perplexity

Add Semantic search abilities for users to be able to query vector databases within the codebase. Users will be able to use voice input through our automatic speech recognition (ASR) using Nvidia Riva for retrieval.

Excellent. This is a logical and powerful extension to the Enterprise Agentic Workflow Engine (EAWE). Integrating semantic search with a voice-first interface transforms the platform from a workflow automation tool into a conversational knowledge and action engine.

Drawing on my experience, here is the detailed system design for adding these capabilities, ensuring it integrates seamlessly with the existing EAWE architecture.

System Design Enhancement: Intelligent Document & Data Retrieval (IDDR) Module

To incorporate these new features, we will introduce a new core component to the EAWE architecture: the **Intelligent Document & Data Retrieval (IDDR) Module**. This module will be responsible for ingesting, indexing, and enabling natural language queries against an enterprise's structured and unstructured data.

Architectural Modifications and New Components

The existing EAWE architecture is well-suited for this extension. The changes will primarily involve adding a new data pipeline, a dedicated voice service, and a new type of agent to the Agent Framework.

1. New Component: Vector Data Pipeline & Store

This is the foundation of our semantic search capability. It's an "offline" process responsible for converting enterprise data into a searchable vector space [1]. The architecture involves three main stages, as is typical for semantic search systems [2].

a. Data Ingestion & Embedding Service:

- **Connectors:** This service will use the existing **Integration & Tooling Layer** to connect to various data sources (e.g., SharePoint, Confluence, Salesforce, network drives).
- **Chunking & Cleaning:** Raw text from documents (PDFs, Word docs, etc.) and database records is extracted, cleaned, and split into smaller, semantically meaningful chunks (e.g., paragraphs) [2]. This is critical for effective retrieval.
- **Embedding Generation:** Each chunk is passed through a sentence-transformer model (e.g., Sentence-BERT or an OpenAI model) to be converted into a high-dimensional vector

embedding. These embeddings capture the semantic meaning of the text [1]. For domain-specific needs, we will support fine-tuning these models on client data to improve accuracy [3].

b. Vector Database Store:

- **Technology:** We will use **Qdrant** as our primary vector database [4] [5]. Its support for rich payload filtering (e.g., searching for vectors that also have a metadata tag `department: "finance"`) and hybrid search capabilities make it ideal for enterprise use cases [4]. It also offers scalable, production-ready deployment options [5].
- **Storage:** The generated vector embeddings are stored in the Qdrant database along with their original text content and relevant metadata (source document, author, creation date, access permissions, etc.).

c. Indexing & Retrieval Service:

- **Indexing:** Qdrant uses advanced indexing algorithms like HNSW (Hierarchical Navigable Small World) to enable extremely fast approximate nearest neighbor (ANN) searches, allowing us to query millions of vectors in milliseconds [2] [6].
- **API Layer:** A dedicated internal API service will provide a clean interface for agents to query the vector database, abstracting the underlying complexity.

2. New Component: Voice Interaction Service

This service acts as the new "ears" of the EAWE, powered by NVIDIA Riva.

- **ASR Endpoint:** A dedicated microservice will expose an endpoint that accepts streaming audio data from the user interface.
- **Riva Integration:** This service will use the **NVIDIA Riva ASR** SDK to transcribe the incoming audio stream into text in real-time [7]. We will leverage Riva's advanced features like automatic punctuation and Inverse Text Normalization (ITN) to produce clean, well-formed text suitable for an LLM [7].
- **Multi-Language & Model Support:** Riva's ability to host multiple models simultaneously allows us to support various languages and dialects, selecting the appropriate model based on user configuration [7].

3. Modification: Agent Framework & Workflow Studio

We will introduce a new agent and tool to leverage the IDDR module.

- **New Agent: The "Retrieval Agent"**
  - This agent is specialized for knowledge retrieval. It receives a text query (either typed or transcribed by the Voice Interaction Service).
  - It uses a "reasoning" LLM to analyze the query, identify key entities, and determine if it needs to perform a simple semantic search, a filtered search, or a hybrid search (combining semantic and keyword methods) [3].
  - It calls the appropriate tool to query the vector database.

