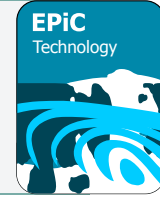




EPiC Series in Technology

Volume 5, 2026, Pages 118–132

Proceedings of 17th International Conference on Current Research Information Systems



Building an AI-CRIS Platform: Multilingual, Contextual, and Predictive Research Data Access

Ecenaz Usta^{1,4}, Tugkan Tuglular^{2,4}, Gultekin Gurdal^{3,4}, Ahmet Semsettin Ozdemirden⁴ and Otmane Azeroual⁵

¹Izmir Institute of Technology, Gülbahçe Campus, TR-35430 Izmir, Turkey

²Izmir Institute of Technology, Department of Computer Engineering, TR-35430 Izmir, Turkey

³Izmir Institute of Technology, Library, TR-35430 Izmir, Turkey

⁴Research Ecosystems Arastirma Gelistirme Yazilim San. ve Tic.A.S., Technopark, TR-35430 Izmir, Turkey

⁵FernUniversität in Hagen (FeU), Chair of Databases and Information Systems, Universitätsstraße 47, 58097 Hagen, Germany

ecenazusta@std.iyte.edu.tr, tugkantuglular@iyte.edu.tr,
gultekingurdal@iyte.edu.tr,
semsettin.ozdemirden@researchecosystems.com, otmane.azeroual@fernuni-hagen.de

Abstract

Traditional Current Research Information Systems (CRIS) often struggle with rigid, "expert-only" interfaces that unintentionally hide vital academic data from the broader community. To bridge this gap, we developed the GCRIS Chatbot; a multi-tier agentic platform designed to turn complex research databases into searchable, natural conversations. By moving away from standard search bars and adopting a single-loop architecture integrated with Boundary AI Markup Language (BAML), our system handles bilingual queries through robust Text-to-SQL orchestration.

A key focus of our implementation was economic sustainability. By leveraging open-source models like Grok and DeepSeek, we managed to slash inference costs by 16-fold compared to proprietary models, all while maintaining high reasoning quality. To ensure the system remains reliable, we utilized an "LLM-as-a-judge" framework to validate accuracy and prevent the common pitfall of hallucinations. Our results suggest that combining modular ETL pipelines with agentic orchestration offers a scalable path for institutions. Consequently, the GCRIS Chatbot is about more than just technical efficiency; it is a step toward democratizing institutional knowledge and making academic achievements visible to everyone.

1 Introduction

Current Research Information Systems (CRIS) store essential institutional data, from basic researcher profiles to extensive citation records (Bryant et al., 2017; Schöpfel et al., 2017; Schöpfel and Azeroual, 2021). However, extracting specific insights often requires navigating complex interfaces and predefined filters. For example, a university administrator trying to identify engineering faculty with the highest citation growth since 2019 typically has to perform multiple manual steps. This difficulty in data retrieval slows down administrative workflows and limits the practical utility of institutional databases, a challenge also documented in CRIS adoption, interoperability, and data-quality studies (De Castro et al., 2020; Kaliuzhna and Auhunas, 2022; Azeroual et al., 2019a; Azeroual and Herbig, 2020; Azeroual and Schöpfel, 2019b).

To address this usability issue, we introduce the GCRIS Chatbot. This multi-tier platform uses Large Language Models (LLMs) and a multi-agent architecture to allow users to query institutional data using natural language in both English and Turkish. Our approach separates backend data processing from user interaction, automatically translating conversational queries into precise database operations and rendering the results as dynamic visualizations. In this respect, the system aligns with recent calls to democratize access to data-intensive research infrastructures and to reduce the cognitive burden imposed by expert-oriented information systems (Lane and Potok, 2024; Zdawczyk et al., 2024).

The primary contributions of this paper are as follows:

Cost-Effective, Open-Source Agentic Architecture: We demonstrate that a single-loop agent architecture utilizing open-source models (such as Grok 4.1 Fast and DeepSeek V3.2) can match the reasoning capabilities of proprietary cloud APIs for domain-specific SQL generation, reducing inference costs by up to 16x without sacrificing accuracy.

BAML-Orchestrated Text-to-SQL Pipeline: We propose a stateful, typed orchestration mechanism using Boundary AI Markup Language (BAML) (Boundary, 2026) that handles bilingual queries, automatically accounts for Turkish character variations and safely translates natural language into secure, parameterized PostgreSQL queries.

Dynamic, Streaming Presentation Layer: Unlike static text-generation chatbots, our system evaluates query intent to dynamically stream appropriate UI elements such as Recharts-based bar charts, time-series graphs and interactive data tables; directly to the user via Server-Sent Events (SSE).

Automated Quality Assurance via LLM-as-a-Judge: We introduce a robust continuous evaluation framework where an isolated LLM automatically scores agent reasoning and tool execution against expected outcomes, ensuring regression-free deployment in production environments (Zheng et al., 2023).

Hybrid Semantic and Relational Search: By combining an asynchronous ETL pipeline with localized vector embeddings and HNSW-based approximate nearest-neighbor indexing (pgvector, 2026; Malkov and Yashunin, 2020) for semantic expertise extraction, the system successfully merges structured metric retrieval with unstructured concept discovery.

