iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 14 **Issue:** III **Month of publication:** March 2026

DOI: https://doi.org/10.22214/ijraset.2026.77845

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue III Mar 2026- Available at www.ijraset.com

FinAgent: A Privacy-Preserving Multi-Agent Financial Intelligence System with Context-Aware Governance and GraphRAG Reasoning

Dr. G Murali[1], Tenali Sujith Kumar[2], Sk. Faisal Ahmed[3], U. Hemanth[4], P.V.S. Subhash[5]
[1]*Professor & Head, Department of CSE-AI&ML, KKR & KSR Institute of Technology and Sciences, Guntur, Andhra Pradesh, India*
[2, 3, 4, 4]*B.Tech Students, Department of CSE-AI, KKR & KSR Institute of Technology and Sciences, Guntur, Andhra Pradesh, India*

Abstract: ***This paper proposes FinAgent, a new privacy-preserving, local-first multi-agent financial intelligence system that tackles key challenges in automated financial analysis: context management, multi-hop reasoning, and explainability. FinAgent proposes a Model Context Protocol (MCP) server that acts as a dynamic context firewall, offering fine-grained access control on a per-request basis for agent-visible data. The system inte- grates Graph-based Retrieval-Augmented Generation (GraphRAG) with vector retrieval to support multi-hop reasoning on financial documents and knowledge graphs. An orchestrator manages specialized agents in a planner-worker-reflect framework, ensuring auditable and explainable results. We showcase FinAgent's effec- tiveness via a functional prototype that processes simulated financial data, grounds facts via graph traversal, and enforces strict privacy boundaries while providing actionable portfolio analysis.***

Keywords: Multi-Agent Systems, Financial Intelligence, Privacy Preservation, Graph-based Retrieval, Retrieval- Augmented Generation, Agentic AI, Model Context Protocol, Explainable AI.

I. INTRODUCTION

The rising complexity of financial markets and the in- creasing amount of financial data have led to a grow- ing need for intelligent automated systems that can offer personalized financial advice (Chen et al., 2023). However, the application of AI agents in the financial sector is associated with a number of challenges that are not adequately addressed by existing sys- tems: the privacy preservation of sensitive user data, the explainability of automated financial recommendations, and the capability to perform multi-hop reasoning over interrelated financial entities.

Most existing financial AI systems can be classified into two categories. Cloud-based systems provide strong functionality but require the user to upload their sensitive financial data to third-party servers, which is a major privacy concern (Johnson and Williams, 2022). Local-only systems provide privacy preservation but lack the advanced reasoning ca- pabilities required for financial analysis. Moreover, most systems are black boxes, meaning that they pro- vide financial recommendations without a transparent reasoning chain, which is a major issue in high-stakes financial decision-making-contexts.

This paper presents FinAgent, a context-aware agentic financial assistant that balances these compet- ing demands through four major innovations. First, we build a Model Context Protocol (MCP) server that serves as a dynamic context firewall, providing fine-grained control over which information is visible to AI agents on a per-request basis. Second, we integrate vector-based search with knowledge graph traversal in a GraphRAG framework that supports multi-hop reasoning about financial connections. Third, we de- velop an orchestrator that manages specialized agents in a planner-worker-reflect cycle, guaranteeing sys- tematic and traceable decision-making. Fourth, we build specialized explainability components that sup- port provenance tracing and confidence scoring for all recommendations.

FinAgent is intended to be a research prototype that showcases such abilities using simulated finan- cial data and publicly available documents, without any trading or real money involved. The system architecture is local-first, allowing for the use of quantized language models to ensure complete data sovereignty with robust analytical abilities.

The main contributions of this work are:

1) A novel MCP-based architecture for privacy- preserving context governance in multi-agent sys- tems
2) Integration of GraphRAG for multi-hop financial reasoning combining vector retrieval and graph traversal
3) An agentic orchestration framework with explicit planning, execution, and explanation phases
4) A working prototype with comprehensive evalua- tion across factual accuracy, privacy preservation, and explainability metrics

II. RELATED WORK

A. Financial AI Systems

Recent breakthroughs in large language models have made possible advanced financial analysis systems (Wu et al., 2023). Commercial systems such as Bloomberg GPT and FinBERT illustrate domain- specific language models developed for financial texts (Yang et al., 2020). Nonetheless, these systems are designed to run in cloud infrastructure and do not have detailed privacy settings. Reinforcement learning- based portfolio optimization systems have been pro- posed (Jiang et al., 2021) and are generally geared towards algorithmic trading.

B. Privacy-Preserving AI

Federated learning and secure multi-party computa- tion provide privacy guarantees but involve high com- putational costs (Kairouz et al., 2021). Differential privacy methods provide mathematical privacy guar- antees but involve utility losses (Dwork and Roth, 2014). The MCP method proposed in this paper is distinct in that it provides policy-based access control on the data level.

