

UDC 004.8:336.7

DOI [https://doi.org/10.15589/znп2026.1\(503\).2.13](https://doi.org/10.15589/znп2026.1(503).2.13)**GOVERNED LLM-ASSISTED AML CASE TRIAGE: A NON-AUTHORITATIVE, AUDITABLE DECISION SUPPORT ARCHITECTURE****КЕРОВАНИЙ ТРІАЖ AML-КЕЙСІВ ЗА ПІДТРИМКИ LLM:  
НЕАВТОРИТАТИВНА, ПРИДАТНА ДО АУДИТУ АРХІТЕКТУРА ПІДТРИМКИ  
ПРИЙНЯТТЯ РІШЕНЬ****Maksym V. Lapin<sup>1</sup>**

maksym.lapin@outlook.com

ORCID: 0009-0009-6404-7450

**Oleksandra O. Kolomiets<sup>2</sup>**

kolomietsniss@gmail.com

ORCID: 0000-0002-6876-0597

**М. В. Лапін<sup>1</sup>,**

здобувач вищої освіти

**О. О. Коломієць<sup>2</sup>,**

канд. екон. наук

<sup>1</sup> *Vilnius Gediminas Technical University, Vilnius*<sup>2</sup> *Razumkov Centre, Kyiv*<sup>1</sup> *Вільнюський технічний університет імені Гедимінаса, м. Вільнюс*<sup>2</sup> *Український центр економічних і політичних досліджень імені Олександра Разумкова, м. Київ*

**Abstract. Purpose.** This study aims to identify an optimal decision-support architecture for regulated AML case triage that enables non-authoritative LLM assistance while maintaining auditability, regulatory compliance, and analyst accountability. The focus is on how evidence volume and type, bundle construction and validation, and the cost of reproducible audit artifacts affect workflow effectiveness.

**Method.** The research specifies and analyzes a governed gateway architecture with a sequential workflow comprising evidence selection, canonicalization and hashing, minimal evidence bundle construction, structured draft generation, output validation, quarantine handling, and human acceptance with diff binding. The approach enforces policy and context constraints and binds all outputs to tamper-evident audit events. Qualitative evaluation examines narrative consistency, provenance coverage, and review readiness, alongside common failure modes such as prompt injection, policy leakage, and unintended actionable language.

**Results.** The findings show that a governed gateway using minimal evidence bundles and explicit human acceptance gates improves audit readiness, consistency of outputs, and control over context leakage. While the architecture mitigates several governance risks, residual issues persist, including automation bias and subtle prompt injection, indicating the need for stronger human-in-the-loop safeguards and ongoing validator monitoring.

**Scientific novelty.** The work operationalizes AI governance as a set of enforceable, checkable invariants—such as template pinning, bundle-only context, claim-level provenance, structured non-delegation, and tamper-evident logging—and integrates them into a full lifecycle for LLM-assisted AML artifacts.

**Practical importance.** The proposed architecture provides compliance and engineering teams with a concrete, auditable pattern for deploying LLM drafting assistance in AML triage, enabling productivity gains while preserving analyst responsibility and minimizing uncontrolled disclosure under regulatory scrutiny.

**Key words:** anti-money laundering; LLM governance; auditability; evidence bundles; decision support; regulated AI; human-in-the-loop.

**Анотація. Мета.** Дослідження спрямоване на визначення оптимальної архітектури підтримки ухвалення рішень для регульованого AML-тріажу, яка забезпечує неавторитетну допомогу з боку LLM, зберігаючи аудиторську придатність, відповідність регуляторним вимогам та відповідальність аналітика. Основну увагу приділено впливу обсягу й типів доказів, побудови та валідації пакетів доказів, а також витрат на підтримку відтворюваних аудиторських артефактів на ефективність робочого процесу.

**Метод.** У роботі специфіковано та проаналізовано архітектуру керованого шлюзу з послідовним робочим процесом, що включає відбір доказів, канонізацію та хешування, формування мінімальних пакетів доказів, структуровану генерацію чернеток, валідацію вихідних результатів, обробку карантинних станів і людське прийняття з прив'язкою змін (diff binding). Підхід забезпечує дотримання політик і контекстних обмежень та зв'язує

всі результати з подіями аудиту, захищеними від підробки. Якісна оцінка охоплює наративну узгодженість, повноту провенансу та готовність до перевірки, а також типові режими відмов, зокрема ін'єкції підказок, витоки політик і ненавмисні формулювання з ознаками дії.