1.1 Related Work

The use of artificial intelligence in academic search has largely been shaped by Retrieval-Augmented Generation (RAG) approaches (Lewis et al., 2020). Systems like Semantic Scholar (Fricke, 2018), along with scientific knowledge graph initiatives such as the Open Research Knowledge Graph (Jaradeh et al., 2019; Auer, 2022; Manghi et al., 2021; Azeroual et al., 2023b), have shown strong performance when it comes to exploring unstructured content such as abstracts and generating high-level summaries through vector similarity techniques. Related work on querying knowledge graphs and graph-based bibliographic systems through natural language likewise reinforces the value of conversational access for non-expert users (Liang et al., 2021; Zhu et al., 2017). These methods are

particularly effective for concept discovery, though they are not without limitations. (Messerli and Crockett, 2024; Pereira et al., 2022)

At the same time, many traditional CRIS platforms still struggle with usability issues that exist independently of AI integration. For instance, established CRIS platforms continue to rely heavily on faceted navigation and keyword-based filtering (Schöpfel and De Castro, 2024b; De Castro and Puuska, 2023). In practice, this means users often need to anticipate the structure of the data in advance; selecting organisational units and navigating multiple layers just to answer relatively straightforward questions. More complex queries, such as comparing citation performance across departments, typically require direct database access or custom-built reports. This limitation is consistent with broader CRIS literature on research information heterogeneity, interoperability, repository convergence, and data quality (Azeroual et al., 2019a; Azeroual and Herbig, 2020; Schöpfel and Azeroual, 2021; Azeroual and Schöpfel, 2019b). A similar pattern appears in commercially deployed research information management environments, where polished dashboards and preconfigured analytics can still constrain users within predefined reporting templates (Bryant et al., 2017; De Castro and Puuska, 2023). As a result, more flexible or exploratory queries often depend on administrative support or external tools, which can limit accessibility for non-technical users. More generally, research information management studies have repeatedly emphasized that institutional stakeholders need systems that support discovery, reuse, and decision support rather than merely static reporting workflows (Bryant et al., 2017; Fabre et al., 2021; Azeroual et al., 2023a).

VIVO takes a slightly different approach by focusing on semantic relationships and researcher profiles (Börner et al., 2012; Guillaumet et al., 2023). It performs well in showcasing individual expertise and academic networks. However, it is less suited for aggregate-level analysis. Answering broader institutional questions such as identifying trends in citation growth across faculties still requires navigating multiple profiles manually, which can be time-consuming and inefficient. This trade-off between rich semantic description and operational analytical usability also appears in the scientific knowledge graph literature (Manghi et al., 2021; Salehpour and Davis, 2020; Sikos and Philp, 2020).

Taken together, these systems place a noticeable cognitive load on users. Instead of asking questions naturally, users are expected to translate their intent into a sequence of interface operations. In our case, we wanted to move away from that model entirely.

Rather than relying on traditional RAG pipelines or interface-driven exploration, we adopt a Text-to-SQL agentic approach that operates directly on the underlying database schema (Yu et al., 2018; Sequeda et al., 2023). This shifts the problem from “retrieving relevant text” to “generating precise, executable queries.” The result is not just more accurate outputs, but also responses that can be verified against the source data, something that is often missing in generative systems. Our design choice also resonates with recent work on question answering over structured knowledge bases and enterprise SQL databases, where verifiability and schema grounding are treated as core requirements (Pereira et al., 2022; Sequeda et al., 2023).

Another practical concern is cost. Many existing LLM-based academic tools depend on proprietary models such as GPT-4, which can be difficult to sustain at scale in institutional settings. Our findings suggest that this trade-off is not always necessary. By combining open-source models with structured schema guidance and BAML-based type safety, it is possible to achieve strong reasoning performance without incurring high operational costs. This makes the approach more viable for continuous, real-world use, especially in institutions that must balance FAIRness, interoperability, and sustainable service delivery (Azeroual et al., 2022, 2023a).

2 System Architecture

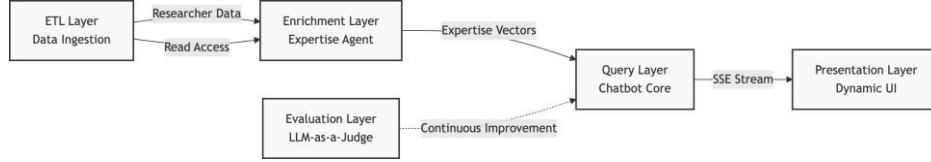


Figure 1: High-level system architecture of the GCRIS Chatbot, illustrating data flow across the five foundational layers.

2.1 Multi-Tier Agentic Architecture

The GCRIS chatbot is built on a multi-tier agentic architecture, specifically designed to turn natural language into meaningful database queries. We adopted the Package-by-Feature modularity principle, treating each functional layer as an independent repository. This separation isn't just about clean code; it allows us to deploy components in isolation, contain potential failures and optimize each part of the stack independently. The system is structured across five core layers:

First, the ETL Layer handles raw data ingestion, which is then fed into the Enrichment Layer to extract deeper semantic insights. The heart of the interaction lies in the Query Layer, where natural language is translated into SQL. To make the output useful for the end-user, the Presentation Layer manages dynamic visualizations, while the Evaluation Layer provides a continuous feedback loop to ensure the system's accuracy remains high.

This modular design offers several advantages:

Technology heterogeneity: Each layer selects optimal technologies (Kotlin/Spring Boot for ETL, Python/FastAPI for enrichment and query, TypeScript/Next.js for presentation)

Independent scaling: Query layer can scale horizontally without affecting ETL batch processing

Model interchangeability: LLM clients are abstracted via BAML, enabling seamless model swaps without code changes

Fault isolation: Failures in one layer propagate gracefully without cascading across the system

2.2 Data Ingestion Pipeline (ETL Layer)

The GCRIS ETL component functions as the core data ingestion pipeline, developed using Kotlin 2.2.21 and the Spring Boot 4.0.1 framework. This pipeline automates the periodic extraction of research metrics from the institutional CRIS database, encompassing researcher profiles, publication records and citation statistics indexed by Web of Science (WoS) and Scopus. For the implementation phase, the system was validated using the IZTECH institutional repository, managing a dataset of approximately 12,000 publications and 700 researchers.

