Balancing Determinism and Stochasticity in a Public-Facing Immigration-Law Chatbot

A Guardrail Architecture for Natural-yet-Predictable Conversational AI

Revised academic draft | anonymized operational context | prepared as a technical white paper

JAIPTEAM Inc. | July 10, 2026

Abstract

Large language models (LLMs) generate text by sampling from learned probability distributions. Stochasticity is central to this process rather than incidental: it is among the conditions that make contemporary LLM interfaces feel responsive, flexible, and natural. In regulated, public-facing applications, however, the same property generates a reliability problem. A fixed user input can produce materially different responses across invocations, and the variation may appear precisely in the elements that a deploying organization most needs to control.

This paper examines that problem in the concrete setting of a public, pre-authentication conversational assistant deployed by a leading Canadian immigration legal technology platform, anonymized here as the Platform. The Platform connects users with licensed immigration lawyers rather than practicing law directly; its public assistant is designed for top-of-funnel triage and engagement. It recognizes visitor intent, reflects the visitor's situation in plain language, communicates the Platform's core differentiator, and routes the visitor toward an appropriate next step. At the same time, it must not provide individualized legal advice, predict immigration outcomes, fabricate procedural details, or omit mandatory professional-review disclosures.

The paper frames the resulting design problem as a determinism-naturalness trade-off. It distinguishes soft constraints, which can be encouraged through prompts and therefore remain probabilistic, from hard constraints, which are mechanically checkable and should be enforced deterministically outside the model. It then proposes a layered guardrail architecture that reserves the model for judgment-bound tasks such as intent recognition and conversational formulation while moving compliance-critical and business-critical facts into code, validation, and canonical resource resolution. The central claim is practical: in regulated conversational systems, prompt engineering is necessary but structurally insufficient. Reliability requires an intentional division of labor between probabilistic language generation and deterministic system design.

---

1. Introduction

Instruction-tuned LLMs have rapidly become credible front-line interfaces for professional-services organizations (Ouyang et al., 2022). Their value is readily apparent. A

public conversational assistant can meet an anonymous visitor at the moment of intent, ask clarifying questions, identify the broad shape of the matter, explain the organization's differentiators, and move the visitor toward a next step without requiring immediate human involvement. In high-consideration services, that early interaction shapes trust as much as it conveys information.

In a regulated domain, however, fluency is not enough. A chatbot operating in a legal-adjacent context cannot be assessed only by whether its responses are readable or helpful. It must also avoid statements that create legal, ethical, privacy, or commercial exposure. It must not provide individualized legal advice in a public channel. It must not predict a user's likelihood of success. It must not invent procedural requirements. It must not obscure the role of licensed professionals. These requirements are qualitatively different from tone preferences; they function as boundary conditions for safe deployment.

This paper examines that tension in the setting of a public, pre-authentication chatbot deployed by a leading Canadian immigration legal technology platform, referred to here as the Platform. The assistant's role is top-of-funnel rather than case-management. It helps distinguish, for example, prospective international students, sponsored workers, employers, and existing applicants. It reflects the visitor's situation, communicates that applications are reviewed by licensed immigration lawyers, and routes the visitor toward an appropriate next step. The system must do this while staying out of advice, avoiding procedural over-specificity, and preserving a natural conversational experience.

The practical question is therefore not whether the system should be deterministic or generative, since neither extreme produces a workable result. A fully scripted assistant is predictable but brittle and wooden; an unconstrained generative assistant is flexible but too variable for a regulated context. The more useful question is where determinism belongs. This paper argues that the answer lies in separating the tasks that require language judgment from the tasks that require exactness.

The paper makes three contributions. First, it explains the determinism-naturalness trade-off in a way that is useful to both technical and governance audiences. Second, it proposes a taxonomy that distinguishes mechanizable constraints from judgment-bound constraints. Third, it describes a guardrail architecture and evaluation methodology appropriate for stochastic systems operating in legal-adjacent, public-facing contexts.

2. Background: Why LLM Output Is Stochastic

Transformer-based language models generate language autoregressively (Vaswani et al., 2017; Brown et al., 2020). At each step in generation, the model computes a probability distribution over possible next tokens conditioned on the preceding context. A decoding strategy then selects the next token. This step is often invisible to non-technical stakeholders, but it is central to how the system behaves.

Deterministic decoding strategies such as greedy search or beam search can maximize likelihood, but they often produce repetitive, bland, or otherwise degenerate text.

