

HEPilot: Integrating Embeddings, Vector Search, and RAG for Physics Use Case

Mike Sokoloff¹

Conor Henderson¹

Mohamed Elashri¹

¹University of Cincinnati
LHCb Collaboration

July 1, 2025



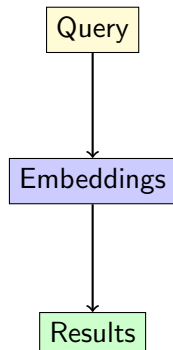
What We Achieved Already: LHCbFinder

Proof of Concept Success

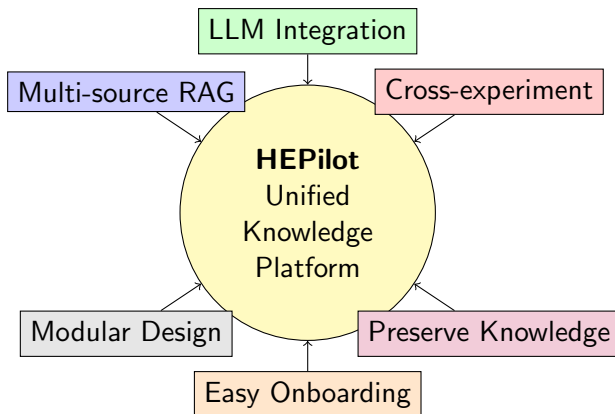
- Semantic search for 4k+ LHCb papers
- Natural language queries
- Sub-100ms response time
- Live demo at lhcbfinder.net.

Key Validation

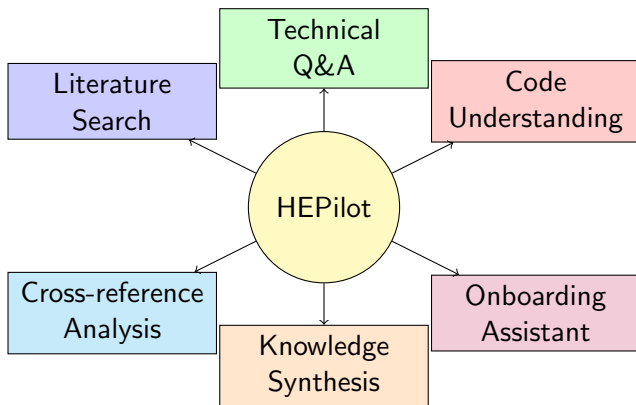
- Vector embeddings work for HEP content
- Users prefer semantic over keyword search
- Foundation ready for expansion



The Vision: HEPilot Framework



Target Use Cases



The Knowledge Discovery Challenge

Current State:

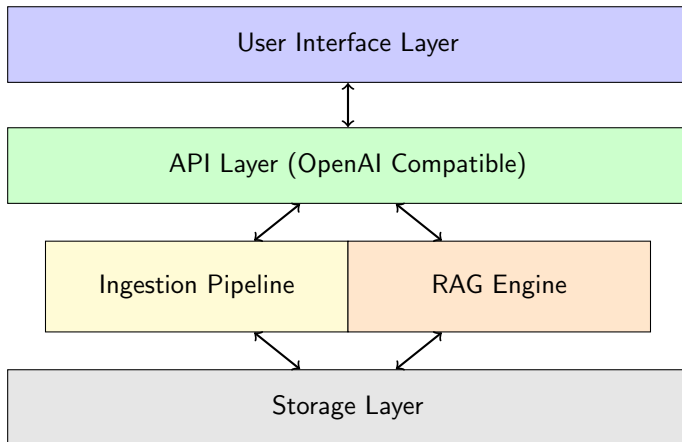
- Information scattered across platforms
- No unified search interface
- Institutional knowledge gets lost
- High barrier for newcomers

HEPilot Solution:

- Single entry point for all knowledge
- Natural language understanding
- Context-aware responses
- Source attribution & traceability

From **"Where do I find...?"** to **"Tell me about..."**

HEPilot Architecture Overview



Key Design Principle: Every component is pluggable via adapters

Technology Stack

Backend:

- FastAPI (async Python)
- PostgreSQL + pgvector
- ChromaDB / Qdrant
- ONNX Runtime

AI/ML:

- BGE embeddings
- OpenAI API
- Ollama (local LLMs)
- LangChain components

Frontend:

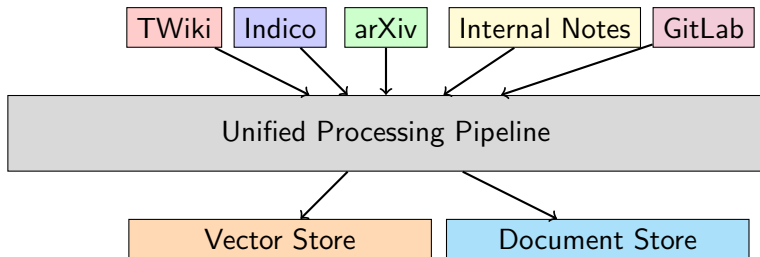
- Chatbot-UI (React)
- Server-sent events
- Real-time streaming

Infrastructure:

- Docker containers
- CI/CD pipelines
- Prometheus metrics
- OpenTelemetry tracing

Philosophy: Use proven open-source tools, avoid reinventing

Multi-Source Ingestion



Smart Processing: LaTeX handling, metadata preservation, deduplication

Retrieval-Augmented Generation

Knowledge Base Platform

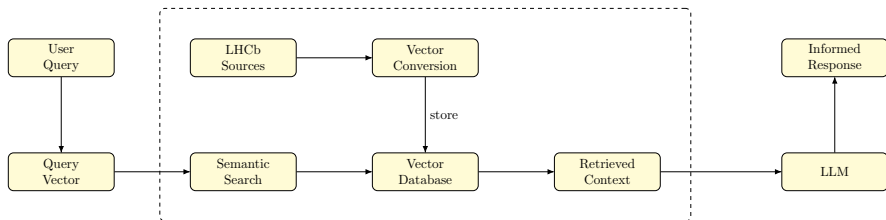


Figure: Retrieval-Augmented Generation architecture

Key Features: Real-time streaming, source attribution, context management

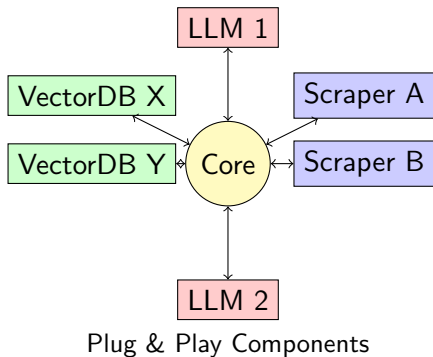
Why Modular Architecture?

Flexibility:

- Swap vector databases
- Change embedding models
- Add new data sources
- Switch LLM providers

Maintainability:

- Test components in isolation
- Update without breaking
- Clear interfaces



Key Framework Features

Search & Retrieval:

- Semantic understanding
- Multi-modal search
- Relevance ranking
- Query expansion

Knowledge Processing:

- Smart chunking
- Metadata preservation
- Citation tracking
- Version control

User Experience:

- Natural language interface
- Real-time responses
- Source transparency
- Feedback integration

Integration:

- OpenAI-compatible API
- REST endpoints
- Streaming support
- Plugin system

Current Development Status

Completed:

- Core architecture design
- Semantic search validation
- Basic RAG implementation
- API framework

40% Complete



PoC Alpha Beta

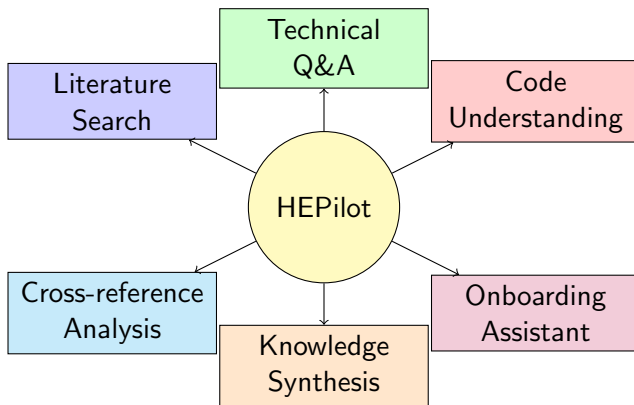
In Progress:

- Multi-source scrapers
- LLM integration fine-tuning
- Performance optimization
- Documentation

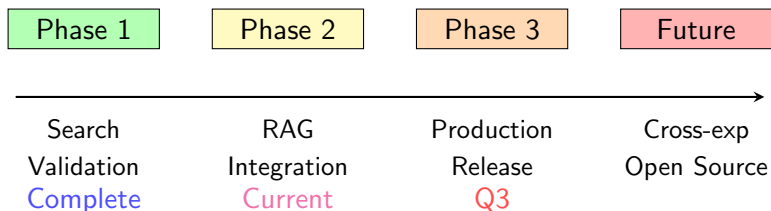
Timeline:

- Q1: Core features
- Q2: Full integration
- Q3: Production ready

Target Use Cases



Development Roadmap



Next Steps:

- Complete adapter implementations for all data sources
- Optimize retrieval quality and response generation
- Build comprehensive test suite and documentation
- Gather user feedback and iterate

Summary

What We've Built:

- Modular, extensible framework with semantic search
- OpenAI-compatible API with full traceability

4k+ papers
indexed

< 100ms
latency

Where We're Going:

- Complete RAG and multi-source integration
- Production deployment and cross-experiment scaling

Live demo
available

lhcbfinder.net

Goal: Make HEP knowledge easily accessible.

Call for Collaboration: Join us to shape the future of physics intelligence.

Backup Slides

For Researchers:

- Faster information discovery
- Better context understanding
- Reduced search time
- Cross-domain insights

For Collaboration:

- Preserved knowledge
- Unified access point
- Better documentation
- Easier collaboration

For Newcomers:

- Lower entry barrier
- Guided exploration
- Context-aware help
- Faster ramp-up

Goal: Transform how we interact with HEP knowledge