Journal of Distributed Computing and Systems(JDCS)
Vol.9, Issue 1, 2026, Pages 73-82

A Scalable Distributed Architecture ifor Semantic Vector-Based Processing of Large-Scale Banking Transactions

Moosa Kalanaki
R&D researcher, Karizab Rayab Pegah Co., Qazvin, Iran

Article History:
Received: 17 January 2026
Received in revised form: 17 March 2026
Accepted: 17 May 2026
Available online: 15 August 2026

Abstract

Traditional relational database–based search approaches often suffer from performance degradation, high query latency, and limited capability in handling large-scale textual and semantic data. These limitations become more pronounced as the volume and complexity of banking transactions grow.

This study proposes the design and implementation of a secure, scalable, and distributed hybrid architecture within banking transaction ecosystem. The proposed system aims to balance security, efficiency, and intelligent data processing. The architecture begins by separating sensitive and non-sensitive data to reduce exposure risks. To improve retrieval capabilities, a semantic analysis layer based on semantic indexing is introduced, enabling concept-based search instead of traditional keyword-based retrieval.

In addition, a distributed architecture is employed in which storage, indexing, and query processing are deployed across multiple independent nodes. A reactive data integration layer connects structured transaction data with semantic indexes using unique identifiers, ensuring consistent and real-time interaction between components.

Experimental evaluation demonstrates that the proposed architecture significantly improves both system performance and retrieval accuracy compared to a conventional Oracle database-based approach. In a pilot implementation involving one million transaction records, the average query response time was reduced from 10,000 milliseconds to 2,000 milliseconds. Furthermore, the distributed design improved system throughput and scalability under high request loads.

Keywords: Semantic Data Processing, System Scalability, Sensitive Data, Transactions.

I. INTRODUCTION

In recent years, the exponential growth of data volumes and the increasing sophistication of cyber threats have introduced substantial challenges to banking systems and payment infrastructures, particularly in critical domains such as fraud detection, risk management, and information security assurance [1]. At the same time, modern financial environments require highly efficient and intelligent data processing capabilities that traditional relational database systems are increasingly unable to provide.

Conventional data analysis and retrieval approaches, which are primarily based on rigid relational schemas and keyword-driven search mechanisms, are no longer sufficient to address the evolving requirements of large-scale banking applications. These systems exhibit inherent limitations in scalability, flexibility, and semantic understanding, leading to performance degradation, increased query latency, and reduced effectiveness in handling complex analytical workloads [3], [4]. Furthermore, their inability to support semantic and concept-based retrieval significantly restricts their applicability in advanced financial analytics and decision-support systems.

To overcome these limitations, recent research has increasingly focused on vector-based processing, neural representation learning, and semantic indexing techniques. By transforming structured and unstructured data into dense vector embeddings, these approaches enable more accurate, context-aware, and intelligent information retrieval. Transformer-based architectures such as BERT and attention mechanisms provide the foundational capability for learning contextual representations of textual data [5], while embedding-based retrieval models such as Sentence-BERT and ColBERT further enhance semantic matching accuracy through dense vector similarity modeling [6].

In addition, modern vector database systems and approximate nearest neighbor (ANN) search techniques have emerged as key enablers for large-scale semantic retrieval, allowing efficient indexing and querying of high-dimensional data [7]. These methods are particularly effective in supporting banking transaction analysis and uncovering hidden behavioral patterns within large and complex datasets. Moreover, their deployment within distributed database architectures facilitates parallel processing, workload distribution across multiple nodes, and improved system responsiveness under high transaction loads [8].

The integration of these technologies into financial system architectures not only enhances scalability, performance, and real-time analytical capabilities but also improves system resilience and fault tolerance, enabling the development of next-generation data-driven banking solutions. Nevertheless, the adoption of vector search technologies in banking

environments requires careful consideration of critical constraints such as data security, customer privacy protection, system integrity, and efficient management of distributed infrastructures [9].

Consequently, the primary challenge lies in designing hybrid architectures that effectively combine vector-based semantic processing with secure, scalable, and highly reliable distributed system designs suitable for mission-critical banking applications. Accordingly, this study aims to design and evaluate secure and distributed vector search architecture for the transaction ecosystem of Tejarat Bank. The proposed framework introduces a hybrid architecture comprising a secure storage layer, semantic data processing mechanisms, vector indexing, and concept-based information retrieval. Within this architecture, storage and processing components are deployed as independent and scalable services to support horizontal scaling, workload distribution, and sustained performance under high transaction volumes. In addition, the study investigates the operational feasibility, implementation considerations, and performance characteristics of the proposed architecture within a real-world banking environment.

Recent studies indicate that approaches based on large language models demonstrate greater flexibility and accuracy in extracting information from unstructured data compared with traditional text processing methods. However, the practical deployment of these models requires the use of quality control mechanisms, output standardization, and data validation procedures to ensure the reliability of the system in real operational environments [10].

This paper is organized into four main sections to present the design rationale, architectural components, and distributed deployment mechanisms of the proposed system. First, the conceptual framework and the primary research problem are introduced and discussed. Next, the relevant literature and related studies are reviewed to establish the position and contribution of the present research within the existing body of knowledge. Subsequently, the proposed architecture and the interactions among its components within a scalable and distributed environment are described in detail. Finally, the performance of the proposed system is evaluated through a comprehensive assessment of its operational and architectural characteristics.