Sampling-based methods, including temperature scaling and nucleus sampling, deliberately preserve randomness in order to produce more varied and human-like output (Holtzman et al., 2020). In many practical deployments, naturalness is partly purchased through stochasticity; the two are causally linked rather than merely compatible.

Instruction tuning, reinforcement learning from human feedback, and related alignment techniques improve model behavior by shifting the probability distribution toward responses that humans prefer (Ouyang et al., 2022; Bai et al., 2022, preprint). They do not, however, convert prompt instructions into hard constraints. A system instruction operates as a strong prior, not as a formal guarantee. Even low-temperature settings can still yield variation because non-zero sampling probability, implementation details, hardware differences, and model-serving architectures may affect output across runs.

Two additional behaviors matter for regulated systems. First, models can hallucinate: they can produce fluent statements that are unsupported, unfaithful, or false (Ji et al., 2023). In legal settings, that can become fabricated authority, invented procedure, or incorrect eligibility guidance (Dahl et al., 2024; Magesh et al., 2024, preprint). Second, models do not reliably use all provided context. Work on long-context behavior shows that relevant information may be underutilized depending on where it appears in the context window, a phenomenon commonly described as being 'lost in the middle' (Liu et al., 2024). A policy may therefore be present in the context without consistently governing the response.

3. The Determinism-Naturalness Trade-Off

The core design problem is a trade-off between predictability and conversational quality. At one pole are fully templated systems. They are straightforward to govern because each response can be reviewed and approved in advance, but their coverage is limited. Users rarely describe their circumstances in standardized language, and scripted flows often fail when the visitor's wording, emotional state, or intent does not match the anticipated path.

At the other pole are unconstrained generative systems. They respond fluidly to open-ended input and can appear significantly more helpful than a decision tree. Yet that flexibility is achieved by allowing the model to author responses probabilistically. In a legal-adjacent context, and particularly for a non-licensee operator whose product sits adjacent to licensed practice, this creates unacceptable exposure if the model improvises around legal boundaries, calls to action, disclosures, or factual claims. The tension is substantive as well as technical, raising questions about what kinds of outputs a deploying organization can legitimately attribute to a stochastic process (Bender et al., 2021).

Most production systems attempt to occupy the middle ground through guided generation. The system prompt, examples, retrieved documents, and operational policies shape the model's output while leaving the model free to compose natural language. This is often the right starting point, but it is not the same as deterministic control. Guided generation, however carefully configured, remains probabilistic generation.

Failure rates that appear acceptable in a small test set may be unacceptable in production. A rule that holds ninety percent of the time still fails approximately one interaction in ten, and the system does not indicate in advance which interaction that will be. For style, tone, or phrasing, that residual variation may be acceptable. For revenue-critical calls to action, mandatory disclosures, link accuracy, and legal scope boundaries, it is not.

4. Constraint Taxonomy: Mechanizable Versus Judgment-Bound

The relevant question is not how important a rule is (many rules are important) but whether compliance can be verified by a deterministic process after the model has produced a response. This paper distinguishes between mechanizable constraints and judgment-bound constraints.

Class	Definition	Examples in this domain	Deterministic enforcement?
Mechanizable	Compliance can be verified by checking the output string, structure, metadata, or selected resource.	Approved call-to-action URL is used; required lawyer-review disclosure is present; banned phrasing is absent; no enumerated procedural checklist appears.	Yes. Enforce through post-hoc validation, substitution, fixed rendering, placeholder resolution, or allowlist checks.
Judgment-bound	Compliance requires semantic interpretation of meaning, audience, context, or intent.	Recognizing visitor persona; identifying when a question crosses into individualized legal advice; matching tone and empathy to the visitor's situation.	No. Must remain model-mediated, although classifier models, second-pass review, and human escalation can reduce risk.

This distinction is the linchpin of the architecture. Mechanizable constraints should be removed from the probabilistic layer and enforced where determinism is available at low cost: in code, structured data, validation logic, or rendering. Judgment-bound constraints must remain model-mediated because they depend on semantic interpretation. They can be improved through prompt design, examples, fine-tuning, classifier checks, and escalation policy, but they cannot be guaranteed by the prompt alone.

A common design error is to keep strengthening prompts for rules that are better handled deterministically. For example, asking a model to 'always use the approved link' is weaker than resolving the link from an allowlist after generation. Asking it to 'always include the disclosure' is weaker than appending or validating the disclosure in code. In these cases, the problem is one of systems design rather than language modeling.

