A Human-in-the-Loop Model Context Protocol Architecture for Automating Actimize-Style Fraud Alert Investigations in Financial Institutions

Edward Shevchenko

Independent Scholar

Author Note

Edward Shevchenko is an independent scholar and holds a Doctor of Business Administration and Analytics from the University of the Incarnate Word. His research interests include fraud analytics, enterprise AI governance, and automation in regulated financial institutions. This manuscript reflects a practitioner-scholar perspective focused on practical control design, measurable operational impact, and responsible AI adoption in financial services.

May 2026

Abstract

Financial institutions are under pressure to detect fraud faster, reduce false positives, improve case documentation, and preserve strong governance over customer-impacting decisions. Fraud platforms such as NICE Actimize support alert generation, risk scoring, case management, and financial-crime workflows; however, detection specialists still perform substantial manual work to gather evidence, interpret rules, compare historical behavior, and prepare defensible disposition narratives. This paper proposes a human-in-the-loop architecture that uses Model Context Protocol (MCP) servers to connect large language model (LLM) applications with governed fraud-alert data, rule documentation, historical case outcomes, and approved analytical tools in an Actimize-style environment. The proposed design separates deterministic fraud logic, Python-based feature engineering and similarity analysis, LLM-based evidence synthesis, and human disposition authority. Drawing on recent academic literature, official MCP documentation, financial AI governance guidance, and Actimize-specific agentic-AI materials, the paper argues that MCP-enabled fraud automation should be framed as bounded decision support rather than autonomous fraud adjudication. The manuscript contributes a conceptual architecture, design-science research approach, evaluation plan, governance framework, and implementation roadmap. It also identifies a research gap: recent sources address LLM agents in finance, fraud-detection models, financial explainable AI, MCP security, and Actimize agentic-AI product direction, but limited independent academic work evaluates MCP-enabled, human-reviewed fraud investigations in production-like banking workflows.

Keywords: Model Context Protocol, MCP, Actimize, fraud detection, fraud operations, agentic AI, large language models, human-in-the-loop, financial crime, model governance

A Human-in-the-Loop Model Context Protocol Architecture for Automating Actimize-Style Fraud Alert Investigations in Financial Institutions

Introduction

Financial institutions operate in an environment where fraud threats evolve quickly, regulatory scrutiny remains high, and fraud teams must process large alert volumes with limited time and uneven contextual information. Fraud platforms can generate alerts from rules, models, scores, customer behavior, and channel-level signals, but the operational work of investigating those alerts often remains manual and fragmented. Detection specialists must review alert metadata, determine which rules fired, inspect account and transaction history, compare behavior against a customer baseline, examine prior case outcomes, and document a defensible disposition. These activities require judgment, yet many of the retrieval, comparison, summarization, and case-narrative tasks are repeatable enough to be improved through carefully governed automation.

The emergence of large language models (LLMs), agentic artificial intelligence, and tool-oriented AI architectures has renewed interest in automating knowledge-intensive financial workflows. However, a regulated financial institution cannot credibly deploy an unrestricted LLM to review fraud alerts, access sensitive data, or determine case outcomes. Fraud investigations involve personally identifiable information, potential account restrictions, customer harm, regulatory reporting consequences, operational losses, and reputational risk. Any automation architecture must therefore preserve human accountability, enforce least-privilege access, log evidence provenance, validate model behavior, and prevent unsupported recommendations. The practical question is not whether an LLM can write a plausible fraud narrative. The practical question is whether an enterprise can design a controlled system that

improves investigation efficiency and consistency without creating unacceptable model, privacy, security, or compliance risk.

The Model Context Protocol (MCP) provides a timely architectural lens for this challenge. Anthropic (2024) introduced MCP as an open standard for connecting AI-powered tools with external data sources. The MCP specification describes the protocol as a mechanism for integrating LLM applications with external data sources and tools through standardized capabilities, including resources, prompts, and tools (Model Context Protocol, 2025a). In a fraud-operations context, MCP can be interpreted not merely as a connector, but as a governed interface layer. Properly designed MCP servers could expose narrowly scoped fraud-alert data, rule documentation, historical dispositions, and analytical functions to an LLM while avoiding direct, unrestricted model access to production databases.

This paper proposes a human-in-the-loop MCP-enabled architecture for automating Actimize-style fraud alert investigations. The term Actimize-style is used because NICE Actimize is a prominent financial-crime and fraud platform associated with detection, alert triage, case management, and investigation workflows. Recent NICE materials describe Xceed AI Agents and X-Sight ActOne agentic AI capabilities for fraud and financial-crime prevention, including analyst-in-the-loop decisioning and investigation support (NICE, 2025a, 2025b). Those vendor materials indicate market direction, but they do not substitute for independent academic validation. The gap addressed by this paper is the absence of a rigorous, governed, workflow-centered architecture and evaluation plan for MCP-enabled fraud investigation automation in a regulated financial institution environment.

The paper’s central argument is that MCP-enabled LLM systems are most defensible when they automate evidence gathering, rule interpretation support, historical comparison,

explanation, and recommendation drafting while preserving human authority over final disposition. The LLM should not become the fraud rules engine, the system of record, or the final decision maker. Deterministic rules, validated scoring logic, Python-based analytics, and human review should remain explicit parts of the control design. This bounded-automation framing aligns with recent financial AI governance expectations, MCP security guidance, and the broader explainable-AI literature (Autio et al., 2024; Model Context Protocol, 2025b; NIST, 2023; Yeo et al., 2025).

Research Problem and Contribution

The research problem is that financial institutions need a secure and auditable way to automate fraud investigation support across fragmented enterprise systems while preserving human decision authority. Existing fraud platforms can detect suspicious activity and generate alerts, but investigation remains a multi-step workflow that depends on the specialist's ability to retrieve context, apply rule knowledge, identify historical patterns, and document a defensible conclusion. When historical case knowledge, rule interpretation, and evidence retrieval are inconsistent, case quality, escalation decisions, and handling time also become inconsistent.

The recent literature supports multiple parts of the proposed solution but not the integrated workflow. Peer-reviewed work has reviewed machine learning for financial fraud detection, LLM agents in finance, LLM-assisted fraud detection, and financial explainable AI (Dong et al., 2025; Hacini et al., 2025; Hernandez Aros et al., 2024; Yeo et al., 2025). Authoritative sources have addressed AI governance and financial-sector AI risk (Autio et al., 2024; NIST, 2023; U.S. Department of the Treasury, 2024). MCP sources define a technical and security model for connecting AI systems to tools and data (Anthropic, 2024; Hou et al., 2026; Model Context Protocol, 2025a, 2025b). NICE Actimize sources describe agentic AI direction in

fraud and financial-crime workflows (NICE, 2025a, 2025b). Yet these streams have not been joined into a single evaluated architecture for human-reviewed fraud alert investigation.

This paper makes four contributions. First, it proposes a conceptual architecture that separates source systems, segmented MCP servers, Python analytics, LLM synthesis, human review, and monitoring. Second, it provides a design-science methodology and evaluation plan that goes beyond classification accuracy to include handling time, documentation quality, analyst override behavior, explanation usefulness, and MCP security testing. Third, it maps the architecture to AI governance expectations, including the NIST AI Risk Management Framework, the NIST Generative AI Profile, and financial-sector guidance. Fourth, it defines a research agenda for independent evaluation of Actimize-style agentic fraud investigation rather than reliance on vendor claims alone.

Original Contribution of This Paper

This paper contributes a design-science framework for MCP-enabled fraud investigation support in Actimize-style environments. Unlike prior fraud-detection research that focuses primarily on predictive classification, this paper focuses on the governed workflow surrounding fraud alerts: evidence retrieval, rule interpretation, historical comparison, LLM-based synthesis, human review, and audit-ready feedback monitoring.

The contribution is intentionally practical and controls-oriented. It treats MCP as a governed orchestration layer, not simply as a technical connector, and it treats the LLM as a synthesis and drafting component rather than as the fraud rules engine, the system of record, or the final decision maker. This framing is useful for financial institutions because the operational problem is not limited to detecting suspicious activity. It also includes producing consistent

investigations, preserving evidence, managing queue pressure, reducing manual rework, and documenting decisions in a way that can survive quality assurance, audit, and regulatory review. The paper also contributes a validation path. It explains how the architecture could be evaluated through retrospective historical alert testing, analyst-in-the-loop studies, workflow metrics, statistical comparisons, and MCP-specific security testing. This is important because the paper does not claim empirical performance improvement from a deployed system. Rather, it proposes a research artifact and validation framework that can be tested using historical alerts, analyst studies, and controlled security evaluation.

