Design of a Unified Microservice Architecture for AI-based Financial Systems Using XAI and RAG*

Anna Shilinh[1,†,*] and Khrystofor-Yaroslav Khlibyshyn[1,†]

[1] *Lviv Polytechnic National University, 12 S. Bandera str., Lviv, 79013, Ukraine*

Abstract

The transition toward data-centric financial ecosystems necessitates the structural integration of Artificial Intelligence (AI) and Machine Learning (ML) beyond isolated algorithmic applications. While individual ML models have shown significant progress in specific tasks, the complex problem of integrating these heterogeneous components into a unified, scalable, and secure information system remains largely unresolved. Authors of the paper propose a comprehensive microservice architecture for an AI-driven personal financial management system. Unlike existing approaches, the proposed architecture provides a holistic blueprint that systematically combines domain logic, security services, ML modules, and Large Language Models (LLMs). By utilizing a centralized API gateway with independent AI microservices, the system ensures flexibility and scalability. Architectural patterns, such as the "Facade", are implemented to integrate Explainable AI (XAI) methods, effectively addressing the "black box" problem and ensuring regulatory compliance. Furthermore, security is guaranteed through Open Banking standard integration, and the Retrieval-Augmented Generation (RAG) approach minimizes LLM hallucinations. The developed microservice framework reduces design risks and establishes a reliable foundation for building transparent, manageable, and secure intelligent financial applications.

Keywords

personal financial management, microservice architecture, artificial intelligence, explainable AI, retrieval-augmented generation [1]

1. Introduction

Artificial Intelligence (AI) and Machine Learning (ML) technologies are fundamentally restructuring financial operations, shifting from auxiliary analytical tools to core components of modern banking architectures. These technologies have evolved from theoretical concepts into essential tools for enhancing competitiveness and operational efficiency [1, 2]. The application of AI in financial systems allows for the resolution of a broad spectrum of tasks: from automating customer service via chatbots and personalizing services [3, 4], to executing complex analytical operations such as credit risk assessment [5], fraud detection [6], and market trend forecasting [7].

Consequently, the development of intelligent information systems for financial management has emerged as a highly relevant research problem. Contemporary studies confirm that such systems significantly improve decision-making accuracy, optimize operational costs, and enhance the overall customer experience [8, 9]. However, despite substantial progress in the development of individual AI models, several complex, system-level challenges remain unresolved. The majority of scientific works focus primarily on the application of specific algorithms (for instance, algorithms designed exclusively for credit scoring [10] or market analysis [11]), but fail to propose a holistic architectural solution for integrating these isolated models into a unified, scalable, and secure information system.

The most critical ongoing challenges include algorithmic bias, data privacy concerns, and the lack of transparency and explainability ("black box" problem) in AI models, which significantly

* *CMIS-2026: The Ninth International Workshop on Computer Modeling and Intelligent Systems, May 5, 2026, Zaporizhzhia, Ukraine*

[1*] Corresponding author.
[†] These authors contributed equally.
anna.y.shilinh@lpnu.ua (A. Shilinh); khrystofor-yaroslav.khlibyshyn.mnsam.2025@lpnu.ua (K.-Y. Khlibyshyn)
0000-0003-1063-3437 (A. Shilinh); 0009-0004-5765-6120 (K.-Y. Khlibyshyn);

complicates their implementation given strict regulatory requirements [12]. Another vital aspect is the necessity of designing a flexible architecture capable of effectively combining centralized management with distributed microservices to execute resource-intensive analytical tasks. Accordingly, there is a clear need to develop a comprehensive approach to the design and construction of a financial management information system. Such a system must systematically address the integration of heterogeneous AI components while ensuring high reliability, security, and manageability.

2. Related Work

Deploying AI within the financial sector necessitates strict alignment between technological innovation and rigorous regulatory and ethical constraints [1]. Comprehensive bibliometric analyses reveal that ML algorithms are actively transforming traditional financial practices, ranging from basic analytics to complex predictive systems [2]. In the realm of personal financial management (PFM), recent research emphasizes the importance of personalized services, where AI-driven behavioral analytics dynamically adapt to customer needs [3]. Furthermore, to enhance user engagement in PFM applications, researchers actively explore gamification strategies that motivate individuals to control their finances effectively [4]. The design and implementation of such personalized finance software increasingly require robust architectural frameworks to handle transaction tracking and expense categorization efficiently [8].

A significant shift in financial technology has been sparked by the adoption of Generative AI (GAI) and Large Language Models (LLMs) [13]. Studies exploring the capabilities of models like ChatGPT demonstrate their proficiency in executing complex financial analyses, data visualization, and autonomous reasoning [4]. These Financial LLMs (FinLLMs) offer unprecedented opportunities for interpreting both structured and unstructured financial data, automating tasks such as sentiment analysis and investment strategy development [7, 11].

However, the inherent risk of hallucination in LLMs has driven the need for advanced prompting and data integration techniques. Retrieval-Augmented Generation (RAG) has emerged as a crucial approach to grounding LLM outputs in factual, domain-specific financial knowledge [14]. Evaluating RAG models in the context of financial report analysis confirms that querying external, reliable databases prior to the generation phase significantly improves answer faithfulness and context relevance [15].