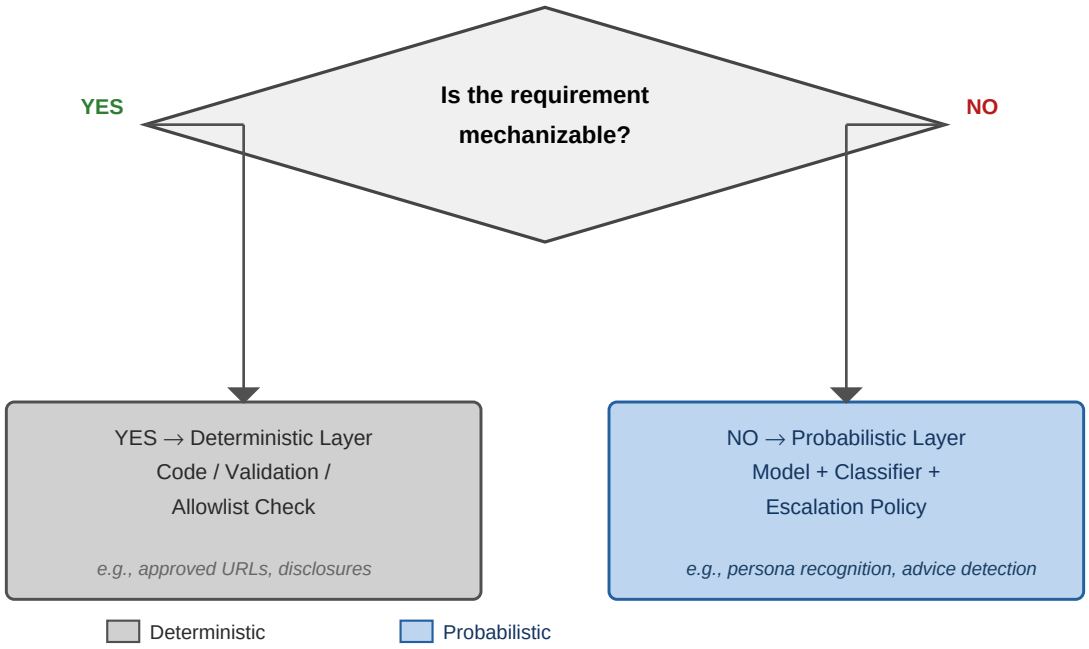

Figure 2. Constraint routing decision tree. Mechanizable requirements are removed from the probabilistic layer; judgment-bound requirements remain model-mediated with classifier and escalation support.

5. Empirical Observations from Structured Quality Assurance

5.0 Methodology

Quality assurance of the Platform's assistant was conducted across two formal evaluation rounds using automated browser-based testing, with prompt revisions applied between rounds. Nine visitor personas were defined to represent the principal user types encountered in a Canadian immigration law context: international student, sponsored worker, visitor/temporary resident, permanent-residence seeker, employer, designated learning institution, education agent, refusal-support seeker, and existing applicant. For each information-seeking persona, three paraphrase variants were constructed (an indirect framing, a direct framing, and a follow-up probe), yielding 45 total test runs across both evaluation rounds. Scoring applied a structured eight-dimension rubric covering persona recognition, professional-review disclosure, call-to-action quality, response conciseness, legal boundaries, competitor/DIY avoidance, returning-user routing, and performance. The permanent-residence persona, which presented the most complex regulatory surface, was tested with four prompt variants in the second round to characterize failure modes more precisely. Testing was conducted by a single automated reviewer; no inter-rater reliability coefficient was computed. Because scoring relied on a fixed rubric applied consistently, within-round consistency is high, but the absence of human validation is a limitation noted further in Section 9.

The goal was to characterize the distribution of responses the system produces under realistic phrasing variation, and to determine whether compliance failures clustered in mechanizable or

judgment-bound dimensions, rather than to confirm that the model could produce any single compliant response.

Table 1. Observed compliance-failure rates across evaluation rounds (proportion of test conversations in which the dimension failed; worst-case judgment applied to risk-bearing rules).