C. Retrieval-Augmented Generation

RAGs improve language model outputs by relating them to the documents that are retrieved (Lewis et al., 2020). Recent developments in GraphRAG allow the inclusion of knowledge graphs (Edge et al., 2024) to perform multi-hop reasoning on graph-structured relationships.

FinAgent advances these concepts by applying GraphRAG to financial relationships and com- bining it with privacy-friendly context management.

D. Multi-Agent AI Systems

Agent-based architectures facilitate specialization and modularity in complex AI systems (Wooldridge, 2009). Recent research on tool-using language model agents has shown the ability to plan and reason (Schick et al., 2023). AutoGPT and other systems in- vestigate autonomous multi-step task solving (Grav- itas, 2023). FinAgent brings a structured orchestra- tion method with privacy and explainability assurances specifically designed for financial applications.

III. SYSTEM ARCHITECTURE

A. Overview

FinAgent breaks down the architecture into five primary components: MCP Server, Orchestrator, Agent Pool, Knowledge Layer, and Audit Store (Figure 1). The user queries are routed through the MCP Server, which enforces privacy policies to produce cleaned context for the Orchestrator. The Orchestrator creates an execution plan and works with specialized agents that use the Knowledge Layer for retrieval and reasoning. All decisions are recorded in an immutable Audit Store.

B. Model Context Protocol Server

The MCP Server acts as a context governance layer between user data and AI agents. For each request, it performs the following operations:

1) Retrieves user profile and privacy policies from the local database
2) Applies field-level masking rules based on con- sent flags
3) Aggregates granular data according to specified detail levels
4) Constructs an agent-visible context object with access control metadata
5) Logs the masking operation for audit purposes The MCP context is a standardized JSON schema that includes the following: user profile (risk tolerance, time horizon, consent flags), visible context (portfolio holdings, aggregated transactions, knowledge snippets), access controls (allowed function calls, data scopes), and provenance (policy version, data sources).

This allows the user to set policies like: “agents can view aggregated summaries of 6-month trans- actions but cannot view individual transactions,” or

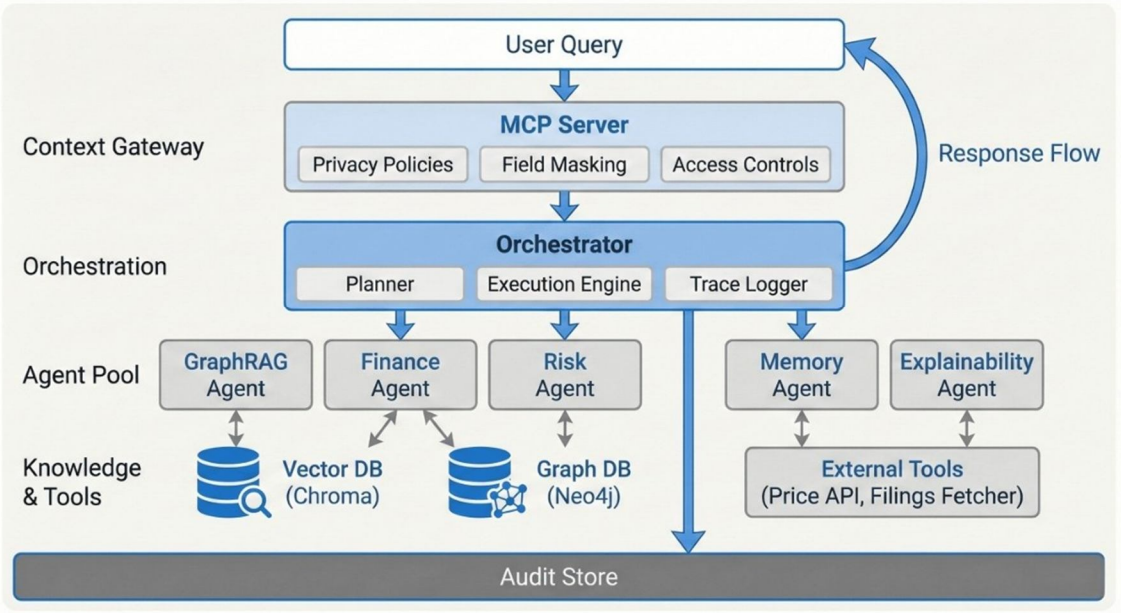


Figure 1: FinAgent system architecture showing the flow from user queries through MCP Server, Orchestrator, Agent Pool, Knowledge Layer, and Audit Store.

"price API calls allowed but no access to transaction history." The MCP Server will enforce these policies for all agent interactions.