Despite these technological advancements, the "black box" nature of deep learning models remains a fundamental barrier to their deployment in highly regulated banking environments [12]. Explainable AI (XAI) addresses this challenge by providing human-understandable justifications for AI-generated outputs. Recent studies demonstrate the successful application of automated machine learning combined with SHAP (SHapley Additive exPlanations) values to enhance transparency in credit decision-making [5] and auto loan default predictions [10]. Similarly, the integration of XAI in fraud detection systems bridges the gap between algorithmic clarity and stakeholder confidence, ensuring regulatory compliance [6].

Finally, integrating these diverse AI, RAG, and XAI components requires a scalable infrastructure. The transition from monolithic systems to microservice architectures allows financial institutions to achieve the necessary agility, fault isolation, and independent deployment of specialized AI modules [16, 17]. Security in such distributed environments, particularly when integrating external bank APIs under the Open Banking (PSD2) directive, is paramount. Architectural models separating business logic from integration layers have proven effective in mitigating web application security threats [18]. To address the lack of a unified blueprint in existing literature, this paper aims to design a comprehensive, scalable microservice architecture that seamlessly integrates LLMs, RAG, XAI, and Open Banking standards for personal finance management.

3. Proposed System Architecture

The design of the proposed personal financial management system, tentatively named "FinAI", is based on the methodology of object-oriented analysis and design using the Unified Modeling Language (UML). This approach allows for a comprehensive, multi-perspective formalization of the system, covering its functional, static, dynamic, and physical dimensions [8]. The following subsections detail the core architectural decisions and system components.

3.1. Functional Model

To determine the functional capabilities of the system and formalize the scenarios of user interaction, a Use Case model was developed. This model visually demonstrates the complete lifecycle of a user's engagement with the system – from initial authentication to receiving complex, AI-driven financial analytics.

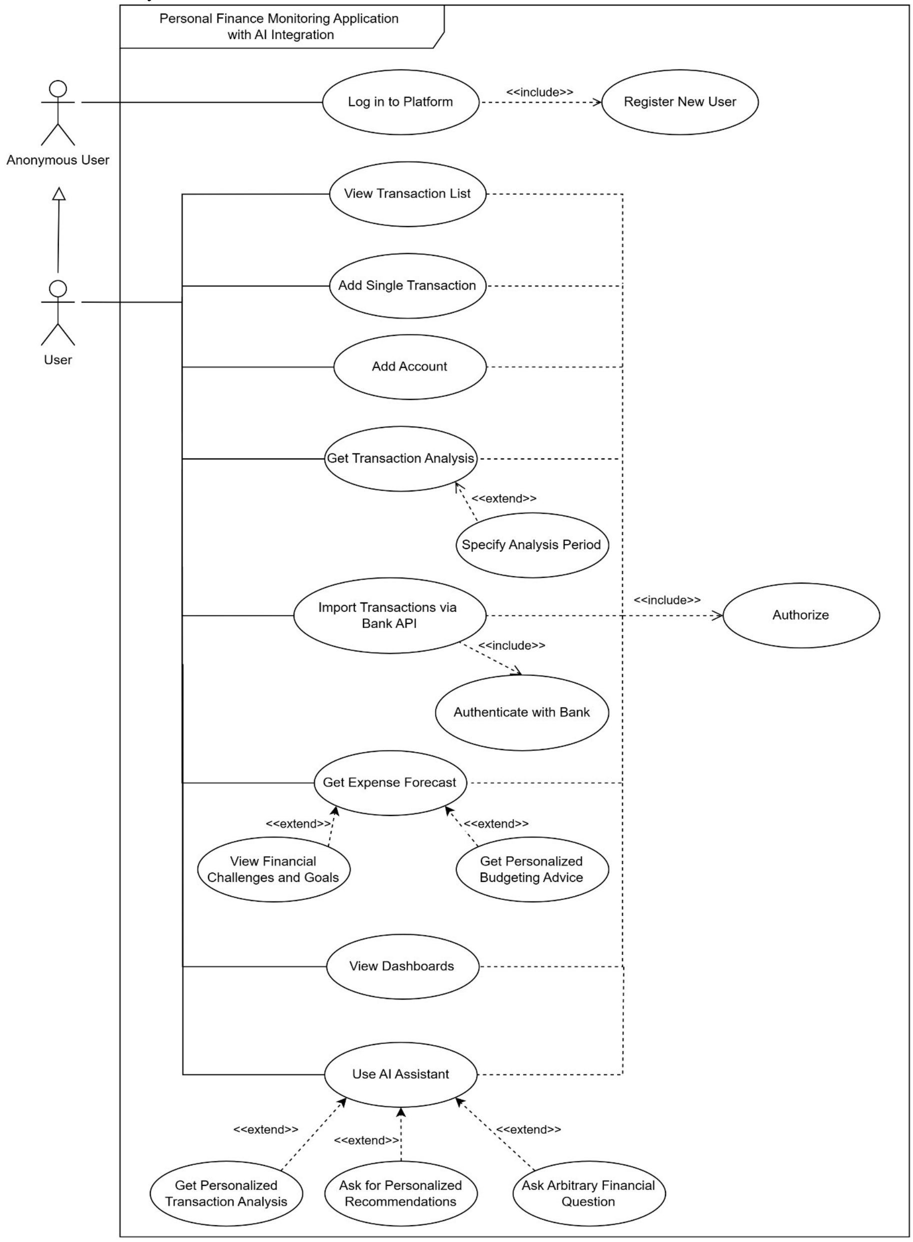

Figure 1: Use case diagram of the proposed FinAI system

As illustrated in Figure 1, the system defines two primary actors: the Anonymous User and the registered User. The Anonymous User has restricted access and can only execute baseline operations, such as registering a new account or logging into the platform. Once authenticated, the fully privileged User gains access to the entire spectrum of functional capabilities. To ensure modularity and clarity, the core use cases are grouped into four logical blocks:

- **System Access and Security**: This block includes baseline scenarios like "Log in to Platform" and "Register New User". From an architectural perspective, the "Authorize" and "Authenticate with Bank" use cases are of critical importance. They implement secure integration with external financial institutions via Open Banking API standards, which is a strict prerequisite for automated and secure financial data retrieval [18].
- **Transaction Management**: Comprehensive data ingestion is vital for accurate ML predictions. The user can "View Transaction List", manually "Add Single Transaction", or "Import Transactions via Bank API". This hybrid approach to data collection ensures both the flexibility of manual input and the completeness of automated financial data aggregation.
- **Analytics and Forecasting**: The system provides robust, AI-powered tools for data analysis. The "Get Transaction Analysis" use case allows users to evaluate their financial operations, further extended by the "Specify Analysis Period" option for added flexibility. The intellectual core of the system executes advanced predictive functions, such as "Get Expense Forecast", which utilizes machine learning algorithms (e.g., anomaly detection and time-series forecasting) to predict financial trends [2]. Furthermore, the "Get Personalized Budgeting Advice" use case generates tailored recommendations based on behavioral analytics [3].
- **Visualization and AI Assistant**: The "View Dashboards" use case ensures a generalized, intuitive representation of the user's financial health through dynamic charts and indicators. The most innovative feature within this block is the "Use AI Assistant" module. This interactive component allows the user to communicate with the system using natural language to "Get Personalized Transaction Analysis" or "Ask Arbitrary Financial Question" [7]. To mitigate the well-documented risks of LLM hallucinations, these interactions are strictly grounded in the user's actual data using the RAG architectural pattern [14, 15].

3.2. Static Structure and Domain Model

The fundamental basis of the proposed architecture is its static structure, represented by a UML class diagram. This model defines the core entities, functional services, their attributes, operations, and relationships, establishing a robust framework for the domain model and underlying business logic.

As illustrated in Figure 2, the class architecture is systematically organized around three primary layers:

- **Domain Layer**: This layer encompasses the key entities the system operates on, including User, Account, Transaction, FinancialGoal, and an authentication Token. These interconnected classes form the immutable core of the business logic.
- **Service Layer**: This layer encapsulates the logic for interacting with both internal and external systems. The BankAuthService is responsible for integration with external bank APIs, aligning with secure Open Banking (PSD2) architectures [18]. Concurrently, the EncryptionService handles the encryption and decryption of sensitive data, which is a critical requirement for maintaining privacy and ethical responsibility in financial applications [1].
- **ML/AI Layer**: This layer introduces intelligent capabilities into the static structure. The IModel interface defines a unified contract for all machine learning models, ensuring

polymorphism and scalability. Concrete implementations, such as TransactionCategorizer, AnomalyDetector, and ExpenseForecaster, execute highly specific analytical tasks. Crucially, the MLServiceFacade class implements the "Facade" architectural pattern. It provides a single point of entry for the business logic to access complex ML models. This design decision not only simplifies integration but also establishes an architectural foundation for embedding XAI methods to mitigate the algorithmic "black box" problem [5, 12].