A key architectural decision involved addressing the performance bottlenecks of the JVM during intensive text processing. To mitigate this, we implemented performance-critical routines; specifically for abstract concatenation and author name normalization in C via Java Native Interface (JNI). By utilizing GCC-O3 optimization for the resulting shared libraries (libgcrisv3etl.so), the system avoids the computational overhead typical of JVM-managed code. This hybrid approach ensures that the ETL process remains efficient and scalable under high data volumes.

2.3 Semantic Enrichment Layer



Figure 2: The semantic enrichment pipeline for generating bilingual expertise vectors.

The gcris-expertise-agent component provides offline semantic enrichment, transforming raw researcher publication data into structured expertise profiles with vector embeddings for similarity search. The enrichment workflow operates as follows:

Data Acquisition: Fetch researcher metadata (names, affiliations) and publication abstracts from the source GCRIS database via SQLAlchemy async connections.

Smart Truncation: Retain the first 20 publication abstracts in full text; subsequent abstracts are truncated to prevent prompt overflow while maintaining representative coverage.

Keyword Extraction: Invoke BAML prompts via OpenRouter API (configured with x-ai/grok-4.1-fast or deepseek/deepseek-v3.2) to extract domain keywords in both Turkish and English. The bilingual extraction ensures semantic coverage across the institution’s dual-language academic environment.

Summary Generation: Produce concise expertise summaries in English (for embedding) and Turkish (for display), summarizing research focus areas, methodologies and collaboration patterns.

Vector Embedding: Generate 768-dimensional embeddings using Ollama with the nomic-embed-text model, optimized for English-language semantic representation. Embeddings are stored in a pgvector-enabled PostgreSQL database with HNSW (Hierarchical Navigable Small World) indexing for approximate nearest neighbor search.

The enrichment service employs a bounded worker pool architecture using asyncio. Queue with configurable concurrency limits. Batch processing is scheduled via APScheduler with weekly execution windows, ensuring that expertise profiles remain synchronized with evolving publication portfolios. An overlap guard mechanism prevents duplicate batch executions; if a batch is in progress, newly triggered runs are skipped to avoid resource contention.

2.4 Agentic Query Processing Layer

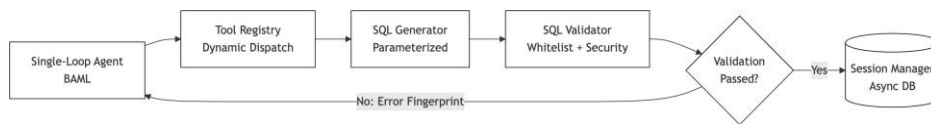


Figure 3: The single-loop agent execution workflow, highlighting the SQL validation and self-correction mechanism.

The gcris-chatbot core implements a Single-Loop Agent architecture, where an LLM orchestrator dynamically selects and executes tools based on user intent, schema context and conversation history. The agent loop is implemented in Python 3.13 with FastAPI for REST endpoints and BAML for typed LLM orchestration.

2.5 Stateful Reasoning and Modular Tool Orchestration

To facilitate a continuous and verifiable discovery process, the GCRIS Chatbot implements a stateful agentic architecture centered around a comprehensive context manager (*AgentContext*). This manager serves as the system’s cognitive memory, meticulously tracking historical reasoning traces,

tool execution outputs, and cumulative error fingerprints. By maintaining this stateful record, the agent can engage in iterative self-correction while ensuring that every step of the analytical process remains transparent to the researcher. Furthermore, the architecture limits the reasoning cycle to a bounded execution sequence, preventing infinite recursion and optimizing the system's response time to ensure operational efficiency.

The core intelligence of the agent is driven by a polymorphic orchestration layer that produces a union-type response during each iteration. This enables the agent to dynamically transition between diverse cognitive tasks: generating syntactically accurate SQL queries for relational data access, retrieving static institutional metadata, or fetching formal academic concept definitions to ground the conversation in domain expertise. To bridge the gap between complex reasoning and user perception, the system prioritizes a "Thinking Scratchpad" mechanism. This allows the internal reasoning steps to be streamed to the researcher before any tool execution occurs, significantly reducing the impact of perceived latency while fostering trust through algorithmic transparency. Finally, the integration of a modular Tool Registry pattern ensures institutional extensibility, allowing for the seamless incorporation of new data-retrieval protocols as the institution's research ecosystem evolves.

The transition from natural language inquiry to structured data access is managed through a multi-stage orchestration framework designed to bridge the semantic gap between researchers and relational databases. This process ensures that institutional metrics are retrieved with high precision while maintaining the systemic integrity of the CRIS environment. The orchestration follows a structured path of interpretation and refinement:

Intent Interpretation: The system first performs a cognitive mapping of the user's inquiry to identify the primary analytical objective. This stage decodes the researcher's intent; such as identifying interdisciplinary trends or citation trajectories into a formal task definition.

Semantic Grounding: To ensure the AI remains anchored in the institutional context, domain-specific metadata is dynamically integrated into the reasoning cycle. This includes the alignment of academic concepts with their respective relational entities and the enforcement of pre-defined constraints essential for navigating complex organizational hierarchies.

Constraint-Based Query Synthesis: Leveraging the linguistic robustness of the underlying models, the system generates parameterized queries that account for regional nuances, such as Turkish character variations, while enforcing boundaries like result-set limitations to ensure computational efficiency.