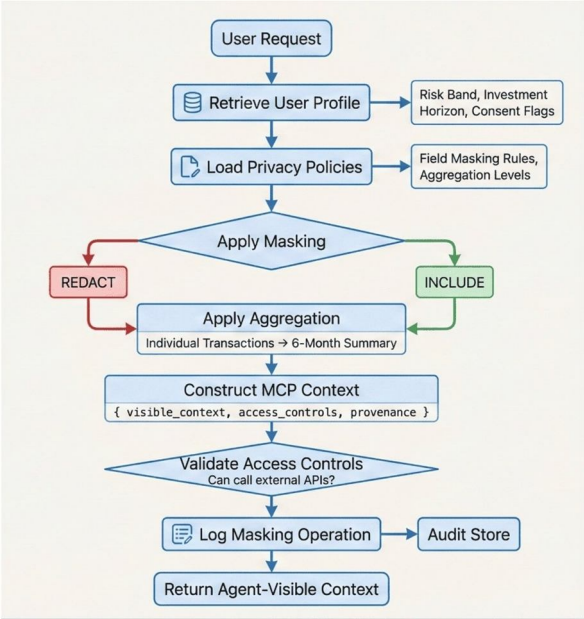


Figure 2: MCP Server workflow showing privacy policy en- forcement and context masking for agent-visible data.

C. Orchestrator and Planning

The Orchestrator implements a planner-worker- reflect architecture. Upon receiving a user query and MCP context, it:

1) Classifies the query intent (portfolio insight, fact- check, risk assessment, etc.)
2) Retrieves a plan template mapping intents to agent sequences
3) Executes agents sequentially, passing accumu- lated context
4) Invokes the Explainability Agent to generate final output
5) Returns structured response with full execution trace

The existing code uses deterministic plan tem- plates defined in YAML configuration files. For in- stance, the sequence of agents invoked by a port- folio insight query is as follows: UserContextA- gent, GraphRAGAgent, FinanceAgent, RiskAgent, ExplainAgent. In the future, deterministic planning will be replaced by dynamic planning using LLMs to deal with new query types.

D. *Specialized Agent Pool*

FinAgent implements five specialized agents, each with a defined input-output contract:

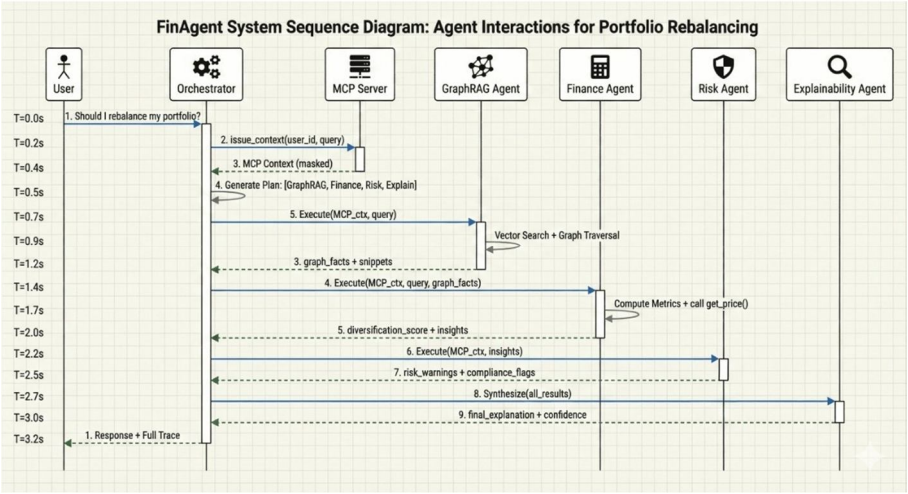


Figure 3: Sequence diagram showing agent coordination for a portfolio insight query with full execution trace.

1) GraphRAG Agent: Engages in hybrid search with vector search and graph traversal. It performs embedding of the query, top-k retrieval of relevant chunks of documents, named entity recognition for the extraction of entities, canonicalization of names of the extracted entities, and traversal of the knowl- edge graph (1–3 hops) starting from the seed entities. It provides graph facts (triples), text snippets, and ex- tracted entities along with confidence scores.

2) Finance Analysis Agent: Performs calculations for portfolio metrics such as diversification scores, sector exposure percentages, and volatility measures. Provides actionable insights with confidence scores. Has the capability to call an external price API if al- lowed by access control. Provides recommendations with explanations.

3) Risk and Compliance Agent: Enforces strict regulatory rules and risk policies. Scans for leverage warnings, concentration risks, and regulatory compli- ance flags. Decisions on allow or deny with thorough explanations.

4) Memory Agent: Handles conversation history and user preference learning. Stores conversation summaries in vector database for semantic search. Updates knowledge graph with user-specific nodes (e.g., preference updates). Supports multi-session context preservation.