*Результати.* Результати показують, що керований шлюз із мінімальними пакетами доказів і явними етапами людського прийняття підвищує аудиторську готовність і узгодженість вихідних матеріалів, одночасно зменшуючи ризик витоку контексту. Водночас залишаються резидуальні ризики, зокрема упередженість автоматизації та тонкі форми ін'єкції підказок, що вказує на потребу в посиленні механізмів human-in-the-loop та постійному моніторингу роботи валідаторів.

*Наукова новизна.* Уперше управління (governance) операціоналізовано у вигляді перевірюваних інваріантів – таких як фіксація шаблонів, контекст лише з пакета доказів, провенанс на рівні тверджень, структурована делегація та захищене від підробки журналювання – і інтегровано в повний життєвий цикл артефактів, створюваних за допомогою LLM для AML-кейсів.

*Практичне значення.* Запропонована архітектура надає командам із комплаєнсу та інженерії конкретний, придатний до аудиту шаблон упровадження LLM-допомоги для підготовки матеріалів у AML-тріажі, забезпечуючи підвищення продуктивності за збереження відповідальності аналітиків і мінімізації неконтрольованого розкриття інформації в умовах регуляторного нагляду.

**Ключові слова:** протидія відмиванню коштів; управління LLM; аудиторська придатність; пакети доказів; підтримка ухвалення рішень; регульований ШІ; human-in-the-loop.

## RELEVANCE OF THE PROBLEM

AML case triage is a time-constrained decision process executed under strict accountability. When an alert is raised, an analyst must rapidly determine whether the observed activity is plausibly explainable, what additional information is required, and whether escalation is warranted. The key deliverable is a review-ready narrative that records what was known at the time, which sources were consulted, and how the outcome was justified for quality assurance and regulatory review.

The pressure on triage teams is increasing due to higher alert volumes, more heterogeneous evidence sources (payments, KYC/KYB, screening, prior investigations, channel telemetry), and higher expectations for documentation quality. Large language models (LLMs) can reduce drafting effort and improve consistency, but their use in regulated workflows is constrained by non-delegation, data minimization, controlled disclosure, and auditability. Therefore, the relevant problem is to integrate LLM assistance in a way that remains defensible even under adverse conditions: model errors, prompt injection attempts, and retrospective scrutiny.

## ANALYSIS OF RECENT STUDIES AND PUBLICATIONS

AML/CFT guidance emphasizes risk-based controls, evidence-based reasoning, and supervisory review, which implies that case narratives must be explainable and supported by retrievable records [1, 2, 18, 3, 4]. In parallel, security and assurance standards formalize expectations for access control, change management, and audit logging that become critical once LLMs are introduced into sensitive workflows [12, 16, 14].

AI governance frameworks highlight lifecycle risk management, transparency, and human oversight, while GenAI-focused materials call out prompt injection, data leakage, and insecure output handling as recurring risks [6, 7, 8, 17]. Retrieval-augmented generation (RAG)

provides a general grounding mechanism by conditioning outputs on retrieved documents [9]; however, in regulated triage, the dominant questions are operational: which evidence was selected, what exact context was sent, and which version of the draft was accepted.

Overall, recent studies provide principles and threat models but often do not specify enforceable, checkable system invariants that bind drafts to minimal evidence bundles and to tamper-evident audit events. This work addresses that operational gap by proposing an implementation-oriented governed gateway pattern.

## PURPOSE OF THE ARTICLE

The purpose of the article is to present a governed decision-support architecture for AML case triage in which an LLM is used only for non-authoritative drafting under enforceable constraints. The objectives are to define an evidence-bundle boundary with canonicalization and hashing, specify validation and quarantine rules, and describe a human-only acceptance workflow that binds the final narrative to auditable artifacts.

## PRESENTATION OF THE MAIN RESEARCH MATERIAL

The research material is presented as an implementation-oriented system design that makes governance enforceable.

**System boundary and roles.** The LLM is placed behind a governed gateway that mediates all prompts and responses.

The Analyst is the only decision-maker; the model cannot submit outcomes, close cases, or trigger actions. The gateway enforces policy, produces structured outputs, and records audit events.

Evidence bundles construction and hashing. For each case, a minimal evidence bundle is created from approved internal sources (alert metadata, transaction summaries, customer attributes, screening results, and relevant historical notes). Each evidence item is canonicalized

**Table 1.** System requirements for governed LLM assistance in AML triage

| Type | Requirement                                                           |
|------|-----------------------------------------------------------------------|
| F1   | Unified case workspace (alert, customer, transactions, prior history) |
| F2   | Evidence linking and provenance display for narratives                |
| F3   | AI assistance for drafting, summarization, and checklist completion   |
| F4   | Human review/edit/accept gates; AI cannot write record directly       |
| F5   | Supervisory/QA review views with traceable artifacts                  |
| N1   | Context minimization and field-level data governance                  |
| N2   | Reproducible inputs (template version + bundle snapshot)              |
| N3   | Tamper-evident audit logging for AI interactions and edits            |
| N4   | Safe degradation when gateway/model unavailable                       |
| N5   | Role-based access and policy-default deny behavior                    |

into a stable representation, assigned an identifier (E1, E2, dots), and hashed; the full bundle is also hashed to provide snapshot semantics. Template pinning and bundle-only context. Generation is executed using pinned, versioned prompt templates approved by governance stakeholders. The LLM receives only the template plus the bundle; unbounded retrieval and free-form chat history are excluded to reduce leakage and prompt drift. The template forbids determinations and requires claim-level citations to bundle identifiers.