II. PROBLEM STATEMENT

In banking transaction systems, the massive volume of data, heterogeneity of information structures, and the requirement for rapid and accurate information retrieval expose fundamental limitations in traditional relational database management systems. Conventional approaches based on keyword matching and lexical search are inherently limited in their ability to capture the semantic relationships embedded within textual and transactional data. As a result, they are unable to support concept-based or meaning-aware retrieval, which is increasingly required in modern intelligent information systems.

Consequently, when applied to large-scale banking datasets, these methods frequently suffer from degraded performance, increased query latency, and reduced retrieval accuracy under high-throughput conditions. Recent advances in deep learning–based natural language processing, particularly transformer architectures such as BERT and self-attention mechanisms, have demonstrated strong capabilities in learning contextual representations of text [5]. Furthermore, embedding-based retrieval models such as Sentence-BERT and ColBERT have shown significant improvements in semantic search tasks by enabling dense vector representations that better capture contextual similarity [6],

In addition, modern retrieval systems increasingly adopt approximate nearest neighbor (ANN) search and vector database technologies to efficiently handle high-dimensional embeddings at scale, addressing the limitations of traditional indexing methods in large-scale environments [7]. These developments collectively highlight the need for hybrid architectures that move beyond lexical search toward semantic, vector-based retrieval systems capable of supporting scalable and high-performance banking applications. Furthermore, the concentration of storage and processing resources within centralized infrastructures introduces significant limitations in terms of scalability, fault tolerance, and the ability to efficiently handle increasing volumes of concurrent user requests. These challenges are widely recognized in large-scale similarity search and distributed retrieval systems, where traditional architectures struggle to maintain performance under high-dimensional data workloads and growing query demands [5]. Such constraints become particularly critical in modern banking environments, where high availability, low-latency processing, and real-time responsiveness are essential operational requirements.

In addition, inadequate integration of security mechanisms in the storage and processing of sensitive financial data continues to represent a major challenge in distributed and cloud-based system design. Research in distributed systems highlights the importance of incorporating privacy-preserving and secure design principles to ensure robustness in data-intensive environments [6].

Accordingly, the central research problem addressed in this study is the design of a scalable and secure framework capable of providing semantic information retrieval while maintaining data integrity, operational efficiency, and system scalability in large-scale environments. To address this challenge, the study adopts a vectorization-based approach supported by dense embedding techniques and approximate nearest neighbor (ANN) search methods, which have been shown to significantly improve retrieval accuracy and efficiency in open-domain information retrieval systems [8].

Beyond the architectural considerations of distributed systems, the application layer of banking services requires robust predictive modeling to handle financial risks. Recent research has demonstrated the effectiveness of machine learning and evolutionary algorithms in credit risk assessment and customer classification [10]. Integrating such predictive models into a distributed semantic processing architecture is critical to maintain real-time decision-making capabilities while handling high-volume transaction streams.

The proposed architecture integrates semantic vector indexing techniques with a hybrid storage model that combines a relational database system for structured transactional data and a vector search engine for high-dimensional semantic representations. This design approach is consistent with modern vector database systems that support hybrid query processing and large-scale embedding management.

Within the framework, unique identifiers are used to establish logical links between original transactional records and their corresponding vector embeddings, ensuring data consistency while enabling efficient semantic retrieval. Furthermore, the storage, indexing, and query-processing components are designed to operate in distributed environments using scalable approximate nearest neighbor search structures, which are essential for maintaining performance in high-dimensional spaces [11].

This architectural design enables workload distribution, improves system availability, and supports horizontal scalability, aligning with recent advances in distributed vector search systems that emphasize both performance optimization and accuracy preservation. Consequently, the proposed framework establishes a foundation for secure, scalable, and semantically enriched information retrieval systems suitable for large-scale banking transaction ecosystems.

III. LITERATURE REVIEW

With the rapid growth of data volumes and the increasing demand for intelligent information analysis and retrieval in financial systems, researchers and practitioners have devoted considerable attention to improving data search mechanisms, accelerating information retrieval, enhancing the security of sensitive data, and developing scalable architectures. A review of the existing literature indicates that traditional relational database–based approaches face significant limitations in meeting the processing requirements of large-scale systems. Consequently, distributed architectures, vector search engines, and parallel processing mechanisms have emerged as promising alternatives for addressing these challenges.

Recent studies have expanded their focus beyond retrieval accuracy and search quality to encompass critical system-level requirements, including workload distribution, fault tolerance, high availability, and horizontal scalability. As a result, architectures based on the separation of processing responsibilities and the distributed deployment of storage and search components have become a dominant paradigm in the design of modern data analytics and information retrieval systems.

The following section reviews the most significant contributions in the areas of semantic search, vector-based data processing, information security, and distributed architectures. The objective is to establish the theoretical foundations of the present study, critically analyze relevant findings, and position the proposed research within the broader body of existing work.

A dynamic partitioning approach has been proposed to address access control challenges in vector databases by leveraging a role-based access control (RBAC) model to organize vectors into overlapping partitions, thereby balancing the reduction of data redundancy with the improvement of search efficiency. Experimental evaluations demonstrated that the proposed method achieved up to a six fold improvement in search performance compared with conventional row-level security mechanisms [11].

Recent research has explored the integration of vector search capabilities into graph database architectures [12]. By incorporating embedding representations directly into graph nodes, these frameworks enable the execution of hybrid graph-vector queries, allowing for the simultaneous processing of complex graph traversals and vector similarity searches. This synergy is particularly valuable in advanced Retrieval-Augmented Generation (RAG) applications and context-aware information retrieval scenarios, as it effectively bridges the gap between structural relational reasoning and unstructured semantic analysis.