Compliance dimension	Pre-revision rate	Post-revision rate	Constraint class
Disclosure leakage (professional-review mention absent)	~100%	~8%	Mechanizable
CTA link non-canonical (wrong or shortened URL)	~100%	~69%*	Mechanizable
Advice-boundary violation (program-naming, eligibility endorsement)	0%	~15%	Judgment-bound
Approval-likelihood drift ('increases your chances')	0%	~15%	Judgment-bound

* Non-canonical CTA URLs were identified as a mechanizable constraint requiring deterministic resolution at the rendering layer, independent of further prompt revision. B2B destination URLs and authenticated-login links that appeared verbatim in the always-present system prompt resolved correctly throughout both rounds, confirming the mechanism described in Section 5.3.

Three patterns in these results are especially relevant to the architecture proposed in this paper.

5.1 Run-to-Run Variance on Identical Inputs

The same input could produce a compliant response in one invocation and a violation in another. In one run, the assistant might remain concise and properly scoped. In another, it might produce a procedural list that approached advice or exceeded the intended level of detail. This pattern demonstrates why single-sample testing is a weak basis for deployment decisions. A single successful response may represent a favorable draw from the distribution rather than reliable system behavior.

5.2 Phrasing-Sensitive Leakage

Rules such as mandatory professional-review disclosures and banned phrasings held under many user phrasings but failed under others. Prompt revisions reduced the rate of failure substantially (disclosure leakage fell from near-universal in the initial configuration to approximately 8% post-revision) but did not eliminate it. This is consistent with the view that prompt instructions adjust probabilities rather than impose invariants. Paraphrase robustness must therefore be part of evaluation, since real users will not phrase the same intent in one standardized way.

5.3 The Literal-String Effect

Canonical values were reproduced most reliably when placed verbatim in the always-present system prompt. Values that lived only in a separate policy document were less reliable. In some cases, the model regenerated a plausible substitute rather than reproducing the canonical value. This is consistent with known limitations in long-context utilization (Liu et al., 2024): even

when a policy is present in the context window, its influence on generation is not uniform and can diminish depending on position and salience. A single source-of-truth document is valuable for governance because it gives policy owners one place to edit and review. It does not, by itself, guarantee deterministic use at generation time.

The second evaluation round confirmed a clarifying contrast. URLs present verbatim in the always-present system prompt resolved correctly across all test conversations. URLs the model was expected to construct compositionally (persona-specific signup destinations) were non-canonical in the majority of cases, and this gap persisted across prompt revisions. The contrast reveals the underlying mechanism: verbatim reproduction from a high-salience location in the prompt is reliable, whereas compositional generation of exact values is not dependable.

Prompting had pushed reliability a substantial distance, and in that sense it succeeded. The remaining failures, however, clustered around the elements most suitable for deterministic enforcement: disclosures, approved links, banned phrasings, and canonical claims. That pattern supports the case for moving those elements outside the model.

6. A Layered Guardrail Architecture

The recommended design is a defense-in-depth architecture in which the model operates inside a deterministic envelope rather than as the entire system. This approach is consistent with emerging work on guardrails, input-output safeguards, and runtime control for LLM applications (Rebedea et al., 2023; Inan et al., 2023, preprint). The layers below are presented in processing order.

6.1 Input Guardrails

Before generation, the system should classify and filter incoming messages. It should detect off-topic, abusive, adversarial, or out-of-scope inputs; prompt-injection and jailbreak attempts, including indirect injection through retrieved or user-supplied content; and sensitive personal information that should not be handled in a pre-authentication channel (Greshake et al., 2023; OWASP Foundation, 2025). Inputs that are clearly unsafe or outside scope should be routed to deterministic responses rather than passed to open generation.

6.2 Grounding and Retrieval

Factual content should be grounded in vetted sources rather than relying on the model's parametric memory. Retrieval-augmented generation can improve factual specificity by connecting model output to external documents (Lewis et al., 2020). In legal and legal-adjacent settings, however, grounding is necessary but not sufficient. Research on legal AI tools shows that retrieval-augmented systems can still hallucinate or misstate legal information (Magesh et al., 2024, preprint). Grounding should therefore be paired with scope limits, output validation, and escalation policy.

6.3 Deterministic Resolution of Canonical Resources

Exact values should not be authored by the model. Approved URLs, required disclosures, product or service names, intake labels, and other canonical resources should live in a machine-readable source of truth and be inserted deterministically. This can be done through fixed UI rendering, placeholder resolution after generation, allowlist validation, or automatic correction.