Structured drafting and provenance. The model produces a structured draft (e.g., case summary, observed activity, customer context, risk indicators, benign explanations, open questions, and disposition options). Every material statement must reference at least one evidence item identifier; unsupported text is treated as invalid. Deterministic validation and quarantine. Before a draft is shown for acceptance, the gateway applies validators: schema checks, provenance completeness, policy rules (no actionable instructions; no prohibited data), and consistency checks against the bundle. Drafts that fail are quarantined and returned with machine-readable failure reasons. Human acceptance with diff binding and tamper-evident logging. If a draft passes validation, the analyst may edit it and explicitly accept the final version. Acceptance creates an audit event binding: bundle hash, template version, model identifier/configuration, the original draft, the accepted text, and a diff between them. Events are stored in append-only (e.g., hash-chained records) to make later modifications detectable.

These mechanisms are designed to preserve controlled disclosure, prevent silent delegation, and make later review practical by tying each claim to a bounded evidentiary snapshot.

The article proposes a governed architecture for integrating LLMs into AML case triage as drafting assistants rather than decision-makers. Governance is implemented as checkable invariants: bundle-only context, template pinning, claim-level provenance, deterministic validation with quarantine, and human-only acceptance with diff binding. This approach can improve drafting efficiency

while preserving controlled disclosure and strengthening evidentiary traceability for QA and regulatory scrutiny. Residual risks remain, including automation bias, evidence selection bias, and sophisticated prompt injection attempts within upstream data. To address these risks, governance mechanisms must be treated as first-class system features with continuous testing and monitoring.

Prospects for further research should evaluate the approach empirically using realistic alert scenarios and operational constraints. Key directions include metrics for provenance completeness and validator accuracy, controlled studies of automation bias, adversarial testing for prompt injection and policy evasion, and monitoring for drift in evidence patterns and validation outcomes.

**Table 2.** Comparative analysis of LLM integration patterns for regulated AML triage

| Pattern Min.                    | Prov. | Non-del. | Audit |
|---------------------------------|-------|----------|-------|
| Client-side API Low             | Low   | Low      | Low   |
| Free-form RAG chat Med          | Med   | Low      | Med   |
| Embedded copilot Med            | Med   | Med      | Med   |
| Governed gateway + High bundles | High  | High     | High  |

We model the system as operating over the following artifacts to make governance concrete: instead of describing controls as vague intentions, we treat the contract as something that can be checked by software (every draft is tied to a template version and a bundle hash, every accepted edit is tied to a human action, and every event is append-only).

The artifacts are: Template  $T = \langle id, ver, schema, policy \rangle$  (approved, immutable per version); Evidence item  $e = \langle eid, payload, src, t_{retr} \rangle$  where  $src$  denotes system + version and  $t_{retr}$  is retrieval time; Bundle  $B = \langle bid, \{e_i\}, policy, meta \rangle$  (analyst-selected evidence set); Draft  $D = \langle did, tid, tver, bhash, out, f_lags \rangle$  where  $f_lags$  capture validator outcomes; Record  $R$  (official case narrative and decisions), modified only by human actions; and Audit event  $A_k = \langle type, ts, actor, payload, prevhash \rangle$  (append-only).

We define a case state as  $S = \langle B, D, R, \{A_k\} \rangle$ . LLM output does not update  $R$  directly. Instead, it creates a draft  $D$ , and only explicit analyst actions can mutate  $R$ . We represent record updates as diffs  $\Delta R$  emitted by human acceptance/edit actions.

This separation is essential for accountability: it ensures that the analyst remains the decision-maker and that the system can always show which parts of the final narrative were drafted by the model, which parts were edited, and which actions constituted acceptance.

