardrails enforce evaluation and fail-safe behavior to verify and ground agent outputs in authorized sources, reducing the need for human review.

Safety First

Enterprise AI agents should only be built on platforms that natively support secure knowledge base access, user authentication, and granular permission controls. For example, on StackAI, knowledge base access is tied directly to user authentication and downstream system permissions, so agents and end users can only retrieve documents they are already authorized to see. Additional safety features include role-based access controls, document-level permissions from connected systems like SharePoint or Confluence, and isolation across teams, departments, or regulated environments.

Because access policies are enforced at the platform level rather than stitched in at the prompt layer, agents operate inside the same trust boundaries as existing enterprise applications. That means outputs remain compliant by construction: no cross-department leakage, no accidental exposure of restricted data, and no audit gaps in how information was retrieved or used. The result is that AI agents can act on real internal knowledge without introducing new security or compliance risk.

True enterprise security isn't just a prompt instruction; it is a multi-layered architecture designed to prevent data leakage at every step:

End-user connection check and SSO

Authenticate users with existing enterprise identity verification providers such as Okta, or within integrations like SharePoint. This ensures that agents can validate user identities and access to data before processing requests or retrieving vectors.

Role- and group-based access control

Assign roles such as “Admin” and “Editor” to control who can run and build agents. Ensure that users in "Marketing" cannot trigger workflows or access tools reserved for "Legal," effectively enforcing the principle of least privilege.

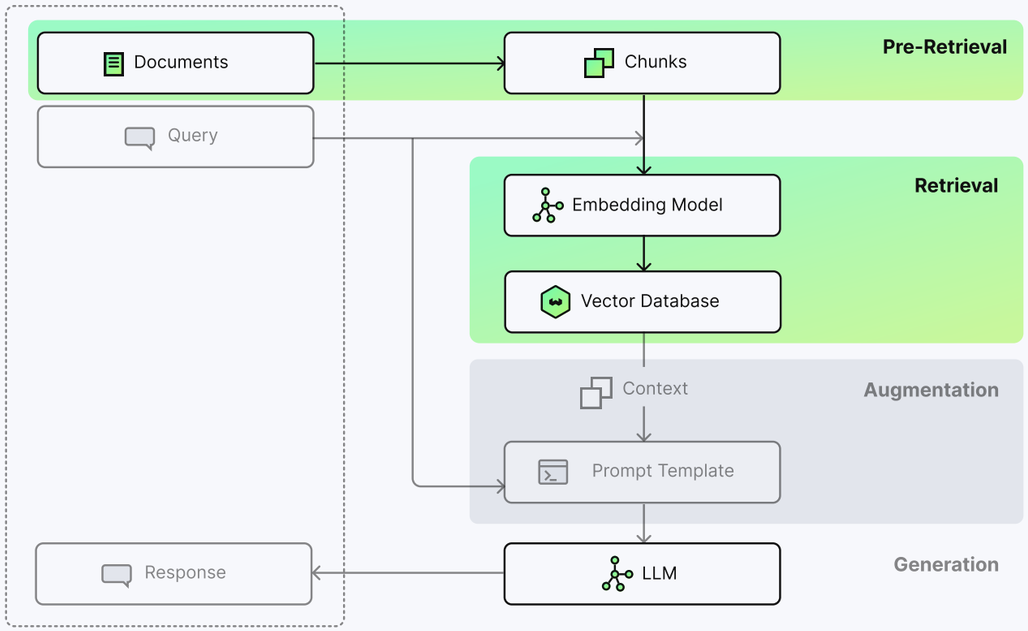
Knowledge base locking and permissions

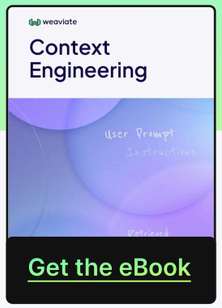
Enforce strict boundaries at the database level. Weaviate’s native multi-tenancy ensures that retrieval queries are physically and logically limited to the specific data subsets the user is authorized to see, preventing cross-department data leakage.

Retrieval Quality
(Finding the Right Information)

A RAG system is only as reliable as the way enterprise knowledge is structured, governed, and indexed. Documents must be broken into machine-readable sections, enriched with metadata (such as department, document type, and time sensitivity), and continuously kept in sync with source systems. This allows agents to retrieve not just a relevant document, but the correct version of the correct document at the correct moment in a workflow.