The resolver should derive from the policy source of truth. Hard-coding the same values separately creates a second source of truth and reintroduces drift. Preserving policy-owner control while ensuring that exact values are not left to probabilistic reproduction is the central design objective of this layer.

The QA results reported in Section 5 illustrate the operational importance of this layer. URLs present verbatim in the always-present system prompt resolved correctly in every test; URLs the model was expected to construct from learned patterns were non-canonical in the majority of cases. This gap persisted across prompt revisions, which is consistent with the view that compositional generation of exact values is an inappropriate use of the probabilistic layer.

6.4 Output Validation and Bounded Correction

After generation and before display, the system should validate the response against mechanizable constraints. It should check required disclosures, banned phrasings, approved links, formatting rules, and prohibited response structures. If validation fails, the system can correct the output, append the missing required element, substitute a canonical value, or regenerate with the specific failure supplied as feedback. Retries should be bounded, and repeated failure should trigger a safe fallback.

An in-prompt self-check may improve results, but it should not be treated as a guarantee. It asks the same stochastic system to police itself during the same probabilistic process. A separate validator (rule-based where possible, model-based only where necessary) offers a more reliable control point.

The latency and engineering costs of the validation layer are real but manageable. A deterministic post-processing pass of the kind described here typically adds on the order of 100–300 ms to total response time, a range consistent with lightweight string-matching and allowlist-lookup operations, representing a small fraction of end-to-end latency in a conversational application where LLM generation itself dominates. Engineering overhead is likewise bounded: because the core validation logic is rule-based and isolated from the model-serving path, its surface area is limited and changes to the model or prompt do not require corresponding changes to the validator. These costs are substantially lower than the reputational, legal, or commercial cost of a compliance failure at scale in a public-facing regulated channel.

6.5 Human Oversight and Escalation

For a legal technology platform operating through a network of licensed professionals, the ultimate guardrail is the lawyer. The public assistant should remain limited to triage, general information, and routing. Anything case-specific should move to an authenticated channel or licensed review. This boundary also reinforces the Platform's value proposition, making clear

that the public assistant exists to support access to professional judgment, not to substitute for it.