Under a two-stage Approximate Nearest Neighbor (ANN) search algorithm, a fast filtering phase is followed by a refinement stage, and an early-termination mechanism is applied to significantly improve system performance under heavy workloads. Additionally, a distributed architecture is introduced to enable efficient scalability across multi-node environments, thereby enhancing both throughput and resource utilization [13].

A novel hybrid framework for fake news detection has been proposed by integrating multiple embedding models and incorporating a FAISS-based retrieval layer directly into a deep neural network architecture, where vector search outputs are utilized as input features for the learning model. In addition, the use of an attention mechanism to prioritize retrieved vectors and combine them with indexed search results substantially improved predictive accuracy and overall model performance [14].

To address scalability challenges in large-scale vector search systems, a hybrid partitioning strategy combining dimension-based and vector-based partitioning techniques has been proposed to maintain load balancing across distributed nodes while minimizing inter-node communication overhead. Experimental results showed that this approach achieved performance gains of up to 4.63 times compared with conventional distributed vector search solutions [15].

A comprehensive overview of the FAISS library has been presented through an examination of its internal architecture, compression techniques, and accuracy–speed trade-offs, and approximate search algorithms. The study also provided an extensive performance analysis and demonstrated the applicability of FAISS in large-scale similarity search and vector retrieval systems, establishing it as one of the most influential frameworks in high-dimensional vector search [16].

The integration of vector search capabilities with graph database systems has attracted significant attention through the development of frameworks that extend graph databases with embedding-aware data structures and massively parallel vector indexing mechanisms. Such systems enable hybrid graph-vector queries, allowing seamless integration of structured and unstructured data, while experimental evaluations have demonstrated superior scalability and performance compared with both traditional graph databases and specialized vector search engines, particularly in Retrieval-Augmented Generation (RAG) applications [17].

Scalability remains one of the primary challenges in billion-scale vector retrieval systems. A comprehensive experimental evaluation of state-of-the-art graph-based vector search algorithms was conducted across datasets containing up to one billion vectors, identifying key design principles that affect search quality, scalability, and computational efficiency, and providing valuable insights for the development of large-scale distributed retrieval systems [18].

A foundational contribution to high-performance vector search was introduced by Johnson et al. through the FAISS framework. Their work presented optimized GPU-based algorithms for similarity search and nearest-neighbor retrieval, achieving substantial performance improvements over previous approaches. FAISS remains one of the most widely adopted platforms for large-scale vector indexing and semantic retrieval applications [19]. From a security perspective, recent studies have emphasized the necessity of embedding access-control policies directly into semantic retrieval pipelines.

An access-controlled semantic search framework has been proposed in which authorization metadata are integrated into vector database operations, enabling the simultaneous execution of access-control decisions and semantic similarity retrieval. This approach reduces security risks in enterprise-scale retrieval systems and Retrieval-Augmented Generation (RAG) deployments [20].

IV. METHODOLOGY

The proposed architecture is designed to integrate the reliability of structured data storage with the efficiency of semantic retrieval and distributed processing capabilities. Financial and textual data are automatically extracted and loaded through a scheduled Extract–Transform–Load (ETL) pipeline and subsequently distributed into two complementary layers: structured storage within the Oracle relational database and semantic vector indexing within the FAISS search engine. To support large-scale operational workloads, the indexing and retrieval services are designed for deployment across multiple processing nodes, enabling workload distribution and parallel execution of search operations. The response layer aggregates, ranks, and combines the outputs of both vector-based and text-based retrieval mechanisms to generate the final search results. This design not only preserves data integrity and freshness but also significantly enhances retrieval quality and semantic relevance.

The FAISS library provides highly efficient similarity search and clustering algorithms for dense vector representations at billion-scale. It enables scalable approximate nearest neighbor (ANN) search through optimized indexing structures and memory-efficient representations, making it suitable for large-scale semantic retrieval systems [6].

Beyond improving search performance, the proposed architecture increases system resilience and availability. The separation of responsibilities across architectural layers facilitates workload distribution and reduces resource contention within individual components. Furthermore, the high-performance vector retrieval capabilities of FAISS enable near real-time query processing, even under heavy operational loads. The autonomous operation of each architectural layer enhances fault tolerance by allowing independent recovery, maintenance, and scaling of individual services. Moreover, the distributed deployment of indexing and search nodes supports horizontal scalability and minimizes dependence on a single processing point, thereby eliminating potential bottlenecks.

The Oracle infrastructure employs replication and clustering mechanisms to ensure data consistency, high availability, and operational continuity. Combined with distributed vector search services, this architecture enables efficient management of large-scale transaction records and textual data under continuously growing workloads. Such capabilities are particularly important in centralized banking environments, where millions of requests must be processed daily while maintaining service quality and operational stability.

From a performance perspective, the computational complexity of traditional text retrieval in Oracle can be approximated as:

$$\boldsymbol{T_{Oracle} = K.N} \qquad (1)$$

where (N) denotes the number of transaction records and (K) represents the average processing cost per record.

In contrast, the approximate retrieval complexity of the FAISS vector search engine can be expressed as:

$$\boldsymbol{T_{FAISS} \approx c.log(N)} \qquad (2)$$

where (c) is a constant dependent on the indexing structure and retrieval configuration.

In a distributed environment, partitioning data and computational workloads across multiple nodes reduces the effective response time and increases concurrent request-processing capacity. This improvement is achieved through resource distribution without requiring modifications to the underlying retrieval algorithms.