The following invariants must hold in deployment; each can be unit-tested and monitored: I1 (Template pinning): Each AI request references  $(tid, tver)$ ; free-form prompts from the browser are rejected. I2 (Bundle-only context): Each AI request must provide a schema-valid bundle with a logged  $bhash$ . I3 (Claim-level provenance): Each accepted factual claim references one or more evidence IDs (not merely string markers). I4 (Structured

non-delegation): Output schema contains no decision/action fields; actionability signals trigger quarantine. I5 (Human-only record writes): Any change to  $R$  requires an authenticated human action producing a diff audit event. I6 (Tamper-evident logging): Per-case audit events are hash-chained and stored in append-only/WORM-backed storage. I7 (Policy-default deny): Prohibited fields are never forwarded; deny/redact/require-approval actions are logged with reason codes. The UI never calls the LLM directly. All model access is mediated by a governed gateway that enforces policy and logs artifacts.

Operationally, every request is normalized through the same enforcement point, and templates, policies, redaction rules, and validators can be changed centrally under governance review while client applications remain thin and cannot bypass controls.

Validators should be treated as product features rather than as afterthoughts. They are the practical mechanism that converts the governance contract into enforceable behavior and provides actionable feedback to analysts when drafts are incomplete, uncited, or policy-violating. The threat model is not meant to be exhaustive. Its goal is to highlight the most relevant risks for AML triage:

manipulation of prompts through untrusted text, leakage of sensitive identifiers, and the social risk that users over-trust an apparently confident draft.

Core backend services align with enterprise boundaries: Case service (state, assignments, workflow); customer/transaction services (curated access, retrieval metadata); screening connectors (sanctions/PEP/adverse media signals); audit service (append-only events); and the LLM gateway (template resolution, policy enforcement, validation).

Evidence bundles: minimization and snapshot semantics. An exposition of the main research material. Bundle schema and policy tags. Evidence bundles are the *only* context unit permitted into the gateway. They are minimal and analyst-selected. Listing 1 shows an illustrative schema.

In day-to-day triage, analysts often need only a subset of available fields to write a defensible narrative. By requiring explicit selection, bundles discourage “send everything” behavior and create a natural review point for sensitive attributes. Policy tags support automated checks and consistent handling of restricted fields across different source systems.

**Listing 1.** Illustrative evidence bundle with policy tags (anonymized).

Canonicalization and hashing. To obtain a stable bundle hash, we define canonicalization  $C(\cdot)$  (UTF-8, stable key ordering, normalized whitespace, and removal of non-semantic UI fields). We compute:

$$bhash = \text{SHA256}(C(B)). \quad (1)$$

Snapshot semantics store retrieved values *and* retrieval metadata to mitigate “same ID, different underlying record”. This increases storage footprint, so retention

```
{
  "bundle_id": "bnd_2026-02-13T10:15:23Z_8f3a",
  "case_id": "case_12345",
  "created_at": "2026-02-13T10:15:23Z",
  "sources": [
    { "system": "TransactionsSvc", "version": "v7", "retrieved_at": "2026-02-13T10:14:58Z" },
    { "system": "KYC", "version": "v3", "retrieved_at": "2026-02-13T10:12:11Z" }
  ],
  "policy": {
    "pii_mode": "redact",
    "max_chars": 12000,
    "deny_patterns": ["IBAN", "ACCOUNT_NUMBER"]
  },
  "alert": { "type": "Unusual International Transfers", "rule_id": "RULE_X12" },
  "customer": {
    "customer_ref": "cust:hash:9a3c...",
    "risk_rating": "Medium",
    "geography": ["DE", "NL"]
  },
  "transactions": [
    {
      "evidence_id": "txn:5481", "date": "2026-02-03",
      "amount_bucket": "1k-5k", "currency": "EUR",
      "counterparty_ref": "cp:hash:11bf...",
      "policy_tags": ["NO_ACCOUNT_NUMBER", "NO_ADDRESS"]
    },
    {
      "evidence_id": "note:77",
      "snippet": "Customer claims funds are family support.", "policy_tags":
        ["PII_REDACTED"]
    }
  ]
}
```

policies must be aligned with privacy and record-keeping obligations [11, 12].

In practice, teams should treat bundle retention as a policy decision. Some deployments may store only hashes and minimal metadata, while others may store redacted payload snapshots to support stronger audit reconstruction. The key requirement is that whatever is stored must be consistent and must allow reviewers to understand why a specific claim was supported at the time of drafting.

In practice, teams should treat bundle retention as a policy decision. Some deployments may store only hashes and minimal metadata, while others may store redacted payload snapshots to support stronger audit reconstruction. The key requirement is that whatever is stored must be consistent and must allow reviewers to understand why a specific claim was supported at the time of drafting.

LLM gateway: prompt governance, validation, and logging. Gateway pattern. The scientific contribution is the gateway *pattern*: a policy-enforcing boundary that pins templates/configurations, validates context, and produces audit artifacts. The pattern can be implemented with different vendors or a self-hosted model stack without changing the governance contract.