Research Questions

The proposed study is guided by four research questions. RQ1 asks how MCP servers can enable controlled LLM access to fraud alert data, rule documentation, historical case outcomes, and approved analytical tools in an Actimize-style environment. RQ2 asks whether a hybrid pipeline combining deterministic rules, Python analytics, historical retrieval, and LLM synthesis can improve fraud investigation efficiency without increasing false negatives or weakening human accountability. RQ3 asks which fraud investigation subtasks are most suitable for automation, including triage, evidence assembly, historical comparison, case-note drafting, recommendation generation, and feedback capture. RQ4 asks which governance and security controls are required for safe deployment of an MCP-enabled fraud investigation assistant in a regulated financial institution.

These questions reflect the paper’s emphasis on workflow augmentation rather than autonomous decisioning. A narrow fraud-classification study would ask whether a model can predict fraud labels. A workflow-centered study asks whether an integrated system can help specialists investigate more consistently, retrieve better evidence, explain decisions more clearly,

and preserve auditable control boundaries. That distinction matters because the operational value of AI in fraud departments is not limited to predictive performance. It also includes repeatability, documentation quality, queue management, escalation accuracy, analyst training, and reuse of institutional knowledge.

Literature Review

Recent financial AI research provides a foundation for MCP-enabled fraud investigation, but the relevant literature is distributed across several domains. This section synthesizes six streams: fraud detection automation, LLM agents in finance, LLM-assisted financial-crime screening, MCP interoperability and security, financial explainable AI, and financial AI governance. It then reviews Actimize-specific authoritative materials and identifies the integrated research gap.

Fraud detection and machine learning. Fraud detection has long relied on rules, statistical methods, supervised learning, unsupervised anomaly detection, graph analysis, and risk scoring. Hernandez Aros et al. (2024) conducted a literature review of machine learning techniques for financial fraud detection and examined 104 articles published between 2012 and 2023. The review emphasized that fraud-detection research has matured around model types, datasets, and performance metrics such as accuracy, precision, recall, F1 score, and sensitivity. It also noted persistent challenges such as privacy, misclassification costs, and the difficulty of detecting complex financial fraud patterns. This literature is important because it establishes that predictive fraud modeling is not new. The novelty of the proposed paper must therefore come from workflow orchestration, governance, explanation, and human-in-the-loop investigation support rather than the generic claim that AI can detect fraud.

Hacini et al. (2025) advanced the fraud detection literature by proposing an LLM-assisted fraud detection framework with reinforcement learning. Their study used LLM embeddings to encode heterogeneous transaction data and evaluated the approach on the European Credit Card Fraud and PaySim datasets. This work is directly relevant because it shows that LLMs can participate in hybrid fraud architectures, but it remains primarily model- and benchmark-oriented. Public datasets generally lack the full operational context of enterprise investigations, including rule documentation, customer histories, prior case notes, analyst rationales, procedural guidance, and escalation decisions. The proposed MCP architecture extends this literature by focusing on the investigative workflow surrounding fraud alerts rather than only the classification of transactions.

A second practitioner source from EdEconomy extends this workflow view to graph analytics for account-takeover fraud. Shevchenko (2025) described ATO fraud as a coordinated network problem in which shared devices, IP ranges, credentials, and session relationships reveal attack patterns that are not visible when sessions are scored independently. This perspective supports the proposed architecture's use of Python analytics and historical matching as deterministic evidence-generation services before LLM synthesis. In other words, graph-derived indicators and network relationships should be computed and validated outside the LLM, then passed into the recommendation package as explainable evidence.

LLM agents in finance. Dong et al. (2025) surveyed large language model agents in finance and connected research capabilities with real-world deployment needs. Their review is useful because financial agents require access to domain data, tools, and dynamic environments, but they also face constraints around privacy, coordination, numerical reasoning, real-time adaptation, and institutional reliability. These concerns are highly relevant to fraud operations. A

fraud investigation assistant must not rely on general model knowledge alone; it must retrieve current case context, validate relevant rules, query historical outcomes, and produce an evidence-grounded narrative. MCP is relevant because it offers a standardized way to expose approved resources and tools to an LLM agent without collapsing all controls into a single unrestricted interface.

Xing (2025) provides a complementary architectural precedent through work on heterogeneous LLM agents for financial sentiment analysis. Although sentiment analysis is not fraud investigation, the study's concept of specialized agent roles is transferable. A fraud workflow can similarly be decomposed into specialized modules or agents: one retrieves alert data, another retrieves rule documentation, another compares historical cases, another executes deterministic analytics, and another synthesizes a recommendation for human review. This decomposition is important because it reduces the temptation to build a monolithic LLM investigator. It also supports independent testing, access scoping, and auditability.

Financial-crime screening as an adjacent empirical analogue. Allen and Hatfield (2025) studied whether LLMs can improve sanctions screening in the financial system. Their Federal Reserve working paper is one of the closest empirical analogues to fraud alert triage because sanctions screening involves high false-positive rates, entity comparison, operational review, and regulated decision-making. The authors found meaningful improvements relative to fuzzy matching, including substantial false-positive reduction, but also highlighted latency costs. This supports a cascaded design: simple deterministic methods should handle routine cases, while LLM-based reasoning should be reserved for ambiguous, complex, or high-risk cases. That cascade logic is directly applicable to Actimize-style fraud workflows. The LLM should not be

invoked for every alert if a rule or score can confidently route the case. Instead, the system should apply LLM reasoning where its contextual synthesis provides marginal value.

MCP as an interoperability layer. Anthropic (2024) introduced MCP as an open standard for connecting AI-powered tools with data sources. The MCP specification further defines an architecture in which clients, servers, resources, prompts, and tools enable LLM applications to interact with external systems in a standardized way (Model Context Protocol, 2025a). In a fraud context, resources might include case summaries, rule documents, customer-risk profiles, or historical disposition records. Tools might include approved functions for retrieving transaction windows, computing velocity metrics, searching similar cases, or generating draft notes. Prompts might encode approved investigation procedures and typology-specific checklists. This layered structure makes MCP attractive for enterprise fraud operations because it can organize access to many systems while maintaining task boundaries.

Practitioner-oriented work on EdEconomy similarly frames MCP as a mechanism for grounding generative AI in live enterprise context rather than relying on a generic chatbot interface. Shevchenko (2026) argued that MCP servers can connect AI assistants to enterprise systems such as data warehouses, code repositories, ticketing platforms, data catalogs, observability tools, and internal documentation, while preserving human review and governance controls. This source is useful as a practitioner example because the same pattern translates directly to fraud operations: the fraud assistant should retrieve governed case context, rules, lineage, health signals, and documentation before generating an analyst-facing recommendation.

MCP also expands the threat surface. The official MCP security guidance emphasizes progressive least privilege and warns against token passthrough as an authorization anti-pattern (Model Context Protocol, 2025b). Hou et al. (2026) analyze MCP’s landscape, security threats,

and future directions, treating MCP as an ecosystem with lifecycle risks involving server creation, operation, and update. These sources are critical because fraud operations data is highly sensitive. An MCP-enabled fraud assistant could access customer identifiers, transaction details, case notes, typologies, and risk signals. Without strong access controls, prompt-injection defenses, logging, and write-back restrictions, the architecture could introduce more risk than it reduces.

Financial explainable AI. Explainability is not a cosmetic feature in fraud operations. It is central to analyst trust, quality assurance, model validation, and auditability. Yeo et al. (2025) reviewed financial explainable AI and emphasized the need for transparency in critical financial decisions. Their review supports the idea that recommendations should be accompanied by evidence, provenance, and uncertainty cues rather than opaque labels. For a fraud investigation assistant, an acceptable output cannot simply state that an alert is likely fraud. It must explain which rules fired, what behavior deviated from the customer baseline, which historical cases appear similar, what evidence supports fraud, what evidence supports a false positive, and what uncertainty remains.

Financial AI governance. NIST's AI Risk Management Framework offers a general framework for mapping, measuring, managing, and governing AI risks (NIST, 2023). The NIST Generative AI Profile extends this framework to GenAI-specific concerns, including hallucination, data leakage, misuse, lack of transparency, and third-party dependencies (Autio et al., 2024). Treasury's report on AI in financial services further indicates that AI is used across financial firms and that firms should evaluate legal compliance before deployment and periodically thereafter (U.S. Department of the Treasury, 2024). These sources support a governance-first approach. In a regulated bank, the key question is not only whether the

architecture works, but whether it can be validated, monitored, challenged, audited, and safely limited.