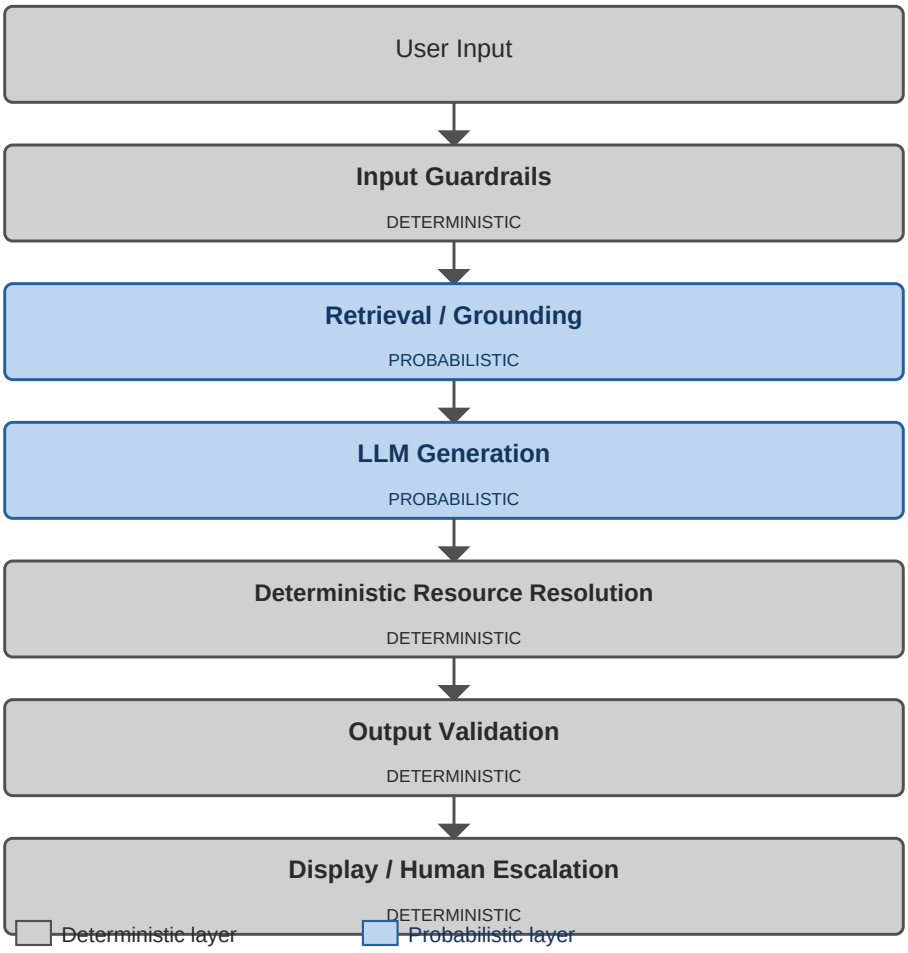

Figure 1. Layered guardrail pipeline. Grey nodes represent deterministic enforcement layers; blue nodes represent probabilistic (model-mediated) layers.

7. Regulated-Domain Considerations

7.1 Unauthorized Practice of Law and Individualized Advice

A public-facing assistant operated by a legal technology company must not cross from general information into individualized legal advice, eligibility determinations, or outcome predictions. This constraint carries particular weight for a non-licensee operator. A licensed law firm's chatbot misbehaving is an internal compliance matter; a non-licensee whose product provides legal advice without authorization faces a distinct and more serious exposure under provincial law society legislation, which prohibits non-licensees from providing legal services regardless of the technology medium through which those services are delivered. The constraint on the

chatbot is therefore structural: the system must remain on the information side of the advice boundary at all times, and no amount of prompt engineering substitutes for building that boundary into the architecture.

This constraint is judgment-bound because it depends on the user's question, the context of the exchange, and the practical effect of the answer. It cannot be solved entirely through deterministic rules. The system should adopt conservative defaults: refuse case assessment in the public channel, avoid predictions, avoid tailored procedural instructions, and escalate when a visitor asks for advice. This matters because legal hallucination research shows that LLMs can produce confident but incorrect legal content, including in ways that users may find persuasive (Dahl et al., 2024).

7.2 Accuracy, Hallucination, and Overconfidence

In immigration contexts, inaccurate information can create real consequences for vulnerable users. The system should avoid fabricated procedural steps, invented eligibility rules, and overconfident statements about a user's prospects. Grounding, refusal-to-speculate defaults, and limits on procedural detail are therefore core controls rather than optional polish. Although calibration research suggests that models may contain some internal signal about uncertainty, that signal is not dependable enough to replace external constraints in high-stakes settings (Kadavath et al., 2022, preprint).

A related risk is what the literature terms overconfident generation: fluent, assertive output that users receive as authoritative even when it is imprecise or wrong (Weidinger et al., 2021, preprint). In immigration settings specifically, this risk extends to probabilistic-sounding claims that imply predictive capacity the system does not possess, such as statements about the likelihood of application approval. The QA results reported in Section 5 included several instances of approval-likelihood drift ('increases your chances of a successful application') that crossed from general encouragement into implied outcome prediction. Guardrail-layer banned-phrase checks are the appropriate mechanism for controlling this failure mode, and they illustrate a broader principle: the distinction between general information and implied advice can turn on a single phrase.

7.3 Privacy and Data Handling

A pre-authentication channel should not solicit or retain sensitive personal information unless there is a clear lawful and operational basis for doing so. Under Canada's Personal Information Protection and Electronic Documents Act (PIPEDA), organizations may collect personal information only for purposes that a reasonable person would consider appropriate in the circumstances, and collection must be limited to what is necessary for those purposes (PIPEDA, Schedule 1, Principle 4.4). For a technology platform handling immigration-adjacent inquiries from potentially vulnerable users in a public channel, this obligation argues for conservative input guardrails that detect and deflect unnecessary personal data before it reaches the model. System design should also reflect broader privacy-by-design expectations and risk-management frameworks such as the NIST AI Risk Management Framework (NIST, 2023).

7.4 Adversarial Robustness

Public endpoints invite manipulation. Prompt injection and jailbreak attempts can subvert prompt-level controls, particularly where the model combines user instructions, system instructions, and retrieved content (Greshake et al., 2023; OWASP Foundation, 2025). Red-teaming, including automated adversarial testing, should therefore be part of the lifecycle (Perez et al., 2022). Deterministic output validation provides a backstop that prompt hardening alone cannot provide.

8. Evaluation Methodology for Stochastic Systems

Single-pass acceptance testing is inadequate for probabilistic systems because it samples the distribution once and may confuse a favorable draw with reliability. Evaluation should be designed to measure variance, identify residual failure modes, and determine whether those failures occur in risk-bearing dimensions.