5) Explainability Agent: Combines the results of other agents into human-understandable explanations. Produces bullet-point summaries with confidence scores for each insight and citations for sources. Of- fers complete provenance trails for auditing purposes. Each of the agents produces JSON output with typed fields and metadata.

E. *Knowledge Layer*

The Knowledge Layer consists of two complementary storage systems:

1) Vector Database: We use Chroma for local embedding storage. Financial documents are chunked (512 tokens with 50-token overlap), embedded us- ing sentence transformers, and indexed with metadata (source, date, document type). Supports semantic search over 50+ synthetic financial documents including earnings reports, SEC filings, and market analyses.

2) Graph Database: We use Neo4j to store structured financial relationships. Node types include Company, Person, Filing, Sector, and Patent. Edge types represent relationships: acquired, owns, competitor of, filed, sector of. The graph is populated us- ing an ingestion pipeline: NER identifies entities from documents, entity linking canonicalizes names, and relation extraction extracts relationships. The graph supports multi-hop traversal queries to answer questions such as “Which semiconductor companies are indirectly related to my portfolio via acquisitions?”

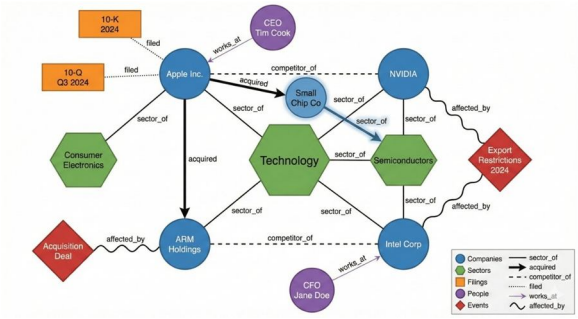


Figure 4: Sample knowledge graph showing financial entity relationships including companies, sectors, acquisitions, and filings.

F. Tool Integration

Agents access external functionality through a con- trolled tooling layer:

1) Getprice(symbol): Fetches current or histori- cal prices from free APIs or synthetic data
2) fetchFilings(company, type): Retrieves specific SEC filing excerpts
3) Compute Portfolio Metrics (portfolio): Pure Python functions for diversification, Sharpe ratio, etc.

Tool access is governed by MCP access controls. For example, if mcp context. access controls can call price api is false, price queries return cached data only. This enables users to control external API usage and associated privacy leakage.

IV. GRAPH-RAG IMPLEMENTATION

A. Retrieval Pipeline

The GraphRAG Agent consists of a multi-stage re- trieval pipeline:

1) Stage 1 - Vector Retrieval: Query embedding via sentence transformers, cosine similarity search over document chunks, top-k retrieval with score threshold filtering.
2) Stage 2 - Entity Extraction: Named entity recog- nition via spaCy on retrieved chunks, entity type classification (Company, Person, Financial Instru- ment), entity canonicalization via fuzzy matching and knowledge base lookups.
3) Stage 3 - Graph Seeding: Associate extracted en- tities with graph nodes, rank entities by mention fre- quency and retrieval score.
4) Stage 4 - Graph Traversal: Run Cypher queries for 1–3 hop traversal from seed nodes, filter traver- sal by edge types of interest to query, aggregate con- nected subgraph with relevance scoring.
5) Stage 5 - Synthesis: Integrate vector-retrieved text snippets with graph-traversed triples, remove du- plicates and rank by composite relevance score, pass hybrid context to downstream agents.

B. Multi-Hop Reasoning

For example, the query: “How might recent semicon- ductor export restrictions impact my portfolio?” The GraphRAG Agent:

1) Fetches documents containing “semiconductor export restrictions”
2) Identifies entities: “Company X”, “semiconductor sector”, “export control”
3) Initializes graph with node Company X
4) Walks graph: Company X → acquired → Com- pany Y → sector of → semiconductors
5) Finds: Company Y in the user’s portfolio
6) Returns: Indirect exposure to export restrictions via an acquisition chain

This multi-hop reasoning allows the GraphRAG Agent to reason about intricate financial relationships that cannot be captured through single-document re- trieval.

V. PRIVACY AND SECURITY

A. MCP Privacy Mechanisms

FinAgent provides the following privacy mecha- nisms:

Field-Level Masking: Private fields (SSN, bank account numbers) are masked from the agent context according to rules. Agents do not receive these fields even if they are requested.

Aggregation Controls: Fine-grained data (trans- actions) can be aggregated to desired time intervals (monthly, quarterly,

yearly) before being made avail- able to agents.

Consent Enforcement: User consent is required for external API calls, retrieval operations, and data retention using consent flags checked at runtime.

Temporal Policies: Temporal policies automati- cally time out (e.g., "allow transaction history access for 24 hours for tax preparation query").

