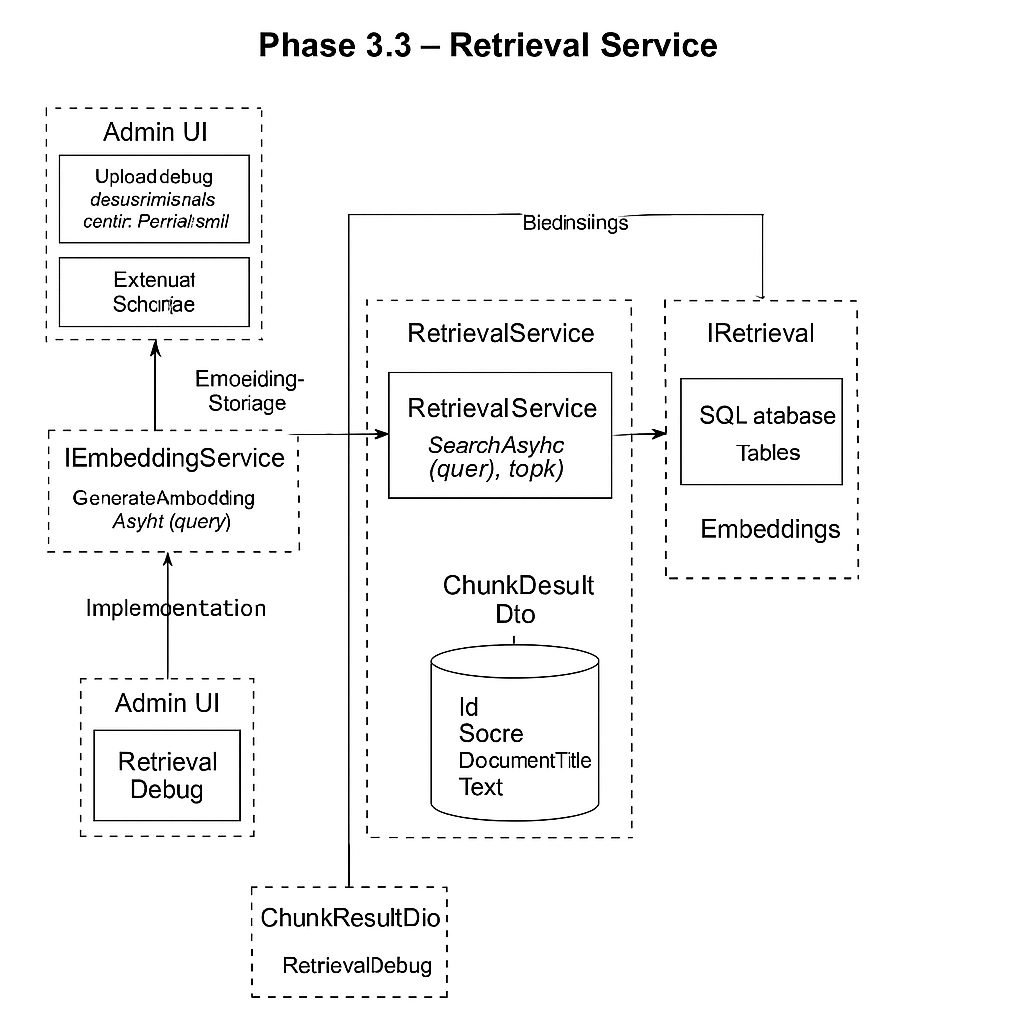
# Phase 3.4 – RAG Pipeline Integration

✅ Status: Completed  
  
This phase integrated the Retrieval Service (Phase 3.3) with LLM augmentation to build a working Retrieval-Augmented Generation (RAG) pipeline. It also introduced an Admin Debug UI (RAG Comparison Page) and history logging into SQL Server.

## Architecture Overview

Flow:  
1. User query is sent to the RagService.  
2. RetrievalService fetches top-K chunks (Semantic / Keyword / Hybrid).  
3. Retrieved chunks are formatted with metadata (doc title, ID, retrieval type).  
4. Two LLM calls are made: Baseline LLM (query only) and RAG LLM (query + retrieved context).  
5. Results are returned with answers, retrieved chunks, latency metrics, and SLA status.  
6. Results are logged into SQL Server (RagComparisonHistories).



## Backend Implementation

- IRagService: defines RunRagAsync(query, topK, useHybrid)  
- RagService: integrates Retrieval + LLM calls  
- DTOs: RagResultDto, RagChunkDto  
- History logging: RagComparisonHistory entity

## Admin Debug UI

Includes Controller, View (Index.cshtml), and JavaScript (rag-comparison.js) with processing spinner, side-by-side answers, context panel, and latency metrics.