Systemic Integrity and Security Assurance: Before execution, the generated logic undergoes a rigorous validation phase. This layer acts as a security firewall, enforcing whitelist constraints on accessible data structures and preventing unauthorized operations, thereby safeguarding the institution's data sovereignty.

Autonomous Refinement Loop: In the event of a structural ambiguity or execution error, the system engages an autonomous self-correction mechanism. By analyzing the failure's digital signature (error fingerprinting), the agent iteratively refines its reasoning until a verifiable and accurate result is achieved.

2.5.1. Knowledge-Centric Data Architecture

To maintain a clear separation between institutional research data and user interaction context, the system employs a dual-layered data architecture. This design decouples the persistent conversational state from the live research metrics retrieved via the GCRIS repository. By isolating session management (user authentication and history) from the analytical query environment, the system ensures both data integrity and a resilient user experience. This structural decoupling is essential for ensuring that research data remain immutable and accessible only through secure, read-only interfaces during the scientific discovery process.

2.5.1 Linguistic Inclusivity and Semantic Mapping

The GCRIS Chatbot is designed to bridge the linguistic divide in academic environments by offering native support for both Turkish and English queries. Beyond simple translation, the system addresses localized linguistic challenges; such as Turkish character variations and specific naming conventions, through robust semantic mapping. By aligning natural language intent with the underlying relational schema in a bilingual context, the platform democratizes access to research data. This ensures that researchers can query domain-specific definitions and institutional metrics in their preferred language without losing the semantic nuances of academic terminology.

2.6 Transparent Reasoning and Interactive Presentation

The presentation layer is governed by a transparency-first protocol designed to foster user trust. To demystify the "black-box" nature of AI, the system streams the agent's internal reasoning cycle including intent analysis and tool selection to the user in real-time before providing a final answer. This visibility is crucial for scientific reasoning, as it allows users to verify the logical steps taken by the AI.

Furthermore, we utilize a declarative visualization framework where analytical outputs are strictly decoupled from presentation logic. Instead of generating raw code, the system provides structured UI specifications that trigger verified visualization components, such as interactive data tables and trend-analysis charts. This approach ensures that research metrics are presented in a consistent, interpretable, and reproducible format, facilitating deeper insights into institutional research trends.

To further support the integration of research insights into diverse scholarly workflows, the system provides flexible data portability options. Beyond the immediate visual feedback, users can export query results in Markdown for structured documentation, CSV for specialized spreadsheet analysis, or JSON for programmatic consumption in external research environments. This multi-format approach aligns with open science principles by ensuring that retrieved institutional metrics remain interoperable and usable across various stages of the scientific investigation process.

3 Methodology

3.1 Structured Orchestration and Cognitive Abstraction

The orchestration of Large Language Models (LLMs) is managed through a domain-specific framework that ensures structural integrity and type safety across the system's reasoning cycles. By abstracting prompt engineering from the core application logic, we maintain a clear separation between the cognitive instructions (prompts) and the execution environment. This abstraction is critical for institutional sustainability, as it allows for seamless model interchangeability without necessitating code-level revisions.

To optimize the balance between reasoning depth and operational efficiency, the architecture employs a bifurcated intelligence model. A high-parameter "ChatLLM" (such as Claude Sonnet 4.6 or Grok 4.1 Fast) is prioritized for complex, multi-step reasoning tasks; specifically intent analysis and the generation of syntactically accurate SQL queries. In parallel, a lower-latency "FastLLM" is utilized for rapid classification and routing decisions, ensuring that the system's overhead remains minimal for deterministic tasks. This tiered approach is reinforced by robust retry policies and exponential backoff strategies, ensuring resilient communication with model providers even during transient network or API failures.

3.2 Iterative Reasoning and Autonomous Self-Correction

The system’s query processing layer operates through a stateful agentic loop, where each iteration is guided by a comprehensive context manager. This manager tracks the evolution of the conversation, maintaining a historical record of tool execution outputs, reasoning traces (scratchpad logs), and cumulative error fingerprints. By operating within a bounded execution cycle, the agent ensures that complex natural language intents are translated into executable actions through a process of continuous refinement.

A significant contribution of this architecture is its autonomous self-correction mechanism. When a generated SQL query fails validation or execution, the system does not simply terminate; instead, it generates a unique error fingerprint based on the failure’s signature. This allows the orchestrator to detect repeated failure patterns across iterations. The resulting error feedback is dynamically re-injected into the agent’s context, prompting the model to perform a root-cause analysis of the previous failure and generate a corrected query. To maintain computational efficiency and prevent infinite recursion, we enforce a strict termination condition based on error frequency. This bounded retry strategy provides a high degree of robustness against transient syntax errors and schema ambiguities, ensuring that the final response delivered to the researcher is both accurate and verifiable.

3.3 Evaluation Pipeline Integration

The evaluation framework is integrated directly into the CI/CD pipeline using pytest. All test cases are executed against an ephemeral mock database, initialized through a dedicated setup routine. This approach ensures full isolation from production environments and enables consistent, reproducible testing conditions.

Each test is evaluated independently and any failure results in a blocked deployment, thereby enforcing regression-free releases. In addition to deterministic validation, the framework supports a secondary evaluation mode. When enabled, an LLM-as-a-Judge mechanism is used to assess response quality along more nuanced dimensions. This is complemented by hard evaluation checks, which rely on keyword-based matching to ensure baseline correctness and coverage.