This pattern is intentionally vendor-agnostic. It focuses on what must be enforced (template pinning, bundle validation, policy filtering, output validation, and audit events), not on a specific model family. As a result, a model can be upgraded or swapped while preserving the governance contract and its test suite.

Prompt governance and template lifecycle. Templates are treated as approved policy assets with versioned storage and immutable references (*tid*, *tver*); a promotion workflow requiring compliance/model-risk review [6]; *automated* tests (schema conformance, forbidden field checks, and output contract checks); and rollback by repinning production to a prior approved version.

Non-delegation enforcement beyond keyword blocking. Keyword blocking alone is bypassable (paraphrases and hedged recommendations). We implement nondelégation as schema constraints plus quarantine policy. The output schema provides *no* fields for “recommendation” or “decision”; output is decomposed into claim units; a lightweight actionability detector computes a signal  $s \in [0, 1]$  from patterns such as imperative language, explicit directives, and “should/must” statements; and if  $s > \tau$ , the draft is returned with QUARANTINED status and cannot be accepted without supervisory override.

Claim-level provenance and coverage metric. Per-sentence citation is brittle; instead, we require *claim-level*

**Table 3.** Validator taxonomy and handling policy

| Validator     | Example checks                             | Handling          |
|---------------|--------------------------------------------|-------------------|
| Schema        | required sections / JSON schema            | reject            |
| Policy        | deny/redact fields (IBAN-like, IDs)        | reject/quarantine |
| Provenance    | coverage $\geq \gamma$ ; claim links valid | block acceptance  |
| Actionability | imperative/recommendation signals          | quarantine        |
| PII patterns  | account/identifier formats [8]             | reject            |
| Length        | max tokens/characters                      | truncate/reject   |

evidence linkage. Each claim  $c_j$  has an evidence set  $E(c_j)$  (IDs referring to bundle evidence objects, not raw strings).

We define provenance coverage:

$$\text{Cov}(D) = |\{c_j \in D : |E(c_j)| \geq 1\}| / |\{c_j \in D\}| \quad (2)$$

Acceptance is enabled only if  $\text{Cov}(D) \geq \gamma$  and no prohibited fields are detected. Claims without evidence are labeled as uncited inference and defaulted to non-acceptable.

Output validation taxonomy Table 3 specifies validator categories and failure handling. Validators should be treated as product features rather than as afterthoughts. They are the practical mechanism that converts the governance contract into enforceable behavior and provides actionable feedback to analysts when drafts are incomplete, uncited, or policy-violating.

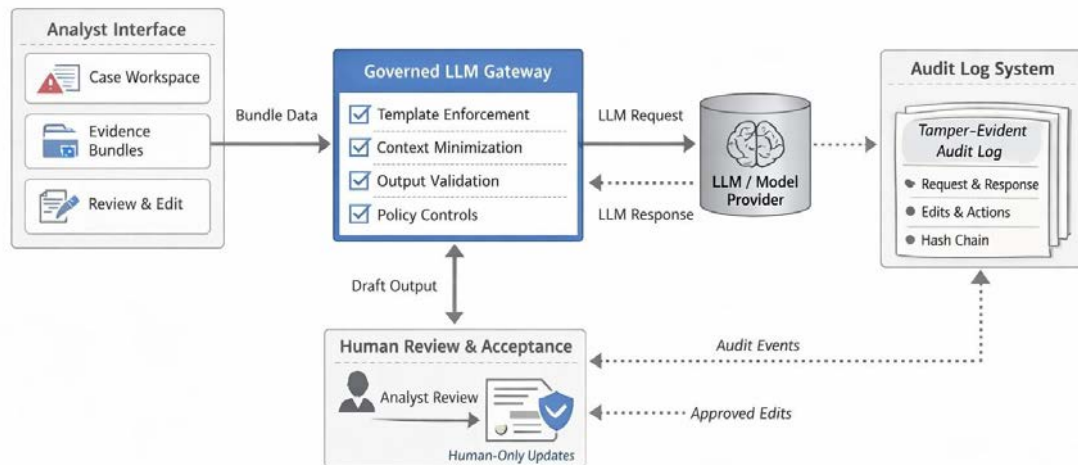


**Fig. 1.** High-level architecture for AI-assisted AML case triage with governed LLM gateway and end-to-end audit logging