OCC model-risk guidance is relevant but must be applied carefully. The OCC's 2026 bulletin addresses model risk management and provides useful principles for validation, monitoring, and governance, but it explicitly states that generative AI and agentic AI are outside the scope of the guidance (Office of the Comptroller of the Currency [OCC], 2026). This distinction matters. A fraud assistant built with LLMs and MCP cannot rely solely on traditional model-risk language. It should borrow model-risk discipline where appropriate, but it also requires GenAI-specific controls, including prompt testing, retrieval validation, hallucination evaluation, tool-use monitoring, and agentic-action limits.

Actimize-specific materials. Recent NICE materials provide the clearest Actimize-specific evidence that agentic AI is becoming part of fraud and financial-crime operations. NICE (2025a) described Xceed AI Agents as specialized for fraud detection and anti-money laundering workflows, including automation of work routines and decisioning with analysts in the loop. NICE (2025b) described X-Sight ActOne agentic AI and InvestigateAI capabilities that can understand policies and procedures, identify data and risk signals, and engage human oversight when needed. These materials are useful for defining the practical workflow surface, but they are vendor-authored. The absence of independent peer-reviewed evaluation of NICE Actimize plus MCP plus human-in-the-loop fraud investigation is a central research gap.

Synthesis of the literature. The literature supports the components of the proposed architecture, but it does not yet validate the integrated system. Fraud-detection studies provide model and metric foundations. LLM-agent studies explain tool-augmented financial agents. Sanctions-screening work demonstrates the value and limitations of LLMs in adjacent financial-

crime review. MCP sources define a new integration layer and associated security risks. Financial XAI research explains why evidence packages matter. Governance sources establish risk-management expectations. Actimize vendor materials show market direction. The unresolved question is how to combine these elements into a secure, auditable, human-reviewed fraud investigation workflow.

Table 1

Summary of Literature Streams Supporting the Proposed Architecture

Literature stream	Representative sources	Relevance to the paper
Fraud detection automation	Hernandez Aros et al. (2024); Hacini et al. (2025)	Establishes fraud detection as a mature ML domain and supports the need to move beyond classification toward workflow augmentation.
LLM agents in finance	Dong et al. (2025); Xing (2025)	Supports tool-augmented and specialized-agent design patterns for financial tasks.
Financial-crime screening	Allen and Hatfield (2025)	Provides adjacent empirical evidence for LLM use in regulated alert-review contexts and supports cascade design.
MCP integration and security	Anthropic (2024); Model Context Protocol (2025a, 2025b); Hou et al. (2026)	Provides the interoperability and threat-model basis for exposing tools and data to LLM systems.
Financial XAI	Yeo et al. (2025)	Supports evidence-rich, intelligible explanations for human reviewers.
Governance and regulation	NIST (2023); Autio et al. (2024); U.S. Department of the Treasury (2024); OCC (2026)	Frames trustworthiness, validation, compliance review, monitoring, and governance responsibilities.
Actimize-specific vendor direction	NICE (2025a, 2025b)	Shows market movement toward agentic financial-crime investigation and analyst-in-the-loop workflows, but requires independent validation.

Note. The table synthesizes the source base used to develop the conceptual model and research design.

Research Gap

The research gap can be stated directly: recent literature covers MCP as an integration and security protocol, LLM agents in finance, machine learning fraud detection, financial

explainable AI, and Actimize agentic-AI product direction, but there is limited independent research evaluating a secure, MCP-enabled, human-in-the-loop architecture for automating Actimize-style fraud alert investigations in production-like banking workflows. This gap is meaningful because fraud investigation is not merely a prediction task. It is a workflow that combines data retrieval, rule interpretation, policy adherence, historical comparison, evidence documentation, and human accountability.

The proposed paper therefore fills a design and evaluation gap rather than a purely algorithmic gap. It does not claim to introduce a new fraud classifier. Instead, it proposes a controlled enterprise architecture and validation strategy. The architecture is designed to test whether MCP can serve as a secure orchestration layer for fraud investigation support, whether LLMs can improve evidence synthesis and narrative quality, and whether human specialists can use the resulting recommendations without inappropriate overreliance. The paper also contributes by specifying MCP-specific threat tests, which are underdeveloped in adjacent fraud-automation studies.

Conceptual Framework

The conceptual framework rests on five principles. First, deterministic rules and validated fraud logic remain first-class controls. In an Actimize-style environment, rules, scores, thresholds, typologies, and alert-generation logic should not be replaced by a general LLM. The LLM may interpret and explain rule triggers, but it should not invent new decision criteria or override validated logic without governance approval.

Second, MCP should operate as a governed access layer. A well-designed MCP server does not simply give an LLM broad database access. It exposes approved resources and tools with specific scopes, access checks, logging, and rate limits. For example, a fraud-alert MCP

server may provide read-only alert metadata for cases assigned to a specialist, while a historical-case MCP server may return de-identified similarity results rather than unrestricted case notes.

Third, deterministic computation should be handled by analytical services rather than language generation. Python or comparable services should calculate transaction velocity, deviation from customer baseline, feature values, similarity scores, peer comparisons, and graph-based link indicators. Those outputs can then be passed to the LLM for synthesis. This separation reduces hallucination risk and supports validation because analytical functions can be versioned, tested, and monitored independently.

Fourth, the LLM should produce evidence-grounded explanations and recommendations rather than final decisions. Its role is to assemble context, compare evidence, identify uncertainty, and draft a recommendation package. The recommendation should be structured, traceable, and easy for the specialist to review. Unsupported assertions should be flagged or prevented by output validation.

Fifth, the human detection specialist retains final disposition authority. Human review is not an afterthought; it is a core control. The system should capture whether the analyst accepted, modified, rejected, or escalated the recommendation. Those feedback signals are essential for model monitoring, trust calibration, training, and continuous improvement.

Proposed MCP-Enabled Actimize-Style Architecture

The proposed architecture consists of six layers: source systems, segmented MCP servers, Python analytics, LLM investigation synthesis, human review, and governance monitoring. Figure 1 illustrates the high-level workflow. The architecture is intentionally modular because modularity supports least privilege, independent validation, and staged

implementation. Each component has a specific responsibility and should be evaluated separately before integration.

Source-system layer. The source-system layer includes Actimize-style alert data, case management records, transaction history, customer profiles, account relationships, channel activity, device data, prior fraud claims, final dispositions, and rule documentation. These data sources may reside in Actimize, an enterprise data warehouse, a fraud data mart, cloud data platforms, document repositories, or case management systems. The proposed system should not require the LLM to connect directly to all source systems. Instead, source access should occur through governed views, APIs, or MCP servers that enforce scope and record retrieval events.

MCP server layer. The MCP server layer should be segmented by function. A fraud-alert server retrieves alert metadata, rule triggers, case status, and assigned specialist context. A rule-and-procedure server retrieves standard operating procedures, rule definitions, fraud typologies, and escalation guidance. A historical-case server performs approved searches over prior cases and returns relevant comparisons. An analytics server exposes deterministic Python functions for feature engineering, similarity matching, anomaly scoring, entity linking, and evidence packaging. A controlled write-back server stages case notes or recommendations for human approval rather than allowing direct autonomous closure. An audit server records prompts, tool calls, retrieved context, model outputs, analyst actions, and final disposition.

Python analytics layer. The Python analytics layer is responsible for reproducible computation. It can calculate customer baselines, compare current activity to historical behavior, identify rapid changes in transaction frequency or amount, examine peer group deviations, compute similarity between current and prior cases, and identify linkages between accounts, devices, IP addresses, phone numbers, addresses, or beneficiaries. The layer can also produce a

structured evidence object for the LLM. That object should include quantitative values, source references, and confidence indicators. By placing computation in Python rather than in the LLM, the architecture improves testability and reduces the risk of fabricated calculations.

LLM investigation synthesis layer. The LLM receives the evidence object, relevant rule documentation, historical case summaries, and an approved prompt template. It drafts a recommendation package with a consistent structure: alert summary, triggered rules, customer context, behavioral comparison, similar historical cases, evidence supporting fraud, evidence supporting false positive, missing information, recommended action, and human review checklist. The LLM should also identify uncertainty and avoid unsupported claims. When a claim cannot be traced to retrieved evidence, the output should either exclude it or flag it for review.

Human review layer. The detection specialist reviews the recommendation package and makes the final decision. The specialist can accept the recommendation, modify the rationale, reject it, request additional information, or escalate the case. The system should capture the specialist's final action and the degree of modification. This creates a feedback dataset that can be used to monitor model performance, identify drift, improve prompts, and determine whether the assistant is being overtrusted or underused.