8.1 Multi-Sample Evaluation

Each scenario should be run multiple times. Compliance should be reported as a rate rather than as a binary pass/fail result. For risk-bearing rules, the worst observed outcome should carry more weight than the representative response, because a single unsafe output can be operationally significant.

8.2 Paraphrase Robustness

Each persona and intent should be tested across several surface phrasings. Phrasing-sensitive leakage is especially important in public chat, where users arrive with different levels of knowledge, urgency, anxiety, and linguistic precision.

8.3 Dimension-Level Scoring

Evaluation should separate recognition, disclosure, advice avoidance, call-to-action correctness, factuality, privacy handling, adversarial robustness, and tone. Composite scores can hide critical failures. A response that is warm and helpful but contains the wrong link or crosses into advice should not pass because the average score looks acceptable.

8.4 Adversarial and Red-Team Suites

Dedicated probes should test injection, advice elicitation, out-of-scope requests, personal-information disclosure, and attempts to override system instructions. These tests should be repeated after prompt changes, model changes, retrieval changes, and policy updates.

8.5 Production Telemetry

Production systems should instrument how often guardrails fire, what they correct, and where residual leakage appears. This telemetry turns drift from an abstract concern into an observable operating signal. It also gives governance stakeholders evidence for whether the system is improving or simply appearing to improve during small-sample review.

9. Limitations and Future Work

Deterministic guardrails solve only the mechanizable part of the problem. They cannot guarantee flawless persona recognition, perfect detection of advice-seeking behavior, or ideal tone. Those behaviors remain probabilistic and bounded by model capability. Classifier models and second-pass review can reduce error, but they introduce their own failure modes and should be evaluated as components rather than assumed to be neutral safeguards.

The findings reported here carry several additional limitations:

- **Single-deployment generalizability.** The observations derive from one deployment context (a specific assistant design, system prompt configuration, and user-facing interface). Failure rates and failure modes may differ substantially across regulatory settings, organizational contexts, platform configurations, or user populations. The architecture proposed here is general, but the specific failure distributions should not be treated as benchmarks for other deployments without independent replication.

- **Model-specificity.** The QA results are specific to the commercial instruction-tuned model used in this deployment. Behavioral patterns such as the literal-string effect and approval-likelihood drift may present differently under other base models, fine-tuning regimes, or sampling parameters. The architecture is intended to be model-agnostic, but the specific failure rates reported in Section 5 should not be generalized beyond this deployment without independent replication on comparable systems.

- **Lack of user-outcome data.** The evaluation rubric assessed compliance as scored by an automated reviewer against predefined criteria. It did not measure actual user decisions, conversion behavior, or downstream legal exposure. Whether rubric-level compliance performance correlates with user-outcome quality remains an open empirical question and a priority for future study.

Output validation also adds latency and engineering surface. Over-aggressive correction can damage naturalness, which is one of the reasons to use an LLM in the first place. The architecture therefore requires tuning. The objective is to protect the specific elements where variation creates material risk, not to render every sentence deterministic.

Future work should examine constrained decoding for hard-format outputs, classifier-guided generation, stronger retrieval-to-citation verification, and more systematic measurement of residual leak rates under real-world traffic. A further research direction is governance design: how non-engineering policy owners can maintain canonical resources without creating operational drift between policy, prompt, and code.

10. Conclusion

The unpredictability of LLM output is a property of the technology that must be managed; it cannot be engineered away. Fully scripting a public assistant sacrifices the adaptability that makes LLMs useful. Trusting unconstrained generation exposes a regulated organization to avoidable legal, reputational, and commercial risk.

The appropriate architecture is a deliberate division of labor. The model should perform the tasks that genuinely require language judgment: recognizing intent, adapting tone, and producing natural conversational responses. Mechanically checkable facts and rules should be placed beyond the reach of sampling through deterministic resolution, validation, and fallback logic.

For a public-facing assistant operated by a legal technology platform, this hybrid architecture is what makes the system operationally credible. It is predictable where predictability matters, flexible where flexibility creates value, and explicit about the boundary between automated triage and the licensed professional judgment the platform exists to connect users with.

References

Bai, Y., Kadavath, S., Kundu, S., Askell, A., et al. (2022). *Constitutional AI: Harmlessness from AI feedback* (preprint). arXiv:2212.08073. https://doi.org/10.48550/arXiv.2212.08073

Bender, E. M., Gebru, T., McMillan-Major, A., & Shmitchell, S. (2021). On the dangers of stochastic parrots: Can language models be too big? *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, 610–623. https://doi.org/10.1145/3442188.3445922

Brown, T. B., Mann, B., Ryder, N., Subbiah, M., et al. (2020). Language models are few-shot learners. *Advances in Neural Information Processing Systems*, *33*, 1877–1901. https://doi.org/10.48550/arXiv.2005.14165

Dahl, M., Magesh, V., Suzgun, M., & Ho, D. E. (2024). Large legal fictions: Profiling legal hallucinations in large language models. *Journal of Legal Analysis*, *16*(1), 64–93. https://doi.org/10.1093/jla/laae003

Greshake, K., Abdelnabi, S., Mishra, S., Endres, C., Holz, T., & Fritz, M. (2023). Not what you've signed up for: Compromising real-world LLM-integrated applications with indirect prompt injection. *Proceedings of the 16th ACM Workshop on Artificial Intelligence and Security*, 79–90. https://doi.org/10.1145/3605764.3623985

Holtzman, A., Buys, J., Du, L., Forbes, M., & Choi, Y. (2020). The curious case of neural text degeneration. *International Conference on Learning Representations*. https://doi.org/10.48550/arXiv.1904.09751

Inan, H., Upasani, K., Chi, J., Rungta, R., et al. (2023). *Llama Guard: LLM-based input-output safeguard for human-AI conversations* (preprint). arXiv:2312.06674. https://doi.org/10.48550/arXiv.2312.06674

Ji, Z., Lee, N., Frieske, R., Yu, T., et al. (2023). Survey of hallucination in natural language generation. *ACM Computing Surveys*, *55*(12), Article 248. https://doi.org/10.1145/3571730

Kadavath, S., Conerly, T., Askell, A., Henighan, T., et al. (2022). *Language models (mostly) know what they know* (preprint). arXiv:2207.05221. https://doi.org/10.48550/arXiv.2207.05221

Lewis, P., Perez, E., Piktus, A., Petroni, F., et al. (2020). Retrieval-augmented generation for knowledge-intensive NLP tasks. *Advances in Neural Information Processing Systems*, *33*, 9459–9474. https://doi.org/10.48550/arXiv.2005.11401

Liu, N. F., Lin, K., Hewitt, J., Paranjape, A., Bevilacqua, M., Petroni, F., & Liang, P. (2024). Lost in the middle: How language models use long contexts. *Transactions of the Association for Computational Linguistics*, *12*, 157–173. https://doi.org/10.1162/tacl_a_00638

Magesh, V., Surani, F., Dahl, M., Suzgun, M., Manning, C. D., & Ho, D. E. (2024). *Hallucination-free? Assessing the reliability of leading AI legal research tools* (preprint). arXiv:2405.20362. https://doi.org/10.48550/arXiv.2405.20362

National Institute of Standards and Technology. (2023). *Artificial Intelligence Risk Management Framework (AI RMF 1.0)*. NIST AI 100-1.

OWASP Foundation. (2025). *OWASP Top 10 for Large Language Model Applications*. Open Worldwide Application Security Project.

Ouyang, L., Wu, J., Jiang, X., Almeida, D., et al. (2022). Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, *35*, 27730–27744. https://doi.org/10.48550/arXiv.2203.02155

Perez, E., Huang, S., Song, F., Cai, T., et al. (2022). Red teaming language models with language models. *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, 3419–3448. https://doi.org/10.48550/arXiv.2202.03286

Personal Information Protection and Electronic Documents Act, S.C. 2000, c. 5 (PIPEDA). https://laws-lois.justice.gc.ca/eng/acts/P-8.6/

Rebedea, T., Dinu, R., Sreedhar, M. N., Parisien, C., & Cohen, J. (2023). NeMo Guardrails: A toolkit for controllable and safe LLM applications with programmable rails. *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, 431–445. https://doi.org/10.18653/v1/2023.emnlp-demo.40

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., et al. (2017). Attention is all you need. *Advances in Neural Information Processing Systems*, *30*. https://doi.org/10.48550/arXiv.1706.03762

Weidinger, L., Mellor, J., Rauh, M., Griffin, C., et al. (2021). *Ethical and social risks of harm from language models* (preprint). arXiv:2112.04359. https://doi.org/10.48550/arXiv.2112.04359