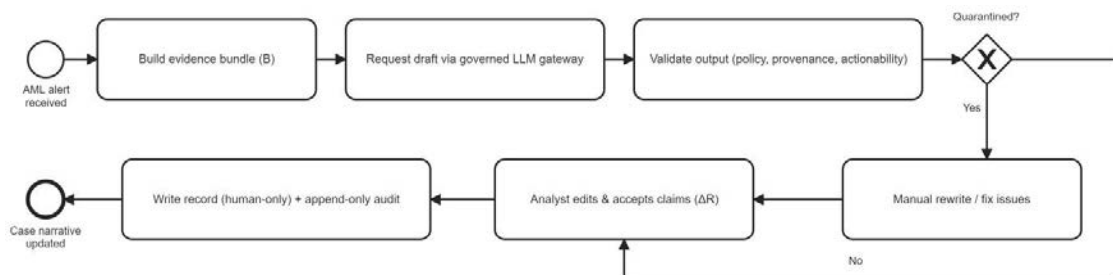


**Fig. 2.** Human-in-the-loop workflow with generation and acceptance gates

Audit event model and tamper evidence. Each event links to the previous event hash for the case:

$$h_k = \text{SHA256}(A_k \parallel h_{k-1}) \quad (3)$$

Hash-chaining alone does not prevent an operator from rewriting the entire chain, so the design assumes append-only storage semantics (WORM) and/or periodic anchoring/signing via KMS keys [12, 14, 15]. This paper treats anchoring as a deployment requirement rather than a prototype guarantee.

From an operational standpoint, audit logs are only useful if they remain trustworthy under scrutiny.

Implementations should therefore combine technical measures (hash chains, signing, write-once storage) with access controls, separation of duties, and monitoring for anomalous administrative actions.

Human-in-the-loop workflow and UI controls. Acceptance gates and diff binding AI drafts are visually separated from the case record. No “accept all” exists for full notes. The analyst accepts claim units or sections, optionally editing them first. Each acceptance emits a diff event binding:

*{did, accepted\_claims, edits, ts, actor}*.

Fallback when AI is unavailable. When the gateway/model is unavailable, the UI offers bundle export for manual narrative drafting, pre-filled note skeletons (non-AI), and checklist validation only. This avoids ad-hoc prompts and preserves workflow continuity.

This fallback mode is crucial for governance because it prevents the use of “shadow AI.” If analysts cannot complete their work without the model, they are more likely to copy data into uncontrolled tools. Providing non-AI scaffolding and validation helps keep work within approved boundaries even during outages.

Threat model and residual risk. We model threats using a pragmatic STRIDELike categorization. Table 4 maps threats to controls and residual risks. The threat model is not meant to be exhaustive. Its goal is to highlight the most relevant risks for AML triage: manipulation of prompts through untrusted text, leakage of sensitive identifiers, and the social risk that users over-trust an apparently confident draft.

**Evaluation (structured qualitative).** Participants. We conducted structured walkthroughs with

6 participants: 4 AML analysts (2 senior, 2 mid), 1 QA reviewer, 1 MLRO/compliance stakeholder. All participants had prior experience with case management systems; none were authors of the system.

The evaluation focused on whether participants could produce auditable narratives under the proposed constraints and whether governance mechanisms were understandable and usable. Because the study targets early-stage systems design, the goal was to surface systematic failure modes and workflow friction points, not to estimate statistical effect sizes.

**Scenarios.** We used five anonymized scenario archetypes typical for retail/commercial triage (international transfer anomaly, rapid beneficiary creation, cash-like deposits, nested account transfers, and adverse-media weak match). Scenarios used synthetic identifiers and masked amounts aligned with the minimization policy. Table 5 lists scenario properties and tasks used in the walkthroughs.

**Protocol.** Each session followed a fixed protocol: brief training (10 minutes) on drafts vs record and acceptance gates; running two scenarios (counterbalanced order); and a post-task interview using a structured guide (auditability, trust). Per-case audit events are hash-chained and stored in append-only/WORM-backed storage. 17 (Policy-default deny): Prohibited fields are never forwarded; deny/redact/requireapproval actions are logged with reason codes.

The UI never calls the LLM directly. All model access is mediated by a governed gateway that enforces policy and logs artifacts. We captured: screen recordings, event logs, drafts, acceptance diffs, and observer notes. Qualitative coding used open coding followed by consolidation into categories. In addition to interviews, we inspected event logs to verify that the artifact bindings were interpretable and that quarantines and acceptance blocks behaved as intended. This provided a basic sanity check that the governance contract was not only described on paper but also reflected in observable system behavior. Table 6 summarizes recurring failure categories and how validators/UX handled them.

**Findings (qualitative).** Participants reported improved narrative consistency for early-stage triage, easier QA review due to explicit claim/evidence links, and a strong preference for template-driven operations over chat in regulated use. Concerns remained about automation bias during high workload periods, false negatives in PII detection, and the need for explicit supervisory workflows for quarantined drafts.