From a cybersecurity perspective, sensitive transaction data remain stored within the Oracle database, whereas only vectorized representations are utilized within the semantic retrieval layer. The overall security risk can be modeled as

$$\boldsymbol{R = R_{Oracle}.(1-\alpha) + R_{FIASS}.\alpha} \qquad (3)$$

where $\boldsymbol{\alpha \ll 1}$ represents the limited exposure of raw sensitive information within the vector-search layer.

the overall risk decreases as (R) approaches lower values, indicating improved system security. Furthermore, the separation of processing layers and the distributed deployment of retrieval services reduce direct exposure to sensitive information and enable centralized access-control management.

To combine the outputs of Oracle-based textual retrieval and FAISS-based semantic retrieval, a weighted fusion model is employed:

$$\boldsymbol{final}_{\boldsymbol{score}_i} = \boldsymbol{w_1 score}_{\boldsymbol{fiass}_i} + \boldsymbol{w_2 . score}_{\boldsymbol{oracle}_i} \quad (4)$$

Where $w_1+w_2=1$ and (w1) and (w2) denote the relative contributions of semantic and textual retrieval scores, respectively.

For semantic retrieval, cosine similarity is used to measure the semantic proximity between a query vector and indexed document vectors:

$$\boldsymbol{Score}_{\boldsymbol{FAISS}}(\boldsymbol{d_i}, \boldsymbol{q}) = \frac{\vec{q}.\vec{d_i}}{\|\vec{q}\|.\|\vec{d_i}\|} \quad (5)$$

where $\vec{q}$ and $\vec{d_i}$ represent the vector embeddings of the query and document, respectively.

For textual retrieval, the TF–IDF weighting scheme is employed:

$$\boldsymbol{Score}_{\boldsymbol{Oracle}}(\boldsymbol{d_i}, \boldsymbol{q}) = \sum_{t \in q} \boldsymbol{TF}(\boldsymbol{t}, \boldsymbol{d_i}) \times \boldsymbol{log}\left(\frac{\boldsymbol{N}}{\boldsymbol{df_t}}\right) \quad (6)$$

where TF(t,di) denotes the term frequency of term (t) in document (di), and (dft) represents the document frequency of term (t).

To evaluate retrieval effectiveness, the Mean Reciprocal Rank (MRR) metric is adopted:

$$MRR = \frac{1}{|Q|}\sum_{i=1}^{|Q|}\frac{1}{rank_i} \quad (7)$$

Where (Q) denotes the set of evaluation queries and $Rank_i$ represents the rank position of the first relevant result if no relevant result is retrieved for a particular query, the reciprocal rank associated with that query is assigned a value of zero. This metric effectively evaluates the ability of a retrieval system to present relevant results at higher ranking positions. Consequently, an MRR value closer to one indicates superior retrieval performance and greater effectiveness in returning relevant results among the top-ranked outputs [21].

Retrieval-augmented approaches combine neural language models with external information retrieval mechanisms to access relevant knowledge during processing, which can improve the handling of knowledge-intensive NLP tasks [22]. Distributed banking systems require secure and scalable data management mechanisms to ensure confidentiality, integrity, and reliable processing of financial transactions.

Overall, the proposed architecture establishes a secure, scalable, and high-performance framework for enterprise-scale information retrieval by integrating semantic vector search and traditional text retrieval within a distributed environment. The combination of TF–IDF-based ranking and cosine-similarity-based semantic matching enhances retrieval precision and user experience. Moreover, workload distribution, horizontal scalability, elimination of processing bottlenecks, and improved fault tolerance enable the system to maintain stable performance under continuously increasing data volumes and heavy operational workloads.

In addition, strict access-control policies, separation of processing layers, and reduced exposure to raw sensitive data significantly strengthen the security posture of the platform and facilitate its deployment within mission-critical banking environments.

Consequently, the proposed architecture achieves an effective balance among performance, security, scalability, and intelligence in the management of large-scale organizational data assets.

V. PROPOSED MODEL

The integration of traditional database infrastructures with advanced natural language processing technologies has become a fundamental requirement for the design of enterprise-scale information systems. In this context, the proposed architecture of this study presents a hybrid model combining the Oracle relational database with the FAISS vector search engine. The primary objective is to provide a secure, scalable, distributed, and intelligent framework for the storage and semantic retrieval of information. Within this architecture, Oracle functions as the core platform for structured and secure data storage, while FAISS leverages vector representations derived from textual embeddings to enable rapid and conceptually meaningful search.

Data are processed automatically through an ETL pipeline and routed along two parallel paths: structured data storage and vector embedding generation. To enhance security,

sensitive information is retained within the database layer, and logical links between the original data and their corresponding vector representations are maintained using unique identifiers. During information retrieval, users can access relevant results through a single text-based query via an intelligent API. Semantic search is first performed on the vector indices, followed by retrieval of the original records from the database, which are decrypted if necessary.

To handle high data volumes and concurrent requests, the processing and search components are deployable across

distributed environments. In this configuration, storage, indexing, and search workloads are distributed among multiple processing nodes and coordinated through queuing and task management mechanisms such as Redis Queue. This approach mitigates processing bottlenecks, supports horizontal scalability, enables workload distribution, improves availability, and enhances fault tolerance.

The architecture is explicitly designed to combine the storage robustness of traditional relational databases with the capabilities of semantic search. It operates along two primary pathways: one for secure and structured data management and another for vector-based semantic processing and retrieval. Coordination between these components is facilitated by a

unified service layer that ensures effective interaction across the system.