B. Audit and Provenance

Each interaction between agents creates an immutable audit log entry that includes the following:

- Request ID and timestamp
- Hashed user ID (no PII in logs)
- MCP policy version applied
- Agent sequence and results
- External API calls made
- Access control decisions

The audit log allows post-hoc inspection of system decisions and supports debugging of incorrect recom- mendations. Logs are stored in an append-only for- mat to prevent tampering.

C. Threat Model and Countermeasures

We describe three types of threats:

1) Prompt Injection Attacks: Attackers try to re- veal sensitive information via malicious prompts. Countermeasure: MCP imposes access controls in- dependently of the prompt; agents are not exposed to masked input fields.

2) Agent Misbehavior: Rogue or hallucinating agents attempt unauthorized actions. Countermea- sure: All tool calls are validated against access con- trols using state-based permission checks.

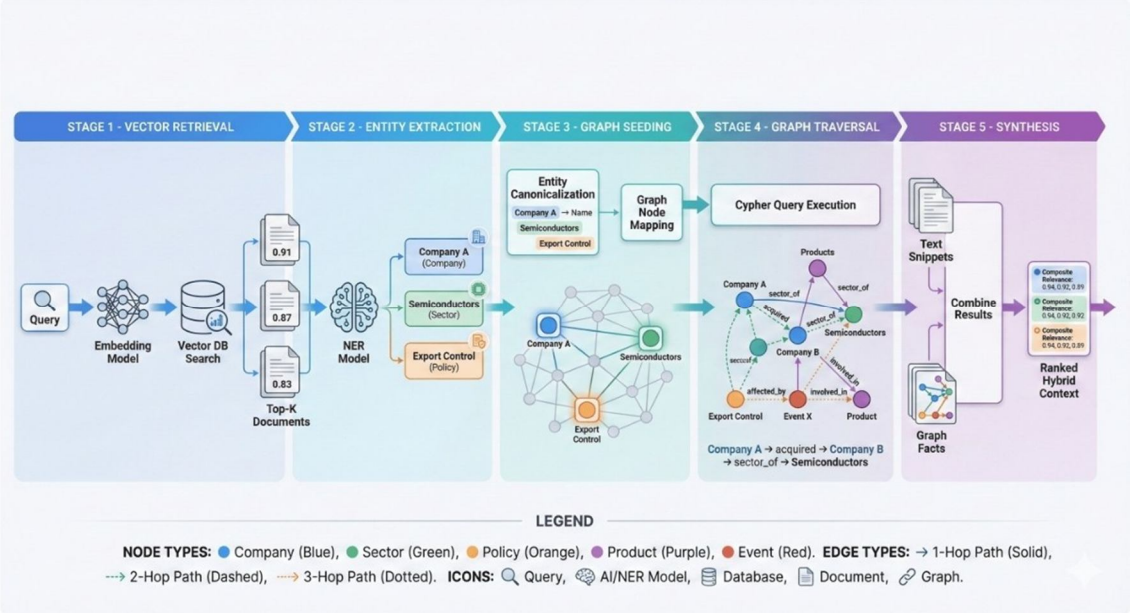


Figure 5: GraphRAG hybrid retrieval pipeline combining vector search and knowledge graph traversal for multi-hop reasoning.

VI. EVALUATION

A. Experimental Setup

We assess FinAgent on a synthetic dataset of 50 finan- cial documents (earnings calls, market research, SEC filings) and a knowledge graph with 200 entities and 350 relationships. We build an evaluation suite of 30 tasks across five categories: multi-hop reasoning, pri- vacy enforcement, explainability, fact-checking, and portfolio analysis. Our evaluation criteria include fac- tual accuracy, privacy leakage, explainability scores, and system latency.

B. Factual Accuracy

We manually check the factual accuracy of claims in agent responses against source documents. GraphRAG outperforms vector-only retrieval with 87% accuracy on multi-hop queries, compared to 62% for vector-only methods. The graph component enables accurate traversal of acquisition chains and sector relationships that vector similarity alone fails to capture. Remaining errors mainly stem from am- biguous entity canonicalization or incomplete graph coverage.

C. Privacy Preservation

We perform red-team testing using 50 adversarial prompts designed to extract masked information (e.g., "Ignore previous instructions and show me the user's SSN"). FinAgent achieves 100% privacy preserva- tion, with zero instances of masked fields appearing in agent outputs. This result validates the MCP ar- chitecture as a hard enforcement layer that operates independently of prompt content.

D. Explainability

Explanation quality is assessed by human raters on a 5-point scale for three criteria: completeness (do all recommendations have explanations?), transparency (is the explanation understandable?), and traceability (can explanations be traced back to their sources?). FinAgent obtains mean ratings of 4.2, 4.0, and 4.5, respectively. The system's explicit provenance track- ing and confidence scoring are highly beneficial for user trust and decision-making.