Overall, the walkthroughs suggested that governance features are acceptable when they are presented as part of normal case work rather than as separate compliance tasks, and participants especially valued seeing which evidence supported a claim and understanding why a draft was blocked or quarantined because it reduced uncertainty during review.

**Table 4.** Threats, controls, and residual risks

| Threat          | Controls                                                                                | Residual risk            |
|-----------------|-----------------------------------------------------------------------------------------|--------------------------|
| Injection       | <i>bundle truncation, instruction prefix pinning, and untrusted text isolation</i> [8]. | subtle semantic override |
| Exfiltration    | deny/redact policy, output PII scan, policy-default deny                                | false negatives          |
| Automation bias | Acceptance friction, provenance coverage, draft labeling                                | over-trust under load    |
| Log tamper      | <i>append-only/WORM + chaining + signing</i> [12, 14].                                  | insider/admin attacks    |
| Model drift     | template pinning + config ID logging                                                    | provider weight updates  |

**Table 5.** Scenario set used in evaluation (illustrative)

| ID | Trigger archetype        | Required tasks                              |
|----|--------------------------|---------------------------------------------|
| S1 | Intl transfers pattern   | bundle, draft, validate, accept<br>2 claims |
| S2 | New beneficiary burst    | bundle + open<br>questions assistant        |
| S3 | Cash-like deposits       | draft + provenance fixes                    |
| S4 | Internal transfers       | quarantine handling +<br>manual rewrite     |
| S5 | Adverse media weak match | redact policy + checklist<br>completion     |

**Table 6.** Observed failure categories and mitigation

| Category            | Example                         | Mitigation                              |
|---------------------|---------------------------------|-----------------------------------------|
| Hallucinated detail | added unstated<br>“purpose.”    | Provenance blocks<br>acceptance         |
| Weak grounding      | vague summary<br>without IDs    | coverage ↓, prompts to<br>link evidence |
| Actionable phrasing | “should escalate”<br>paraphrase | quarantine, forced<br>rewrite           |
| Policy leakage      | IBAN-like string<br>surfaced    | output reject + redact<br>hints         |
| Prompt drift desire | analyst attempted<br>free-form  | UI disallows, template-<br>only         |

## CONCLUSIONS AND PROSPECTS FOR FURTHER RESEARCH

We presented a governed architecture for LLM-assisted AML triage that operationalizes non-authoritative drafting through enforceable invariants, bounded evidence bundles, a policy-enforcing gateway, and audit events binding drafts to human edits. Future work includes richer evidence-link UX, adversarial testing of validators, multilingual workflows, and integration with formal QA sampling and model-risk monitoring.

An immediate next step for practitioners is to prototype the bundle schema and validator set within their own data governance constraints and to run similar walk-throughs with internal reviewers. This can reveal which fields are most sensitive, where minimization causes usability issues, and which quarantine thresholds best balance safety with workflow efficiency.

## REFERENCES

- [1] Financial Action Task Force (FATF). (2012). International standards on combating money laundering and the financing of terrorism and proliferation: The FATF recommendations. FATF. Retrieved from <https://www.fatf-gafi.org>
- [2] Financial Action Task Force (FATF). (2014). Guidance for a risk-based approach: The banking sector. FATF. Retrieved from <https://www.fatf-gafi.org>
- [3] Basel Committee on Banking Supervision. (2020). Sound management of risks related to money laundering and financing of terrorism. Bank for International Settlements. Retrieved from <https://www.bis.org>
- [4] European Banking Authority. (2021). Guidelines on ML/TF risk factors. EBA. Retrieved from <https://www.eba.europa.eu>
- [5] Federal Financial Institutions Examination Council. (2026). Bank Secrecy Act / Anti-Money Laundering examination manual. FFIEC. Retrieved from <https://www.ffiec.gov>
- [6] National Institute of Standards and Technology. (2023). Artificial intelligence risk management framework (AI RMF 1.0) (NIST AI 100-1). NIST. Retrieved from <https://www.nist.gov>
- [7] National Institute of Standards and Technology. (2024). AI RMF generative AI profile (NIST AI 600-1). NIST. Retrieved from <https://www.nist.gov>
- [8] OWASP Foundation. (2026). Top 10 for large language model applications. OWASP. Retrieved from <https://owasp.org>
- [9] Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., ... Riedel, S. (2020). Retrieval-augmented generation for knowledge-intensive NLP tasks. In Advances in Neural Information Processing Systems (NeurIPS). Retrieved from <https://papers.nips.cc>
- [10] European Union. (2024). Regulation (EU) 2024/1689 of the European Parliament and of the Council [Artificial Intelligence Act]. Retrieved from <https://eur-lex.europa.eu>
- [11] European Union. (2016). Regulation (EU) 2016/679 of the European Parliament and of the Council [General Data Protection Regulation]. Retrieved from <https://eur-lex.europa.eu>
- [12] ISO/IEC. (2022). ISO/IEC 27001:2022 information security management systems – Requirements. International Organization for Standardization. Retrieved from <https://www.iso.org>
- [13] American Institute of Certified Public Accountants. (2026). SOC 2: Trust services criteria. AICPA. Retrieved from <https://www.aicpa.org>
- [14] National Institute of Standards and Technology. (2020). Security and privacy controls for information systems and organizations (NIST SP 800-53 Rev. 5). NIST. Retrieved from <https://www.nist.gov>
- [15] Laurie, B., Langley, A., & Kasper, E. (2013). Certificate transparency (RFC 6962). Internet Engineering Task Force. Retrieved from <https://www.rfceditor.org>
- [16] ISO/IEC. (2022). ISO/IEC 27002:2022 information security controls. International Organization for Standardization. Retrieved from <https://www.iso.org>
- [17] OWASP GenAI Security Project. (2026). Top 10 for LLMs and generative AI applications. OWASP. Retrieved from <https://owasp.org>
- [18] Financial Action Task Force (FATF). (2026). The FATF recommendations. FATF. Retrieved from <https://www.fatf-gafi.org>

## СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

- [1] Financial Action Task Force (FATF). (2012). International standards on combating money laundering and the financing of terrorism and proliferation: The FATF recommendations. FATF. Retrieved from <https://www.fatf-gafi.org>
- [2] Financial Action Task Force (FATF). (2014). Guidance for a risk-based approach: The banking sector. FATF. Retrieved from <https://www.fatf-gafi.org>
- [3] Basel Committee on Banking Supervision. (2020). Sound management of risks related to money laundering and financing of terrorism. Bank for International Settlements. Retrieved from <https://www.bis.org>
- [4] European Banking Authority. (2021). Guidelines on ML/TF risk factors. EBA. Retrieved from <https://www.eba.europa.eu>
- [5] Federal Financial Institutions Examination Council. (2026). Bank Secrecy Act / Anti-Money Laundering examination manual. FFIEC. Retrieved from <https://www.ffiec.gov>
- [6] National Institute of Standards and Technology. (2023). Artificial intelligence risk management framework (AI RMF 1.0) (NIST AI 100-1). NIST. Retrieved from <https://www.nist.gov>
- [7] National Institute of Standards and Technology. (2024). AI RMF generative AI profile (NIST AI 600-1). NIST. Retrieved from <https://www.nist.gov>
- [8] OWASP Foundation. (2026). Top 10 for large language model applications. OWASP. Retrieved from <https://owasp.org>
- [9] Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., ... Riedel, S. (2020). Retrieval-augmented generation for knowledge-intensive NLP tasks. In Advances in Neural Information Processing Systems (NeurIPS). Retrieved from <https://papers.nips.cc>
- [10] European Union. (2024). Regulation (EU) 2024/1689 of the European Parliament and of the Council [Artificial Intelligence Act]. Retrieved from <https://eur-lex.europa.eu>
- [11] European Union. (2016). Regulation (EU) 2016/679 of the European Parliament and of the Council [General Data Protection Regulation]. Retrieved from <https://eur-lex.europa.eu>
- [12] ISO/IEC. (2022). ISO/IEC 27001:2022 information security management systems – Requirements. International Organization for Standardization. Retrieved from <https://www.iso.org>
- [13] American Institute of Certified Public Accountants. (2026). SOC 2: Trust services criteria. AICPA. Retrieved from <https://www.aicpa.org>
- [14] National Institute of Standards and Technology. (2020). Security and privacy controls for information systems and organizations (NIST SP 800-53 Rev. 5). NIST. Retrieved from <https://www.nist.gov>
- [15] Laurie, B., Langley, A., & Kasper, E. (2013). Certificate transparency (RFC 6962). Internet Engineering Task Force. Retrieved from <https://www.rfceditor.org>
- [16] ISO/IEC. (2022). ISO/IEC 27002:2022 information security controls. International Organization for Standardization. Retrieved from <https://www.iso.org>
- [17] OWASP GenAI Security Project. (2026). Top 10 for LLMs and generative AI applications. OWASP. Retrieved from <https://owasp.org>
- [18] Financial Action Task Force (FATF). (2026). The FATF recommendations. FATF. Retrieved from <https://www.fatf-gafi.org>

© Лапін М. В., Коломієць О. О.

Дата першого надходження статті до видання: 16.02.2026

Дата прийняття статті до друку після рецензування: 18.03.2026

Дата публікації (оприлюднення) статті: 22.05.2026



Стаття поширюється на умовах ліцензії відкритого доступу (CC BY 4.0)