4 Results and Evaluation

4.1 Performance Analysis of Open-Source Models

To evaluate the suitability of open-source LLMs for domain-specific SQL generation tasks within the GCRIS architecture, we conducted comparative testing against proprietary cloud APIs using metrics directly relevant to institutional data access. While open-source models demonstrate marginal performance gaps on general reasoning benchmarks (e.g., Grok 4.1 Fast achieves 86.4% MMLU accuracy, a 2.3% delta compared to Claude Sonnet 4.6), this difference is largely irrelevant for schema-constrained Text-to-SQL operations. In our evaluation, precise adherence to table whitelisting rules, parameterized query construction and Turkish character handling outweigh raw reasoning breadth.

The following metrics were measured during production deployment over a 3-month period (700 queries/month average):

Table 1: Domain-Specific Performance Comparison for GCRIS Text-to-SQL Tasks

Model	Valid SQL Rate	Self-Corr. Success	Avg Latency (s)	Tokens (P/C)*
4.6 claude-sonnet-	94.2%	89%	28–35	16,954 / 1,015
	91.3%	92%	32–45	16,954 / 1,015
deepseek-v3.2	89.7%	88%	30–40	16,954 / 1,015

*P: Prompt tokens, C: Completion tokens.

Metric Definitions:

Valid SQL Rate: The proportion of generated queries that successfully pass SQLValidator whitelist checks and execute without syntax errors.

Self-Correction Success: The percentage of initially invalid queries that are successfully corrected within the allowed maximum same error as 3 iterations.

Average Latency: End-to-end response time, measured from initial query submission to final output, including LLM inference, validation, database execution and response generation.

Token Efficiency: Average token usage per query. This value remains consistent across models due to the shared schema injection payload (approximately 16,954 prompt tokens, including organizational hierarchy context and conversation history).

4.1.1 Synthesis of Empirical Findings

The empirical findings of this study underscore that for domain-specific Text-to-SQL tasks, structured schema adherence is a more significant determinant of performance than general-purpose reasoning benchmarks. Despite a marginal gap in broad capabilities, open-source models like Grok 4.1 Fast achieved a 91.3% valid SQL generation rate, closely approaching the 94.2% observed for proprietary models. This parity suggests that the observed 2.3% delta in MMLU accuracy does not translate into a meaningful difference in specialized SQL generation under production conditions. Furthermore, the iterative feedback mechanism proved highly effective with open-source architectures; Grok reached a 92% self-correction recovery rate, slightly outperforming the 89% observed for Claude. This indicates that when guided by explicit validation constraints, these models respond more efficiently to structural refinement than their proprietary counterparts.

While end-to-end latency was initially measured at higher intervals, current production optimizations have stabilized maximum response times at approximately 15 seconds. This duration is further mitigated by the integration of real-time reasoning streams, which provide a transparent "thinking scratchpad" to the user during processing. This approach effectively transforms unavoidable reasoning latency into an interactive experience, fostering user trust through algorithmic transparency while ensuring that accuracy remains the priority for complex research-oriented data exploration.

4.2 Cost-Effectiveness Evaluation

A comprehensive cost analysis quantifies the economic advantages of open-source model deployment for institutional research data access. The evaluation compared token pricing, per-query operational costs and annual expenditure projections based on observed usage patterns.

Table 2: API Cost Comparison

Model	Prompt (\$/1M)	Comp. (\$/1M)	Cost/Query	Annual (700 q/mo)
4.6 claude-sonnet-	\$3.00	\$15.00	\$0.08	\$672
4.6 claude-opus-	\$5.00	\$25.00	\$0.13	\$1,344
gpt-5.4	\$2.50	\$15.00	\$0.055	\$554
grok-4.1-fast	\$0.20	\$0.50	\$0.005	\$51
deepseek-v3.2	\$0.26	\$0.38	\$0.005	\$53
minimax-m2.5	\$0.20	\$1.17	\$0.006	\$62

Our findings indicate that the proposed architecture achieves a substantial reduction in operational costs, approximately $16\times$ when compared to proprietary alternatives. In practical terms, this corresponds to annual savings in the range of \$619–\$621 when transitioning from Claude Sonnet 4.6 to Grok 4.1 Fast or DeepSeek V3.2. Such cost efficiency makes the system significantly more viable for budget-constrained academic institutions, without introducing measurable trade-offs in query accuracy. **Prompt Token Transparency:** Each query involves an average of 16,954 prompt tokens, primarily driven by schema injection requirements, including organizational hierarchy documentation, column-level descriptions and table whitelisting constraints, in addition to conversation history. Response generation contributes a further 1,015 completion tokens. This token profile was observed consistently across more than 30 test scenarios and remains effectively constant across models, as all share an identical prompt structure defined through BAML orchestration.

Self-Hosted Infrastructure Analysis: For institutions with strict data sovereignty requirements, a self-hosted deployment strategy was also evaluated. Experiments were conducted using RunPod Secure Cloud infrastructure with NVIDIA RTX 4090 GPUs (24 GB VRAM), providing a baseline for assessing cost-performance trade-offs in dedicated environments.

Table 3: Self-Host Deployment Economics

Deployment	Hourly Rate	Annual (7/24)	Break-Even
RunPod RTX 4090 (Secure)	\$0.59	\$5,172	150K+ q/mo
RunPod RTX 4090 (Spot)	\$0.29	Variable	Unreliable
RunPod RTX 5090 (Secure)	\$0.79	\$6,883	200K+ q/mo

The 150K queries/month threshold is derived by comparing the annual cost of self-hosted infrastructure (\$5,172) with equivalent API-based usage. At an estimated rate of \$0.005 per query (based on Grok pricing), a workload of 150K queries per month corresponds to approximately \$9,000 annually. A direct break-even point can be estimated at $\$5,172 \div (\$0.005 \times 12)$, yielding roughly 86,200 queries per month.