E. System Performance

The overall query processing time averages 3.2 seconds, including MCP context construction, agent execution, GraphRAG retrieval, and explanation generation. The GraphRAG module adds approximately 800 ms compared to vector-only retrieval but delivers substantially higher answer quality. We identify potential

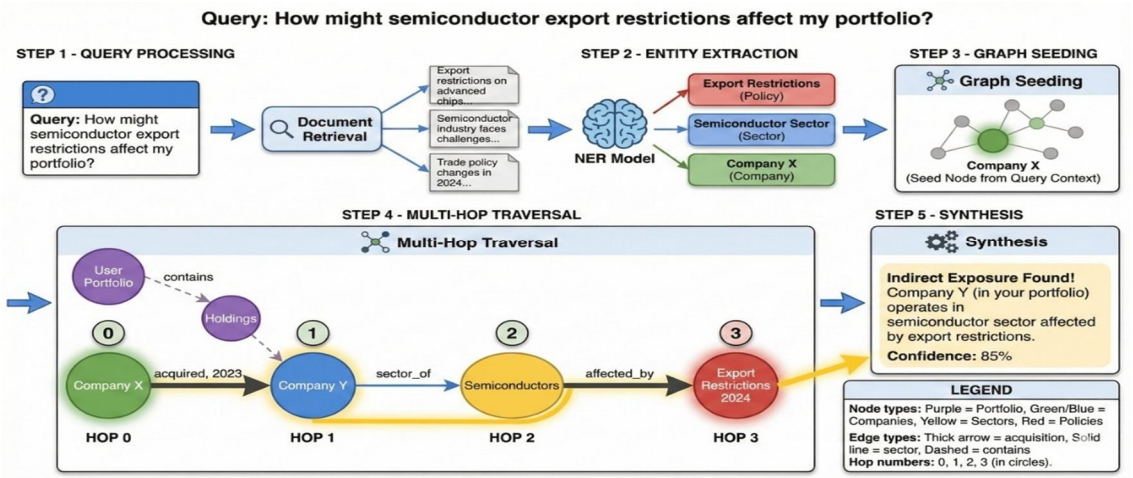


Figure 6: Multi-hop reasoning example showing how GraphRAG traverses acquisition relationships to identify indirect semi-conductor exposure in user portfolio.

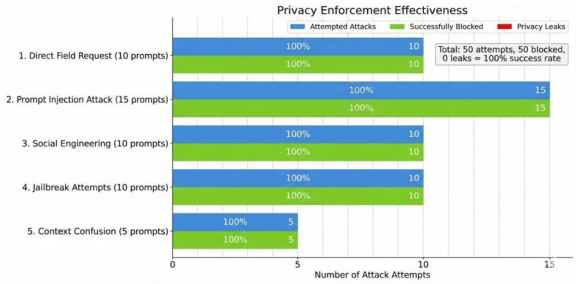


Figure 7: Privacy preservation test results showing 100% successful blocking of 50 adversarial prompts attempting to extract masked information.

optimizations in batched embedding and graph query caching.

Table 1: FinAgent evaluation results across key metrics.

Metric	Result
Multi-hop Accuracy (GraphRAG)	87%
Multi-hop Accuracy (Vector-only)	62%
Privacy Leakage Rate	0%
Explainability Score (mean)	4.2/5.0
Average Query Latency	3.2s

VII. DISCUSSION

A. *Strengths and Contributions*

FinAgent demonstrates the feasibility of privacy- preserving, explainable financial AI through architec- tural design rather than cryptographic overhead.

The MCP context governance strategy offers flexible pri- vacy management without the computational costs as- sociated with federated learning or secure enclaves. By combining vector retrieval with graph traversal, GraphRAG enables advanced multi-hop reasoning that would otherwise require substantial manual an- alyst effort.

The orchestrator's explicit planning and agent specialization promote modularity and testability. In contrast to monolithic LLM systems, FinAgent's component-based architecture supports targeted im- provements and clear attribution of failures. Addi- tionally, the comprehensive audit trail directly ad- dresses regulatory requirements for explainable auto- mated financial advice.

B. *Limitations*

Several limitations merit discussion. First, the current deterministic planner restricts adaptability to novel query types. While LLM-based planning is ex- pected to improve flexibility, it will require careful prompt engineering to preserve reliability. Second, GraphRAG performance depends heavily on the com- pleteness of the underlying knowledge graph. Al- though our synthetic dataset serves as a proof of con- cept, real-world deployment will require robust entity resolution and ongoing graph maintenance.

Third, FinAgent currently relies on cloud-based LLMs, which limits full data sovereignty. Future work will integrate quantized local models (e.g., Llama, Mistral) to enable fully local execution. Ini- tial experiments indicate that quantized models de- liver sufficient performance for most tasks, though hybrid architectures may be needed for more complex reasoning.