In the proposed framework, the combination of relational database capacity and vector search capability provides an efficient solution for semantic information retrieval. As illustrated in Figure I, Oracle is employed for secure structured data storage, while the FAISS vector search engine, utilizing the HNSW indexing structure, enables fast and accurate information retrieval. Moreover, the distributed deployment of search and processing components allows the system to manage large-scale transactional data and respond to millions of daily requests, making it fully operational and suitable for deployment in enterprise and banking environments.

The Oracle database serves as a persistence layer responsible for ensuring data integrity, security, and applications such as semantic search engines, question-answering systems, and large-scale text analytics.

The data ingestion, preprocessing, and storage workflow is managed through a structured ETL pipeline implemented using Oracle Data Integrator (ODI) version 14. As one of Oracle's advanced data integration solutions, ODI enables efficient design, scheduling, and execution of extraction, transformation, and loading processes. After ingestion and cleansing, data flows through two parallel pipelines: in the first, textual data is forwarded to an embedding service and indexed in FAISS using the HNSW structure together with a unique document identifier; in the second, sensitive data is encrypted using the AES algorithm and stored in Oracle tables. At this stage, the document identifier acts as a linkage key, ensuring consistency between encrypted records and vectorized representations.

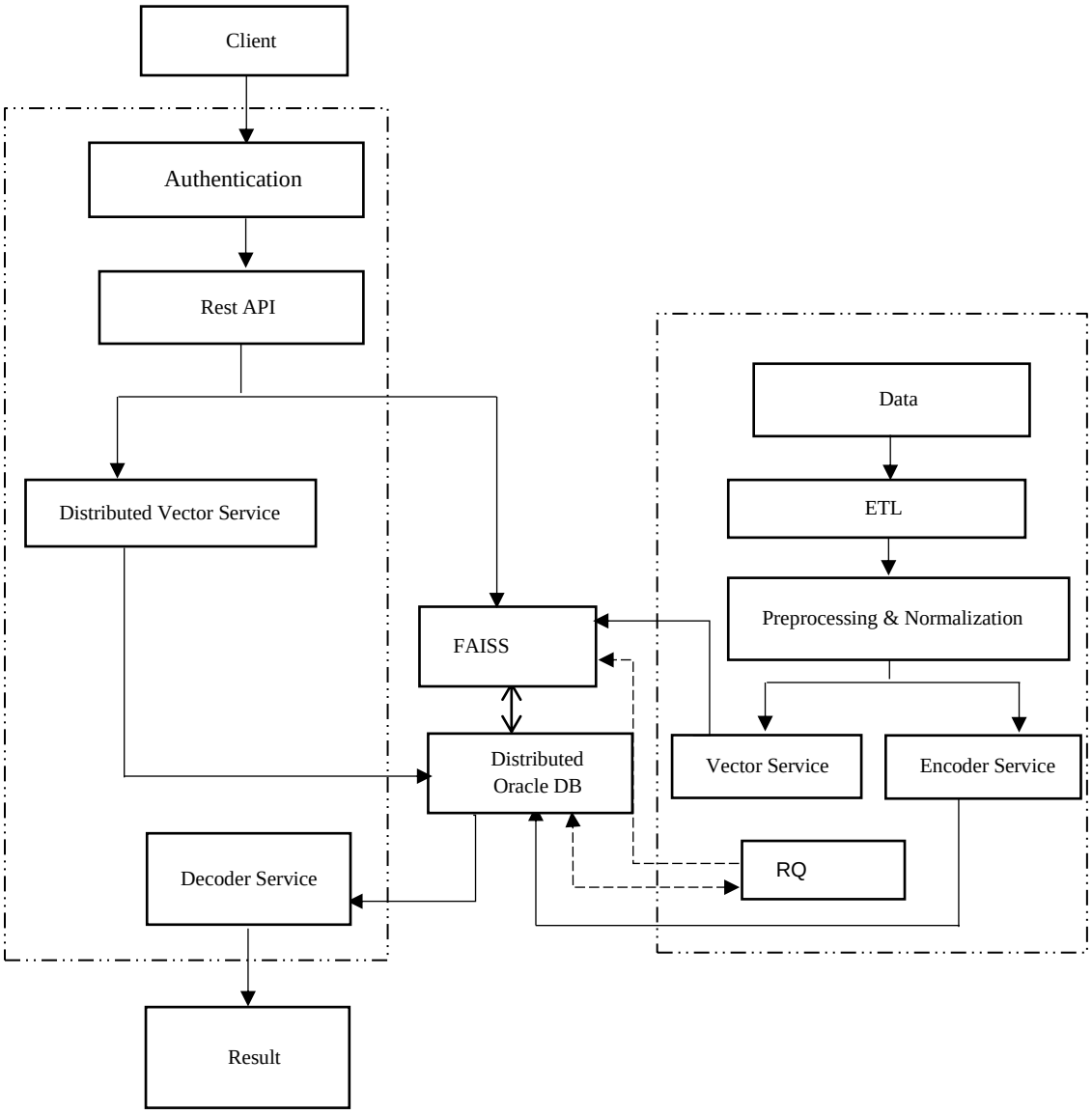

Fig. I: Proposed Hybrid Architecture

The embedding service is responsible for converting textual inputs into dense numerical representations, enabling semantic search and content similarity computation. For simplicity, efficiency, and CPU-based execution, a lightweight multilingual model, paraphrase-multilingual-MiniLM-L12-v2 from the Sentence Transformers framework, is employed, which demonstrates strong performance for the Persian language. The service preprocesses the required text fields and generates 384-dimensional embeddings. These embeddings are subsequently used for building and updating the FAISS index and enabling fast semantic retrieval. The service is designed as an independent and scalable component capable of handling concurrent requests in CPU-only environments.