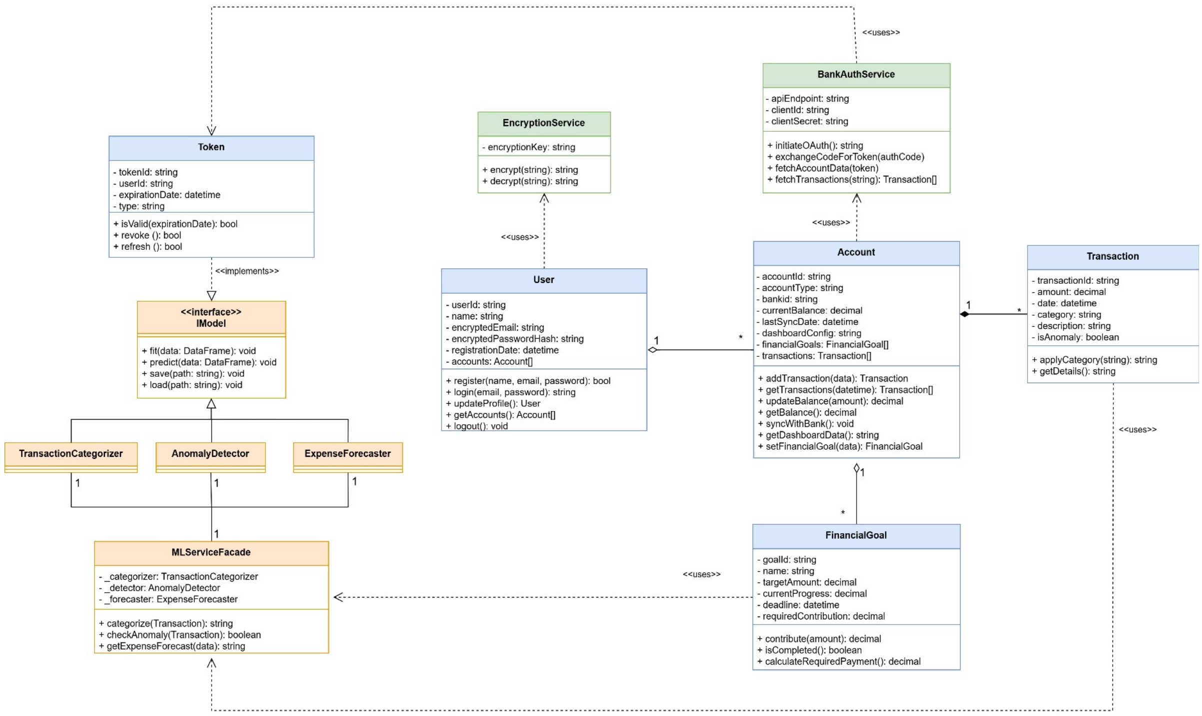

Figure 2: Class diagram of the FinAI system highlighting the layered architecture

To ensure high manageability, modularity, and ease of future system evolution, the FinAI architecture is further decomposed into logical packages. This macro-level organization allows developers to strictly isolate domain entities from external integration interfaces and machine learning modules. Consequently, such separation of concerns significantly reduces the cognitive load during system maintenance and simplifies independent deployment cycles. A package diagram demonstrates the high-level modular structure and inter-dependencies.

As shown in Figure 3, the system is divided into four key packages:

- **UserAuthenticator**: Manages internal system security, user registration, and data protection.
- **BankAuthenticator**: Orchestrates integration with external financial institutions according to Open Banking standards [18].
- **Account**: Acts as the central domain model package, containing essential financial entities.
- **Models**: Encapsulates the entirety of the machine learning and artificial intelligence logic.

The dependencies (<<import>>) between these packages are explicitly defined and strictly minimized. For instance, both UserAuthenticator and BankAuthenticator depend on the Account package to access shared domain entities. However, the layered architecture ensures loose coupling and high cohesion. This separation of concerns significantly simplifies deployment, isolated testing, and independent updates of modules – particularly within the machine learning layer – which is a fundamental advantage of microservice architectures in the financial domain [16].