## Database Schema (SQL Server)

Table: RagComparisonHistories  
- Id, UserQuery, BaselineAnswer, RagAnswer, RetrievedChunksJson, RetrievalLatencyMs, LlmLatencyMs, TotalLatencyMs, IsWithinSla, CreatedAt

## Features Implemented in Phase 3.4

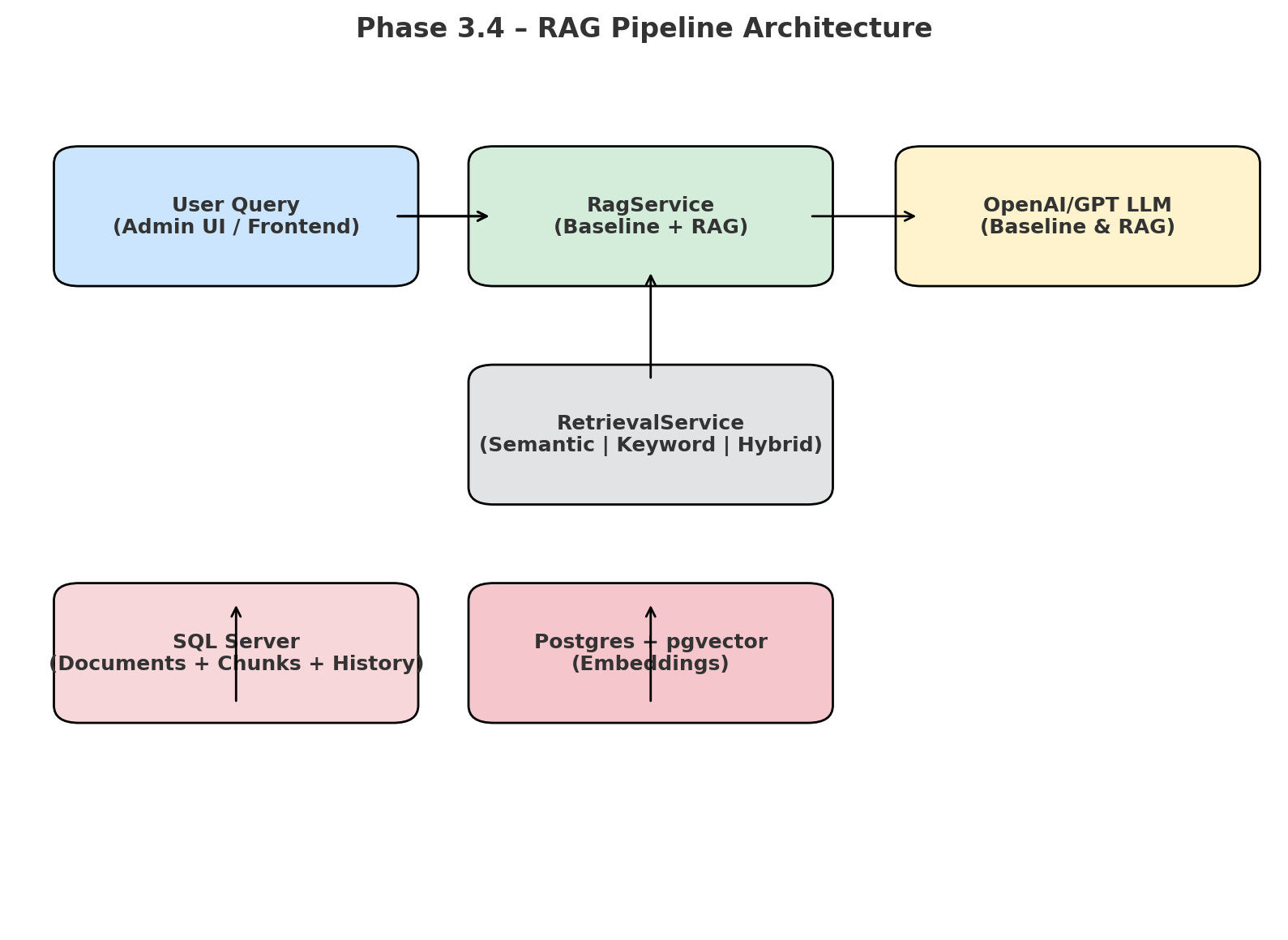
✅ Backend RAG pipeline  
✅ DTOs for structured results  
✅ History logging  
✅ Admin Debug UI with spinner  
✅ Clean migrations separation

## Expected Outcomes

A fully working RAG pipeline in .NET, transparent & debuggable, foundation for Phase 3.5 (RAG History Page).

## Project Structure (Updated)

/AirNir  
├── Library  
│ ├── ArNir.Core → Entities, DTOs, Config, Validations  
│ ├── ArNir.Data → DbContexts (SQL Server + Postgres), EF Migrations  
│ └── ArNir.Services → EmbeddingService, RetrievalService, RagService  
├── Presentation  
│ ├── ArNir.Admin → AdminLTE UI (embedding, retrieval, RAG comparison)  
│ └── ArNir.Frontend → End-user chat (Phase 3.5+)  
├── sql  
└── docs



Updated Architecture Diagram (Phase 3.4):