During the retrieval phase, user queries are submitted through a .NET 9-based API. The query is forwarded to the embedding service, where its vector representation is generated. This vector is then passed to FAISS, where approximate nearest neighbor search is performed using the HNSW index. The identifiers of retrieved documents are used to fetch the corresponding records from Oracle. After decryption, the final response is returned to the user.

This architecture maintains operational independence between Oracle and FAISS while enabling synchronization between them. Any updates in Oracle data (e.g., modifications or deletions) must be reflected in the HNSW index. This synchronization is handled by a coordination service based on Redis Queue (version 2.6), preventing inconsistencies between database records and vector indexes.

Overall, the proposed hybrid architecture leverages the secure and persistent storage capabilities of Oracle alongside the high-performance vector processing capabilities of FAISS with HNSW, achieving a balanced trade-off between security, efficiency, and intelligent information retrieval. This design is highly scalable and suitable for complex enterprise applications such as semantic search engines, question-answering systems, and large-scale text analytics.

Web services and business logic are implemented using C# and the .NET 9 framework, ensuring strong integration with Oracle and vector-based services. In the vector processing layer, Python (version 3.1) is employed to leverage advanced machine learning and natural language processing libraries. Inter-service communication is handled via APIs, while PL/SQL in Oracle 19c is used for data-layer logic management.

To ensure operational stability and horizontal scalability under heavy workloads or failure scenarios, critical architectural components—including APIs and embedding services—are reinforced with load balancing mechanisms. Tools such as NGINX or HAProxy are used to distribute incoming requests across multiple service instances, preventing processing bottlenecks.

At the storage layer, failover mechanisms are implemented using Oracle Data Guard to enable automatic and uninterrupted recovery in case of primary database failure. Additionally, to enhance availability and fault tolerance in intermediate layers, replication is applied in Redis and read-only FAISS deployments, improving system performance and reducing retrieval latency in critical scenarios.

Finally, Prometheus and Grafana are employed for comprehensive system monitoring, enabling metric collection and analytical dashboards. This monitoring infrastructure supports real-time fault detection, automated alerting, and immediate corrective actions, playing a critical role in maintaining service quality and operational reliability.

VI. Comparative Analysis of Results

In comparison with the traditional Oracle-based database architecture, the proposed architecture demonstrates a significant improvement in response times under high data volumes. The most notable enhancement is observed in search latency: as the number of records increases, the traditional architecture experiences a substantial growth in processing time, whereas the proposed architecture, leveraging the FAISS vector search engine and the HNSW structure, maintains a more stable and scalable performance. Furthermore, the distributed deployment of search components allows computational load to be balanced across multiple nodes, enabling the system to sustain optimal performance even under increased data volumes and concurrent request rates.

From the perspective of semantic search accuracy, the Mean Reciprocal Rank (MRR) for the proposed architecture ranges between 0.90 and 0.95, compared to an average of approximately 0.60 for the traditional architecture. This improvement underscores the system’s enhanced capability for semantic information retrieval, which is particularly valuable in applications such as text analytics, question answering, semantic search, and knowledge extraction from transactional data.

However, the proposed architecture incurs higher resource requirements during the preprocessing and data ingestion stages due to vectorization, encryption, and semantic indexing processes. These computational costs primarily occur in offline or scheduled operations and have minimal impact on the real-time user experience.

Another significant advantage of the proposed architecture is its scalability and flexibility. Its service-oriented design and the use of independent APIs enable the integration of new capabilities—such as AI models, intelligent question-answering systems, natural language processing modules, and advanced data analytics tools—without requiring major changes to the underlying infrastructure. In contrast, adding such capabilities to traditional centralized database architectures typically encounters more technical limitations.

From a system architecture standpoint, the distributed deployment of processing and search components enhances fault tolerance, availability, and horizontal scalability. In this setup, the failure of a single processing node does not necessarily halt service provision, as other nodes can continue to operate. This feature is particularly critical in banking and

mission-critical environments that demand continuous service availability.

Table I presents a comparison of key performance indicators between the traditional Oracle-based architecture and the proposed architecture.

Overall, the evaluation results indicate that the proposed architecture not only outperforms the traditional model in technical performance—encompassing search speed, information retrieval quality, and semantic accuracy—but also offers significant advantages in scalability, availability, fault tolerance, and readiness for modern AI- and NLP-based applications. These findings suggest that the proposed architecture represents a robust solution for intelligent management and retrieval of large-scale data in banking and enterprise information systems.

In Table I, the performance of the two architectures is evaluated and compared based on key performance indicators. The results indicate that the proposed hybrid architecture, which simultaneously leverages the Oracle database for secure and structured data storage and the FAISS vector search engine for semantic information retrieval, outperforms the traditional Oracle-based architecture. In an evaluation conducted on a dataset comprising one million transactional records, the average search response time in the proposed architecture was measured at approximately 2000 milliseconds, demonstrating a substantial improvement over the traditional architecture.

Beyond response time improvements, the proposed architecture also demonstrates strong performance in semantic search. The Mean Reciprocal Rank (MRR) was observed in the range of 0.90 to 0.95, compared to approximately 0.60 in the traditional architecture, indicating the system's enhanced ability to retrieve results that are relevant and semantically aligned with user queries.

Although the proposed architecture requires additional resources such as RAM and CPU capacity during the ETL stages due to vectorization, embedding generation, data encryption, and semantic indexing, these computational costs primarily occur in offline and scheduled processes, and do not significantly impact real-time user response times. The improvements in retrieval accuracy, reduced search latency, and system extensibility justify these additional processing requirements.