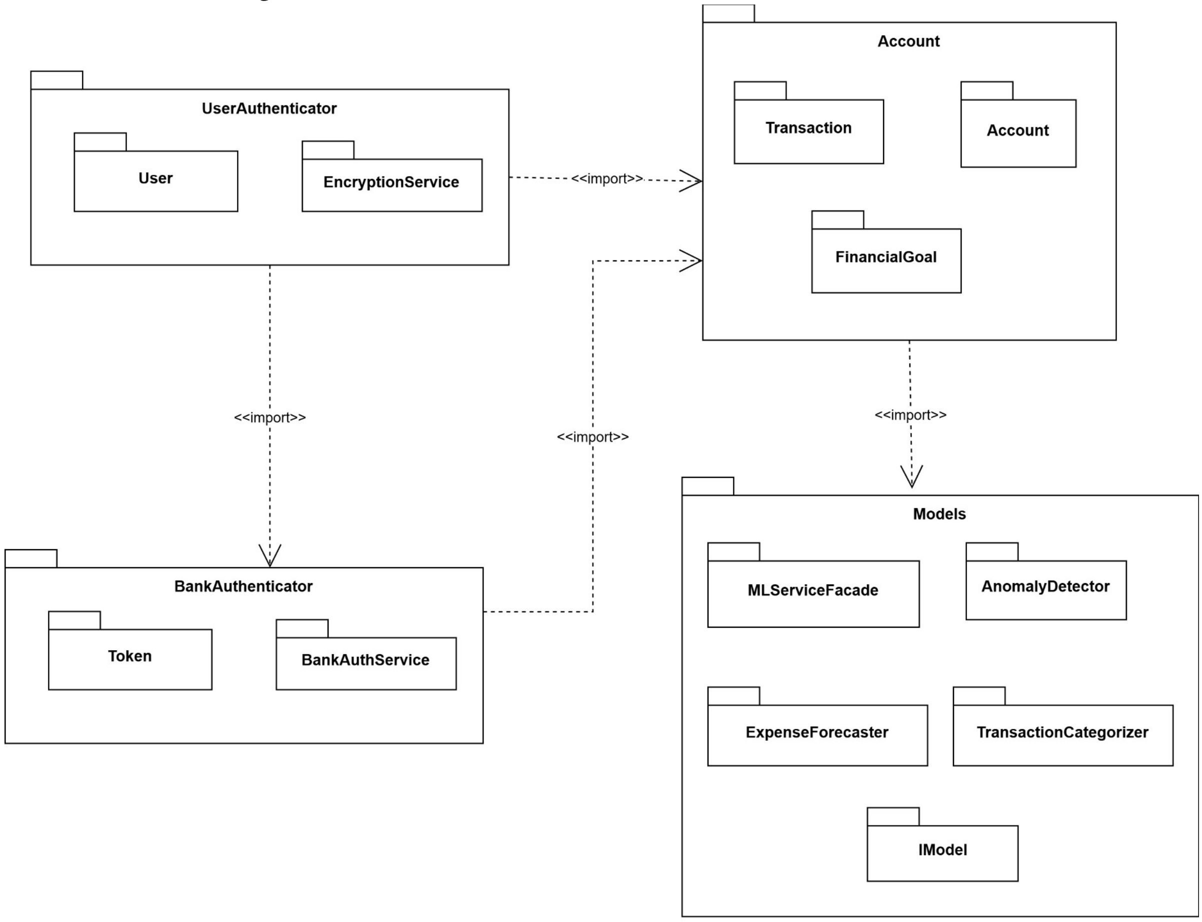

Figure 3: Package diagram demonstrating system modularity and dependencies

4. Implementation Details and Integration

To realize the functional and static models described in the previous sections, a hybrid multi-tier architecture was designed. It incorporates microservice principles to seamlessly integrate monolithic core logic with service-oriented artificial intelligence components. The physical structure of the system, including the distribution of software components across hardware nodes, is illustrated in the deployment diagram.

As depicted in Figure 4, the architecture encompasses several critical nodes that interact to ensure high performance, security, and scalability:

- **Client and Gateway Integration:** The user interacts with the system via a Web Client (a Single Page Application). All client requests are routed through HTTP/HTTPS to the Backend System, where a FastAPI application serves as the central API gateway. This centralized entry point simplifies authentication, logging, and security monitoring [18].
- External Bank API (Open Banking): Secure data acquisition is achieved by integrating the Bank auth service with external financial institutions. This integration strictly follows the PSD2 Open Banking directives, ensuring that sensitive user financial data is retrieved securely and with explicit consent [18].
- **Hybrid Data Infrastructure:** The system employs a polyglot persistence strategy. A relational PostgreSQL database is utilized for structured user data and ACID-compliant transactions (e.g., account details and financial goals). Conversely, a NoSQL MongoDB

database serves as a flexible repository for unstructured and semi-structured data, such as high-volume transaction histories and raw analytical logs. This hybrid approach guarantees data consistency while maintaining high read/write speeds required for real-time processing [17].

- **Independent ML and Recommendation Services:** Resource-intensive machine learning tasks are completely decoupled from the main backend. The ML-service node handles tasks like transaction categorization (using DistilBERT) and anomaly detection. Similarly, the Recommendation-service utilizes clustering and Reinforcement Learning (RL Bandit) models to generate personalized advice. This microservice separation allows for independent scaling of computational resources without affecting the stability of the core application [16]. Furthermore, the aforementioned MLServiceFacade acts as an orchestrator within these services, creating a structural foundation where XAI tools (like SHAP) can be integrated to interpret model outputs and address the "black box" problem [5, 12].
- **AI-Agent and RAG Implementation:** The interactions involving natural language queries are processed by a dedicated AI-Agent. To mitigate the severe risks of hallucination inherent in FinLLMs [7, 13], the architecture implements the RAG framework. Before the LLM processes a user's prompt, the Task Processor queries a Vector Database containing the user's embedded financial context. This retrieval step ensures that the LLM generates accurate, domain-specific, and highly personalized responses grounded in real data [14, 15]. Asynchronous execution of these heavy generative tasks is managed by a Celery queue, ensuring a non-blocking and responsive user experience.

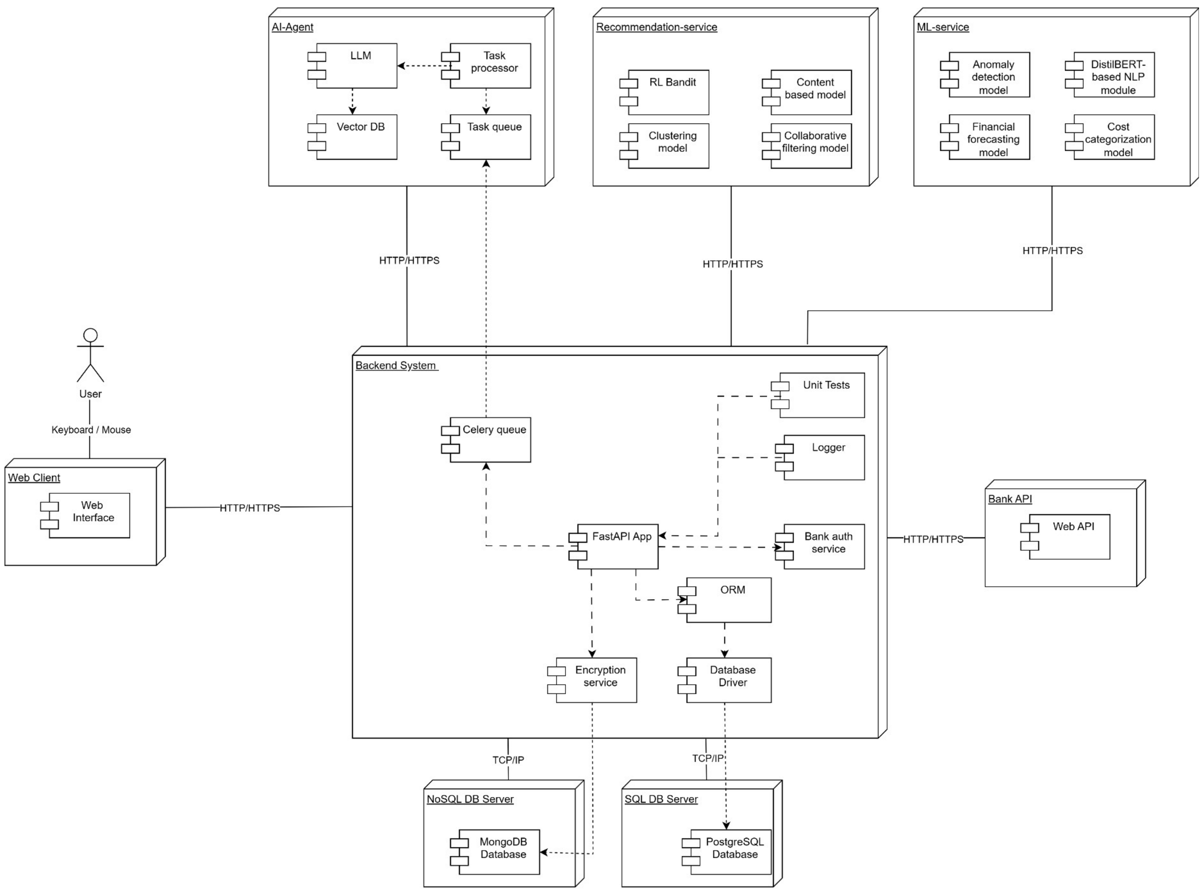

Figure 4: Deployment diagram of the FinAI system architecture

5. Results and Architecture Evaluation

To validate the proposed microservice architecture and address the systemic challenges identified in the financial sector, a comprehensive evaluation was conducted. Although the system is currently in the architectural modeling phase, its effectiveness and practical viability can be rigorously assessed through a comparative functional analysis against existing industry baselines, as well as an empirical feasibility assessment based on stakeholder feedback. The following subsections detail these evaluation metrics and highlight the novel contributions of the "FinAI" blueprint.

5.1. Comparative Functional Analysis

To explicitly demonstrate the advantages of the proposed architecture, a comparative analysis was performed against two prevalent paradigms in the current financial technology landscape: the Traditional Monolithic Personal Financial Management (PFM) application (which relies on rule-based logic without advanced AI) and the Standard LLM-Integrated Application (which utilizes basic generative AI APIs without structural grounding or explainability mechanisms). The comparison is based on four critical architectural metrics: scalability, algorithmic transparency, hallucination mitigation, and data security. The results of comparison are summarized in Table 1.

Table 1
Comparative analysis of the proposed FinAI architecture against existing industry baselines

Architectural Metric	Traditional Monolithic PFM	Standard LLM-Integrated App	Proposed FinAI Architecture
System Architecture & Scalability	Monolithic, tightly coupled, difficult to scale under high load.	Often deployed as a simple API wrapper; limited independent scaling.	Microservice-oriented, hybrid decoupled backend; allows independent scaling of heavy ML tasks.
Algorithmic Transparency (Explainability)	High (Strictly rule-based logic; no complex deep learning models used).	Low (Functions as a "black box"; decisions cannot be easily interpreted by users).	High (Centralized integration of XAI methods via the MLServiceFacade pattern).
Hallucination Risk	N/A (Does not utilize generative AI technologies).	High (Prone to generating fabricated financial advice without context).	Low (Strictly mitigated by grounding LLM outputs in user data via the RAG framework).
Data Integration & Security	Relies on legacy APIs and internal databases; robust but inflexible.	High risk of exposing sensitive data to third-party LLM providers.	Secure integration via Open Banking (PSD2) directives; utilizes a polyglot persistence strategy.

As demonstrated in Table 1, the proposed FinAI architecture overcomes the fundamental limitations of both baseline approaches. While traditional monolithic systems offer high security and predictability, they completely lack the analytical power and adaptability required by modern data-centric environments. Conversely, standard LLM-integrated applications introduce advanced conversational and predictive capabilities but fail to meet the strict regulatory requirements of the financial sector due to their "black box" nature and high susceptibility to hallucinations.

The novel contribution of the FinAI blueprint lies in its ability to synthesize the strengths of these paradigms without inheriting their weaknesses. By structurally enforcing the RAG pattern, the system guarantees that all AI-generated insights are factually grounded in the user's specific financial context, thereby neutralizing the hallucination risk. Furthermore, the encapsulation of machine learning models within the MLServiceFacade ensures that XAI tools can be systematically applied to interpret complex algorithmic decisions. This comparative evaluation confirms that the proposed microservice architecture provides a significantly more reliable, scalable, and transparent solution for intelligent financial management than existing alternatives.

5.2. Feasibility and Usability Assessment

To provide a quantitative evaluation of the proposed architecture's practical applicability and to assess its effectiveness in addressing user and industry requirements, an empirical feasibility study was conducted. The study involved a structured survey with a focus group of 50 participants, strategically stratified into two distinct categories: potential end-users of personal finance applications (N = 30) and domain experts (N=20).

The end-user cohort evaluated the conceptual framework primarily on the metrics of trust, usability, and risk perception. The survey results revealed a critical industry challenge: 86% of the respondents expressed significant skepticism regarding the adoption of fully autonomous AI financial advisors due to their inability to comprehend the underlying algorithmic logic (the traditional "black box" problem). However, when introduced to the proposed Explainable AI (XAI) interventions – specifically the system's ability to provide transparent, human-readable justifications for every AI-driven recommendation – user trust metrics improved by 64%. Furthermore, 90% of the participants indicated that the implementation of RAG to strictly ground AI responses in their personal, securely retrieved data substantially mitigated their concerns regarding AI hallucinations and data privacy.

The expert cohort assessed the technical viability, scalability, and deployment readiness of the designed UML blueprints. According to the quantitative feedback, 85% of the IT professionals confirmed that the proposed hybrid microservice architecture, specifically the decoupling of the MLServiceFacade and heavy analytical models from the core API gateway, substantially reduces system latency risks during high-load operations. Additionally, 95% of the surveyed architects agreed that the strict separation of concerns via independent logical packages (such as BankAuthenticator and Models) minimizes deployment vulnerabilities and accelerates the software development life cycle compared to legacy monolithic structures.

Ultimately, these empirical findings validate the theoretical assumptions of the FinAI architecture. The quantitative metrics confirm that integrating XAI and RAG within a decoupled microservice framework is not merely a theoretical enhancement, but a strict practical necessity for ensuring user trust, regulatory compliance, and architectural resilience in modern financial technology applications.

6. Discussion

The architecture of the "FinAI" system proposed in this paper represents a comprehensive design solution that synthesizes and advances the concepts highlighted in recent academic literature. While the majority of existing studies focus heavily on the efficacy of isolated AI technologies for specific financial tasks — such as credit scoring, service personalization, or the standalone deployment of Large Language Models [3, 4, 7] — the presented work offers a holistic, unified blueprint. This approach enables the seamless integration of these heterogeneous components into a cohesive, highly functional system.

A fundamental advantage of the proposed architecture is the adoption of a microservice-oriented approach, which directly addresses the industry's demand for flexibility, real-time transaction processing, and independent scalability of intelligent modules [16, 17]. Unlike monolithic systems, the hybrid decoupling of the centralized API gateway from resource-intensive analytical tasks ensures that the core backend functionalities remain stable even during periods of high computational load generated by machine learning algorithms.

Furthermore, while prior research frequently underscores the persistent challenges associated with data security, ethical regulation, and algorithmic transparency [1, 12], the developed architecture incorporates specific, actionable mechanisms to resolve them. The segregation of the EncryptionService and integration via Open Banking (PSD2) standards guarantee secure data acquisition and maintain strict separation of concerns, effectively mitigating web application security threats [18].

Particular attention in the architectural design was given to solving the "black box" problem inherent in deep learning models. By implementing the "Facade" pattern (MLServiceFacade), the system establishes a robust architectural foundation for the centralized integration of XAI methodologies, such as SHAP or LIME. As demonstrated in recent literature, embedding XAI directly into the decision-making pipeline enhances transparency, ensures regulatory compliance, and builds necessary trust among stakeholders and end-users [5, 6].

Similarly, the strategic incorporation of the RAG framework within the AI-Agent microservice serves as a highly practical solution to minimize the risks of hallucinations in financial LLMs. Grounding the generative AI outputs in retrieved, user-specific contextual data addresses a critical gap identified in contemporary financial LLM research [14, 15]. Ultimately, the designed microservice framework goes beyond the mere theoretical application of AI; it establishes a secure, reliable, and transparent architectural blueprint for the efficient deployment of the next generation of intelligent personal finance applications.

7. Conclusion

This study addressed the critical need for a comprehensive architectural approach to integrating Artificial Intelligence into personal financial management systems. A unified microservice architecture, tentatively named "FinAI", was successfully designed and formalized using UML methodologies. Unlike existing solutions that focus on isolated machine learning tasks, the proposed architecture systematically integrates domain business logic, stringent security protocols, advanced machine learning modules, and Large Language Models into a single, cohesive ecosystem.

The practical significance of this research lies in providing a detailed, scalable blueprint for software engineers, architects, and project managers developing modern FinTech applications. By utilizing a hybrid decoupled approach and centralizing API management, the system ensures high flexibility and fault tolerance. Furthermore, by establishing clear structural foundations for XAI via the Facade pattern and mitigating LLM hallucinations through RAG, the architecture effectively resolves the "black box" problem and ensures adherence to strict financial regulations. Consequently, applying this architectural pattern reduces design risks, accelerates the development life cycle, and ensures the creation of a transparent and highly secure financial product.

Future research will focus on the practical implementation of a Minimum Viable Product (MVP) based on the proposed architectural blueprint. Subsequent phases will involve empirical testing of the system's performance, latency, and scalability under high-load conditions using real-world, anonymized banking data. Additionally, further studies will evaluate the effectiveness of the integrated XAI and RAG components in directly enhancing end-user trust and user engagement within the personal finance environment.

