Write the functional and non functional requirements

Based on the comprehensive system design, business logic, and the initial problem analysis, this document outlines the functional and non-functional requirements for the Enterprise Agentic Workflow Engine (EAWE).

Functional Requirements

Functional requirements define what the system must *do*. They describe the specific behaviors, features, and functions of the platform.

FR1: User and Tenant Management

- **FR1.1 Multi-Tenancy:** The system shall support a multi-tenant architecture, ensuring complete logical and data isolation between different enterprise clients [1].
- **FR1.2 Role-Based Access Control (RBAC):** The system shall implement a role-based access control model with at least three distinct roles [2]:
  - **Administrator:** Manages users, billing, system-wide integrations, and security settings for their tenant.
  - **Designer:** Creates, edits, tests, and deploys workflow definitions.
  - **Operator:** Executes workflows, monitors runs, and handles tasks in the Human-in-the-Loop (HITL) queue.
- **FR1.3 User Authentication:** The system shall provide secure user authentication via email and password.

FR2: Workflow Studio (Design & Creation)

- **FR2.1 Visual Workflow Designer:** The system shall provide a visual, no-code interface with drag-and-drop functionality for users to design and map business processes [3] [4].
- **FR2.2 Modular Components:** The designer shall provide a toolbox of pre-built, reusable components, including agents, logic operators (e.g., conditions), and triggers [5] [1].
- **FR2.3 Workflow Templates:** The system shall offer a library of pre-built workflow templates for common enterprise tasks (e.g., "AP Invoice Processing," "Employee Onboarding") to accelerate adoption [1] [6].
- **FR2.4 Workflow Configuration:** Users shall be able to configure the specific properties of each node in a workflow, such as defining an agent's goal or setting parameters for a tool.
- **FR2.5 Workflow Versioning:** The system shall allow workflows to be saved in a `DRAFT` state for design and testing before being activated to a production `ACTIVE` state.

FR3: Workflow Engine (Execution)

- **FR3.1 Workflow Triggers:** The system shall support initiating workflows based on three types of triggers [1]:
  - **Event-Driven:** Triggered by an event from an integrated third-party system (e.g., a new CRM lead).
  - **Scheduled:** Triggered at a predefined time or interval.
  - **Manual:** Triggered on-demand by a user.
- **FR3.2 State Management:** The engine shall maintain the state of every running workflow instance, ensuring that processes can be paused, resumed, and are resilient to system restarts [7].
- **FR3.3 Conditional Logic:** The engine shall support conditional branching (if/then/else logic) within workflows based on the data payload.
- **FR3.4 Error Handling:** The system shall have robust error-handling mechanisms. If an agent fails a task, it shall retry a configurable number of times before escalating the issue to the HITL queue.

FR4: Conversational AI & Retrieval (IDDR Module)

- **FR4.1 Voice Input:** The system shall accept user queries via real-time voice input using NVIDIA Riva for Automatic Speech Recognition (ASR).
- **FR4.2 Semantic Search:** The system shall allow users to perform natural language queries against an indexed knowledge base of enterprise documents and data [7].
- **FR4.3 Retrieval-Augmented Generation (RAG):** Upon receiving a query, the system shall retrieve relevant context from its vector database and use an LLM to synthesize a coherent, accurate answer, citing the sources used.
- **FR4.4 Permission-Aware Retrieval:** The system shall enforce data permissions at the point of retrieval, ensuring a user can only receive information they are authorized to access.
- **FR4.5 Document Ingestion:** Administrators shall be able to connect to data sources (e.g., SharePoint, network drives) to initiate the ingestion, chunking, and vectorization of documents.

FR5: Human-in-the-Loop (HITL)

- **FR5.1 Approval Task Generation:** The system shall automatically generate a task in a human's approval inbox when a workflow reaches a designated approval checkpoint, encounters an unrecoverable error, or an agent's confidence score is below a set threshold [5].
- **FR5.2 Task Resolution:** An approver shall be able to perform one of three actions on a task:
  - **Approve:** Allow the workflow to proceed.
  - **Reject:** Terminate the workflow, providing a mandatory reason.
  - **Edit and Approve:** Correct data processed by an agent and then approve the workflow.

- **FR5.3 Contextual Information:** The HITL interface shall provide the approver with all necessary context, including the agent's actions, links to source documents, and the specific data requiring review.
- **FR5.4 Feedback Loop:** When a user performs an "Edit and Approve" action, the system shall log both the original agent output and the human-corrected version to create a dataset for future model fine-tuning and optimization [5].

FR6: Auditing and Administration

- **FR6.1 Immutable Audit Trail:** The system shall maintain an immutable, time-stamped audit trail for every workflow run, logging every action taken by agents and users [1].
- **FR6.2 Secure Credential Management:** The system shall provide a secure vault for administrators to store and manage encrypted credentials for third-party system integrations.
- **FR6.3 Performance Dashboards:** The system shall provide real-time dashboards for monitoring key metrics such as workflow completion rates, task processing times, and error rates [3].

Non-Functional Requirements

Non-functional requirements define *how* the system should operate. They are quality attributes that ensure the system is performant, secure, reliable, and usable [8] [9].

NFR1: Performance & Scalability

- **NFR1.1 API Response Time:** Core REST API endpoints shall have a median response time of less than 500ms under expected load.
- **NFR1.2 UI Responsiveness:** User interface screens and interactive components shall fully render in under 2 seconds.
- **NFR1.3 Throughput:** The system shall be able to process a minimum of [e.g., 1000] concurrent workflow instances without performance degradation.
- **NFR1.4 Scalability:** The system architecture shall support horizontal scaling of its microservices to handle increased load as the business grows [2] [10].
- **NFR1.5 Ingestion Speed:** The document ingestion pipeline shall be capable of processing at least [e.g., 1 GB] of text-based documents per hour.

NFR2: Security

- **NFR2.1 Data Encryption:** All sensitive customer data shall be encrypted both at rest (using AES-256 or equivalent) and in transit (using TLS 1.2 or higher) [8] [11].
- **NFR2.2 Vulnerability Management:** The system shall be regularly scanned for common security vulnerabilities (e.g., OWASP Top 10), and any critical vulnerabilities must be patched within 30 days of discovery.

- **NFR2.3 Authentication Security:** User passwords shall be salted and hashed. The system shall enforce strong password policies and support session timeouts.
- **NFR2.4 Data Privacy:** The system must comply with relevant data protection regulations (e.g., GDPR, CCPA) based on the client's jurisdiction.

NFR3: Reliability & Availability

- **NFR3.1 Availability:** The system shall have a minimum uptime of 99.9% ("three nines"), excluding scheduled maintenance windows [11] [10].
- **NFR3.2 Durability:** The state of active workflows must be durable. A service restart or failover must not result in the loss of an in-progress workflow [12].
- **NFR3.3 Backup and Recovery:** The system shall have automated daily backups of the primary database. A documented disaster recovery plan must be in place to restore service within 4 hours of a catastrophic failure.

NFR4: Usability

- **NFR4.1 Intuitiveness:** The no-code workflow designer shall be usable by non-technical business analysts with a maximum of 4 hours of training [3] [9].
- **NFR4.2 Accessibility:** The web application user interface shall comply with Web Content Accessibility Guidelines (WCAG) 2.1 Level AA standards.
- **NFR4.3 Consistency:** The user interface shall maintain a consistent design language, layout, and interaction patterns across all modules.

NFR5: Maintainability & Supportability

- **NFR5.1 Modularity:** The system shall be developed using a modular, microservices-based architecture to allow for independent updates and maintenance of components [5] [8].
- **NFR5.2 Logging:** All services shall produce structured, machine-readable logs to facilitate debugging and monitoring.
- **NFR5.3 Monitoring:** The system shall expose key health and performance metrics (e.g., CPU usage, memory, error rates) for each service to a centralized monitoring solution.

NFR6: Integration & Compatibility

- **NFR6.1 API Strategy:** The system shall provide a well-documented, versioned REST API to allow for external integrations [2].
- **NFR6.2 Browser Compatibility:** The web application shall be fully functional on the latest two versions of major web browsers (Google Chrome, Mozilla Firefox, Microsoft Edge, and Safari) [10].
- **NFR6.3 Model Agnosticism:** The architecture should allow for swapping underlying LLMs or embedding models with minimal changes to the core agent logic, providing flexibility and future-proofing [12].

⁂

1. https://stormid.com/agentic-workflow-engine/
2. https://apix-drive.com/en/blog/other/business-automation-workflow-system-requirements
3. https://www.flowforma.com/blog/enterprise-workflow-automation
4. https://www.cmwlab.com/blog/complete-basics-workflow-automation/
5. Organizational-Workflow-Automation-Analysis_-Ident.pdf
6. https://invgate.com/itsm/enterprise-service-management/enterprise-workflow-automation
7. https://orkes.io/blog/what-are-agentic-workflows/
8. https://www.perforce.com/blog/alm/what-are-non-functional-requirements-examples
9. https://www.altexsoft.com/blog/functional-and-non-functional-requirements-specification-and-types/
10. https://www.altexsoft.com/blog/non-functional-requirements/
11. https://www.businessanalyststoolkit.com/non-functional-requirements-checklist/
12. https://www.diagrid.io/blog/the-agentic-spectrum-why-its-not-agents-vs-workflows