One of the most important advantages of the proposed architecture is its high scalability and modular design. Its service-oriented structure and independent APIs allow the integration of new capabilities including natural language processing (NLP) modules, AI models, recommendation engines, and data analytics tools without requiring major modifications to the underlying infrastructure. This feature provides greater flexibility and adaptability for future developments compared to traditional centralized database architectures.

Furthermore, the distributed deployment of search and processing components enhances horizontal scalability, balanced computational load distribution, availability, and system fault tolerance. These characteristics ensure stable performance under high volumes of concurrent requests and continuous data growth, facilitating deployment in operational banking environments.

To accurately measure performance and monitor system behavior, a comprehensive set of specialized tools was employed across infrastructure, software, and data layers. Prometheus was used as a metrics collection system to monitor CPU and memory usage, disk I/O, request rates, response latency, and the status of processing nodes. The collected data was visualized through Grafana, enabling trend analysis and performance comparisons over time.

At the software level, JetBrains dotTrace was utilized for profiling and performance analysis of .NET-based services, allowing identification of processing bottlenecks, function execution times, and resource consumption across different system components. To evaluate the FAISS vector search engine, its internal tools were employed to measure retrieval time, search success rate, result accuracy, and vector matching quality.

For system event and log analysis, Splunk was used to provide real-time search, analysis, and visualization of log data, enabling monitoring, error detection, and operational behavior analysis. At the database level, Oracle Enterprise Manager (OEM) was employed to monitor database health, evaluate transactional load, analyze SQL query performance, manage resources, and assess the data infrastructure.

The combination of these monitoring and evaluation tools provided a comprehensive view of system performance across multiple layers, enabling precise measurement of efficiency, scalability, availability, and information retrieval quality. The results of these evaluations indicate that the proposed architecture not only improves search speed and semantic accuracy but also offers significant advantages in extensibility, operational stability, and readiness for AI-based applications compared to the traditional architecture.

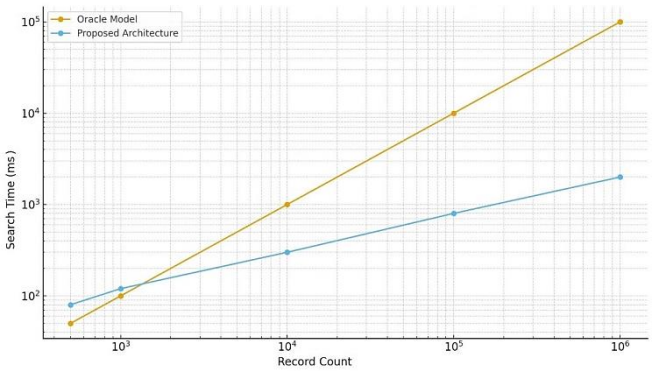

Fig. II: Response Time in the Traditional and Proposed Models

Figure II presents a comparison of text-based search latency as a function of dataset size for two different architectures: a traditional Oracle relational database architecture and the proposed hybrid architecture integrating Oracle with the FAISS vector search engine.

As illustrated in the figure, search response time in the conventional architecture increases in an approximately linear manner as the data volume grows. Consequently,

search operations become increasingly expensive at larger scales, resulting in a substantial degradation in retrieval performance. In contrast, the proposed architecture exhibits significantly better scalability characteristics by leveraging optimized vector-search structures, particularly the

Hierarchical Navigable Small World (HNSW) index implemented within FAISS. As a result, the growth rate of search latency remains considerably lower and more stable across increasing dataset sizes.

The observed divergence in search-time growth rates highlights the efficiency gains achieved through the proposed architecture, especially in large-scale data environments. Experimental results indicate that for a dataset containing one million records, the search response time of the traditional architecture approaches approximately 10^5 milliseconds. In comparison, the proposed architecture maintains a substantially lower response time of approximately 2×10^3 milliseconds. This significant performance gap demonstrates the effectiveness of vector indexing and approximate nearest-neighbor search techniques in reducing computational complexity and accelerating information retrieval.

The results further suggest that the proposed architecture can sustain efficient search performance under large-scale workloads, making it particularly suitable for enterprise and banking environments characterized by massive transactional datasets and high query throughput requirements.

VII. CONCLUSIONS

The proposed hybrid architecture, leveraging both the Oracle database and the FAISS vector search engine, provides an efficient approach to enhancing performance, accuracy, and scalability in information retrieval systems. While maintaining data security and integrity, this architecture significantly reduces response times for both textual and semantic queries and substantially improves the quality of information retrieval. The separation of responsibilities between storage and search layers, combined with the use of optimized algorithms such as HNSW, increases system resilience under heavy workloads and enhances overall scalability.

Although computational costs are higher during preprocessing and vector representation generation, these costs are primarily incurred in offline and asynchronous operations, having negligible impact on online query response times and being justified by the overall performance gains. Furthermore, the modular and service-oriented design of the proposed architecture enables seamless integration with artificial intelligence and natural language processing systems, providing a foundation for future development of more advanced capabilities. From a security perspective, the use of sensitive data encryption and access-level controls strengthens organizational data protection.

Overall, by balancing performance, security, and extensibility, the proposed architecture establishes a robust foundation for deploying intelligent information analysis and retrieval systems at the enterprise scale, capable of addressing both current and future requirements simultaneously.

