AI/ML Technology Stack

Overview

The AI Persona Platform leverages a sophisticated AI/ML stack combining multiple LLM providers, advanced RAG frameworks, and multi-agent orchestration. This approach ensures optimal performance, cost efficiency, and regulatory compliance for pharmaceutical applications.

Core Technologies

LLM Providers

OpenAI Models

OpenAI offers a range of models from lightweight to advanced reasoning capabilities. **Critical requirement**: Use models with function calling capabilities for agentic tasks.

Model Tiers: - **GPT-4o**: Base multimodal model with solid performance across tasks - **GPT-4.1-mini**: Enhanced version with better reasoning and efficiency - **GPT-4.1-nano**: Ultra-lightweight and fast model for basic tasks - **GPT-o3/o4**: Advanced reasoning models for complex problem-solving - **GPT-4o-mini**: Cost-effective option for routine operations

Key Features: - **Function Calling**: Essential for tool integration and structured data extraction - **Multimodal Processing**: Text, images, and audio understanding - **128K Context Window**: Process entire research papers - **Streaming Responses**: Real-time response generation - **Enterprise SLA**: 99.9% uptime guarantees

Anthropic Claude 4 Family

Anthropic's Claude 4 models excel in reasoning, analysis, and safety-critical applications ideal for pharmaceutical use.

Model Options: - **Claude 4 Sonnet**: Balanced performance for general pharmaceutical tasks - **Claude 4 Opus**: Maximum capability for complex analysis and reasoning - **Claude 4 Haiku**: Fast responses for routine queries and support tasks

Key Features: - **Constitutional AI**: Built-in safety and alignment for regulated industries - **Long Context**: Extended context windows for comprehensive document analysis - **Reasoning Excellence**: Superior performance on complex analytical tasks - **Function Calling**: Full tool integration capabilities - **Safety Focus**: Designed for responsible AI deployment in critical domains

Ollama

Local LLM deployment solution for sensitive pharmaceutical data that cannot leave corporate infrastructure.

Benefits: - **Data Sovereignty**: Keep confidential data on-premise - **Zero API Costs**: No per-token charges for internal usage - **Air-Gapped Deployment**: Works without internet connectivity - **Model Flexibility**: Supports Llama, Code Llama, Mistral, and custom models - **OpenAI-Compatible API**: Easy integration with existing code

Embedding Providers

FastEmbed (Recommended Default)

Lightweight library for local embedding generation without API calls.

Benefits: - **Local Processing**: No API calls or internet dependency - **Cost Effective**: Zero per-request charges - **Fast Inference**: Optimized for production speed - **Multiple Models**: Support for various embedding architectures - **Minimal Dependencies**: Lightweight deployment

OpenAI Embeddings

Cloud-based embeddings via text-embedding-3-large model for advanced semantic understanding.

Apollo Embeddings

Enterprise gateway access to OpenAI embeddings with corporate compliance and monitoring.

miniCOIL (Optional Recommendation)

Advanced sparse-dense hybrid embeddings optimized for long pharmaceutical documents via Qdrant.

Benefits: - **Hybrid Search**: Combines keyword precision with semantic understanding - **Long Document Optimization**: Superior handling of clinical protocols and regulatory documents
- **Pharmaceutical Terminology**: Excellent for drug names, clinical terms, and complex medical literature - **Computational Efficiency**: Smaller models with faster inference than traditional COIL - **Seamless Integration**: Works within existing Qdrant/Apollo infrastructure - **Cost Effective**: Reduced computational overhead compared to large embedding models

Use Cases: Consider for applications requiring precise retrieval from long pharmaceutical documents, regulatory submissions, or clinical trial protocols where both exact terminology matching and semantic understanding are critical.

SPLADE (Optional Recommendation)

Sparse lexical and expansion model that can be pretrained on domain-specific pharmaceutical data for enhanced retrieval.

Benefits: - **Domain-Specific Training**: Can be fine-tuned on pharmaceutical literature and internal documents - **Sparse Representations**: Interpretable keyword-based embeddings with learned expansions - **Terminology Expansion**: Automatically expands queries with relevant pharmaceutical synonyms and abbreviations - **Exact Match Preservation**: Maintains precise matching for drug names, dosages, and clinical terms - **Explainable Results**: Clear understanding of why documents were retrieved - **Custom Vocabulary**: Incorporates proprietary terminology and internal naming conventions

Use Cases: Ideal for organizations with large pharmaceutical datasets wanting custom-trained models that understand company-specific terminology, drug portfolios, and internal documentation patterns. Particularly valuable for regulatory affairs and medical information teams requiring explainable retrieval results.

Agentic Frameworks

Note: Agentic frameworks are optional. For simple use cases, lightweight custom implementations may be more suitable than full frameworks.

AutoGen

Microsoft's multi-agent conversation framework for complex collaborative workflows.

Strengths: - **Microsoft Backing**: Enterprise support and infrastructure alignment - **Human-in-the-Loop**: Native oversight capabilities for pharmaceutical compliance - **Code Execution**: Agents can write and run analysis code - **Group Chat**: Natural conversation flows between expert personas - **Healthcare Proven**: Used by major health systems and pharmaceutical companies

CrewAI

Modern framework for building AI agent teams with role-based collaboration.

Strengths: - **Role Definition**: Clear agent roles with specific responsibilities - **Task Orchestration**: Structured workflow management - **Memory Management**: Persistent agent memory across interactions - **Integration Ready**: Easy integration with existing tools and APIs

Semantic Kernel

Microsoft's SDK for integrating LLMs with conventional programming languages.

Strengths: - **Enterprise Integration**: Native .NET and Python support - **Plugin Architecture**: Modular skill development - **Memory Stores**: Built-in vector and semantic memory - **Planning**: Automatic task decomposition and execution

Agno

Enterprise-focused platform for building production AI agents.

Strengths: - **Enterprise Features**: Built-in compliance and monitoring - **Visual Development**: Low-code agent creation interface - **Production Ready**: Scalable deployment and management - **Integration Hub**: Pre-built connectors for enterprise systems

LangChain

Popular framework for building LLM applications with extensive ecosystem of integrations.

Strengths: - **Extensive Ecosystem**: Large community and integration library - **Chain Abstraction**: Composable building blocks for LLM workflows - **Memory Management**: Built-in conversation and document memory - **Tool Integration**: Rich set of pre-built tools and connectors - **Agent Templates**: Ready-made agent patterns and examples

LlamaIndex

Specialized framework for building RAG applications with advanced indexing capabilities.

Strengths: - **RAG Focus**: Purpose-built for retrieval-augmented generation - **Advanced Indexing**: Sophisticated document chunking and indexing strategies - **Query Engine**: Optimized retrieval and synthesis pipelines - **Agent Support**: Built-in agent capabilities with tool integration - **Multi-modal**: Support for text, images, and structured data

Haystack Agents

Haystack's agent system built on their component-pipeline architecture.

Strengths: - **Component-Based**: Modular agent design with reusable components - **Production Ready**: Enterprise-grade stability and performance - **Tool Integration**: Native support for custom tools and APIs - **Pipeline Flexibility**: Agent workflows as configurable pipelines - **Conversational Memory**: Built-in memory management for multi-turn conversations

Vector Stores

Qdrant (via Apollo)

High-performance vector database optimized for pharmaceutical applications through Apollo's managed service.

Benefits: - **Managed Infrastructure**: No deployment or maintenance overhead - **Multi-tenancy**: Automatic application isolation - **High Performance**: Optimized HNSW algorithm for fast similarity search - **Hybrid Search**: Vector similarity with metadata

filtering - **Enterprise Scale**: Supports millions of vectors - **Apollo Integration**: Unified authentication with LLM services

Document Processing

MarkitDown

Microsoft's document conversion tool for transforming various formats into AI-ready markdown.

Multi-Provider LLM Strategy

OpenAI GPT-4o Family as Primary

- **Quality Leadership**: Consistently highest scores on medical and scientific benchmarks
- **Cost Efficiency**: GPT-4o-mini provides 90% of quality at 15% of the cost for routine queries
- **Function Calling**: Native support for tool integration and structured data extraction
- **Streaming Support**: Real-time response generation for better user experience
- **Enterprise SLA**: 99.9% uptime guarantees with dedicated support

Ollama for Sensitive Data

- **Data Sovereignty**: Keeps confidential pharmaceutical data on-premise
- **Zero Latency**: No internet dependency for critical operations
- **Cost Predictability**: No per-token costs for high-volume internal usage
- **Air-Gapped Deployment**: Can operate in completely isolated network environments

Boehringer Ingelheim Apollo Services Integration

Apollo v2 serves as Boehringer Ingelheim's comprehensive AI platform, providing three integrated components under unified authentication:

1. LLM API (Powered by LiteLLM)

- **Enterprise Gateway**: Centralized access point for approved AI models (GPT-4o, embeddings) with corporate authentication
- **OAuth2 Security**: Client credentials flow with automatic token refresh for enterprise-grade security
- **Model Flexibility**: Access to OpenAI's latest models (GPT-4o, text-embedding-3-large) through corporate infrastructure
- **Cost Management**: Per-application usage tracking and budget controls via `/apollo/llm-api/customer/info`

2. Vector Store (Powered by Qdrant)

- **Managed Infrastructure**: Multi-node Qdrant cluster on OpenShift, no infrastructure to maintain
- **Multi-tenancy**: Automatic collection isolation with application ID prefixing
- **Enterprise Scale**: Supports millions of vectors with distributed architecture
- **Unified Authentication**: Same OAuth2 tokens as LLM API, no separate credentials needed
- **Optimized Settings**: Centrally managed shard/replication configuration for performance

3. Data Curation (Powered by Argilla)

- **Collaborative Labeling**: Multi-user annotation workflows for training data
- **Automatic Workspace**: Default workspace created on first API call
- **Integrated Authentication**: Same OAuth2 flow as other Apollo components

Key Benefits: - **Single Authentication**: One set of credentials for LLM, vector store, and data curation - **Compliance Built-in**: Pre-configured for pharmaceutical regulatory requirements - **Hybrid Deployment**: Mix Apollo services with local models based on

data sensitivity - **Python SDK**: Official SDK handles OAuth2 complexity and certificate issues

Multi-Provider Architecture Implementation

Universal LLM Client Architecture

The platform implements a **unified adapter pattern** that enables seamless switching between LLM providers through configuration rather than code changes:

- **Provider Abstraction**: Single interface adapts to OpenAI, Groq, Ollama, and Apollo APIs
- **Dynamic Authentication**: Handles different auth methods (Bearer tokens for OpenAI/Groq, OAuth2 for Apollo, basic auth for Ollama)
- **Resilience Built-in**: Automatic retry with exponential backoff for transient failures
- **Configuration-Driven**: Provider selection via environment variables enables runtime flexibility
- **Consistent Interface**: Unified error handling and response formatting across all providers

Multi-Provider Embedding Solution

The embedding service implements a **hybrid provider pattern** that optimizes for both cost and performance:

- **Default Local Processing**: FastEmbed runs locally for zero-cost embeddings without API calls
- **Remote Provider Support**: Falls back to OpenAI, Ollama, or Apollo when advanced embeddings are needed
- **Intelligent Fallback**: Automatically switches to local embeddings if remote providers fail
- **Provider-Specific Caching**: Disk-based cache with provider-aware keys prevents mixing embeddings
- **Batch Optimization**: Efficient batch processing for both local and remote providers

Architectural Benefits

Provider Independence

- **No Vendor Lock-in**: Switch between providers without code changes
- **Cost Optimization**: Route queries to the most cost-effective provider based on complexity
- **Compliance Flexibility**: Use local models for sensitive data, cloud for general queries
- **Performance Tuning**: Select providers based on latency requirements

Enterprise Readiness

- **Apollo Gateway Integration**: Native support for enterprise API gateways with OAuth2
- **Token Management**: Sophisticated token lifecycle handling with automatic refresh
- **Observability**: Comprehensive logging across all providers for debugging and monitoring
- **Type Safety**: Pydantic models ensure data consistency across provider boundaries

Operational Excellence

- **Zero Downtime Switching**: Change providers via configuration without redeployment
- **Gradual Rollout**: Test new providers with subset of queries before full migration
- **Cost Attribution**: Track usage per provider for accurate cost allocation
- **Performance Monitoring**: Provider-specific metrics for optimization

Framework Comparisons

Why AutoGen over Haystack Agents/CrewAI for Multi-Agent

AutoGen Advantages

- **Microsoft Backing**: Enterprise support and roadmap alignment with Boehringer's existing Microsoft infrastructure
- **Human-in-the-Loop**: Native support for human oversight and intervention (critical for pharma compliance)
- **Code Generation**: Superior code execution capabilities for data analysis personas
- **Group Chat**: Natural conversation flows between multiple expert personas
- **Proven in Healthcare**: Used by major health systems and pharmaceutical companies

Why Haystack for RAG Implementation

- **RAG-Specific Design**: Purpose-built for retrieval-augmented generation with superior performance
- **Component Architecture**: Intuitive, modular design that's easier to maintain and debug
- **Production Stability**: More stable and reliable for enterprise pharmaceutical applications
- **Visual Pipeline Builder**: deepset Studio provides drag-and-drop pipeline creation
- **Better Documentation**: Clearer, more comprehensive documentation than alternatives
- **Qdrant Integration**: Native QdrantEmbeddingRetriever and QdrantDocumentStore components
- **Apollo Compatibility**: Works seamlessly with Apollo's managed Qdrant instance
- **Citation Tracking**: Built-in source attribution and provenance tracking for regulatory compliance

Why MarkitDown for Document Processing

MarkitDown is Microsoft's document processing tool that converts various file formats (PDF, Word, PowerPoint, etc.) into clean, structured markdown. For enterprise deployment, it can be implemented as a dedicated microservice.

Core Benefits: - **Microsoft Integration**: Native support for Office formats (critical for pharma workflows) - **Structured Extraction**: Preserves tables, images, and formatting context - **Clinical Trial Documents**: Excellent handling of complex PDF protocols and reports - **Regulatory Submissions**: Processes CTD and eCTD documents accurately

Microservice Architecture: - **RESTful API**: FastAPI-based service with async processing - **Batch Processing**: Handle multiple documents simultaneously with error isolation - **Format Validation**: Pre-processing validation for supported file types - **Optional AI Enhancement**: OpenAI integration for advanced image OCR - **Containerized Deployment**: Docker-ready with multi-stage builds - **Comprehensive Testing**: Hurl-based API tests for format compatibility

Supported Formats: - **Documents**: PDF, DOCX, PPTX, XLSX, HTML, XML, CSV, JSON, TXT, MD, RTF, EPUB - **Media**: PNG, JPG, GIF, BMP, TIFF, MP3, WAV, M4A, OGG - **Archives**: ZIP files with automatic extraction

Enterprise Features: - **Async Processing**: Non-blocking document conversion for high throughput - **Error Handling**: Graceful failure handling with detailed error responses - **Security**: CORS configuration, input validation, non-root container execution - **Monitoring**: Health checks and conversion time tracking - **Scalability**: Stateless design suitable for horizontal scaling

Technical Dependencies: - **Google Magika**: Advanced file type detection using deep learning models - **Heavy Dependencies**: Large ML models require containerization over serverless functions - **Container-Only Deployment**: Not suitable for AWS Lambda due to dependency size constraints - **Infrastructure Requirements**: Requires dedicated container infrastructure for reliable operation

RAG Architecture with Qdrant

Vector Store Implementation via Apollo

The platform leverages Apollo's managed Qdrant instance for all vector storage needs:

- **Unified Access**: Single OAuth2 authentication for both LLM and vector operations
- **Collection Management**: Automatic application ID prefixing for multi-tenancy

- **Optimized Configuration**: Centrally managed HNSW parameters for best performance
- **Scalability**: Supports growth from thousands to millions of vectors
- **No Infrastructure**: Fully managed service, no Qdrant deployment needed

Qdrant Collection Strategy

- **Single Collection Pattern**: One collection per application for efficiency
- **Metadata Filtering**: Use Qdrant's payload filtering for access control
- **Hybrid Search**: Combine vector similarity with metadata conditions
- **Batch Operations**: Efficient bulk ingestion for large document sets
- **In-Memory Testing**: Use Qdrant's in-memory mode for development

Medical Literature RAG

- **Quality Filtering**: Qdrant payload filters for journal impact factor, publication date
- **Source Validation**: Store credibility scores as vector metadata
- **Citation Management**: Full reference data in Qdrant payloads
- **Confidence Scoring**: Distance metrics for retrieval confidence

Internal Knowledge RAG

- **Access Control**: Qdrant payload conditions for user-based filtering
- **Version Tracking**: Document versions as separate vectors with metadata
- **Department Restrictions**: Department IDs in payloads for filtering
- **Metadata Enhancement**: Rich payloads with tags, categories, timestamps

Hybrid RAG Strategy with Haystack + Qdrant

- **Semantic Search**: Qdrant's HNSW algorithm for vector similarity
- **Metadata Filtering**: Qdrant's filtering DSL for precise retrieval
- **Haystack Pipeline**: QdrantEmbeddingRetriever for seamless integration

- **Reranking**: Haystack's ranker components with Qdrant results

Performance Characteristics

LLM Performance Metrics

- **GPT-4o**: 200+ tokens/second streaming, 128K context window
- **GPT-4o-mini**: 500+ tokens/second streaming, 128K context window
- **Ollama**: 50-100 tokens/second (hardware dependent), 8K-32K context
- **Response Time**: <2s for simple queries, <10s for complex RAG queries

RAG Performance

- **Document Retrieval**: <500ms for semantic search
- **Embedding Generation**: <100ms for 1K token documents
- **Citation Accuracy**: >95% source attribution accuracy
- **Relevance Score**: >85% user satisfaction with retrieved content

Multi-Agent Performance

- **Agent Coordination**: <1s for simple agent interactions
- **Complex Workflows**: <30s for multi-step agent collaborations
- **Human Escalation**: <5s to route to human expert
- **Quality Validation**: >90% accuracy with guardrail agents

Cost Optimization

Token Usage Strategy

- **Smart Routing**: Route simple queries to cheaper models
- **Context Optimization**: Truncate irrelevant context

- **Caching**: Cache common responses for 24-48 hours
- **Batch Processing**: Group similar requests for efficiency

Infrastructure Costs

- **Local Models**: 60-80% cost reduction for high-volume queries
- **Hybrid Approach**: Balance quality and cost based on query type
- **Auto-scaling**: Scale resources based on demand patterns
- **Cost Monitoring**: Real-time tracking and alerting for budget control