    - It receives a set of relevant data chunks (the "context") from the database.
    - Finally, it passes the original query and the retrieved context to an LLM to synthesize a coherent, natural-language answer. This is a classic Retrieval-Augmented Generation (RAG) pattern [5] [6].
- **New Tool in the Integration & Tooling Layer:**
    - `semantic_search(query: str, filters: dict)`: This tool will be available to the Retrieval Agent. It queries the Indexing & Retrieval Service API to find the most relevant vectors and their associated metadata.
- **Workflow Studio Update:**
    - The UI will be updated with a microphone icon and a global search bar, allowing users to initiate a query from anywhere in the application.

Detailed End-to-End Workflow Example

Let's trace a user query through the enhanced system, referencing the sales workflow bottlenecks identified in the analysis document [8].

User Scenario: A sales manager wants to prep for a weekly team meeting.

1. **Voice Input:** The manager clicks the microphone icon and asks:
   > "Show me the latest sales performance reports for the West region and find any proposals generated last week that are still awaiting approval."
2. **Transcription (Voice Interaction Service):** The audio is streamed to the Riva ASR service. Riva transcribes it into clean text: `"Show me the latest sales performance reports for the West region and find any proposals generated last week that are still awaiting approval."` [7] [9].
3. **Routing (Core Orchestration Engine):** The engine identifies this as a query and routes it to the **Retrieval Agent**.
4. **Query Decomposition & Planning (Retrieval Agent):** The agent's reasoning LLM breaks the query into two parts:
    - Part A: A semantic search for "latest sales performance reports" with a metadata filter `region: "West"`.
    - Part B: A semantic search for "proposals" with a date filter `creation_date: "last week"` and a status filter `status: "awaiting_approval"`.
5. **Execution (Retrieval Agent & IDDR Module):**
    - The agent calls the `semantic_search` tool twice with the decomposed queries.
    - The **Indexing & Retrieval Service** queries the **Qdrant** database, returning the most relevant documents for both parts of the query.
6. **Synthesis (Retrieval Agent & LLM):** The agent compiles the retrieved documents (e.g., a link to a PowerBI dashboard and a list of three proposals) into a context prompt. It sends this context along with the original query to a powerful LLM (like GPT-4).
7. **Response Generation & Display:** The LLM generates a clear, actionable summary:

> "Here is the link to the latest West Region Sales Performance Dashboard.
> I also found three proposals from last week that are still awaiting approval:
>
> - Proposal #781 for Acme Corp (awaiting legal review)
> - Proposal #783 for Stark Industries (awaiting finance sign-off)
> - Proposal #785 for Wayne Enterprises (awaiting client signature)"

This response is then displayed in the user interface.

Updated Technology Stack & Implementation Notes

- **Voice Recognition: NVIDIA Riva** for high-performance, real-time ASR [7]. Integration will be streamlined using the `NVIDIARivaASR` runnable within our existing **LangChain**-based agent framework [9].
- **Vector Database: Qdrant** (Open Source) for its powerful filtering and production readiness [4] [5]. **Milvus** is a strong alternative, especially for workflows requiring extreme performance, due to its integration with GPU-accelerated search via NVIDIA's cuVS [6].
- **Performance Optimization:** For enterprise-scale clients with massive datasets, we will implement GPU-accelerated indexing and search using **NVIDIA RAPIDS cuVS**. This can drastically reduce both index build times and query latency, which is a critical competitive advantage [6].

⁂

1. https://www.kdnuggets.com/semantic-search-with-vector-databases
2. https://milvus.io/ai-quick-reference/whats-the-typical-architecture-for-a-semantic-search-system
3. https://www.instaclustr.com/education/vector-database/vector-search-vs-semantic-search-4-key-differences-and-how-to-choose/
4. https://www.instaclustr.com/education/vector-database/top-10-open-source-vector-databases/
5. https://www.techtarget.com/searchdatamanagement/tip/Top-vector-database-options-for-similarity-searches
6. https://developer.nvidia.com/blog/accelerating-vector-search-using-gpu-powered-indexes-with-rapids-raft/
7. https://docs.nvidia.com/deeplearning/riva/user-guide/docs/asr/asr-overview.html
8. Organizational-Workflow-Automation-Analysis_-Ident.pdf
9. https://python.langchain.com/docs/integrations/tools/nvidia_riva/