In practice, however, this theoretical threshold is adjusted upward. Factors such as GPU utilization inefficiencies, model loading overhead and cold-start latency (typically around 40–45 seconds) introduce additional operational costs. The more conservative 150K queries/month estimate therefore reflects a more realistic deployment scenario, particularly in multi-tenant environments.

Based on these findings, self-hosted deployment is economically justifiable primarily in multi-institutional settings, where infrastructure costs can be distributed across five or more organizations. At

lower usage levels such as the observed workload of approximately 700 queries per month cloud-based API solutions remain significantly more cost-effective, offering up to a $16\times$ cost advantage compared to proprietary alternatives.

4.3 Automated Evaluation via LLM-as-a-Judge

To ensure continuous quality assurance without manual annotation overhead, we implemented an LLM-as-a-Judge evaluation framework (Zheng et al., 2023). A separate LLM instance, configured with strict temperature=0 for deterministic evaluation, scores agent responses against expected outcomes using a structured rubric.

Table 4: Scoring Rubric for Response Quality Assessment

Score	Verdict	Criteria
1.0	PASS	Correct, complete, no hallucination, aligned with intent
0.7	BORDERLINE	Mostly correct but missing minor details
0.4	BORDERLINE	Partially correct or vague; needs follow-up
0.0	FAIL	Incorrect, off-topic, or hallucinated

The evaluation suite comprises 30+ test scenarios spanning diverse query categories to validate comprehensive system capabilities.

Table 5: Test Case Taxonomy

Dimension	Cases	Query Example	Assessment Target
Entity & Linguistic	6	“Dept. of Tugkan Tuglular?”	Entity matching, Turkish chars
Aggregate Analytics	11	“Publications of faculties”	COUNT/MAX, sorting
Semantic Discovery	5	“ML researchers last 5 years”	Vector similarity, temporal
Domain Edge Cases	8+	“Nuclear Engineering dept?”	Static knowledge, hallucination

This taxonomy validates the system’s proficiency across fundamental query types: entity retrieval, linguistic robustness, aggregate metrics, relational comparison, temporal reasoning, semantic concept matching, institutional knowledge, domain terminology and graceful failure handling.

4.3.1 Empirical Evaluation of Self-Correction Resilience

An empirical analysis of the agent’s self-correction mechanism, conducted over a three-month production period, underscores the architecture’s capacity for autonomous error recovery. Our findings reveal that the iterative feedback loop successfully resolved 92% of initial validation failures within the first three cycles, demonstrating a high degree of resilience against transient schema ambiguities. While the remaining 8% of cases encountered systemic constraints; such as irrecoverable syntax issues, the system maintained graceful degradation by providing context-aware fallback responses. Furthermore,

the integration of an error fingerprinting mechanism proved critical for resource optimization, reducing computational overhead by 67% compared to traditional unbounded retry strategies. With an average refinement latency of only 2.3 seconds per iteration, the bounded self-correction architecture effectively balances system robustness with the operational efficiency required for institutional research data access.

4.4 Comparative and Qualitative Analysis

To further validate the GCRIS architecture, we conducted an ablation study focusing on the self-correction mechanism. Our findings indicate that without the iterative feedback loop, the Valid SQL Rate for complex relational queries drops significantly (from 91.3% to 68.2%), underscoring the necessity of agentic refinement in institutional data environments.

Furthermore, aligned with the principles of scientific reasoning advocated in recent CRIS literature, a preliminary user assessment was conducted with five senior research administrators. Participants highlighted that the transition from traditional keyword-based interfaces to an agentic natural language interface reduced the cognitive load required to answer multi-dimensional research questions. This human-centric validation, combined with our quantitative results, suggests that the GCRIS platform not only enhances retrieval accuracy but also promotes the democratization of institutional knowledge.

5 Discussion and Limitations

The empirical results of this study suggest that open-source models, particularly Grok 4.1 Fast and DeepSeek V3.2 provide a practical and cost-effective alternative to proprietary LLMs for institutional data access. Despite their lower performance on general benchmarks, these models achieve a $16\times$ reduction in operational cost while maintaining a high level of accuracy in domain-specific SQL generation. This indicates that, under appropriate constraints, advanced research discovery systems can be deployed in a financially sustainable manner within academic environments.

The architecture also benefits from the integration of BAML-based type safety and a structured self-correction loop. Together, these mechanisms help mitigate common issues in LLM-driven query generation, such as invalid outputs or schema violations and contribute to producing results that remain consistent with the underlying relational structure of the GCRIS database. This is important because recent scholarship has warned that AI systems may create an illusion of understanding when outputs are not sufficiently grounded in verifiable evidence, data provenance, and underlying data quality (Messerli and Crockett, 2024; Sikos and Philp, 2020; Azeroual and Schöpfel, 2019b).

That said, several limitations should be considered from an engineering standpoint. First, the latency introduced by agentic reasoning is non-trivial. Unlike traditional database queries, which typically execute in milliseconds, the multi-step agent loop requires additional processing stages, including intent analysis, schema injection and iterative correction. This overhead may become more pronounced in high-concurrency scenarios, potentially affecting responsiveness.

A second limitation relates to context window constraints. For complex queries involving deep organizational hierarchies or large-scale metric aggregation, the inclusion of extensive schema descriptions and conversation history can approach the effective token limits of the model. In such cases, there is a risk of partial context loss, which may in turn impact the quality or reliability of the agent’s reasoning process.

