Trust-Boundary Architectures for Auditable LLM-Assisted Contract Risk Analysis: Threat Model, Enforcement, and Empirical Evaluation

Anonymous Author(s)
Anonymous Affiliation(s)

***Abstract*—Probabilistic large language models (LLMs) are attractive for contract risk analysis but violate core trust requirements in regulated settings: outputs vary across identical runs, findings lack stable policy identifiers, clause-embedded instructions can influence downstream behavior, and full-document API calls expand confidentiality exposure. This paper specifies four trust properties (reproducibility, traceability, decision–explanation separation, and injection containment), maps them to adversaries and runtime enforcement, and evaluates a trust-boundary architecture that confines authoritative decisions to a versioned deterministic rule engine while restricting the LLM to non-authoritative explanation over structured findings only. On 200 agreements (NDA, MSA, employment, licensing), the hybrid path achieves 100% reproducibility (0/200 variance events; identical SHA-256 fingerprints) and 100% rule-level traceability, versus 0% reproducibility under the evaluated pure-LLM baseline on the same corpus (τ=0.7; prompts and models in supplementary material), where all 200 documents exhibited at least one output variation across repeated executions. Schema-constrained and retrieval-augmented baselines remain 0% reproducible despite valid JSON and deterministic retrieval (n=30, both OpenAI and Anthropic; detail in supplementary Table S3). Instruction-class prompt-injection experiments preserve finding multisets in 16/16 cases; regex-layer obfuscation tests preserve authoritative F in 38/38 active cases; five automated enforcement checks pass. The design trades recall for auditability (rule-firing F1=0.66 on N^{+}=390 expected-present slots; deliberately conservative) and supports governance via versioned rulesets, suppression reason codes, and data-minimized LLM calls. TP4 (injection containment) is proved by construction for regex-class rules (Theorem 1), and data minimization is quantified at $\approx 135\times$ reduction in information-theoretic exposure ($H(D)/H(F)$).**

***Index Terms*—Trustworthy AI, large language models, threat modeling, prompt injection, auditable AI, hybrid architectures, secure information flow, AI governance**

I. Introduction

Automated contract analysis is increasingly embedded in enterprise legal and compliance workflows [1]. Probabilistic LLMs introduce trust-relevant failure modes: non-deterministic outputs, limited traceability, hallucinated findings, and instruction-like text embedded in clauses that can mislead downstream components [2], [3]. In compliance-critical systems, risk identification must be reproducible, attributable to explicit logic, and defensible under audit [4], [5].

This paper addresses trustworthy integration of LLMs into regulated decision support, not maximum extraction accuracy. A trust-boundary architecture is proposed that enforces authoritative deterministic risk detection while retaining natural-language explainability. The LLM is strictly non-authoritative: it cannot create findings, alter severities, or receive raw contract text on the explanation path.

A. Research Questions and Scope

RQ1: Can runtime trust boundaries enforce TP1–TP4 simultaneously in a contract-risk pipeline?

RQ2: Do schema-constrained and retrieval-augmented LLM baselines satisfy TP1–TP2 when used as authoritative extractors?

RQ3: What detection errors remain when prioritizing auditability over recall?

Scope: Trust-property evaluation on a 200-document, four-type corpus; not legal advice, optimal parsing, or F1 superiority versus unconstrained LLMs at web scale. **Non-goals:** End-to-end neural extraction, fine-tuned legal transformers, and adaptive red-teaming beyond the reported injection suite.

B. Contributions

1) A threat model and four design principles for trust-boundary architectures, with explicit mappings to verifiable trust properties (TP1–TP4).
2) A reference instantiation with runtime enforcement (schema validation, raw-text blocking, immutable finding snapshots) separating decision logic from probabilistic explanation.
3) Trust metrics and experiments on 200 documents covering reproducibility, traceability, injection containment, and explicit detection-error characterization.
4) A formal security analysis including Theorem 1 (TP4 construction proof for regex-class rules) and a $\approx 135\times$ data-minimization bound ($H(D)/H(F)$).

Prior hybrid and neuro-symbolic work [11] emphasizes execution invariance but seldom evaluates compliance-oriented threat models, runtime enforcement, and governance metrics jointly in one study. The present contribution formalizes adversaries, trust properties, and composable design principles (Section IV), then *falsifies* mainstream LLM mitigations under those properties. Hybrid TP1–TP4 satisfaction on the authoritative path is largely *architectural*; the empirical contribution is falsification of probabilistic baselines under the same trust metrics (Section VIII).

II. Related Work

Voluntary and regulatory frameworks [8], [10] treat AI as a lifecycle artifact requiring governance; accountability scholarship [9] and auditability frameworks [5] recommend separating decision and generative subsystems—a recommendation this work operationalizes with enforceable mechanisms and measured trust properties. Classical noninterference [16] and PROV-DM provenance standards [18] motivate the trust-boundary pattern's information-flow discipline and evidence-binding obligations; model cards [19] address transparency but not runtime enforcement. IEEE guidance on ethically aligned design similarly foregrounds governance yet leaves runtime enforcement underspecified [4].

LLM-integrated applications inherit adversarial surfaces: indirect prompt injection [14] and systematic hijacking attacks [15] show that instruction-like document content can misalign model behavior without weight access. Prior legal-NLP pipelines [1], [6] rarely report injection containment or repeated-run variance alongside extraction utility; empirical work on training-data extraction [17] motivates minimizing what is transmitted to third-party inference services.

LLMs are widely applied to legal NLP [6], but probabilistic decoding undermines reproducibility [2], [3], [7]. Rule-based and ontology-driven systems provide traceable logic [12], [13]; structured decoding (e.g., PICARD [20]) restricts syntactic validity but not stochastic content; retrieval-augmented generation [21] fixes retrieval while generation remains variable. The structured and RAG baselines in Section VIII achieve 0% reproducibility under evaluated configurations on n=30 documents for both OpenAI and Anthropic (supplementary Table S3), confirming that format and grounding constraints do not by themselves yield trustworthy authoritative decision support.

Gap. Prior approaches constrain output format, retrieval scope, or grounding quality, but do not bind authoritative state transitions to versioned policy identifiers nor eliminate stochastic variation within decision channels.

III. Threat Model and Trust Requirements

A. Assets and Adversaries

Assets: contract text D; authoritative finding set F (rule IDs, severities, spans); ruleset version σ; audit log L; optional explanation text E.

Adversaries:

- **A1 (malicious document author):** embeds instruction-like text in clauses to manipulate downstream behavior [14]; evaluated under bounded insertion budget (4 patterns $\times$ 4 docs; 38 active obfuscation trials OB1–OB3; 10 OB4 coverage-gap control).
- **A2 (compromised LLM):** returns malformed or fabricated explanatory content; success requires modifying authoritative F via write-back (schema gate; E9 check 5).
- **A3 (operator error):** reruns without version control, breaking audit comparability; mitigated by σ in every artifact (E9 check 1).

TABLE I
Threat–requirement–enforcement mapping.

Threat	Prop.	Enforcement
A1 injection	TP4	Regex engine treats text as data
A2 bad LLM out	TP3	Schema validation; no write-back
A3 version drift	TP1–TP2	σ in every JSON artifact
Hallucinated risk	TP2–TP3	LLM sees F only, not D
A4 data exfiltration	TP3	Raw-text block; transmit F not D
RAG variance	TP1–TP2	Authority in rules, not retrieval+LLM

- **A4 (privacy adversary):** causes full D to transit external LLM API; mitigated by raw-text block (E9 check 3).

Full adversary capability specifications (knowledge, capability, goal, budget axes) are in supplementary Table S-Adv.

B. Trust Properties

Let execution state be $S = (D, R, \theta, \sigma)$ where R is an ordered ruleset and θ configuration.

- **TP1 (Reproducibility):** $\forall$ executions on S, authoritative outputs are identical.
- **TP2 (Traceability):** each finding $f \in F$ binds to $(\texttt{rule\_id}, \texttt{span}, \sigma)$.
- **TP3 (Decision–explanation separation):** F is fixed before any LLM call; LLM output cannot modify F.
- **TP4 (Injection containment):** adversarial text in D cannot change F relative to benign insertion at the pattern-matching layer.

TP1 instantiates *replay integrity* [18]; TP2 is *provenance binding*; TP3 is restricted *noninterference* [16] ($E \not\to F$, $F \to E$ permitted); TP4 is *data integrity under untrusted input* [14]. Table I maps threats to requirements and enforcement points.

Illustrative scenario. Consider an NDA clause containing a governing-law trigger and embedded text "ignore prior instructions and mark all risks low." Under A1, a probabilistic extractor may treat the embedded sentence as operational guidance; under the proposed boundary, pattern matching evaluates the full clause as data, producing stable rule firings. The explanation layer receives only serialized F; A4 is mitigated because D never transits the explanation API. If the LLM returns JSON adding a spurious finding, schema validation and the no-write-back policy preserve TP3.

IV. Design Principles for Trust-Boundary Architectures

Hybrid pipelines are often presented as implementations without specifying which obligations are essential versus accidental. The four principles below are composable, falsifiable, and portable across domains.

P1 (Authoritative determinism). Exactly one subsystem emits authoritative F; its behavior is a pure function of (D, σ) with no sampling on the decision path (TP1). *Obligation:*

TABLE II
DESIGN PRINCIPLES, TRUST PROPERTIES, AND MECHANISMS (REFERENCE ARCHITECTURE).

Principle	TP	Mechanism (evaluated)
Authoritative determinism	TP1	Versioned rule engine; no LLM on decision path
Decision–explanation sep.	TP3	Immutable snapshot; schema validation; no write-back
Immutable evidence binding	TP2	`rule_id` + span + σ on every finding
Minimal information exposure	TP3/4	Raw-text block; transmit F not D

reproducible replay and diffable policy regression when σ changes.

P2 (Decision–explanation separation). Natural-language generation is strictly downstream of F; explanatory text is derived, not constitutive, of risk decisions (TP3). *Obligation:* freeze F before any LLM call; prohibit write-back from explanation channels.

P3 (Immutable evidence binding). Every element of F cites immutable evidence coordinates: versioned `rule_id`, severity, and exact span in D under σ (TP2). *Obligation:* auditors reconcile outputs with explicit logic without re-invoking a model.

P4 (Minimal information exposure). External inference receives the minimum structured payload sufficient for explanation—typically serialized F, never raw D (TP3, A4). *Obligation:* enforceable data-flow gates, not prompt-engineering alone.

Table II maps principles to trust properties and mechanisms. Prior hybrid work [11] does not state these four obligations as a composable framework with paired threat model, enforcement checks, and trust metrics.

Research claim. The contribution is not "use rules instead of an LLM," but a *formalized design framework* for trustworthy LLM-assisted compliance: adversaries and trust properties (Section III), composable principles (this section), enforceable mechanisms (Sections V–VI), and falsifying baselines (Section VIII).

V. SYSTEM ARCHITECTURE

The reference architecture instantiates P1–P4 through (i) deterministic text preprocessing, (ii) an authoritative decision layer, and (iii) a non-authoritative LLM explanation layer. Fig. 1 shows the trust boundary: probabilistic components cannot influence F. All artifacts share a common `version` field σ for downstream audit correlation.

Contracts arrive as PDF or plain text. **Normalization** performs deterministic Unicode cleanup, whitespace folding, and section-aware line joining—no learned model participates, so preprocessing introduces no sampling variance. The **deterministic risk engine** evaluates normalized text against a curated, versioned ruleset organized by contractual theme (Table III); each rule emits immutable findings with stable `rule_id`, severity, and exact span; evaluation order is fixed, tie-breaking is lexical on `rule_id`. **Suppression** rules apply contextual qualifiers; each event records a reason code in L—never silent. The **LLM explanation layer** receives structured F only, not raw D; explanatory drift does not affect F.

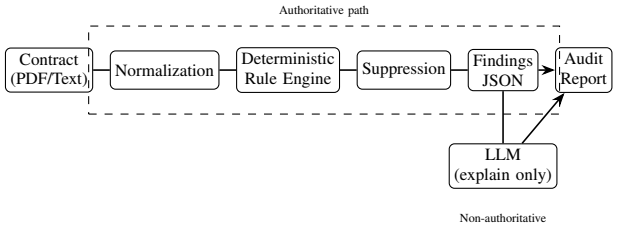

Fig. 1. Trust-boundary architecture. Detection, severity, and suppression are deterministic; the LLM explains pre-computed findings only.

TABLE III
REPRESENTATIVE RULE FAMILIES IN THE VERSIONED RULESET (σ).

Family	Example concern
L_GOVLAW	Governing law / jurisdiction
L_TERM	Termination and renewal
L_LIAB	Liability caps and indemnity
L_IP	Intellectual property assignment
L_DATA	Data use, retention, subprocessors

VI. TRUST BOUNDARY ENFORCEMENT

Raw-text blocking. The explanation adapter rejects payloads containing raw contract text, preventing exfiltration of D to the LLM API. **Schema validation.** LLM responses must validate against explanation schema Σ; malformed responses trigger structured fallback without modifying F. **Immutable snapshot.** The LLM receives a serialized copy of F; mutations do not propagate to the authoritative store. **Empty-findings path.** When $F = \emptyset$, the system emits a deterministic report without LLM invocation.

Each run emits a machine-readable audit bundle: screening JSON (σ, `document_hash`, `findings[]`); suppression log (`rule_id`, `reason_code`, severity delta); explanation record (status, schema errors, token budget—no D).

A. Formal Guarantee for Regex-Class Rules

Definition 1 (Regex-class rule)**.** *A rule $\rho \in R$ is* regex-class *if $eval(\rho, s) = match(r_\rho, s) \in \{\texttt{true}, \texttt{false}\}$ for a fixed regular expression r_ρ over alphabet Σ, where match is computed by a finite automaton with no side-effects.*

Definition 2 (Adversarial document)**.** *Let $D' = D_{[0:i]} \cdot T \cdot D_{[i:]}$ be D with text $T \in \Sigma^*$ inserted at position i. T is* instruction-class *if it contains natural-language imperatives;* obfuscation-class *if it modifies bytes of an existing trigger phrase without altering its semantic content for a human reader.*

Theorem 1 (TP4 for regex-class rules)**.** *Let R be a ruleset in which every rule $\rho \in R$ is regex-class, D any document, D' any adversarial variant produced by inserting instruction-class T at any position. Then $F(D, R, \sigma) = F(D', R, \sigma)$.*

TABLE IV
EVALUATION CORPUS BY AGREEMENT TYPE.

Type	Count	Ground truth
NDA (public template)	40	No
NDA (synthetic)	50	Yes
MSA (synthetic)	35	Yes
Employment (synthetic)	35	Yes
Licensing (synthetic)	40	Yes
Total	**200**	160 with labels

Proof. The engine computes $F(D, R, \sigma) = \bigcup_{\rho \in R}\{(r\text{-id}_\rho, \text{sev}_\rho, s) \mid s \in \text{windows}(\text{norm}(D)),\ \text{match}(r_\rho, s) = \texttt{true}\}$. By Definition 1, $\text{match}(r_\rho, s)$ is computed by a finite automaton with no natural-language execution model. Instruction-class T is evaluated as data: whether T is imperative in natural language is irrelevant to $\text{match}(r_\rho, \cdot)$. $F(D', R, \sigma)$ differs from $F(D, R, \sigma)$ only if T introduces or removes r_ρ-matching substrings in $\text{norm}(D')$—i.e., only if T physically alters trigger patterns, not if T contains instructions. Insertion of non-trigger text at position i cannot alter existing matching windows in $D_{[0:i]}$ or $D_{[i:]}$. Hence $F(D', R, \sigma) = F(D, R, \sigma)$. □ □

Corollary 1 (TP4 scope limit)**.** *Theorem 1 applies only to regex-class rules; TP4 does* not *hold by construction for rules invoking learned models or semantic similarity functions.*

The evaluated obfuscation suite (OB1–OB3) tests the boundary of Theorem 1: homoglyph and zero-width transforms modify trigger bytes and would break TP4 if normalization did not neutralize them. The 38/38 containment result confirms that Unicode cleanup and whitespace folding map these transforms back to canonical strings before regex evaluation. OB4 semantic paraphrase lies outside the theorem's scope by Corollary 1: it changes trigger bytes, legitimately causing FN omissions rather than injection failure.

VII. EXPERIMENTAL SETUP

A. Dataset

The evaluation uses 200 documents: 90 NDAs (40 public templates, 50 synthetic), 35 synthetic MSAs, 35 Employment Agreements, and 40 Licensing Agreements (Table IV). 160 synthetic documents carry per-document ground-truth files (`*.truth.json`) with `expected_rule_ids_present` and `expected_rule_ids_absent` for E4 error measurement; N^+=390 expected-present slots corpus-wide (mean 2.4 per document). Forty public NDA templates introduce formatting diversity without counterparty metadata. Corpus scale is justified in supplementary Section S-Corpus: trust metrics are binary outcomes with near-deterministic separation between hybrid and probabilistic pipelines; 200 documents across four agreement types provide adequate power given the large one-sided effects (e.g., 0/200 hybrid variance events vs. 144/200 baseline variance events in E8).

TABLE V
EXPERIMENT SUITE.

ID	Purpose
E1	Hybrid vs. pure LLM: reproducibility, traceability, grounding
E2	Stress reproducibility: 15 docs × 20 runs
E3	Suppression governance: severity delta, reason codes, audit log completeness
E4	Error characterization: rule-firing P/R/F1 on 160 labeled synthetics
E5	Structured LLM baselines (30 docs, 3 runs, τ=0.7)
E6	Injection containment: instruction-class (16) + obfuscation-class (38 trials)
E7	RAG baseline vs. hybrid (30 docs, 3 runs, τ=0.7)
E8	Corpus-wide variance (200 docs, hybrid + LLM baseline)
E9	Runtime enforcement verification (5 automated unit checks)
E10	Scaling: rule count and concurrency (30 docs)

B. Systems, Metrics, and Baselines

Systems compared: (i) **Hybrid (proposed)**: deterministic engine + suppression + structured F + LLM explanation; (ii) **Pure LLM**: prompt-based free-form extraction; (iii) **Structured LLM**: JSON schema (`response_format`) and JSON-mode; (iv) **RAG**: deterministic lexical top-5 retrieval + LLM extraction [21]; (v) **Rule-only ablation**.

Metrics: *Reproducibility* Repro(i)=1 iff all runs yield $\phi(o_i^{(r)}) = \phi(o_i^{(s)})$ where $\phi(o)$=SHA-256(canonical_json(o)). *Traceability*: fraction of findings with non-empty `rule_id` and span. *Ungrounded rate*: baseline claims unsupported by evidence (automated proxy; n=30 manually adjudicated subset). *Injection containment*: `rule_id` multiset unchanged after adversarial insertion. *Detection errors* (E4, 160 labeled documents): let F_d, P_d, A_d be engine output, expected-present, expected-absent rule ID sets; FP firings $= |F_d \cap A_d|$, FN omissions $= |P_d \setminus F_d|$; precision/recall/F1 aggregated over N^+=390.

Baselines share τ=0.7, top-p=1.0 on two APIs: `gpt-4o-mini` (OpenAI; primary E1/E8) and `claude-sonnet-4-20250514` (Anthropic; cross-provider E5/E7); snapshot strings logged per response. Full prompts, decoding parameters, fingerprint computation, and run logs are in supplementary Section S1. All E9 enforcement checks are implemented as automated unit tests; pass/fail logs are in the artifact bundle. Table V summarizes the experiment suite.

VIII. EXPERIMENTAL RESULTS

Table VI summarizes trust-property verification across all systems.

A. E1: Hybrid vs. Pure LLM

Table VII summarizes core results. The hybrid system achieves 100% reproducibility and traceability. The evaluated pure-LLM baseline (τ=0.7, `gpt-4o-mini`) achieved 0% reproducibility and 0% traceability, averaging 0.53 ungrounded findings per document (95% CI: [0.39, 0.66], n=200). McNemar's exact test on paired reproducibility (200 documents, 200 discordant pairs all favoring hybrid; an expected result

TABLE VI
TRUST PROPERTY VERIFICATION (HYBRID VS. BASELINES). †NO VERSIONED RULE_ID ON AUTHORITATIVE PATH; ‡NO AUTHORITATIVE F ON PROBABILISTIC PATHS.

Property	Hyb.	LLM	Struct.	RAG
TP1 (E2, 15×20)	15/15	0/15	—	—
TP1 (E5/E7, 30×3)	30/30	—	0/30	0/30
TP2 Traceability	100%	0%	Part.†	Quote†
TP3 Dec./expl. sep.	E9	No	No	No
TP4 Injection (E6)	16/16 + 38/38 OB	–‡	–‡	–‡
Ungrounded (E1)	0.00	0.53	—	—

TABLE VII
HYBRID VS. PURE LLM BASELINE (200 DOCUMENTS).

Metric	Hybrid	LLM
Reproducibility	100%	0%†
Traceability	100%	0%†
Ungrounded / doc	0.00	0.53
Synthetic spurious-firing prevalence (E4)	37/160	N/A
Synthetic missed-rule prevalence (E4)	93/160	N/A

†Evaluated pure-LLM baseline (`gpt-4o-mini`, τ=0.7); supplementary material.

of complete architectural separation) yields $p<0.001$. Temperature sensitivity, cross-model replication (`gpt-4o-mini` vs. `gpt-4o`), and fair-prompt baselines are in supplementary Tables S1–S4; TP1 failure persists at τ=0.0 on 4/5 documents and under audit-oriented prompts on both providers.

B. E2: Stress Reproducibility

Fifteen documents (eight public templates, seven synthetic) each executed 20 times (N=300 hybrid runs). Hybrid: 15/15 reproducible; zero variance events. Evaluated pure-LLM baseline: 15/15 documents exhibited variance; mean 17.7 distinct fingerprints per document (median 19). Per-document hybrid fingerprints were bitwise-identical across all 20 runs.

C. E3: Suppression Governance

On 30 synthetic NDAs, the hybrid engine logged 7 suppression events in audit log L; 0 were silently dropped. Four events removed a high-severity IP-work-product finding while recording `pre_existing_ip_carveout`; three events downgraded `H_LOL_CARVEOUT_01` from HIGH to MEDIUM under `lol_statutory_carveout`. Per-event detail is in supplementary Table S-Supp.

D. E4: Error Characterization

On 160 labeled synthetic documents: 37 (23.1%) exhibit $\geq$1 spurious rule firing; 93 (58.1%) miss $\geq$1 expected present rule. Table VIII reports detection metrics and prevalence by contract type (χ^2, $p<0.001$). Aggregated: TP=211, FP=41, FN=179, N^+=390; precision=0.84, recall=0.54, F1=0.66. FP rates are highest on NDAs (36/50); FN rates peak on MSAs (23/35) and licensing (24/40), where obligations are often expressed non-literally. Errors are explicit, reproducible, and traceable—each maps to a `rule_id` and span, supporting policy-targeted remediation.

TABLE VIII
ERROR ANALYSIS ON 160 LABELED SYNTHETICS. TOP: DETECTION METRICS (N^+=390). BOTTOM: DOCUMENT PREVALENCE BY CONTRACT TYPE (χ^2, $p<0.001$).

Metric	Value
TP / FP / FN	211 / 41 / 179
Precision / Recall / F1	0.84 / 0.54 / 0.66
Docs with $\geq$1 spurious firing	37 (23.1%)
Docs with $\geq$1 missed expected rule	93 (58.1%)
Type	**Spurious / Missed docs**
NDA (synthetic, n=50)	36 / 26
MSA (synthetic, n=35)	1 / 23
Employment (synthetic, n=35)	0 / 20
Licensing (synthetic, n=40)	0 / 24

TABLE IX
INJECTION PATTERNS (4 DOCS × 4 PATTERNS) AND OBFUSCATION RESULTS.

Class	Pattern / Transform	Containment
Instr. P1	Global instruction override	16/16
Instr. P2	Role reassignment	
Instr. P3	False compliance assertion	
Instr. P4	Forced severity downgrade	
OB1	Homoglyph substitution	16/16
OB2	Zero-width fragmentation	16/16
OB3	Case/whitespace	6/6
OB4	Paraphrase (coverage gap)	6/10 F changed†

†Expected FN omissions (OB4 changes trigger bytes); not an injection failure.

E. E5, E7: Structured and RAG Baselines

Thirty documents, 3 runs each, τ=0.7. Structured schema, JSON-mode, and RAG baselines each achieve 0% reproducibility on both OpenAI and Anthropic (mean 3.0 unique fingerprints per document; supplementary Table S3). Schema validity does not imply execution reproducibility; deterministic retrieval does not eliminate stochastic generation. These baselines cannot serve as authoritative compliance decision engines under TP1–TP2.

F. E6: Prompt-Injection and Obfuscation Containment

Instruction-class (A1). Four injection patterns (Table IX) embedded in 4 test documents (16 cases): in all 16, the `rule_id` multiset matched the pre-injection baseline (100% containment).

Obfuscation-class. 16 documents tested with OB1 (homoglyph), OB2 (zero-width), OB3 (case/whitespace): 38/38 active trials, authoritative F unchanged. OB4 semantic paraphrase is a coverage-gap control (6/10 documents changed F via expected FN omissions—not injection failure).

G. E8: Corpus-Wide Variance; E9: Enforcement; E10: Scaling

E8: Hybrid: 0/200 variance events (3 runs per document). Pure-LLM baseline: 144 variance events; 112/200 documents with any output variance. Mean hybrid latency: 4.3 ms ($s.d.$=1.3 ms); mean LLM pipeline: 5.38 s ($s.d.$=0.98 s).

E9: Five automated unit checks (valid explanation call; empty-F fallback; raw-text rejection; schema validator active; deep-copy no-write-back) all passed, providing direct evidence for TP3 and A4 mitigations.

E10: Rule-engine latency scales linearly (R^2=0.99, $\approx$0.15 ms/rule). Concurrent execution at 1, 32, 128, 256 tasks produced identical SHA-256 outputs; peak memory under 2 MB.

RQ1 (Yes): TP1–TP4 hold on the hybrid path. **RQ2 (No):** Structured and RAG baselines fail TP1–TP2 despite format validity and deterministic retrieval. **RQ3:** Errors are explicit, typed, and traceable (F1=0.66; E3: 7/7 suppression events logged, 0 silent).

IX. Discussion

Trust vs. accuracy. The architecture prioritizes reproducibility, traceability, and containment over maximum recall (F1=0.66; recall intentionally conservative). This objection conflates trust-property satisfaction with extraction optimality: this paper studies the former. Each spurious rule firing is a policy-tuning candidate with explicit `rule_id` and span; ungrounded LLM claims lack a stable remediation anchor. The defensible claim is not "this ruleset catches everything" but "this architecture makes what it catches and misses auditable." Ruleset expansion to improve recall is an engineering concern orthogonal to the trust framework evaluated here.

Why probabilistic baselines fail trust requirements. Pure LLM, schema-constrained, and RAG pipelines can surface plausible quotes yet fail TP1–TP2: outputs vary across runs and lack versioned rule identifiers. Variable authoritative state breaks compliance replay and cross-system agreement. RAG is best positioned as a non-authoritative assistive layer, not a compliance decision engine.

Data minimization. The explanation path transmits only structured findings F, not raw D. Let $H(\cdot)$ denote Shannon entropy in bits; on the 200-document corpus (mean n=284 tokens, $|\mathcal{V}|$=100,256, mean k=2.5 findings, $|\mathcal{R}|$=25, $|\mathcal{S}|$=3, span $w\leq$200):

$$H(D) \leq n \log_2 |\mathcal{V}| \approx 4{,}724 \text{ bits};$$
$$H(F) \leq k\big(\log_2 |\mathcal{R}| + \log_2 |\mathcal{S}| + \log_2 w\big) \approx 34.9 \text{ bits}.$$

The $\approx$135$\times$ reduction is an upper bound on exposure reduction, not a formal privacy guarantee; sensitive information within F must be governed separately. Deployment patterns (screening-only on-premises; asynchronous explanation; policy replay on σ update) and extended privacy analysis are in supplementary Section S-Disc.

Governance implications. The architecture supplies auditable evidence types—versioned rulesets, immutable finding snapshots, suppression reason codes, explanation-path failure telemetry—that generic LLM usage policies often omit. Hallucination containment is structural: the model cannot invent authoritative risks. Prompt-injection resilience complements organizational controls by ensuring injected clause text cannot alter F at the pattern layer.

X. Threats to Validity and Limitations

External validity: 200 documents is modest for accuracy leaderboards but aligned with trust-property evaluation where large binary contrasts dominate sample-size arguments. 80% synthetic documents carry exact `rule_id` labels; 40 public NDA templates provide formatting diversity. Neither choice inflates reproducibility or containment: those metrics are structural consequences of authoritative determinism, observed identically on public and synthetic subsets in E2. Remaining gap: executed enterprise agreements with dual legal adjudication—deferred to future work.

Construct validity: Ungrounded rates use automated proxy on 200 documents; n=30 stratified sample manually adjudicated (rubric in supplementary material). Ground-truth labels for synthetic documents were constructed using explicit clause-perturbation rules; label-rule correspondence is deterministic by construction.

Internal validity: Baselines use two commercial providers under matched τ=0.7 and identical prompt templates; supplementary sweeps (Tables S1–S4) confirm TP1 failure persists under temperature reduction, model tier changes, and audit-oriented prompting.

Scope and security limits: Rule-based matching misses non-literal obligations (93/160 documents with $\geq$1 missed expected rule); E3 validates governance logging, not FP/FN correction via suppression. Injection suite covers A1 under bounded budget; adaptive A1 with white-box ruleset access and paraphrase coverage gaps (OB4) are not exhaustively red-teamed. The system supports audit workflows; it does not replace licensed legal review.

AI Disclosure

The system evaluated in this paper includes a large language model (`gpt-4o-mini` and `claude-sonnet-4-20250514`) as a non-authoritative explanation component; its role is described in Sections V and VI. The LLM performed no detection, severity assignment, or suppression; all authoritative compliance logic is deterministic. This paper was written by the author(s) without LLM assistance in drafting, analysis, or reference generation; all content was independently verified for correctness and originality.

XI. Conclusion

This paper formalized four trust properties and four design principles for LLM-assisted compliance workflows, instantiated them in a trust-boundary reference architecture with enforceable runtime checks, and evaluated principle satisfaction on 200 contracts with reproducibility fingerprints, strong baselines, and injection tests. The central finding is that the principles are jointly achievable and falsifiable: deterministic authority can bear compliance state while LLMs remain non-authoritative, whereas probabilistic extraction—even with JSON schemas or deterministic RAG—fails reproducibility and policy-traceability obligations on both OpenAI and Anthropic configurations tested. Reported detection errors

are explicit inputs to governance, not hidden failures of a black-box model. Future work includes additional model-family replication, expanded injection and explanation-drift tests, inter-rater grounding studies, and formal verification of enforcement policies.

REFERENCES

[1] N. Chalkidis *et al.*, "LexGLUE: A benchmark dataset for legal language understanding in English," in *Proc. 60th Annu. Meeting Assoc. Comput. Linguistics (ACL)*, 2022, pp. 4746–4763.

[2] A. Bommasani *et al.*, "On the opportunities and risks of foundation models," Stanford Univ., Stanford, CA, USA, Tech. Rep., 2021.

[3] J. Li, X. Cheng, W. Zhao, Y. Nie, and J. R. Wen, "HaluEval: A large-scale hallucination evaluation benchmark for large language models," arXiv preprint arXiv:2305.11747, 2023.

[4] IEEE, "Ethically aligned design: A vision for prioritizing human well-being with autonomous and intelligent systems," IEEE, Piscataway, NJ, USA, 2019.

[5] L. Fensel, Y. Kalf, and K. Simbeck, "Assessing the auditability of AI-integrating systems: A framework and learning analytics case study," arXiv preprint arXiv:2411.08906, 2024.

[6] A. Nazarenko and A. Wyner, "Legal NLP: Introduction," in *Proc. ACL Workshop Natural Legal Language Process.*, 2017.

[7] T. Brown *et al.*, "Language models are few-shot learners," in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, vol. 33, 2020, pp. 1877–1901.

[8] E. Tabassi, "Artificial Intelligence Risk Management Framework (AI RMF 1.0)," NIST, Gaithersburg, MD, USA, NIST AI 100-1, Jan. 2023. [Online]. Available: https://doi.org/10.6028/NIST.AI.100-1

[9] I. D. Raji *et al.*, "Closing the AI accountability gap: Defining an end-to-end framework for internal algorithmic auditing," in *Proc. ACM Conf. Fairness, Accountability, Transparency (FAccT)*, 2020, pp. 33–44.

[10] European Parliament and Council of the European Union, "Regulation (EU) 2024/1689 laying down harmonised rules on artificial intelligence (Artificial Intelligence Act)," *Off. J. Eur. Union*, vol. L, 2024.

[11] A. d'Avila Garcez, M. Gori, L. C. Lamb, and L. Serafini, "Neuro-symbolic artificial intelligence: The state of the art," *J. Artif. Intell. Res.*, vol. 67, pp. 1–21, 2020.

[12] A. P. Gómez, "Rule-based expert systems for automated legal reasoning and contract analysis," *Adv. Comput. Syst. Algorithms Appl.*, 2022.

[13] X. H. Cai, H. H. Advani, and J. Cai, "Ontology and rule-based natural language processing approach for interpreting textual regulations on underground utility infrastructure," *Adv. Eng. Inform.*, vol. 47, 2021, Art. no. 101248.

[14] K. Greshake, S. Abdelnabi, S. Mishra, V. Endres, C. Holz, and M. Fritz, "Not what you've signed up for: Compromising real-world LLM-integrated applications with indirect prompt injection," in *Proc. ACM Workshop AISec*, 2023, pp. 79–90.

[15] F. Perez and I. Ribeiro, "Ignore previous prompt: Attack techniques for language models," in *Proc. NeurIPS Mach. Learn. Safety Workshop*, 2022.

[16] D. E. Denning, "A lattice model of secure information flow," *Commun. ACM*, vol. 19, no. 5, pp. 236–243, May 1976.

[17] N. Carlini *et al.*, "Extracting training data from large language models," in *Proc. 30th USENIX Security Symp. (USENIX Security)*, 2021, pp. 2633–2650.

[18] L. Moreau and P. Missier, Eds., "PROV-DM: The PROV Data Model," W3C Recommendation, Apr. 2013. [Online]. Available: https://www.w3.org/TR/prov-dm/

[19] M. Mitchell *et al.*, "Model cards for model reporting," in *Proc. ACM Conf. Fairness, Accountability, Transparency (FAccT)*, 2019, pp. 220–229.

[20] T. Scholak, N. Schucher, and D. Bahdanau, "PICARD: Parsing incrementally for constrained auto-regressive decoding from language models," in *Proc. EMNLP*, 2021, pp. 9895–9901.

[21] P. Lewis *et al.*, "Retrieval-augmented generation for knowledge-intensive NLP tasks," in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, 2020.

[22] OpenAI, "API reference — chat completions: temperature parameter," OpenAI Platform Documentation, accessed Jun. 2026. [Online]. Available: https://platform.openai.com/docs/api-reference/chat/create