Weaviate combines semantic search with traditional keyword and metadata filtering, so AI agents on StackAI can reason over meaning while still respecting exact language, policy boundaries, and regulatory constraints. Agents can search by intent, filter by role or business unit, and return only authorized, contextually valid material.





Optimize Your Chunking Strategy

Getting retrieval *right* starts with how you **chunk your data**.

Should you split documents by paragraph, by token count, or by semantic meaning? The wrong choice can ruin your retrieval accuracy.

Learn to balance fixed-size, recursive, and semantic chunking strategies and more to find the 'sweet spot' for your data.

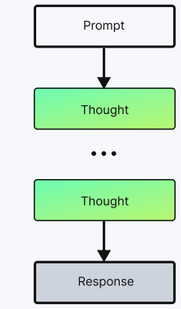
Prompt Design for Stability
(Clear Instructions, Predictable Outputs)

In real-world settings, the instructions given to AI agents need to be treated like operating rules. On StackAI, teams define clear boundaries for what each agent is allowed to do, and tasks are broken into simple, focused steps within a workflow. Requiring consistent output formats (such as fixed report templates or structured bullet points) makes results easier to trust and automate downstream. Just as important, agents are explicitly instructed not to guess: when the required information isn't available from authorized sources, the system returns a clear "no result" state rather than fabricating an answer.

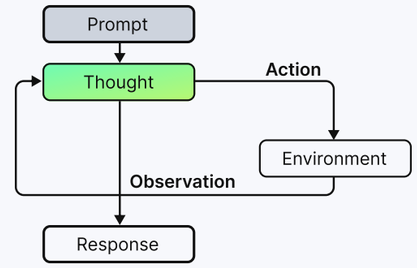
Classic Prompting Techniques

Advanced Prompting Strategies

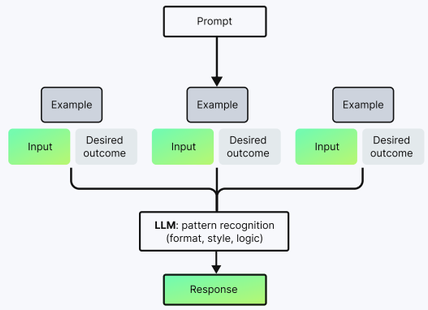
Chain of Thought



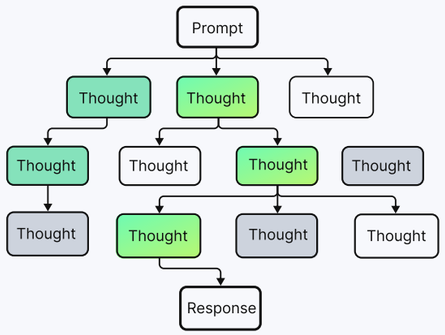
ReAct Prompting



Few-Shot Prompting



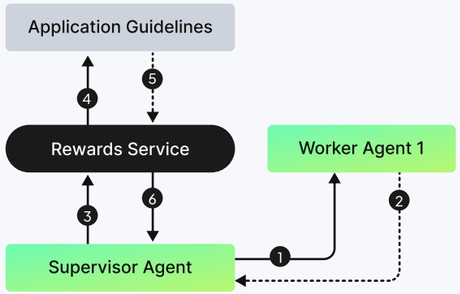
Tree of Thought (ToT)



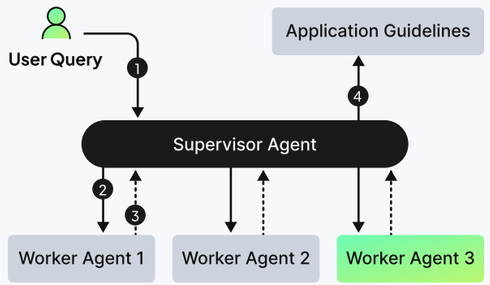
Guardrails That Enforce Grounding (Failing Safely)

Enterprise agents must be engineered to fail safely. Every response should be explicitly grounded in retrieved sources, and agents should be trained to acknowledge missing or insufficient context rather than fabricate answers. High-trust systems often implement multi-stage evaluation loops, where outputs pass through structured draft, critique, and verify stages before being finalized. This layered approach mirrors human review processes while preserving automation at scale.