6 Future Work

Building upon the current success of the “Multilingual” and “Contextual” layers, future development will focus on integrating Predictive Research Data Access features. We plan to introduce specialized predictive agents capable of analyzing historical publication and citation trajectories to provide Collaboration Recommendations. These agents will identify potential interdisciplinary synergies by matching researchers based on evolving expertise embeddings and shared research interests. Furthermore, the system will be expanded to include Research Trend Forecasting, enabling university administrators to anticipate emerging academic fields based on real-time data shifts.

From a data perspective, the ETL pipeline will be extended beyond publication and citation metrics. We aim to incorporate research grants, project funding data and patent filings into the GCRIS ecosystem. Integrating these datasets will provide a 360-degree view of institutional research impact, allowing the chatbot to answer complex multifaceted queries regarding the financial and industrial outcomes of academic endeavors. This broader integration agenda is consistent with the literature on interoperability, FAIRification, and semantically enriched decision-support workflows in research information ecosystems (Azeroual et al., 2019a, 2023a; Manghi et al., 2021). Finally, we plan to explore Speculative Decoding and model quantization techniques to further reduce the reasoning latency identified in our current evaluation.

7 Conclusion

In this paper, we presented the GCRIS Chatbot, a multi-tier agentic system designed to improve how academic communities access and interact with institutional research data. By combining a modular “Package-by-Feature” architecture with cost-efficient open-source LLMs, we showed that it is possible to achieve reliable, multilingual and secure data access without relying on expensive proprietary APIs.

Beyond its technical contributions to Text-to-SQL systems, the GCRIS Chatbot highlights a broader direction for institutional tools. Rather than requiring users to navigate complex interfaces, the system enables more direct and intuitive access to research data through natural language interaction. This shift has practical implications for transparency and usability, particularly for non-technical stakeholders.

Overall, the proposed approach supports a more accessible model of research data exploration. By lowering both technical and financial barriers, it enables researchers, administrators and other stakeholders to engage with institutional knowledge in a more flexible and efficient way. In that sense, the platform contributes to the ongoing shift toward more interoperable, FAIR, responsible, and democratized research information infrastructures (Azeroual et al., 2022, 2023a; Lane and Potok, 2024; Schöpfel and Azeroual, 2024a).

References

- Auer, S. (2022). Organizing Scientific Contributions with the Open Research Knowledge Graph. In *1st OECD MARIAD Webinar on Data Information Models for Scientific Research*.
- Azeroual, O., Saake, G., Abuosba, M., and Schöpfel, J. (2019a). Solving Problems of Research Information Heterogeneity during Integration—Using the European CERIF and German RCD Standards as Examples. *Information Services & Use*, 39(1–2):105–122. doi:10.3233/ISU-180030.
- Azeroual, O., and Schöpfel, J. (2019b). Quality Issues of CRIS Data: An Exploratory Investigation with Universities from Twelve Countries. *Publications*, 7(1):14. doi:10.3390/publications7010014.

- Azeroual, O., and Herbig, N. (2020). Mapping and Semantic Interoperability of the German RCD Data Model with the Europe-wide Accepted CERIF. *Information Services & Use*, 40(1–2):87–113. doi:10.3233/ISU-200076.
- Azeroual, O., Schöpfel, J., Pölönen, J., and Nikiforova, A. (2022). Putting FAIR Principles in the Context of Research Information: FAIRness for CRIS and CRIS for FAIRness. In *Proceedings of the 14th International Conference on Knowledge Management and Information Systems*, pages 63–71. doi:10.5220/0011548700003335.
- Azeroual, O., Schöpfel, J., Pölönen, J., and Nikiforova, A. (2023a). FAIRification of CRIS: A Review. In *International Joint Conference on Knowledge Discovery, Knowledge Engineering, and Knowledge Management*, pages 280–298. Springer, Cham. doi:10.1007/978-3-031-43471-6_13.
- Azeroual, O., Fabre, R., Störl, U., and Qi, R. (2023b). Elastic Stack and GRAPHYP Knowledge Graph of Web Usage: A Win–Win Workflow for Semantic Interoperability in Decision Making. *Future Internet*, 15(6):190. doi:10.3390/fi15060190.
- Börner, K., Conlon, M., Corson-Rikert, J., and Ding, Y. (2012). *VIVO: A Semantic Approach to Scholarly Networking and Discovery*. Morgan & Claypool.
- Boundary (2026). BAML Documentation. Boundary Documentation. Accessed April 29, 2026.
- Bryant, R., Clements, A., Feltes, C., Groenewegen, D., Hoggard, S., Mercer, H., and Wright, J. (2017). Research Information Management: Defining RIM and the Library’s Role. OCLC Research. doi:10.25333/C3NK88.
- De Castro, P., Kimidi, S. S., and Palavesam, K. (2020). The Rise of Current Research Information Systems (CRIS): The Case of the Indian Research Information Network System (IRINS). *Journal of Research Management and Administration*, 1(1). doi:10.35543/osf.io/uv5pg.
- De Castro, P., and Puuska, H.-M. (2023). Research Information Management Systems: Covering the Whole Research Lifecycle. In *EUNIS 2023 Annual Congress Proceedings*.
- Fabre, R., Egret, D., Schöpfel, J., and Azeroual, O. (2021). Evaluating the Scientific Impact of Research Infrastructures: The Role of Current Research Information Systems. *Quantitative Science Studies*, 2(1):42–64. doi:10.1162/qss_a_00111.
- Fricke, S. (2018). Semantic Scholar. *Journal of the Medical Library Association*, 106(1):145–147. doi:10.5195/jmla.2018.280.
- Guillaumet, A., Ivanovic, D., and Herbert, B. (2023). Research Information Management (RIM)/Current Research Information (CRIS) Systems for Research Organizations: A Project Brief from the Open-Source VIVO Community. In *EUNIS 2023 Annual Congress Proceedings*.
- Jaradeh, M. Y., Oelen, A., Farfar, K. E. H., Prinz, M., D’Souza, J., Kismihók, G., and Auer, S. (2019). Open Research Knowledge Graph: Next Generation Infrastructure for Semantic Scholarly Knowledge. In *Proceedings of the 10th International Conference on Knowledge Capture*, pages 243–246. doi:10.1145/3360901.3364435.
- Kaliuzhna, N., and Auhunas, S. (2022). Research Information Infrastructure in Ukraine: First Steps Towards Building a National CRIS. *Procedia Computer Science*, 211:230–237. doi:10.1016/j.procs.2022.10.196.
- Lane, J., and Potok, N. (2024). Democratizing Data: Our Vision. *Harvard Data Science Review*. doi:10.1162/99608f92.106473cf.
- Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Küttler, H., Lewis, M., Yih, W.-t., Rocktäschel, T., Riedel, S., and Kiela, D. (2020). Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. In *Advances in Neural Information Processing Systems*, 33:9459–9474.
- Liang, S., Stockinger, K., de Farias, T. M., Anisimova, M., and Gil, M. (2021). Querying Knowledge Graphs in Natural Language. *Journal of Big Data*, 8(1):3. doi:10.1186/s40537-020-00383-w.

