Use Case: Customer Support

Prompt: Can you tell me the delivery status of my order?

System Process (Single-Agent Workflow):

1. **Query Submission and Evaluation:**
   - The user submits the query, which is received by the coordinating agent.
   - The coordinating agent analyzes the query and determines the most appropriate sources of information.
2. **Knowledge Source Selection:**
   - Retrieves tracking details from the order management database.
   - Fetches real-time updates from the shipping provider's API.
   - Optionally conducts a web search to identify local conditions affecting delivery, such as weather or logistical delays.
3. **Data Integration and LLM Synthesis:**
   - The relevant data is passed to the LLM, which synthesizes the information into a coherent response.
4. **Output Generation:**
   - The system generates an actionable and concise response, providing live tracking updates and potential alternatives.

Response:
Integrated Response: "Your package is currently in transit and expected to arrive tomorrow evening. The live tracking from UPS indicates it is at the regional distribution center."

5.2 Multi-Agent Agentic RAG Systems:

Multi-Agent RAG [30] represents a modular and scalable evolution of single-agent architectures, designed to handle complex workflows and diverse query types by leveraging multiple specialized agents (as shown in Figure 17). Instead of relying on a single agent to manage all tasks—reasoning, retrieval, and response generation—this system distributes responsibilities across multiple agents, each optimized for a specific role or data source.

Workflow

1. **Query Submission**: The process begins with a user query, which is received by a *coordinator agent* or master retrieval agent. This agent acts as the central orchestrator, delegating the query to specialized retrieval agents based on the query's requirements.
2. **Specialized Retrieval Agents**: The query is distributed among multiple retrieval agents, each focusing on a specific type of data source or task. Examples include:
   - **Agent 1**: Handles structured queries, such as interacting with SQL-based databases like PostgreSQL or MySQL.
   - **Agent 2**: Manages semantic searches for retrieving unstructured data from sources like PDFs, books, or internal records.
   - **Agent 3**: Focuses on retrieving real-time public information from web searches or APIs.
   - **Agent 4**: Specializes in recommendation systems, delivering context-aware suggestions based on user behavior or profiles.
3. **Tool Access and Data Retrieval**: Each agent routes the query to the appropriate tools or data sources within its domain, such as:
   - *Vector Search*: For semantic relevance.
   - *Text-to-SQL*: For structured data.
   - *Web Search*: For real-time public information.
   - *APIs*: For accessing external services or proprietary systems.

   The retrieval process is executed in parallel, allowing for efficient processing of diverse query types.

Use Case: Multi-Domain Research Assistant

Prompt: What are the economic and environmental impacts of renewable energy adoption in Europe?

System Process (Multi-Agent Workflow):

- **Agent 1:** Retrieves statistical data from economic databases using SQL-based queries.
- **Agent 2:** Searches for relevant academic papers using semantic search tools.
- **Agent 3:** Performs a web search for recent news and policy updates on renewable energy.
- **Agent 4:** Consults a recommendation system to suggest related content, such as reports or expert commentary.

Response:
Integrated Response: "Adopting renewable energy in Europe has led to a 20% reduction in greenhouse gas emissions over the past decade, according to EU policy reports. Economically, renewable energy investments have generated approximately 1.2 million jobs, with significant growth in solar and wind sectors. Recent academic studies also highlight potential trade-offs in grid stability and energy storage costs."

5.3 Hierarchical Agentic RAG Systems

Hierarchical Agentic RAG: [14] systems employ a structured, multi-tiered approach to information retrieval and processing, enhancing both efficiency and strategic decision-making as shown in Figure 18. Agents are organized in a hierarchy, with higher-level agents overseeing and directing lower-level agents. This structure enables multi-level decision-making, ensuring that queries are handled by the most appropriate resources.

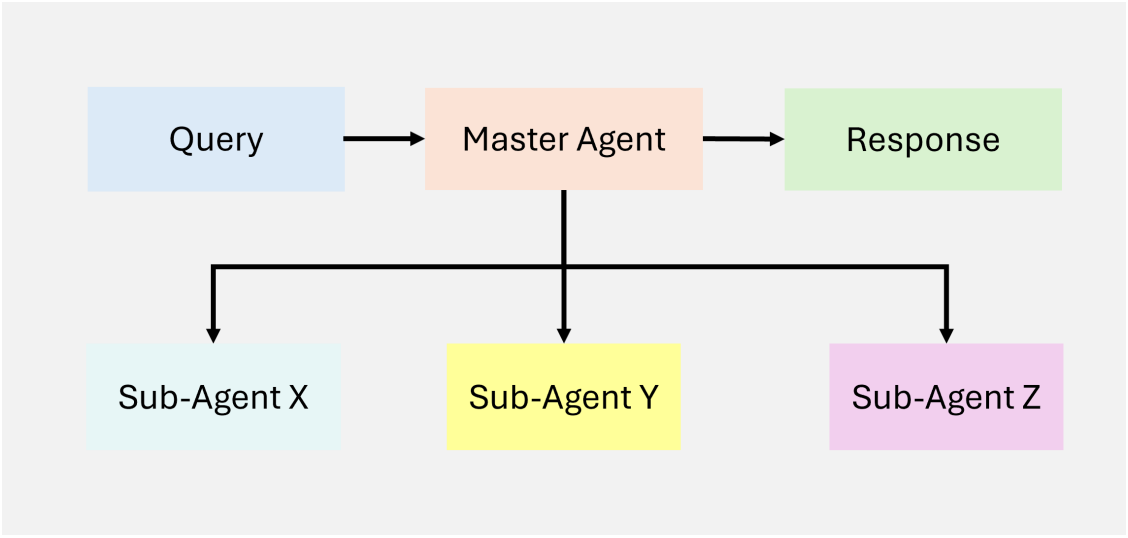

Figure 18: An illustration of Hierarchical Agentic RAG

Workflow

1. **Query Reception**: A user submits a query, received by a *top-tier agent* responsible for initial assessment and delegation.
2. **Strategic Decision-Making**: The top-tier agent evaluates the query's complexity and decides which subordinate agents or data sources to prioritize. Certain databases, APIs, or retrieval tools may be deemed more reliable or relevant based on the query's domain.
3. **Delegation to Subordinate Agents**: The top-tier agent assigns tasks to lower-level agents specialized in particular retrieval methods (e.g., SQL databases, web search, or proprietary systems). These agents execute their assigned tasks independently.

4. **Aggregation and Synthesis**: The results from subordinate agents are collected and integrated by the higher-level agent, which synthesizes the information into a coherent response.
5. **Response Delivery**: The final, synthesized answer is returned to the user, ensuring that the response is both comprehensive and contextually relevant.

Key Features and Advantages.

- **Strategic Prioritization**: Top-tier agents can prioritize data sources or tasks based on query complexity, reliability, or context.
- **Scalability**: Distributing tasks across multiple agent tiers enables handling of highly complex or multi-faceted queries.
- **Enhanced Decision-Making**: Higher-level agents apply strategic oversight to improve overall accuracy and coherence of responses.

Challenges

- **Coordination Complexity**: Maintaining robust inter-agent communication across multiple levels can increase orchestration overhead.
- **Resource Allocation**: Efficiently distributing tasks among tiers to avoid bottlenecks is non-trivial.

Use Case: Financial Analysis System

Prompt: What are the best investment options given the current market trends in renewable energy?

System Process (Hierarchical Agentic Workflow):

1. **Top-Tier Agent**: Assesses the query's complexity and prioritizes reliable financial databases and economic indicators over less validated data sources.
2. **Mid-Level Agent**: Retrieves real-time market data (e.g., stock prices, sector performance) from proprietary APIs and structured SQL databases.
3. **Lower-Level Agent(s)**: Conducts web searches for recent policy announcements and consults recommendation systems that track expert opinions and news analytics.
4. **Aggregation and Synthesis**: The top-tier agent compiles the results, integrating quantitative data with policy insights.

Response:
Integrated Response: "Based on current market data, renewable energy stocks have shown a 15% growth over the past quarter, driven by supportive government policies and heightened investor interest. Analysts suggest that wind and solar sectors, in particular, may experience continued momentum, while emerging technologies like green hydrogen present moderate risk but potentially high returns."

5.4 Agentic Corrective RAG

Corrective RAG : introduces mechanisms to self-correct retrieval results, enhancing document utilization and improving response generation quality as demonstrated in Figure 19. By embedding intelligent agents into the workflow, Corrective RAG [31] [32] ensures iterative refinement of context documents and responses, minimizing errors and maximizing relevance.

Key Idea of Corrective RAG: The core principle of Corrective RAG lies in its ability to evaluate retrieved documents dynamically, perform corrective actions, and refine queries to enhance the quality of generated responses. Corrective RAG adjusts its approach as follows:

- **Document Relevance Evaluation:** Retrieved documents are assessed for relevance by the *Relevance Evaluation Agent*. Documents below the relevance threshold trigger corrective steps.
- **Query Refinement and Augmentation:** Queries are refined by the *Query Refinement Agent*, which leverages semantic understanding to optimize retrieval for better results.

Use Case: Academic Research Assistant

Prompt: What are the latest findings in generative AI research?

System Process (Corrective RAG Workflow):

1. **Query Submission:** A user submits the query to the system.
2. **Context Retrieval:**
   - The *Context Retrieval Agent* retrieves initial documents from a database of published papers on generative AI.
   - The retrieved documents are passed to the next step for evaluation.
3. **Relevance Evaluation:**
   - The *Relevance Evaluation Agent* assesses the documents for alignment with the query.
   - Documents are classified into relevant, ambiguous, or irrelevant categories. Irrelevant documents are flagged for corrective actions.
4. **Corrective Actions (if needed):**
   - The *Query Refinement Agent* rewrites the query to improve specificity and relevance.
   - The *External Knowledge Retrieval Agent* performs web searches to fetch additional papers and reports from external sources.
5. **Response Synthesis:**
   - The *Response Synthesis Agent* integrates validated documents into a coherent and comprehensive summary.

Response:
Integrated Response: "Recent findings in generative AI highlight advancements in diffusion models, reinforcement learning for text-to-video tasks, and optimization techniques for large-scale model training. For more details, refer to studies published in NeurIPS 2024 and AAAI 2025."

5.5 Adaptive Agentic RAG

Adaptive Retrieval-Augmented Generation (Adaptive RAG) [33] enhances the flexibility and efficiency of large language models (LLMs) by dynamically adjusting query handling strategies based on the complexity of the incoming query. Unlike static retrieval workflows, Adaptive RAG [34] employs a classifier to assess query complexity and determine the most appropriate approach, ranging from single-step retrieval to multi-step reasoning, or even bypassing retrieval altogether for straightforward queries as illustrated in Figure 20.

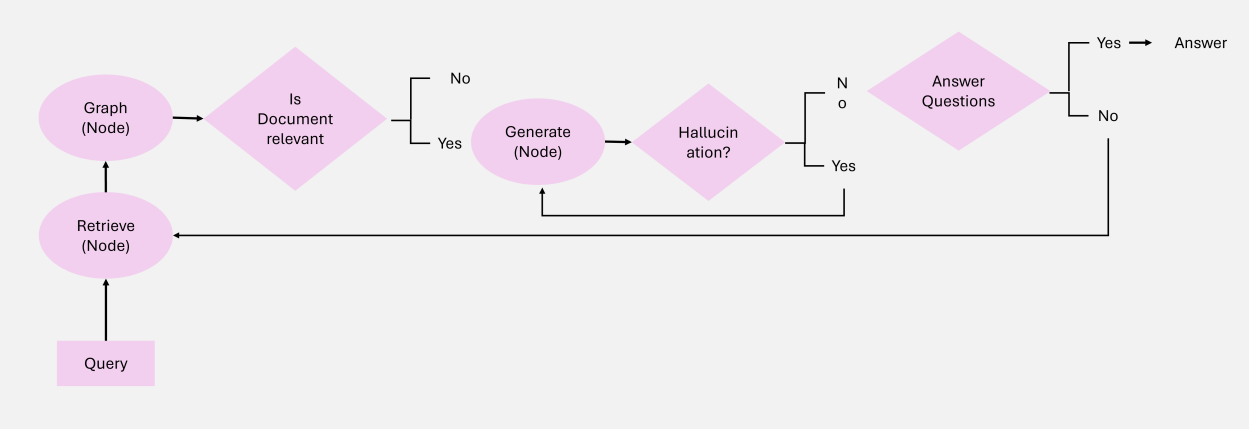

Figure 20: An Overview of Adaptive Agentic RAG

Key Idea of Adaptive RAG The core principle of Adaptive RAG lies in its ability to dynamically tailor retrieval strategies based on the complexity of the query. Adaptive RAG adjusts its approach as follows:

Use Case: Customer Support Assistant

Prompt: Why is my package delayed, and what alternatives do I have?

System Process (Adaptive RAG Workflow):

1. **Query Classification:**
   - The classifier analyzes the query and determines it to be complex, requiring multi-step reasoning.
2. **Dynamic Strategy Selection:**
   - The system activates a multi-step retrieval process based on the complexity classification.
3. **Multi-Step Retrieval:**
   - Retrieves tracking details from the order database.
   - Fetches real-time status updates from the shipping provider API.
   - Conducts a web search for external factors such as weather conditions or local disruptions.
4. **Response Synthesis:**
   - The LLM integrates all retrieved information, synthesizing a comprehensive and actionable response.

Response:
Integrated Response: "Your package is delayed due to severe weather conditions in your region. It is currently at the local distribution center and will be delivered in 2 days. Alternatively, you may opt for a local pickup from the facility."

5.6 Graph-Based Agentic RAG

5.6.1 Agent-G: Agentic Framework for Graph RAG

Agent-G [8]: introduces a novel agentic architecture that integrates graph knowledge bases with unstructured document retrieval. By combining structured and unstructured data sources, this framework enhances retrieval-augmented generation (RAG) systems with improved reasoning and retrieval accuracy. It employs modular retriever banks, dynamic agent interaction, and feedback loops to ensure high-quality outputs as shown in Figure 21.

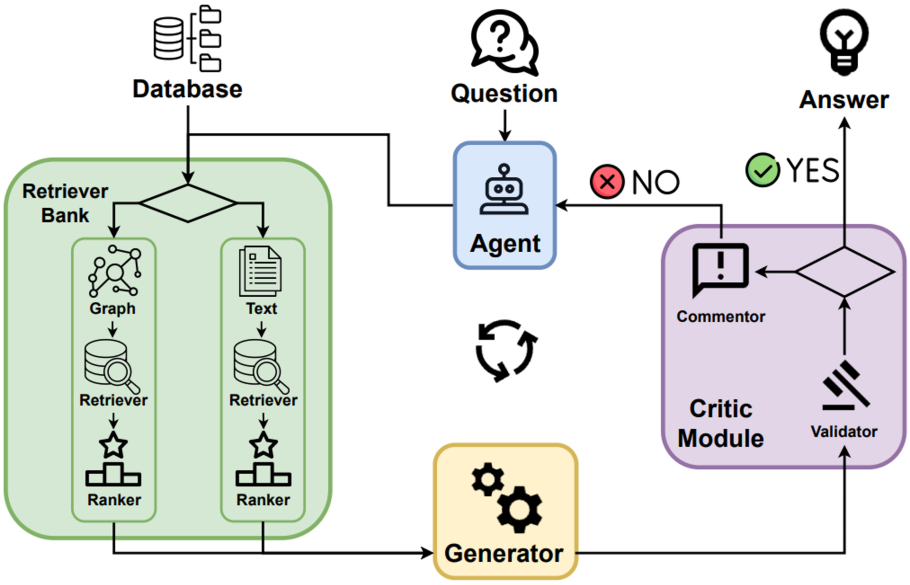

Figure 21: An Overview of Agent-G: Agentic Framework for Graph RAG [8]

Use Case: Healthcare Diagnostics

Prompt: What are the common symptoms of Type 2 Diabetes, and how are they related to heart disease?

System Process (Agent-G Workflow):

1. **Query Reception and Assignment:** The system receives the query and identifies the need for both graph-structured and unstructured data to answer the question comprehensively.
2. **Graph Retriever:**
   - Extracts relationships between Type 2 Diabetes and heart disease from a medical knowledge graph.
   - Identifies shared risk factors such as obesity and high blood pressure by exploring graph hierarchies and relationships.
3. **Document Retriever:**
   - Retrieves descriptions of Type 2 Diabetes symptoms (e.g., increased thirst, frequent urination, fatigue) from medical literature.
   - Adds contextual information to complement the graph-based insights.
4. **Critic Module:**
   - Evaluates the relevance and quality of the retrieved graph data and document data.
   - Flags low-confidence results for refinement or re-querying.
5. **Response Synthesis:** The LLM integrates validated data from the Graph Retriever and Document Retriever into a coherent response, ensuring alignment with the query's intent.

Response:
Integrated Response: "Type 2 Diabetes symptoms include increased thirst, frequent urination, and fatigue. Studies show a 50% correlation between diabetes and heart disease, primarily through shared risk factors such as obesity and high blood pressure."

5.6.2 GeAR: Graph-Enhanced Agent for Retrieval-Augmented Generation

GeAR [35]: introduces an agentic framework that enhances traditional Retrieval-Augmented Generation (RAG) systems by incorporating graph-based retrieval mechanisms. By leveraging graph expansion techniques and an agent-based architecture, GeAR addresses challenges in multi-hop retrieval scenarios, improving the system's ability to handle complex queries as shown in Figure 22.

Key Idea of GeAR GeAR advances RAG performance through two primary innovations:

- **Graph Expansion:** Enhances conventional base retrievers (e.g., BM25) by expanding the retrieval process to include graph-structured data, enabling the system to capture complex relationships and dependencies between entities.
- **Agent Framework:** Incorporates an agent-based architecture that utilizes graph expansion to manage retrieval tasks more effectively, allowing for dynamic and autonomous decision-making in the retrieval process.

Workflow: The GeAR system operates through the following components:

1. **Graph Expansion Module:**
   - Integrates graph-based data into the retrieval process, allowing the system to consider relationships between entities during retrieval.
   - Enhances the base retriever's ability to handle multi-hop queries by expanding the search space to include connected entities.
2. **Agent-Based Retrieval:**
   - Employs an agent framework to manage the retrieval process, enabling dynamic selection and combination of retrieval strategies based on the query's complexity.
   - Agents can autonomously decide to utilize graph-expanded retrieval paths to improve the relevance and accuracy of retrieved information.

Use Case: Multi-Hop Question Answering

Prompt: Which author influenced the mentor of J.K. Rowling?

System Process (GeAR Workflow):

1. **Top-Tier Agent**: Evaluates the query's multi-hop nature and determines that a combination of graph expansion and document retrieval is necessary to answer the question.
2. **Graph Expansion Module**:
   - Identifies that J.K. Rowling's mentor is a key entity in the query.
   - Traces the literary influences on that mentor by exploring graph-structured data on literary relationships.
3. **Agent-Based Retrieval**:
   - An agent autonomously selects the graph-expanded retrieval path to gather relevant information about the mentor's influences.
   - Integrates additional context by querying textual data sources for unstructured details about the mentor and their influences.
4. **Response Synthesis**: Combines insights from the graph and document retrieval processes using the LLM to generate a response that accurately reflects the complex relationships in the query.

Response:
Integrated Response: "J.K. Rowling's mentor, [Mentor Name], was heavily influenced by [Author Name], known for their [notable works or genre]. This connection highlights the layered relationships in literary history, where influential ideas often pass through multiple generations of authors."

5.7 Agentic Document Workflows in Agentic RAG

Agentic Document Workflows (ADW) [36] extend traditional Retrieval-Augmented Generation (RAG) paradigms by enabling end-to-end knowledge work automation. These workflows orchestrate complex document-centric processes, integrating document parsing, retrieval, reasoning, and structured outputs with intelligent agents (see Figure 23). ADW systems address limitations of Intelligent Document Processing (IDP) and RAG by maintaining state, coordinating multi-step workflows, and applying domain-specific logic to documents.

Workflow

1. **Document Parsing and Information Structuring:**
   - Documents are parsed using enterprise-grade tools (e.g., LlamaParse) to extract relevant data fields such as invoice numbers, dates, vendor information, line items, and payment terms.
   - Structured data is organized for downstream processing.
2. **State Maintenance Across Processes:**
   - The system maintains state about document context, ensuring consistency and relevance across multi-step workflows.
   - Tracks the progression of the document through various processing stages.
3. **Knowledge Retrieval:**
   - Relevant references are retrieved from external knowledge bases (e.g., LlamaCloud) or vector indexes.
   - Retrieves real-time, domain-specific guidelines for enhanced decision-making.
4. **Agentic Orchestration:**
   - Intelligent agents apply business rules, perform multi-hop reasoning, and generate actionable recommendations.
   - Orchestrates components such as parsers, retrievers, and external APIs for seamless integration.
5. **Actionable Output Generation:**
   - Outputs are presented in structured formats, tailored to specific use cases.
   - Recommendations and extracted insights are synthesized into concise and actionable reports.

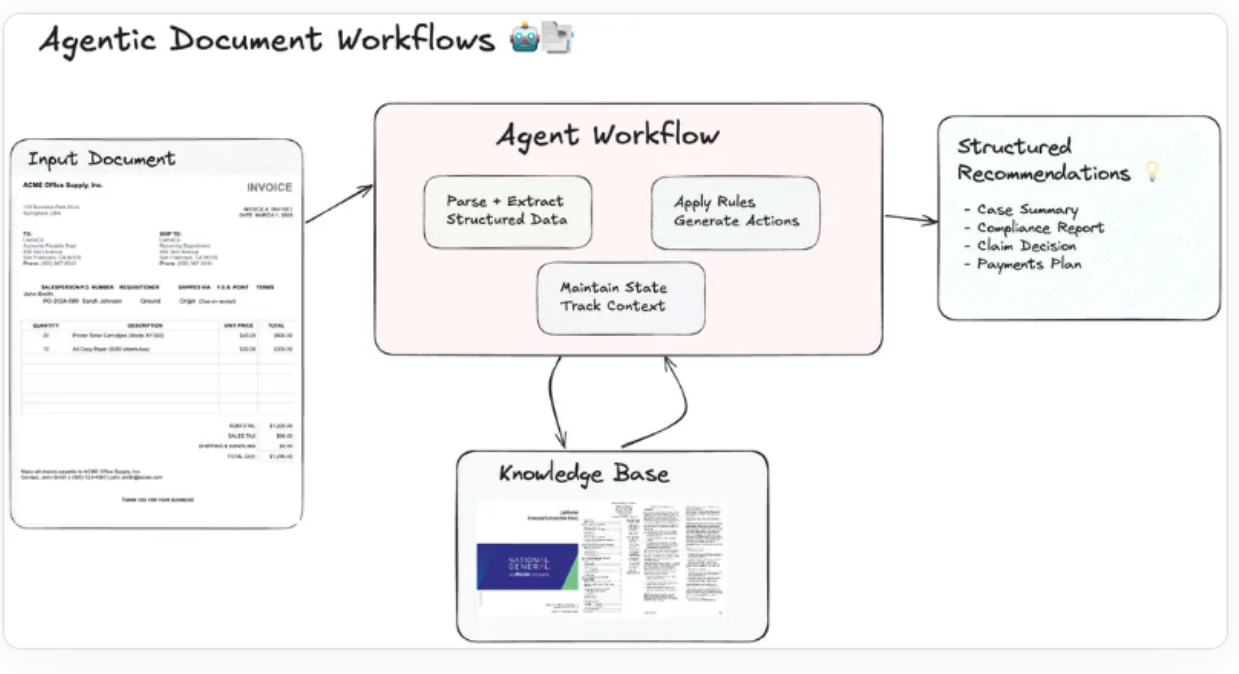

Figure 23: An Overview of Agentic Document Workflows (ADW) [36]

Use Case: Invoice Payments Workflow
Prompt: Generate a payment recommendation report based on the submitted invoice and associated vendor contract terms. **System Process (ADW Workflow):** 1. Parse the invoice to extract key details such as invoice number, date, vendor information, line items, and payment terms. 2. Retrieve the corresponding vendor contract to verify payment terms and identify any applicable discounts or compliance requirements. 3. Generate a payment recommendation report that includes original amount due, potential early payment discounts, budget impact analysis, and strategic payment actions. **Response:** *Integrated Response:* "Invoice INV-2025-045 for $15,000.00 has been processed. An early payment discount of 2% is available if paid by 2025-04-10, reducing the amount due to $14,700.00. A bulk order discount of 5% was applied as the subtotal exceeded $10,000.00. It is recommended to approve early payment to save 2% and ensure timely fund allocation for upcoming project phases."

Key Features and Advantages

- **State Maintenance:** Tracks document context and workflow stage, ensuring consistency across processes.
- **Multi-Step Orchestration:** Handles complex workflows involving multiple components and external tools.
- **Domain-Specific Intelligence:** Applies tailored business rules and guidelines for precise recommendations.
- **Scalability:** Supports large-scale document processing with modular and dynamic agent integration.
- **Enhanced Productivity:** Automates repetitive tasks while augmenting human expertise in decision-making.