Governance and monitoring layer. Governance applies across the full architecture. It includes identity and access management, data minimization, role-based authorization, audit logging, model validation, prompt management, retrieval quality controls, output testing, override monitoring, and incident response. It also includes security testing for prompt injection, poisoned retrieval content, over-permissioned tool access, and unauthorized write-back attempts.

Table 2

Roles and Control Concerns by Architecture Component

Architecture component	**Primary responsibility**	**Primary control concern**
Actimize-style source systems	Generate alerts, store cases, preserve rule triggers and investigation records.	Data quality, lineage, privacy, and system-of-record integrity.
Fraud-alert MCP server	Retrieve alert metadata and assigned-case context.	Role-based access and read-only scoping.
Rules and procedures MCP server	Retrieve approved rules, SOPs, typologies, and escalation guidance.	Document currency and prevention of outdated procedure use.
Historical-case MCP server	Return comparable prior cases and disposition patterns.	De-identification, retrieval quality, and bias control.
Python analytics engine	Compute baselines, features, similarity, and anomaly indicators.	Versioning, validation, reproducibility, and monitoring.
LLM investigation assistant	Synthesize evidence, draft rationale, identify uncertainty, and recommend next action.	Hallucination, overreliance, explainability, and output validation.
Human detection specialist	Approve, modify, reject, or escalate recommendations.	Accountability, quality assurance, and override capture.
Monitoring and audit layer	Log prompts, tool calls, sources, recommendations, and final outcomes.	Audit completeness, drift monitoring, and incident response.

Note. The modular architecture supports least privilege, independent validation, and staged deployment.

Figure 1

Proposed MCP-enabled human-in-the-loop fraud investigation architecture.

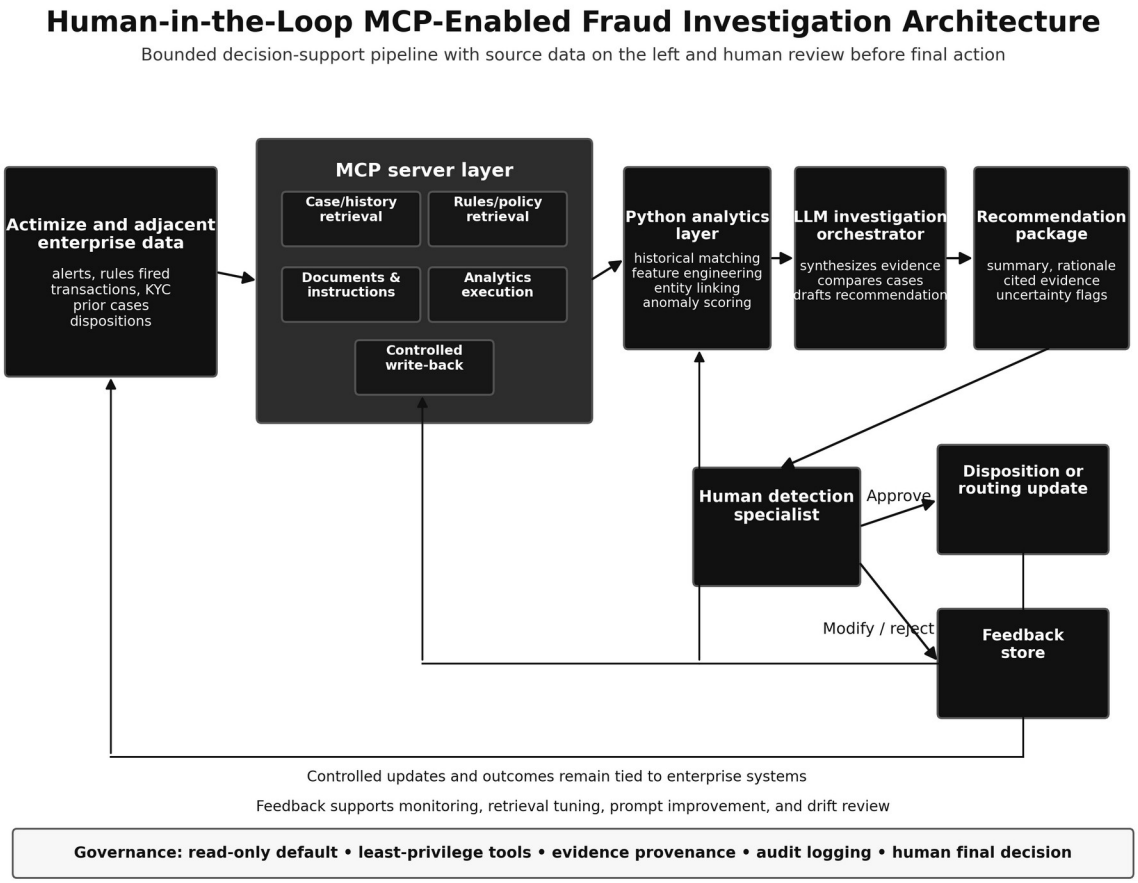


Figure 1 presents the architecture as a bounded decision-support pipeline with Actimize and adjacent enterprise data placed at the beginning of the workflow. The diagram shows a controlled sequence in which MCP enables governed access, Python performs reproducible analytics, the LLM synthesizes evidence, and the human detection specialist retains final authority. The feedback path illustrates monitoring and learning from outcomes, not autonomous case closure.

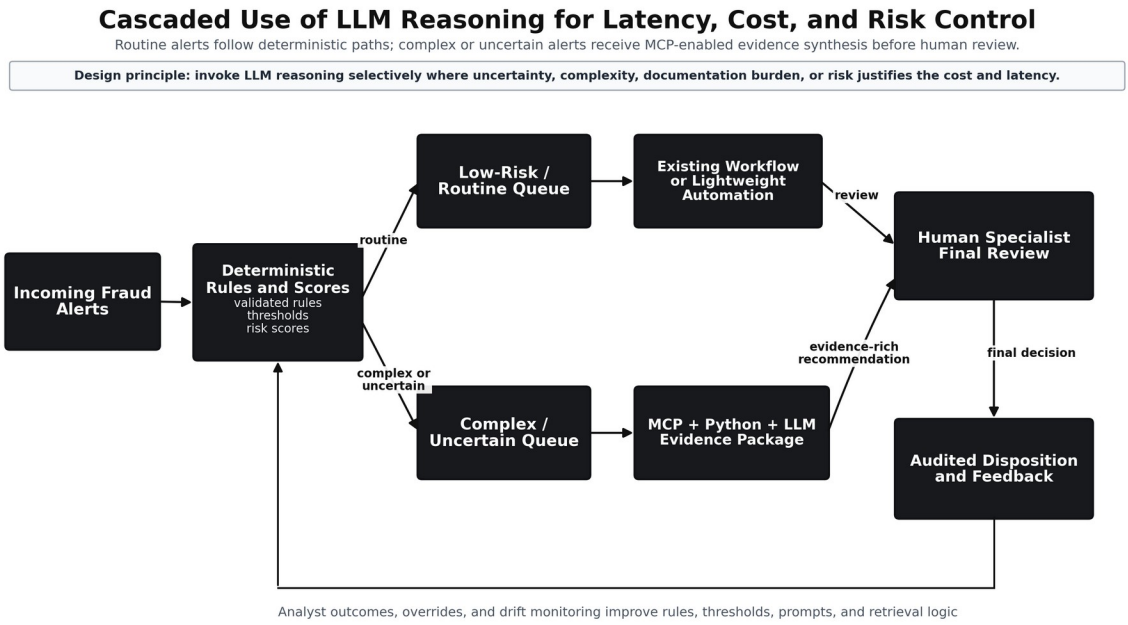
Cascaded Automation Model

The architecture should use a cascaded automation model rather than invoking LLM reasoning for every alert. Allen and Hatfield (2025) showed that LLMs can improve financial-crime screening performance but also introduce latency tradeoffs. In fraud operations, latency, cost, and operational risk matter. Routine alerts that can be handled confidently by deterministic rules or existing scores should not necessarily consume LLM resources. Conversely, complex alerts with conflicting evidence, unusual historical context, or high documentation burden may benefit substantially from LLM-based synthesis.

Figure 2 illustrates a cascaded model. The first stage routes incoming alerts through existing rules, scores, and thresholds. Low-risk or routine alerts can be routed through existing workflows or lighter automation. Complex, uncertain, high-risk, or high-value alerts are sent to the MCP-enabled evidence engine and LLM assistant. Human review remains the final stage for all customer-impacting decisions. This design balances efficiency and control. It also reduces the risk of automation bias by preventing the LLM from becoming the default answer for every case.

Figure 2

Cascaded use of LLM reasoning for latency, cost, and risk control.



Methodology

A design-science research approach is appropriate for this paper because the primary contribution is an artifact: a proposed architecture and evaluation framework for MCP-enabled fraud investigation. Design science is appropriate when research produces and evaluates an artifact intended to solve an organizational problem. In this case, the artifact is not a single model but a socio-technical system that includes MCP servers, analytical functions, prompt templates, evidence packages, human-review controls, and monitoring processes.

The proposed methodology includes four phases: architecture specification, retrospective evaluation, human-in-the-loop evaluation, and MCP security evaluation. A fifth optional phase involves longitudinal monitoring if the system is piloted over time. Each phase addresses a different aspect of validity. Architecture specification evaluates whether the system is conceptually coherent and governable. Retrospective evaluation tests recommendation quality

against historical outcomes. Human-in-the-loop evaluation tests whether specialists actually benefit from the assistant. Security evaluation tests whether MCP integration introduces unacceptable risk. Longitudinal monitoring tests whether performance and trust remain stable over time.

Phase 1: architecture specification. The first phase documents the system boundary, data sources, MCP server scopes, analytical functions, prompt templates, evidence package schema, human review workflow, and audit logs. This phase should produce an architecture diagram, data-flow inventory, control matrix, model inventory entry, and operational procedure. The key output is a system design that can be reviewed by fraud operations, compliance, model risk, information security, privacy, and technology stakeholders.

Phase 2: retrospective historical alert study. The second phase uses de-identified historical alerts and final dispositions. The system processes each case as if it were being reviewed in real time. Inputs include alert metadata, rule triggers, transaction history, customer baseline features, historical case summaries, and rule documentation. Outputs include a recommendation, rationale, uncertainty flags, and suggested next action. Metrics include agreement with final disposition, precision, recall, false-negative rate, false-positive rate, area under the precision-recall curve, recommendation confidence calibration, evidence completeness, and narrative quality. Because fraud data is often imbalanced, precision-recall metrics should be emphasized over accuracy alone.

Phase 3: human-in-the-loop analyst study. The third phase evaluates analyst performance under multiple conditions. A baseline group performs investigations using the current workflow. A second group receives recommendation-only outputs. A third group receives evidence-rich recommendation packages that include source citations, rule explanations, historical

comparisons, and uncertainty flags. The study should measure investigation time, documentation quality, final decision agreement with known outcomes, escalation appropriateness, analyst trust, perceived workload, override rate, and error severity. The hypothesis is that evidence-rich outputs will outperform recommendation-only outputs because they support critical review rather than blind acceptance.

Phase 4: MCP security and reliability assessment. The fourth phase tests MCP-specific risks. Test cases should include prompt-injection attempts hidden in case notes or documents, poisoned retrieval records, attempts to access unauthorized customer data, excessive tool calls, unauthorized write-back attempts, and token-handling failures. Metrics include unauthorized retrieval rate, incorrect tool invocation rate, scope-amplification incidents, write-action failures, audit-log completeness, and recovery from tool errors. The design should compare a segmented least-privilege MCP configuration with a hypothetical monolithic broad-access configuration. The expectation is that segmentation will reduce risk and improve auditability.

Phase 5: longitudinal monitoring. If piloted in a controlled environment, the system should be monitored over time. Metrics should include recommendation acceptance rate, modification rate, override rate, false-negative severity, queue impact, latency, cost per case, subgroup performance, retrieval failures, prompt failures, and explanation drift. Analyst feedback should be used to improve retrieval ranking, prompt templates, rule-document coverage, and analytical functions. However, any learning loop should be governed. Feedback should not automatically update production behavior without review.

The proposed hypotheses are as follows. H1: An MCP-enabled evidence package will reduce average alert handling time compared with the baseline workflow. H2: Evidence-rich recommendation packages will improve analyst decision consistency compared with

recommendation-only outputs. H3: Cascaded LLM use will provide a better latency-cost-risk tradeoff than always-on LLM use. H4: Segmented MCP servers will reduce unauthorized data exposure risk compared with monolithic MCP access. H5: Analyst feedback and override monitoring will improve recommendation quality and trust calibration over time.

Statistical Evaluation Plan

A stronger empirical version of this study should use statistical testing that reflects both the imbalanced nature of fraud data and the repeated-measures structure of analyst work. For historical alert testing, the main outcome variables should include recommendation agreement with final disposition, false-negative rate, false-positive rate, precision, recall, F1 score, calibration, and area under the precision-recall curve. Because fraud cases are typically imbalanced, accuracy should not be used as the primary performance metric. Precision-recall measures are more appropriate because they focus attention on the minority class and the tradeoff between missed fraud and unnecessary friction.

For workflow-efficiency outcomes, the study should compare baseline handling time with assistant-supported handling time. If the same analysts review comparable cases across conditions, paired t tests may be appropriate when the distribution is approximately normal. If handling-time distributions are skewed, which is common in operations data, the Wilcoxon signed-rank test or bootstrapped confidence intervals should be used. Median handling time, interquartile range, and tail outcomes such as the 90th percentile handling time should be reported because fraud queues are often affected by outlier cases.

Agreement between the assistant recommendation and human disposition can be measured with Cohen’s kappa when the outcome categories are categorical, such as approve, reject, escalate, or request additional information. When multiple analysts score case-note quality or explanation

usefulness, inter-rater reliability should be reported to determine whether quality judgments are consistent. If the study compares analysts, alert typologies, or complexity levels, mixed-effects models can be used to account for repeated observations by analyst and case type. Logistic regression may also be useful for estimating whether the assistant changes the odds of escalation, override, or false-negative outcomes after controlling for alert complexity.

The statistical plan should not treat time savings as the only success measure. A system that reduces handling time but increases false negatives, weakens documentation, or creates overreliance would not be acceptable in a regulated financial institution. Therefore, the preferred evaluation design should combine operational metrics, predictive metrics, documentation-quality ratings, analyst trust measures, and security outcomes. This mixed evaluation approach is consistent with the paper's central position that MCP-enabled fraud automation must be judged as a socio-technical workflow, not only as a model-performance exercise.

Evaluation Metrics and Measures

The evaluation should use both predictive and workflow metrics. Predictive metrics include precision, recall, false-positive rate, false-negative rate, F1 score, area under the precision-recall curve, calibration, and agreement with historical dispositions. These metrics are necessary but insufficient. Fraud operations also require workflow metrics such as median handling time, time to evidence package, queue aging, case-note completeness, escalation quality, and rework rate.

Human-factors metrics are also essential. The assistant could appear accurate but still create poor outcomes if analysts overtrust it, ignore uncertainty, or spend more time reviewing model output than conducting the investigation. Measures should include trust calibration, perceived workload, usefulness ratings, override frequency, modification patterns, and

qualitative analyst feedback. The evaluation should also examine whether the assistant helps new analysts more than experienced specialists, because one potential benefit is faster onboarding through consistent rule explanations and historical comparisons.

Security metrics should not be treated as secondary. An MCP-enabled system connected to fraud data creates a powerful pathway for data access and tool execution. Security evaluation should measure unauthorized data exposure, failed access checks, prompt-injection success, poisoned-retrieval influence, write-back prevention, audit completeness, and alerting effectiveness. The most important question is not only whether the system improves productivity, but whether it does so while maintaining bank-grade controls.

Table 3

Proposed Evaluation Measures for an MCP-Enabled Fraud Investigation Assistant

Evaluation area	Example measures	Purpose
Predictive performance	Precision, recall, F1, AUCPR, false-negative rate, calibration.	Determine whether recommendations align with known outcomes and risk appetite.
Workflow efficiency	Median handling time, queue aging, time to evidence package, rework rate.	Determine whether the assistant improves operational productivity.
Documentation quality	Completeness, source coverage, rule explanation quality, uncertainty clarity.	Determine whether case narratives are stronger and more auditable.
Human factors	Trust calibration, override rate, perceived workload, usefulness ratings.	Determine whether specialists can use the assistant appropriately.
MCP security	Unauthorized retrievals, prompt-injection success, write-back failures, audit gaps.	Determine whether MCP access creates unacceptable security risk.
Longitudinal monitoring	Drift, override drift, latency, cost per case, subgroup performance.	Determine whether system behavior remains stable over time.

Note. Workflow, security, and human-factors metrics are included because predictive accuracy alone is insufficient for regulated fraud operations.

Governance Framework

The governance model should be built around the NIST AI Risk Management Framework functions of govern, map, measure, and manage (NIST, 2023). The govern function

requires accountability structures, policies, roles, and oversight. In this architecture, governance should define who owns MCP servers, who approves prompt templates, who validates analytical functions, who monitors recommendations, and who approves access to sensitive fraud data. The map function requires understanding the context and risks of use. Fraud investigation support should be classified as a high-sensitivity operational decision-support use case because it can influence customer-impacting actions and regulatory processes. The measure function requires testing performance, robustness, explainability, and risk. The manage function requires risk treatment, monitoring, and incident response.

The NIST Generative AI Profile adds controls specific to GenAI risks (Autio et al., 2024). For this paper, the most relevant risks are hallucination, data leakage, misuse, opacity, overreliance, and third-party dependency. Hallucination controls require evidence-grounded output templates and automated checks for unsupported claims. Data leakage controls require data minimization, masking, and role-based retrieval. Misuse controls require restricted tools and human approval for write-back. Opacity controls require source provenance and explanation structure. Overreliance controls require uncertainty flags, analyst training, and override monitoring. Third-party dependency controls require vendor-risk assessment and contingency planning.

Treasury’s report on AI in financial services provides additional institutional context. The report indicates that AI is used across financial functions and emphasizes review of legal compliance before deployment and periodic reevaluation afterward (U.S. Department of the Treasury, 2024). For the proposed architecture, this means governance review should occur before pilot deployment, before expansion to new fraud typologies, before enabling write-back capabilities, and after material model or prompt changes. Compliance review should include

privacy, fair treatment, consumer harm, record retention, complaint response, and regulatory reporting implications.

OCC model-risk guidance should inform validation and monitoring, but it should not be overstated. The OCC (2026) bulletin provides important model-risk principles but states that GenAI and agentic AI are outside its direct scope. Therefore, the proposed system should use model-risk discipline while also adding GenAI-specific controls. Analytical functions should be validated like models or quantitative tools where applicable. LLM prompts and outputs should be evaluated using test suites, red-team cases, and human review studies. MCP servers should be reviewed as part of information-security and access-control governance.

The governance model should include a three-lines-of-defense structure. First-line fraud operations owns the use case, business procedures, and daily monitoring. Second-line risk, compliance, privacy, and model risk provide challenge, policy interpretation, validation standards, and control testing. Third-line audit reviews the control environment, evidence retention, and governance effectiveness. Technology and information security cut across all three lines because MCP server design, access management, and logging are technical controls that directly affect business risk.

A minimum control set includes role-based access, least-privilege MCP scopes, read-only access by default, segregation of retrieval and write-back tools, data masking, prompt and tool-call logging, evidence provenance, human approval for disposition, quality assurance sampling, model and prompt versioning, incident response, third-party review, red-team testing, and ongoing monitoring of analyst override behavior. These controls should be documented before production deployment.

Table 4

Mapping the Proposed Architecture to NIST AI RMF Functions

NIST AI RMF function	**Application to MCP-enabled fraud investigation**
Govern	Define ownership, policies, approval gates, role responsibilities, audit expectations, and model-risk challenge.
Map	Classify the use case, identify stakeholders, document data flows, and assess customer-impacting risks.
Measure	Test recommendation quality, explanation quality, security controls, hallucination risk, and analyst reliance.
Manage	Prioritize risks, implement controls, monitor drift, respond to incidents, and govern expansion.

Note. The mapping is intended as a practical governance scaffold rather than a substitute for institution-specific risk review.

Ethical and Customer-Impact Considerations

Fraud investigation automation has direct customer-impact implications. A false positive may delay a legitimate transaction, restrict account access, create customer frustration, or cause financial inconvenience. A false negative may allow fraud losses, create regulatory exposure, and reduce customer trust in the institution's ability to protect accounts. The ethical design challenge is to improve investigation speed and consistency without shifting unreasonable risk to customers or analysts.

The proposed architecture reduces but does not eliminate ethical risk. Human-in-the-loop review is necessary because fraud decisions often require judgment, context, and proportionality. A recommendation package should make evidence easier to evaluate, not create pressure to accept the AI output. The interface should therefore present uncertainty, missing evidence, and countervailing facts. If the evidence supports both fraud and false-positive interpretations, the system should make that tension visible to the specialist rather than forcing a confident label.

Historical case data must also be treated carefully. Prior dispositions can contain human error, inconsistent documentation, or embedded bias. If a historical-case server retrieves prior cases without quality controls, the assistant may reinforce past mistakes. For that reason, historical comparisons should be accompanied by source quality indicators, recency, typology match, and data completeness. The system should avoid treating prior outcomes as unquestioned truth.

Customer-impacting actions should remain subject to human approval and institutional policy. The assistant may draft a recommendation, but account restriction, payment decline, customer contact, regulatory referral, or case closure should require explicit human action unless a separate, formally approved automation policy exists. Institutions should also preserve review and appeal paths where customer harm is possible. These controls support fairness, transparency,

and accountability while still allowing the institution to benefit from faster evidence gathering and more consistent case narratives.

Threat Model for MCP-Connected Fraud Systems

A threat model is necessary because MCP-connected fraud systems can access sensitive data and execute tools. The first threat is prompt injection through retrieved content. A case note, document, or historical narrative could contain text instructing the LLM to ignore rules, retrieve unrelated data, or change a recommendation. The system should treat retrieved content as untrusted input and should separate system instructions from data content. It should also use output validation and tool-call policies that prevent the model from following data-originating instructions.

The second threat is overbroad authorization. If an MCP server exposes unrestricted database queries, the LLM or a compromised client could retrieve data beyond the assigned case or analyst role. Least privilege requires scoped tools, parameter validation, and identity-aware access. The assistant should only retrieve cases assigned to the reviewer or records needed for approved comparison. Historical similarity tools can return de-identified summaries when full case details are unnecessary.

The third threat is unauthorized write-back. Fraud case systems contain customer-impacting actions. A write-back MCP server should not permit autonomous case closure, account restriction, regulatory filing, or customer messaging. Draft notes and staged recommendations may be allowed only after explicit human approval. Write-back events should be logged with the analyst identity, model output version, and evidence package identifier.

The fourth threat is poisoned retrieval or corrupted historical examples. If a historical case repository contains inaccurate dispositions or biased case notes, the assistant may retrieve

misleading analogues. Retrieval quality should be evaluated, and the system should avoid treating prior cases as unquestioned truth. Historical cases should be weighted based on final disposition quality, recency, typology match, and data completeness.

The fifth threat is automation bias. Analysts may accept confident recommendations even when evidence is weak. The system should display uncertainty, highlight missing data, and require the specialist to confirm key evidence before final disposition. Quality assurance reviews should examine whether analysts are over-accepting AI recommendations or failing to challenge weak rationales.

The sixth threat is model drift and fraud adaptation. Fraudsters change behavior in response to controls, and a static recommendation system may degrade. Drift monitoring should track false negatives, override rates, confidence calibration, subgroup patterns, and typology-specific performance. Any observed deterioration should trigger review of rules, features, prompts, retrieval sources, or model configuration.

Actimize-Style Workflow Application

A practical workflow begins when an Actimize-style system generates an alert. The alert includes a case identifier, customer or account identifiers, rule triggers, alert score, transaction details, channel indicators, and case status. In the current-state workflow, the detection specialist may need to move between the alert screen, transaction history, customer profile systems, prior case records, rule documentation, and policy guidance. The MCP-enabled assistant reduces fragmentation by retrieving approved context and packaging it in a standard review format.

For example, an account-takeover alert might trigger because of a new device, unusual login geography, rapid change in beneficiary, and high-value external transfer. The fraud-alert MCP server retrieves the alert details and relevant transactions. The rules server retrieves the rule

descriptions and account-takeover investigation checklist. The historical-case server searches for similar patterns among prior account-takeover and false-positive cases. The Python analytics layer compares the customer's recent activity against historical baselines, computes deviations, and identifies similar cases. The LLM then drafts a structured recommendation.

The recommendation package might state that the alert is elevated because the transaction sequence is inconsistent with the customer's prior activity, the device is new, the transfer beneficiary has no prior relationship, and similar historical cases were frequently confirmed as account takeover. It should also list evidence supporting a false-positive interpretation, such as successful multifactor authentication or a prior pattern of legitimate high-value transfers, if applicable. The assistant should not close the case. The human specialist reviews the evidence, confirms system data, considers customer context, and determines whether to approve, decline, escalate, or request additional information.

This workflow illustrates why the LLM should function as a synthesis layer. The rules fired because deterministic logic detected risk indicators. Python computed deviations and similarity. MCP provided controlled retrieval. The LLM organized evidence and drafted a rationale. The specialist made the decision. Each layer can be reviewed independently, which is essential for governance and auditability.

Illustrative Alert Scenario

Consider an account-takeover alert generated after a customer logs in from a new device, changes a beneficiary, and initiates a high-value external transfer from an unusual geography. In a traditional workflow, the detection specialist may need to open the alert, review the transaction sequence, inspect recent login behavior, search customer history, locate the applicable account-

takeover procedure, compare similar cases, and write a case narrative. Each step is reasonable, but the workflow is fragmented and time consuming.

In the proposed architecture, the fraud-alert MCP server retrieves the alert details, triggered rules, transaction window, and case status. The rules and procedures server retrieves the account-takeover review checklist and escalation guidance. The historical-case server searches for prior alerts involving similar combinations of new device, beneficiary change, geography shift, and external transfer activity. The Python analytics layer compares the current behavior against the customer's historical baseline, evaluates velocity and amount deviations, and identifies entity links such as shared devices, IP addresses, phone numbers, beneficiaries, or addresses. Graph-style relationship evidence is useful here because account-takeover activity often appears as a network pattern rather than a single isolated event (Shevchenko, 2025).

The LLM investigation orchestrator then produces a recommendation package. The package might explain that the alert is elevated because the customer's current activity differs sharply from prior behavior, the device has no known history with the account, the beneficiary is new, and similar historical cases were often confirmed as account takeover. It should also identify facts that could support a false-positive interpretation, such as recent travel indicators, successful multifactor authentication, or prior legitimate high-value transfers. The detection specialist reviews the package, confirms the relevant evidence, and decides whether to approve, modify, reject, escalate, or request additional information.

This scenario illustrates the intended division of labor. MCP retrieves governed enterprise context. Python computes deterministic signals. The LLM organizes the evidence into an understandable narrative. The specialist remains accountable for final disposition. That division is what makes the architecture more credible than a fully autonomous fraud agent.

Implementation Roadmap

A phased implementation roadmap reduces risk and improves stakeholder confidence. Phase 0 is use-case selection and governance approval. The institution should begin with a narrow fraud typology and a clearly bounded pilot. Suitable initial use cases include read-only alert summarization, historical case comparison, or draft case-note generation. The pilot should avoid autonomous disposition or customer-impacting actions.

Phase 1 is data and access preparation. The team should identify source systems, define approved data fields, create governed views, map data lineage, and classify sensitive fields. Access should be tied to analyst roles and case assignments. Historical data should be de-identified where possible. Rule documentation should be curated so the assistant retrieves approved policy language rather than outdated informal notes.

Phase 2 is MCP server development. The first MCP servers should be read-only and segmented by function. A fraud-alert server, rules server, historical-case server, and analytics server can be developed before any write-back server. Each tool should have narrow parameters, validation checks, logging, and rate limits. Security review should occur before user testing.

Phase 3 is evidence package design. Fraud operations should define the standard output template, including alert summary, triggered rules, customer context, historical comparison, evidence supporting fraud, evidence supporting false positive, uncertainty, recommendation, and human checklist. The template should be tested with specialists to ensure it supports decision-making rather than adding cognitive burden.

Phase 4 is retrospective testing. The system should process historical alerts and compare recommendations with final outcomes and analyst notes. The team should identify where the

assistant agrees, disagrees, hallucinates, omits key evidence, or provides helpful explanations. Prompt templates, retrieval filters, and analytical functions should be refined before live use.

Phase 5 is a controlled human-in-the-loop pilot. A small group of specialists should use the assistant on selected cases while all final decisions remain human. The pilot should measure productivity, recommendation usefulness, quality, overrides, and analyst trust. It should also monitor whether the assistant creates unexpected work, delays, or confusion.

Phase 6 is controlled write-back and monitoring. Only after evidence-package quality is proven should the institution consider staged case-note write-back. Even then, write-back should require human approval. Ongoing monitoring should track performance, drift, security events, user feedback, and compliance issues. Expansion to additional fraud typologies should require renewed governance review.

Discussion

The proposed architecture offers several potential benefits. The first is reduced handling time. Detection specialists spend substantial time gathering context and writing case notes. Automating retrieval, summarization, and evidence packaging can allow specialists to focus on judgment. The second benefit is consistency. Standardized evidence packages can reduce variation in how analysts document cases and interpret rule triggers. The third benefit is improved knowledge reuse. Historical cases and prior dispositions often contain valuable institutional knowledge, but that knowledge is difficult to search manually. MCP-enabled retrieval and LLM synthesis can make historical patterns more accessible.

The fourth benefit is better onboarding and training. New specialists often need time to learn rules, typologies, procedures, and common false-positive patterns. An assistant that explains why rules fired and compares cases to prior outcomes can function as a training aid. The

fifth benefit is improved governance evidence. If the system logs tool calls, retrieved sources, prompts, recommendations, human edits, and final dispositions, it can create a richer audit trail than many manual workflows.

The risks are equally important. The system could increase harm if analysts overtrust recommendations, if data retrieval is wrong, if historical cases are biased, if prompts are manipulated, or if write-back tools are misused. The architecture therefore emphasizes bounded automation. LLMs should not be positioned as independent fraud investigators. They should be treated as tools that synthesize evidence under strict control. Human specialists remain responsible for final decisions, and governance teams remain responsible for validating and monitoring system behavior.

The architecture also has implications for vendor strategy. NICE Actimize's recent agentic-AI materials show that vendors are moving toward AI-assisted investigation, but financial institutions should not rely only on vendor claims. Independent validation should test whether such tools actually reduce time, improve quality, and maintain controls in a specific institutional environment. The MCP concept may be useful even when the underlying fraud platform has native AI capabilities because MCP can connect broader enterprise context, including policy repositories, data warehouses, model documentation, quality controls, and audit systems.

A key insight is that the highest-value use cases may not be the ones that fully automate fraud detection. Instead, the strongest early use cases are likely evidence retrieval, rule explanation, case summarization, similar-case comparison, and draft narratives. These tasks are repetitive, context-heavy, and documentation-heavy. They also preserve a clear human review

point. This makes them better candidates for early deployment than autonomous final adjudication.

This operational framing is consistent with a practitioner-scholar view of AI adoption in financial institutions: automation should improve speed and consistency, but it must also strengthen evidence quality, accountability, and control transparency. A system that saves time but weakens explainability or auditability would not be a successful fraud operations solution. The target state is a workflow that helps specialists work faster and more consistently while making the final decision easier to review, defend, and govern.

Fraud leaders should also treat the assistant as an operational control object, not only as a technology tool. Each recommendation should be traceable to a case, a rule, a retrieved source, an analytical output, and a human action. The feedback loop should capture whether the recommendation was accepted, modified, rejected, or escalated. Over time, those signals can show where the assistant creates value, where it creates friction, and where prompts, retrieval sources, or analytical functions require remediation.

A practical implementation should begin with narrow, high-value tasks. The strongest starting point is not autonomous disposition; it is read-only evidence packaging, similar-case retrieval, rule explanation, and draft case-note generation. These tasks are documentation-heavy, repeatable, and valuable to specialists, but they still leave the final judgment with the human reviewer. That sequencing also gives governance teams a safer path to evaluate the assistant before any write-back or customer-impacting capability is enabled.

Practical Implications for Fraud Operations

The proposed architecture has practical implications for fraud operations leaders, technology teams, and governance stakeholders. Its value is not limited to whether an LLM can classify an

alert. The larger operational opportunity is to reduce the manual friction between alert generation, evidence retrieval, rule interpretation, historical comparison, and case documentation. In many institutions, those steps are distributed across different tools, teams, and data environments. MCP-enabled orchestration can help standardize that workflow without removing the specialist from the decision.

Limitations

This manuscript is a conceptual and design-oriented study rather than an empirical validation study. It proposes an architecture and evaluation plan but does not present original test results from a live financial institution. The next stage should involve retrospective testing with de-identified historical alerts and analyst studies. Until that work is completed, claims about time savings, quality improvement, and risk reduction should be treated as hypotheses.

A second limitation is the limited availability of peer-reviewed Actimize-specific research. Recent NICE materials provide valuable product-context evidence, but they are vendor-authored and should not be treated as independent validation. This limitation is also part of the paper's contribution because it highlights the need for independent academic or practitioner research on Actimize-style agentic workflows.

A third limitation is data access. Real fraud investigations involve sensitive customer data, case notes, and operational procedures. Public datasets do not fully represent these workflows. Future research must address de-identification, privacy, data quality, and representativeness. Synthetic data may help in early testing, but production-like validation is necessary before operational conclusions can be drawn.

A fourth limitation is the rapid evolution of MCP and LLM tooling. MCP specifications, security practices, vendor implementations, and regulatory expectations may change. Any

architecture based on MCP must therefore be revisited periodically. Governance should include version control, change management, and reassessment after material technology changes.

A fifth limitation is automation bias. Human-in-the-loop design does not automatically eliminate overreliance. Analysts may defer to recommendations if they appear authoritative. Future evaluation should measure trust calibration and not merely ask whether analysts liked the tool.

Future Research

Future research should first validate the architecture using historical Actimize-style alert data. The most important empirical question is whether evidence packages improve specialist performance compared with baseline workflows. Researchers should measure handling time, decision quality, narrative completeness, escalation accuracy, and analyst trust. They should also evaluate whether benefits differ across fraud typologies, analyst experience levels, and alert complexity.

Second, future research should compare multiple architectural patterns. A monolithic LLM assistant should be compared with segmented MCP servers and specialized analytical tools. A recommendation-only design should be compared with an evidence-rich design. An always-on LLM design should be compared with a cascade design. These comparisons would help determine which design choices matter most for performance, cost, latency, and risk.

Third, researchers should develop MCP-specific security benchmarks for financial use cases. Current AI security work often focuses on prompt injection generally, but MCP-connected fraud systems require tests for tool scoping, retrieval poisoning, audit completeness, write-back controls, and role-based access enforcement. A standardized evaluation suite would be valuable for banks, vendors, and regulators.

Fourth, future research should examine feedback loops. Analyst corrections and final dispositions can improve future recommendations, but uncontrolled feedback can amplify bias or degrade quality. Research should test whether human feedback improves retrieval, prompts, or secondary models and determine which governance controls are needed before feedback affects production behavior.

Finally, future work should study organizational adoption. Fraud automation is not only a technical problem. Specialists may resist AI tools that appear to threaten their role, increase surveillance, or produce unhelpful narratives. Successful adoption will require training, transparency, clear accountability, and a design that supports human expertise rather than replacing it.

Conclusion

MCP-enabled fraud investigation support is a promising direction for modernizing financial crime operations, but it should be framed as governed decision support rather than autonomous fraud adjudication. The architecture proposed in this paper uses MCP servers to provide controlled access to fraud alerts, rules, historical cases, and analytics tools; uses Python services for deterministic computation; uses an LLM for evidence synthesis and recommendation drafting; and preserves human specialist authority over final disposition.

The recent literature supports this bounded-automation approach. Fraud-detection research demonstrates the maturity of machine learning methods, LLM-agent research shows the potential of tool-augmented financial agents, sanctions-screening evidence supports cascaded LLM deployment, MCP documentation and research define integration and security concerns, financial XAI research supports evidence-rich explanations, and governance sources emphasize

trustworthiness, compliance, and monitoring. Actimize vendor materials show that agentic investigation is timely, but independent validation remains limited.

The manuscript's central contribution is the integration of these streams into a practical, auditable, human-reviewed architecture for Actimize-style fraud investigations. The strongest future contribution will come from empirical testing with production-like alert data, analyst studies, MCP security red teaming, and longitudinal monitoring. A well-designed system can reduce manual burden, improve consistency, and strengthen documentation, but only if it preserves human accountability, evidence provenance, least-privilege access, and rigorous governance.

References

Allen, J., & Hatfield, J. W. (2025). Can LLMs improve sanctions screening in the financial system? Evidence from a fuzzy matching assessment (Finance and Economics Discussion Series 2025-092). Board of Governors of the Federal Reserve System. https://doi.org/10.17016/FEDS.2025.092

Anthropic. (2024, November 25). Introducing the Model Context Protocol. https://www.anthropic.com/news/model-context-protocol

Autio, C., Schwartz, R., Dunietz, J., Jain, S., Stanley, M., Tabassi, E., Hall, P., & Roberts, K. (2024). Artificial Intelligence Risk Management Framework: Generative Artificial Intelligence Profile (NIST AI 600-1). National Institute of Standards and Technology. https://doi.org/10.6028/NIST.AI.600-1

Dong, Y., Wu, F., Zhang, K., Dai, Y., Zhang, S., Ye, W., Chen, S., & Cheng, Z.-Q. (2025). Large language model agents in finance: A survey bridging research, practice, and real-world deployment. Findings of the Association for Computational Linguistics: EMNLP 2025, 17889-17907. https://doi.org/10.18653/v1/2025.findings-emnlp.972

Hacini, A. D., Benabdelouahad, M., Abassi, I., Houhou, S., Boulmerka, A., & Farhi, N. (2025). LLM-assisted financial fraud detection with reinforcement learning. Algorithms, 18(12), 792. https://doi.org/10.3390/a18120792

Hernandez Aros, L., Bustamante Molano, L. X., Gutierrez-Portela, F., Moreno Hernandez, J. J., & Rodríguez Barrero, M. S. (2024). Financial fraud detection through the application of machine learning techniques: A literature review. Humanities and Social Sciences Communications, 11, 1130. https://doi.org/10.1057/s41599-024-03606-0

Hou, X., Zhao, Y., Wang, S., & Wang, H. (2026). Model Context Protocol: Landscape, security threats, and future research directions. ACM Transactions on Software Engineering and Methodology. https://doi.org/10.1145/3796519

Model Context Protocol. (2025a). Model Context Protocol specification (Version 2025-11-25). https://modelcontextprotocol.io/specification/2025-11-25

Model Context Protocol. (2025b). Security best practices. https://modelcontextprotocol.io/docs/tutorials/security/security_best_practices

NICE. (2025a). NICE Actimize introduces Xceed AI Agents for faster, smarter fraud and fincrime prevention. https://www.nice.com/press-releases/nice-actimize-introduces-xceed-ai-agents-for-faster-smarter-fraud-and-fincrime-prevention

NICE. (2025b). NICE Actimize X-Sight ActOne platform redefines financial crime investigations with agentic AI. https://www.nice.com/press-releases/nice-actimize-x-sight-actone-platform-redefines-financial-crime-investigations-with-agentic-ai

NIST. (2023). Artificial Intelligence Risk Management Framework (AI RMF 1.0) (NIST AI 100-1). National Institute of Standards and Technology. https://doi.org/10.6028/NIST.AI.100-1

Office of the Comptroller of the Currency. (2026). Model risk management: Revised guidance (OCC Bulletin 2026-13). https://www.occ.treas.gov/news-issuances/bulletins/2026/bulletin-2026-13.html

Shevchenko, E. (2025, December 17). Graph analytics ATO fraud: From ATLAS research to systems. EdEconomy Publishing. https://edeconomy.com/graph-analytics-account-takeover-fraud/

Shevchenko, E. (2026, March 9). How generative AI is transforming enterprise analytics using MCP servers. EdEconomy Publishing. https://edeconomy.com/how-generative-ai-is-transforming-enterprise-analytics-using-mcp-servers/

U.S. Department of the Treasury. (2024). Uses, opportunities, and risks of artificial intelligence in financial services. https://home.treasury.gov/system/files/136/Artificial-Intelligence-in-Financial-Services.pdf

Vuković, D. B., Dekpo-Adza, S., & Matović, S. (2025). AI integration in financial services: A systematic review of trends and regulatory challenges. Humanities and Social Sciences Communications, 12, 562. https://doi.org/10.1057/s41599-025-04850-8

Xing, F. (2025). Designing heterogeneous LLM agents for financial sentiment analysis. ACM Transactions on Management Information Systems, 16(1), Article 5. https://doi.org/10.1145/3688399

Yeo, W. J., Van Der Heever, W., Mao, R., Cambria, E., Satapathy, R., & Mengaldo, G. (2025). A comprehensive review on financial explainable AI. Artificial Intelligence Review, 58, 189. https://doi.org/10.1007/s10462-024-11077-7

Appendix A

Recommended Fraud Recommendation Package Template

Alert summary: plain-language description of the alert and why it was generated.

Rules triggered: rule identifiers, rule descriptions, triggering conditions, and relevant thresholds.

Customer context: baseline behavior, account history, channel activity, and relevant prior cases.

Historical comparison: similar cases, typology match, final dispositions, and differences from the current alert.

Evidence supporting fraud: facts that support escalation or adverse action, with source references.

Evidence supporting false positive: facts that support legitimate activity or lower-risk disposition.

Uncertainty and missing information: unresolved issues that require specialist review.

Recommended action: approve, decline, monitor, escalate, request more information, or refer to another queue.

Human review checklist: required confirmations before final disposition or case-system update.