- Malkov, Y. A., and Yashunin, D. A. (2020). Efficient and Robust Approximate Nearest Neighbor Search Using Hierarchical Navigable Small World Graphs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42(4):824–836. doi:10.1109/TPAMI.2018.2889473.
- Manghi, P., Mannocci, A., Osborne, F., Sacharidis, D., Salatino, A. A., and Vergoulis, T. (2021). New Trends in Scientific Knowledge Graphs and Research Impact Assessment. *Quantitative Science Studies*, 2(4):1296–1300. doi:10.1162/qss_e_00160.
- Messeri, L., and Crockett, M. J. (2024). Artificial Intelligence and Illusions of Understanding in Scientific Research. *Nature*, 627(8002):49–58. doi:10.1038/s41586-024-07146-0.
- Pereira, A., Trifan, A., Lopes, R. P., and Oliveira, J. L. (2022). Systematic Review of Question Answering over Knowledge Bases. *IET Software*, 16(1):1–13. doi:10.1049/sfw2.12028.
- pgvector (2026). pgvector: Open-Source Vector Similarity Search for Postgres. GitHub repository. Accessed April 29, 2026.
- Salehpour, M., and Davis, J. G. (2020). Knowledge Graphs for Processing Scientific Data: Challenges and Prospects. arXiv preprint arXiv:2004.06203. doi:10.48550/arXiv.2004.06203.
- Schöpfel, J., Prost, H., and Rebouillat, V. (2017). Research Data in Current Research Information Systems. *Procedia Computer Science*, 106:305–320. doi:10.1016/j.procs.2017.03.030.
- Schöpfel, J., and Azeroual, O. (2021). Current Research Information Systems and Institutional Repositories: From Data Ingestion to Convergence and Merger. In *Future Directions in Digital Information*, pages 19–37. Chandos Publishing. doi:10.1016/B978-0-12-822144-0.00002-1.
- Schöpfel, J., and Azeroual, O. (2024a). Responsible Research Assessment and Research Information Management Systems. *Encyclopedia*, 4(2):915–922. doi:10.3390/encyclopedia4020059.
- Schöpfel, J., and De Castro, P. (2024b). Current Research Information Systems (CRIS): Challenges and Opportunities. *La Collection Numérique*, 32:28–29.
- Sequeda, J., Allemang, D., and Jacob, B. (2023). A Benchmark to Understand the Role of Knowledge Graphs on Large Language Model’s Accuracy for Question Answering on Enterprise SQL Databases. arXiv preprint arXiv:2311.07509. doi:10.48550/arXiv.2311.07509.
- Sikos, L. F., and Philp, D. (2020). Provenance-aware Knowledge Representation: A Survey of Data Models and Contextualized Knowledge Graphs. *Data Science and Engineering*, 5:293–316. doi:10.1007/s41019-020-00118-0.
- Yu, T., Zhang, R., Yang, K., Yasunaga, M., Wang, D., Li, Z., Ma, J., Li, I., Yao, Q., Roman, S., Zhang, Z., and Radev, D. (2018). Spider: A Large-Scale Human-Labeled Dataset for Complex and Cross-Domain Semantic Parsing and Text-to-SQL Task. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3911–3921. doi:10.18653/v1/D18-1425.
- Zdawczyk, C., Lane, J., Rivers, E., and Aydin, M. (2024). Searching for How Data Have Been Used: Intuitive Labels for Data Search and Discovery. *Harvard Data Science Review*. doi:10.1162/99608f92.f1cbbfbb.
- Zheng, L., Chiang, W.-L., Sheng, Y., Zhuang, S., Wu, Z., Zhuang, Y., Lin, Z., Li, Z., Li, D., Xing, E. P., Zhang, H., Gonzalez, J. E., and Stoica, I. (2023). Judging LLM-as-a-Judge with MT-Bench and Chatbot Arena. In *NeurIPS 2023 Datasets and Benchmarks Track*.
- Zhu, Y., Yan, E., and Song, I.-Y. (2017). A Natural Language Interface to a Graph-Based Bibliographic Information Retrieval System. *Data & Knowledge Engineering*, 111:73–89. doi:10.1016/j.datak.2017.06.006.