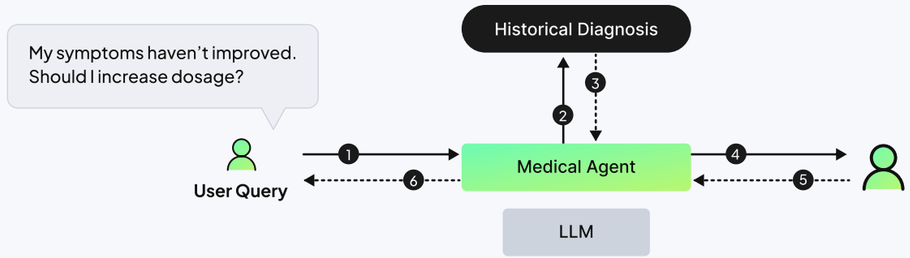
Adaptive feedback loops



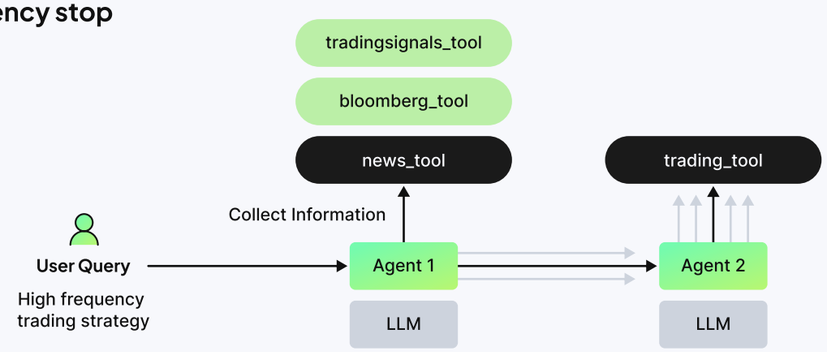
Corrective action



Human in the loop

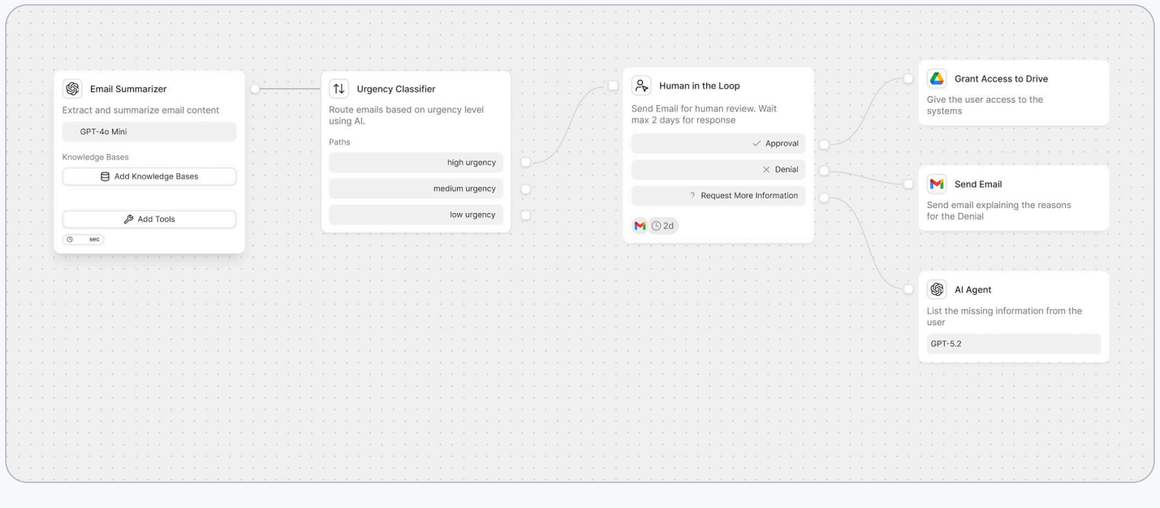


Emergency stop



Continuous Monitoring and Optimization

On StackAI, AI agents with RAG are not "set and forget" systems—teams can easily track retrieval failures, hallucination risk, and output stability over time, and evaluate test instructions to measure which prompt patterns consistently produce the most reliable results using a built-in LLM-as-a-judge feature. Retrieval quality is continuously evaluated to ensure that the most relevant, authoritative documents are still being surfaced as knowledge bases evolve.