Table I. Presents a comparative evaluation of the two architectures based on key performance indicators

Performance Metric / Feature	**Traditional Model**	**Proposed Architecture**	**Differences**
Average search time on 1 million records	10,000 ms	2,000 ms	Measured under identical query load
Semantic search accuracy (MRR)	0.55 – 0.65	0.90 – 0.95	Scale: 0–1, based on Mean Reciprocal Rank evaluation
RAM usage	6 GB	10 GB	Higher usage due to vector embeddings and indexing
CPU utilization during ETL	70%	87%	Particularly during ETL and embedding generation processes
API / Modular extensibility	Limited	High	Scalable for NLP and AI requirements

VIII. References

[1] P. Lewis et al., “Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks,” NeurIPS, 2020.

[2] H. Rahmani, A. Amiri, and N. Aghaei Meybodi, “Automatic software requirements extraction from natural language texts using natural language processing and large language models,” Journal of Distributed Computing and Distributed Systems, vol. 8, no. 2, pp. 174–184, 2025.

[3] Y. Zhang, L. Chen, and X. Wu, “Hybrid dimension- and vector-based partitioning for scalable distributed vector search,” IEEE Transactions on Knowledge and Data Engineering, 2025.

[4] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones,A. N. Gomez, L. Kaiser, and I. Polosukhin, “Attention is all you need,” in Proc. Advances in Neural Information Processing Systems (NeurIPS), 2017, pp. 5998–6008.

[5] N. Reimers and I. Gurevych, “Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks,” EMNLP, 2019.

[6] O. Khattab and M. Zaharia, “ColBERT: Efficient and Effective Passage Search via Contextualized Late Interaction over BERT,” Proc. SIGIR, 2020.

[7] J. Zhao, G. Zhang, S. Li, and L. Wang, “Towards efficient and scalable vector similarity search on distributed

systems," Journal of Systems Architecture, vol. 121, p. 102315, 2021.

[8] S. Liu, Z. Zeng, L. Chen, A. Ainihaer, A. Ramasami, S. Chen, Y. Xu, M. Wu, and J. Wang, "TigerVector: Supporting Vector Search in Graph Databases for Advanced RAGs," arXiv:2501.11216, Jan. 2025.

[9] H. Ferhatosmanoglu, E. Tuncel, D. Agrawal, and A. El Abbadi, "High dimensional nearest neighbor searching," Information Systems, vol. 31, no. 6, pp. 512–540, 2006.

[10] S. Karimlou Sayyah, "Banking customer credit risk: Proposing machine learning and evolutionary algorithm-based models for credit risk prediction and assessment for banking customer classification," Journal of Distributed Computing and Distributed Systems, vol. 8, no. 2, pp. 69–87, 2025.

[11] X. Zhong, Y. Liang, H. Chen, and J. Wang, "HoneyBee: Dynamic partitioning for role-based access control in vector databases," Proc. VLDB Endowment, vol. 18, no. 4, 2025.

[12] Z. Liu, H. Wang, Q. Zhang, and Y. Sun, "TigerVector: Embedding-aware graph databases with parallel vector indexing," Proc. VLDB Endowment, vol. 18, no. 3, 2025.

[13] M. Aumüller, E. Bernhardsson, and A. Faithfull, "ANN-Benchmarks: A benchmarking tool for approximate nearest neighbor algorithms, "Information Systems, vol. 87, pp. 101–118, 2020

[14] S. S. Monir and D. Zhao, "NexusIndex: Integrating advanced vector indexing and multi-model embeddings for robust fake news detection," arXiv preprint arXiv:2410.18294, 2024.

[15] Q. Xu, F. Zhang, C. Li, L. Cao, and Z. Chen, "HARMONY: A Scalable Distributed Vector Database for High-Throughput Approximate Nearest Neighbor Search," 2025.

[16] M. Douze, J. Johnson, H. Jégou, and L. Matuszyk, "The Faiss library," arXiv preprint arXiv:2401.08281, 2024.

[17] J. Johnson, M. Douze, and H. Jégou, "Billion-scale similarity search with GPUs," IEEE Transactions on Big Data, vol. 7, no. 3, pp. 535–547, 2017.

[18] R. Roman, J. Zhou, and J. Lopez, "On the features and challenges of security and privacy in distributed systems," IEEE Systems Journal, vol. 7, no. 3, pp. 475–485, 2013.

[19] I. Azizi, K. Echihabi, and T. Palpanas, "Graph-Based Vector Search: An Experimental Evaluation of the State-of-the-Art," arXiv:2502.05575, Feb. 2025.

[20] A. Amanbayev et al., "Filtered approximate nearest neighbor search in vector databases: System design and performance analysis," arXiv preprint arXiv:2602.11443, 2026.

[21] H. Zhong, M. Lentz, N. Narodytska, A. Szekeres, and K. Rong, "HoneyBee: Efficient Role-Based Access Control for Vector Databases via Dynamic Partitioning," arXiv:2505.01538, May 2025.

[22] J. J. Pan, J. Wang, and G. Li, "Survey of vector database management systems," The VLDB Journal, 2024.

How to cite: M. Kalanaki, **A Scalable Distributed Architecture for Semantic Vector-Based Processing of Large-Scale Banking Transactions,** Journal of Distributed Computing and Systems (JDCS), vol 9, No 1, pages 73-82

Moosa Kalanaki holds an M.Sc. in Artificial Intelligence and has over 8 years of experience as a Senior .NET Developer in the financial technology sector. Specialized in designing secure, scalable enterprise architectures, his research focuses on machine learning, generative AI, and LLM fine-tuning. He is actively involved in publishing academic papers and technical articles that bridge the gap between robust backend engineering and advanced, data-driven AI solutions. Email: moses.kalanaki@gmail.com