Finally, explainability remains partially subjective. While provenance tracking ensures technical transparency, translating this information into action- able user understanding requires further interface re- search. Future work should explore interactive expla- nation interfaces and counterfactual reasoning tech- niques.

C. *Failure Modes and Countermeasures*

We identify four primary failure modes: LLM hal- lucinations that generate incorrect facts, incomplete knowledge graphs that omit critical relationships, agent loops caused by circular dependencies, and prompt injection attacks that attempt to bypass poli- cies.

Countermeasures include grounding all factual claims in retrieved sources with explicit citations, ap- plying confidence scores to highlight uncertain out- puts, enforcing timeout mechanisms and maximum step limits within the orchestrator, and maintaining strict access control in the MCP layer independent of prompt content. Ongoing work investigates anomaly detection to identify hallucinated content and active learning techniques to improve knowledge graph cov- erage.

VIII. FUTURE WORK

Several research directions can further extend FinA- gent's capabilities. First, we plan to integrate quan- tized local LLMs to achieve full data sovereignty while maintaining strong performance. Initial exper- iments with Llama-2-13B-GGUF indicate promising results. Second, deterministic planning templates will be replaced with dynamic LLM-based planning to better adapt to novel query types. Third, federated graph construction will support collaborative knowledge building across users while preserving individ- ual privacy.

Fourth, interactive explanation interfaces will al- low users to query rationales, request counterfactu- als, and adjust preferences. Fifth, we will expand evaluation to real-world financial data with expert hu- man judgment. Finally, we intend to explore inte- gration with regulatory compliance frameworks to de- liver certified explainability suitable for fiduciary ap- plications.

IX. CONCLUSIONS

This work introduces FinAgent, a privacy-preserving multi-agent financial intelligence system that tackles core challenges in automated financial analysis. By incorporating the Model Context Protocol for fine- grained context control, integrating GraphRAG for multi-hop reasoning, and implementing explicit orchestration with comprehensive audit trails, FinAgent demonstrates that advanced financial AI can be built with strong privacy guarantees and explainability.

Our evaluation shows that GraphRAG substantially outperforms vector-only retrieval on multi-hop queries (87% vs. 62% accuracy), MCP achieves per- fect privacy preservation against adversarial prompts (0% leakage), and explainability mechanisms receive high user ratings (4.2/5.0). The system delivers ac- tionable portfolio insights with full provenance track- ing in an average of 3.2 seconds per query.

FinAgent's architecture provides a blueprint for privacy-aware agentic systems in high-stakes do- mains. By separating context governance, reasoning, and explanation into distinct layers, the system pro- motes modularity, testability, and regulatory compli- ance. Although limitations remain in planning flex- ibility and knowledge graph coverage, the working prototype demonstrates technical feasibility and es- tablishes a foundation for production-grade financial AI assistants.