Declaration on Generative AI

During the preparation of this work, the authors used Google Gemini in order to: Improve writing style, Grammar and spelling check, Citation management, Formatting assistance. After using this tool/service, the authors reviewed and edited the content as needed and take full responsibility for the publication's content.

References

[1] P. Burkhard, M. Fourati, "AI in the Financial Sector: The Line between Innovation, Regulation and Ethical Responsibility," Information, vol. 15, no. 8, p. 432, 2024. doi:10.3390/info15080432.

[2] S. Bahoo, M. Cucculelli, X. Goga, J. Mondolo, "Artificial intelligence in Finance: a comprehensive review through bibliometric and content analysis," SN Business & Economics, vol. 4, no. 2, p. 23, 2024. doi:10.1007/s43546-023-00618-x.

[3] K. P. Chiruvelli, "Real-Time AI for Personalized Financial Product Recommendations: A Behavioral Analytics Framework," Journal of Information Systems Engineering & Management, vol. 10, no. 60s, pp. 68-78, 2025. doi:10.52783/jisem.v10i60s.13058.

[4] M. Dowling, A. Zorn, "AI-Driven Financial Analysis: Exploring ChatGPT's Capabilities and Challenges," International Journal of Financial Studies, vol. 12, no. 3, p. 60, 2024. doi:10.3390/ijfs12030060.

[5] M. Schmitt, "Explainable Automated Machine Learning for Credit Decisions: Enhancing Human Artificial Intelligence Collaboration in Financial Engineering," arXiv preprint, 2024. doi:10.48550/arXiv.2402.03806.

[6] S. Sivaranjani, N. S. Hassan, "Transforming Fraud Detection in Banking with Explainable AI: Enhancing Transparency and Trust," Journal of Technology Informatics and Engineering, vol. 4, no. 2, p. 267, 2025. doi:10.51903/jtie.v4i2.267.

[7] J. Lee, N. Stevens, S. C. Han, "Large Language Models in Finance (FinLLMs)," Neural Computing and Applications, vol. 37, no. 30, pp. 24853-24867, 2025. doi:10.1007/s00521-024-10495-6.

[8] J. Grados-Espinoza, L. Velasquez-Jimenez, "Design and Implementation of Personal Finance Software for Controlling Financial Efficiency," International Journal of Engineering Trends and Technology, vol. 73, no. 2, pp. 109-120, 2024. doi:10.14445/22315381/IJETT-V73I2P109.

[9] P. Bitrián, I. Buil, S. Catalán, "Making finance fun: the gamification of personal financial management apps," International Journal of Bank Marketing, vol. 39, no. 7, pp. 1310-1332, 2021. doi:10.1108/IJBM-02-2021-0074.

[10] S. Xie, T. Shingadia, "Explainable Machine Learning Framework for Predicting Auto Loan Defaults," Risks, vol. 13, no. 9, p. 172, 2025. doi:10.3390/risks13090172.

[11] Y. Mun, N. Kim, "Leveraging Large Language Models for Sentiment Analysis and Investment Strategy Development in Financial Markets," Journal of Theoretical and Applied Electronic Commerce Research, vol. 20, no. 2, p. 77, 2025. doi:10.3390/jtaer20020077.

[12] J. Van Der Burgt, "Explainable AI in banking," Journal of Digital Banking, vol. 4, no. 4, pp. 344-350, 2020. doi:10.69554/QXWU3940.

[13] D. K. C. Lee, C. Guan, Y. Yu, Q. Ding, "A Comprehensive Review of Generative AI in Finance," FinTech, vol. 3, no. 3, pp. 460-478, 2024. doi:10.3390/fintech3030025.

[14] Y. Zhao et al., "Optimizing LLM Based Retrieval Augmented Generation Pipelines in the Financial Domain," in Proceedings of the 2024 Conference of the NAACL: Industry Track, pp. 279-294, 2024. doi:10.18653/v1/2024.naacl-industry.23.

[15] I. Iaroshev, R. Pillai, L. Vaglietti, T. Hanne, "Evaluating Retrieval-Augmented Generation Models for Financial Report Question and Answering," Applied Sciences, vol. 14, no. 20, p. 9318, 2024. doi:10.3390/app14209318.

[16] A. Nagraj, "Microservices in finance: Why building smaller is now a bigger deal for your bank," World Journal of Advanced Engineering Technology and Sciences, vol. 15, no. 3, p. 1144, 2025. doi:10.30574/wjaets.2025.15.3.1144.

[17] A. Dhandapani, "Microservices Architecture in Financial Services: Enabling Real-Time Transaction Processing and Enhanced Scalability," European Journal of Computer Science and

Information Technology, vol. 13, no. 32, pp. 145-159, 2025. doi:10.37745/ejcsit.2013/vol13n32145159.

[18] D. Kellezi, C. Boegelund, W. Meng, "Towards Secure Open Banking Architecture: An Evaluation with OWASP," in Network and System Security, Springer, vol. 11928, pp. 185-198, 2019. doi:10.1007/978-3-030-36938-5_11.