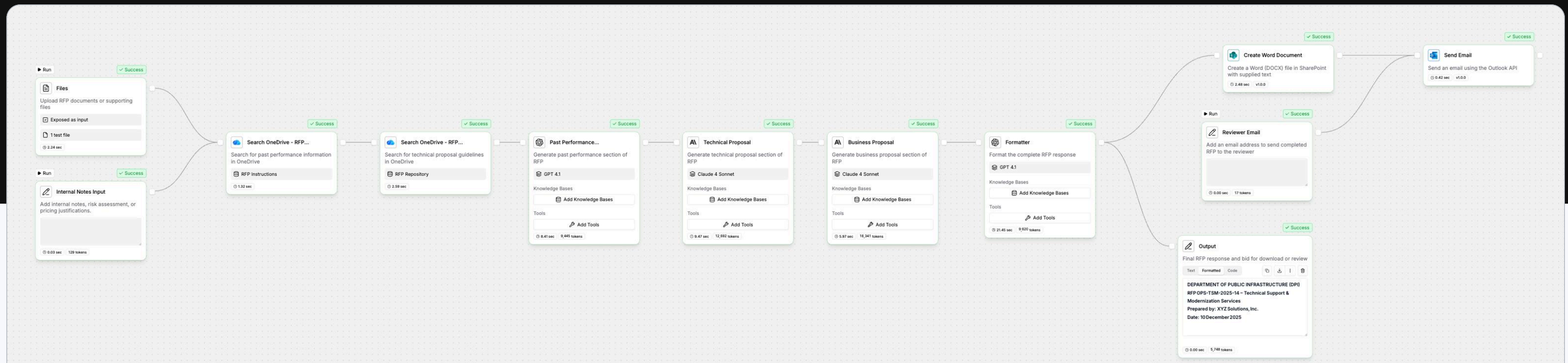
For high-impact steps, StackAI enables Human-in-the-Loop directly within the workflow. An agent can route a draft, decision, or action to a human reviewer via Teams or email, require explicit approval or denial, and only then proceed with a predefined downstream action. Once the human responds, the workflow resumes automatically, creating a controlled handoff between automated reasoning and real operational authority. This structure allows teams to scale agentic AI without sacrificing accountability or oversight.

Real-World Use Cases

Where RAG AI Agents Are Already Delivering Value

RFP Drafting Agent

Enterprises spend enormous amounts of time reconstructing institutional knowledge during proposal cycles, slowing down proposal development and introducing inconsistencies. A RAG-powered RFP agent retrieves relevant historical proposals, reference architectures, pricing or capability summaries, and prior win narratives, then assembles a first-pass draft aligned to past successes. With responses grounded in approved materials rather than improvised text, firms can accelerate proposal turnaround and maintain consistent messaging and standards.



1

Files → Accepts new uploaded RFP documents and extracts text using OCR.

2

Internal Notes → Captures user-provided context like risk assessments or pricing notes.

3

Reviewer Email → Collects the email address for sending the completed draft.

4

Search OneDrive - RFP Instructions → Retrieves formatting guidelines and evaluation criteria from OneDrive.

5

Search OneDrive - Past RFP Repository → Retrieves all relevant past RFPs from OneDrive.

6

Technical Proposal (AI) → LLM generates the section covering covering methodology, approach, and technical capabilities.

7

Business Proposal (AI) → LLM generates the section with pricing, value proposition, and compliance details.

8

Formatter (AI) → LLM assembles all sections into a cohesive, professionally structured document.

9

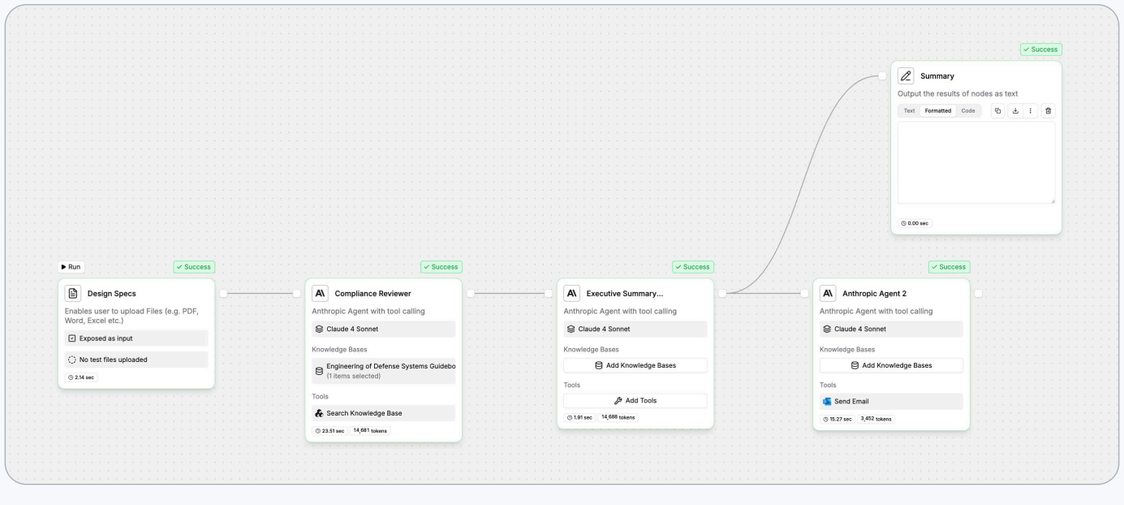
Create Word Document → Saves the draft as a DOCX file in a designated SharePoint.

10

Send Email → Notifies the reviewer that a draft is completed via Outlook with a link to the document.

Safety and Compliance Agent

Traditional compliance chatbots tend to hallucinate or generalize when they lack context, which is unacceptable in regulated environments where answers must be precise, traceable, and defensible. RAG-based compliance and safety agents avoid this problem by grounding every response in approved source material, including regulatory documents, internal compliance policies, official interpretations, audit frameworks, and historical rulings so enterprise teams can operate from the same authoritative guidance.



1. **Design Specs (Files Input)** → Accepts uploaded engineering design specification documents from users.

2. **Compliance Reviewer (AI)** → LLM acts as a senior engineering compliance reviewer. References internal engineering guidelines via Knowledge Base Search (e.g., Eng-Def-Sys-2022.pdf) and outputs a structured compliance assessment.

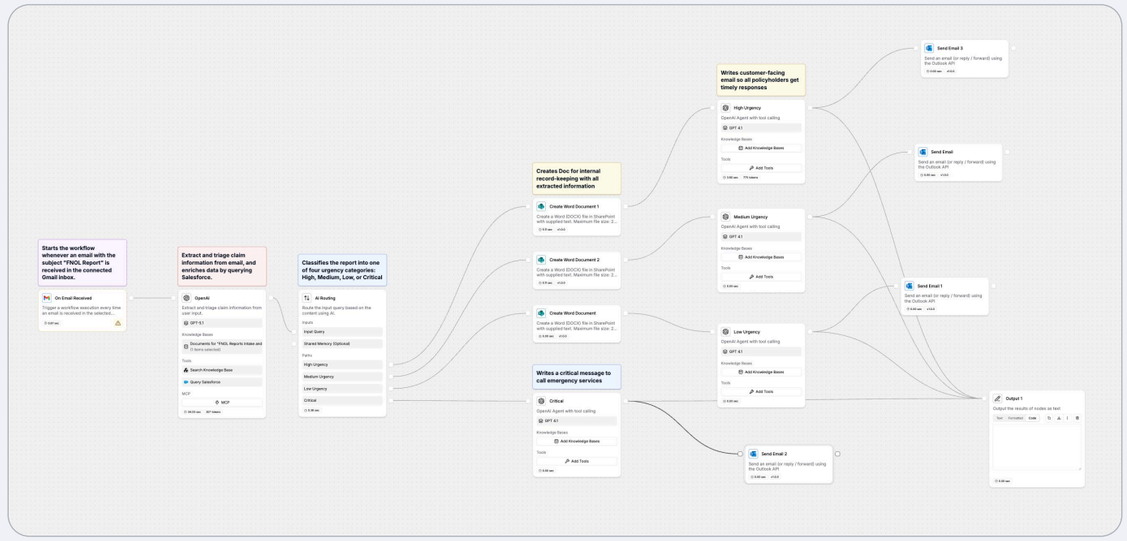
3. **Executive Summary Generator (AI)** → LLM produces a 2–4 paragraph executive summary for non-technical stakeholders (business leaders, project managers).

4. **Report Assembly & Delivery (AI)** → LLM combines the compliance review and executive summary into a formatted HTML report. Uses Outlook email integration to send the final report to the designated reviewer.

Weaviate combines semantic search with traditional keyword and metadata filtering, so AI agents on StackAI can reason over meaning while still respecting exact language, policy boundaries, and regulatory constraints. Agents can search by intent, filter by role or business unit, and return only authorized, contextually valid material.

Triage and Response Agent

Claims and ticketing workflows hinge on precise interpretation of coverage rules, exclusions, and procedural language. Generic chatbots are unsafe here because they improvise; firms need systems that cite the actual documents. RAG-powered agents solve this by retrieving official policy text and operational guidelines before generating an answer, ensuring every response is grounded in authoritative language. The result is standardized decisions, shorter cycle times, and consistent regulatory alignment.



1. **Email Trigger** → Automatically starts the workflow when a form with the subject “FNOL Report” arrives in the connected Gmail inbox. Captures sender, subject, and body metadata for downstream processing.

2. **Claim Extraction & Enrichment (AI)** → LLM parses the incoming email to extract key claim details.

3. **AI Routing: Urgency Classification (AI)** → LLM classifies the claim into one of four urgency tiers and routes accordingly.
   - **High Urgency Branch** → LLM sends the priority response to the policyholder.
   - **Medium Urgency Branch** → LLM sends the medium-priority response.
   - **Low Urgency Branch** → LLM sends the low-priority acknowledgment.
   - **Critical Branch** → LLM immediately sends an emergency advisory message instructing the policyholder to contact emergency services if in danger.

4. **Final Output Node** → Displays the formatted claim triage summary for internal review.

From AI Experimentation to Production in 2026

The first phase of enterprise AI was defined by experimentation: chat interfaces, pilot tools, and one-off automations that showed what models could do, but not how they would actually operate inside the business. 2026 will be the year of agentic AI systems that don't just respond to prompts, but take responsibility for real, multi-step work.

This is where platforms like StackAI, built on a high-performance retrieval layer from Weaviate, change the equation. Together, they allow agents to reason over authoritative internal knowledge, execute workflows across real enterprise systems, and operate inside defined permissions and authorizations. Further, StackAI's hardened governance and security features ensure safe, seamless data handling and transparency at every step when dealing with even thousands of official internal documents.

In the year ahead, the advantage won't belong to the companies that tried AI first, but to the ones that learned how to deploy agents as dependable collaborators inside real business workflows.

Want to learn more about how StackAI and Weaviate can help you automate manual, document-heavy processes at scale?

Get a Demo

About StackAI

StackAI

StackAI is the no-code, drag-and-drop platform for building enterprise-grade AI agents that automate manual work across every vertical. Connect 100+ tools like SAP, SharePoint, and ServiceNow with cutting-edge generative AI, then export agents as chatbots, forms, APIs, and more. Manage roles, access, and analytics in one secure, compliant environment—SOC 2, HIPAA, ISO, and GDPR approved. With white-glove support from AI experts, StackAI helps enterprises build, deploy, and scale AI safely for the fastest time to value. Built by MIT PhDs; trusted by finance, operations, and risk teams worldwide.

About Weaviate

weaviate

Weaviate is an AI database designed to store both data objects and their vector embeddings, enabling semantic and hybrid search, retrieval augmented generation (RAG), and agent-driven workflows for modern AI applications. Supporting text, images, and multimodal data, integrating with leading model providers (like OpenAI, Cohere, Google, Hugging Face, and AWS), and can automatically handle embedding generation, reranking, and generative AI through built-in model integrations.

Weaviate's use cases range from semantic search and recommendation systems to question answering, chatbots, content discovery, and anomaly detection, scaling from thousands to billions of objects with high-performance vector indexes.

Weaviate is available as open source for self-hosting and as Weaviate Cloud, a fully managed service with enterprise-grade security and compliance options, including SOC 2, ISO 27001, and HIPAA-ready offerings for sensitive workloads.