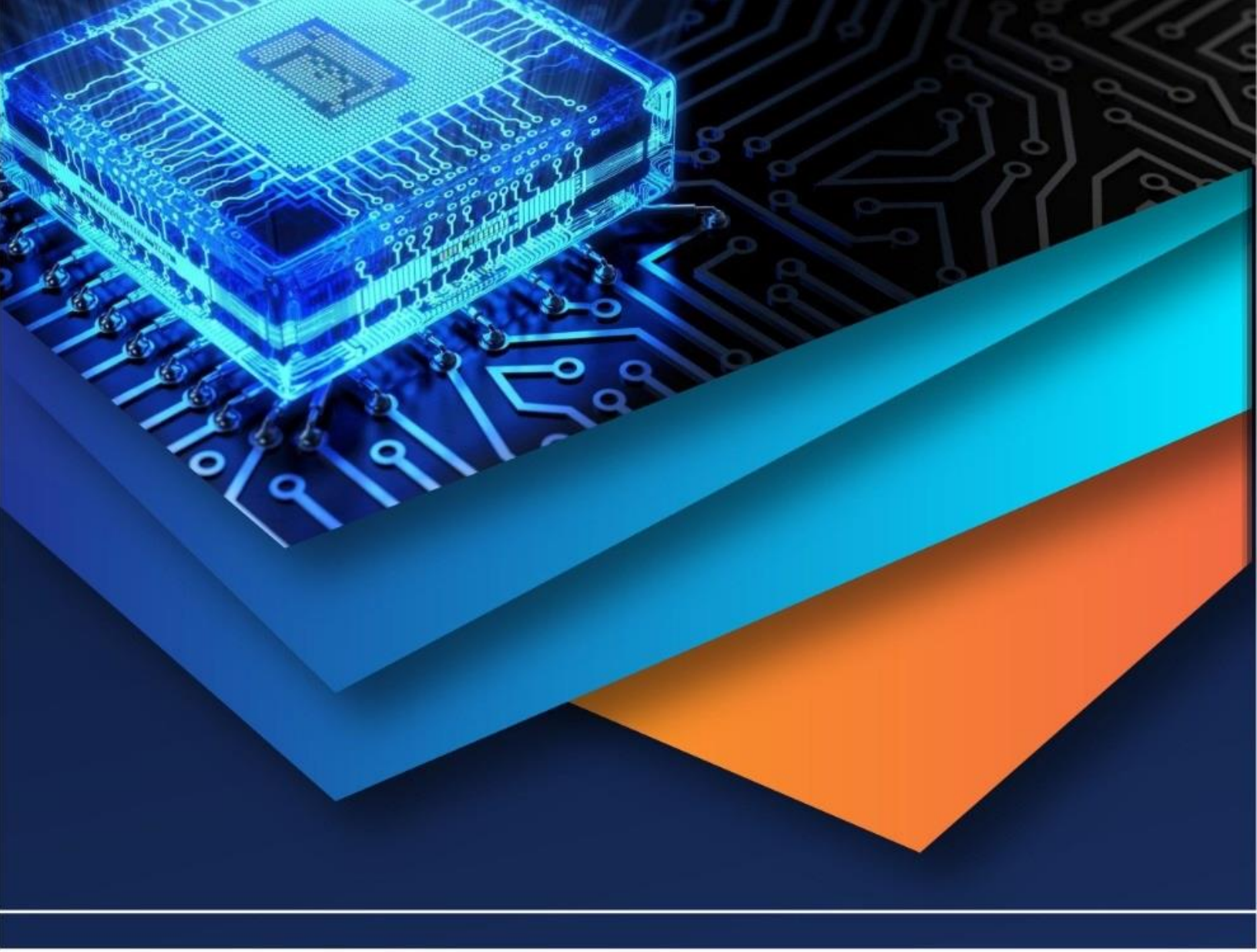
As AI systems increasingly participate in con- sequential decision-making, FinAgent's emphasis on privacy, explainability, and auditability addresses key requirements for responsible deployment. Future work will extend these principles to other high-stakes domains that demand trustworthy automated assistance.

X. ACKNOWLEDGEMENTS

We thank the open-source communities behind Neo4j, Chroma, FastAPI, and spaCy for their excellent tools that enabled this research.

REFERENCES

[1] Dwork, C. and Roth, A. (2014). The algorithmic founda- tions of differential privacy. Foundations and Trends in Theoretical Computer Science, 9(3-4):211–407.
[2] Edge, D., Trinh, H., Cheng, N., Bradley, J., Chao, A., Mody, A., Truitt, S., and Larson, J. (2024). From local to global: A graph rag approach to query-focused sum- marization. arXiv preprint arXiv:2404.16130.
[3] Gravitas, S. (2023). Autogpt: An autonomous gpt-4 exper- iment.
[4] urlhttps://github.com/Significant-Gravitas/AutoGPT. Jiang, Z., Xu, D., and Liang, J. (2021). A deep reinforce-
[5] ment learning framework for the financial portfolio
[6] management problem. IEEE Transactions on Neural Networks and Learning Systems, 32(8):3590–3602.
[7] Kairouz, P., McMahan, H. B., Avent, B., et al. (2021). Advances and open problems in federated learning. Foundations and Trends in Machine Learning, 14(1-2):1–210.
[8] Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Ku¨ttler, H., Lewis, M., Yih, W., Rockta¨schel, T., Riedel, S., and Kiela,
[9] D. (2020). Retrieval-augmented generation for knowledge-intensive nlp tasks. In Advances in Neu- ral Information Processing Systems, volume 33, pages 9459–9474.
[10] Schick, T., Dwivedi-Yu, J., Dess`ı, R., Raileanu, R., Lomeli, M., Zettlemoyer, L., Cancedda, N., and Scialom, T. (2023). Toolformer: Language models can teach themselves to use tools. arXiv preprint arXiv:2302.04761.
[11] Wooldridge, M. (2009). An Introduction to MultiAgent Sys- tems. John Wiley & Sons, 2nd edition.
[12] Wu, S., Irsoy, O., Lu, S., Dabravolski, V., Dredze, M., Gehrmann, S., Kambadur, P., Rosenberg, D., and Mann, G. (2023). Bloomberggpt: A large language model for finance. In Proceedings of the 61st An- nual Meeting of the Association for Computational Linguistics, pages 5023–5043.
[13] Yang, Y., Uy, M. C. S., and Huang, A. (2020). Finbert: A pretrained language model for financial communica- tions. arXiv preprint arXiv:2006.08097.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY