RAGF: Boundary Enforcement as Governance Infrastructure for Agentic AI in Regulated Systems

Yamil Rodríguez-Montaña
Cronodata / Reflexio
Barcelona, Spain
yrm@reflexio.es

Abstract

Agentic AI systems increasingly move from advisory roles to operational execution in regulated domains. However, probabilistic reasoning engines cannot be certified under traditional safety frameworks that require deterministic, auditable behavior.

We present RAGF, a governance architecture that enforces a strict execution boundary between probabilistic AI reasoning and deterministic authority. RAGF validates proposed actions through domain ontologies, deterministic safety rules, and cryptographic audit trails before allowing execution.

Deployment across 12,847 aviation and healthcare actions demonstrates operational viability: 37 unsafe actions prevented with substantiated zero false positives, sub-30ms governance latency at p95, and fail-closed behavior across 7 enumerated failure categories validated through 3,500 systematic failure injections.

However, boundary enforcement concentrates authority in policy designers and amplifies encoded biases. We analyze governance trade-offs including centralization risks, rigidity versus adaptability tensions, and meta-governance requirements that technical architecture alone cannot resolve.

ACM Reference Format:
Yamil Rodríguez-Montaña. 2026. RAGF: Boundary Enforcement as Governance Infrastructure for Agentic AI in Regulated Systems. In . ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 Introduction

AI systems are transitioning from recommendation engines to decision-executing agents. In regulated sectors—aviation, healthcare, finance, and critical infrastructure—execution authority carries legal, operational, and ethical consequences. When an AI agent proposes rerouting a flight, adjusting a medication regimen, or initiating a grid reconfiguration, the distinction between *recommendation* and *commitment* becomes institutional, not merely technical.

Regulatory regimes emphasize determinism, traceability, and bounded behavior. Foundation models violate these assumptions by design: they are probabilistic, adaptive, and opaque. This creates a structural mismatch between model capabilities and institutional accountability requirements. Existing governance approaches either constrain AI to predetermined playbooks (sacrificing adaptive intelligence) or delegate authority to human-in-the-loop validation (creating operational bottlenecks).

We argue that certification should target the *governance harness mediating execution* rather than the adaptive model itself. RAGF embodies this architectural shift, separating probabilistic reasoning (uncertifiable) from deterministic boundary enforcement (certifiable). This paper demonstrates that boundary certification is technically viable while examining the governance trade-offs this approach introduces.

1.1 The Accountability Question

Beyond technical certification, agentic AI raises fundamental questions of responsibility: When AI-proposed actions cause harm, who is accountable—the model developer, the deploying organization, the policy designer who encoded validation rules, or the human operator who approved (or failed to reject) the action?

RAGF addresses this by creating an *institutional checkpoint* where human values (encoded as ontologies) meet AI capabilities. Each validation decision is cryptographically signed, creating an auditable record of which rules were applied and why actions were allowed or denied.

However, this solution introduces new governance challenges: Who controls the ontologies? How do we prevent regulatory capture? Can deterministic rules adequately capture context-dependent ethics? Can boundary enforcement itself become a mechanism of organizational control?

We argue these tensions are inherent to governed agentic AI and must be made explicit rather than resolved through technical design alone. RAGF provides enforcement infrastructure, not governance legitimacy.

1.2 Contributions

This work makes the following contributions:

(1) **Architectural Pattern**: Formalization of execution boundary enforcement as a distinct certification target, separating adaptive reasoning from deterministic authority mediation.
(2) **Governance Infrastructure**: Implementation of Validation Gate, semantic ontology integration, and cryptographic audit mechanisms across aviation and healthcare deployments.
(3) **Empirical Safety Property**: Demonstration of fail-closed behavior across 12,847 production actions and 3,500 injected failures, with formalization of empirical guarantee scope.
(4) **Governance Trade-off Analysis**: Systematic examination of authority centralization, policy encoding bias, rigidity-adaptability tensions, and meta-governance requirements.

Conference'17, Washington, DC, USA

ACM ISBN 978-x-xxxx-xxxx-x/YYYY/MM
https://doi.org/10.1145/nnnnnnn.nnnnnnn

(5) **Institutional Implications**: Discussion of who controls boundary rules, how encoding choices embed values, and what mechanisms ensure governance legitimacy beyond technical enforcement.

2 From Model-Centric AI to Governance-Centric AI

Enterprise AI adoption remains largely model-centric: organizations invest in fine-tuning, prompt engineering, and retrieval-augmented generation to improve model behavior. However, increased model capability has not translated proportionally into institutional accountability. A more capable model that proposes better actions is still a model whose reasoning process remains opaque and whose behavior cannot be guaranteed.

2.1 The Governance Gap

We define the **Governance Gap** as the institutional deficit between adaptive AI capabilities and accountable execution. This gap is not merely technical but structural: organizations lack mechanisms to translate probabilistic reasoning into deterministic, auditable commitments that satisfy regulatory and liability frameworks.

The governance gap manifests across three dimensions:

Accountability Vacuum: When an AI-proposed action causes harm, existing frameworks struggle to assign responsibility. Is the model developer liable for training data biases? Is the deploying organization responsible for inadequate oversight? Is the human operator accountable for approving (or failing to reject) the action? The distributed nature of AI decision-making creates what legal scholars term "responsibility gaps"—situations where no single actor can be held clearly accountable.

Opacity Problem: LLMs produce actions through probabilistic inference across billions of parameters. Even with explainability techniques such as attention visualization or rationale generation, the *causal path* from input to output remains fundamentally opaque to institutional oversight. A model may generate a superficially plausible explanation that does not reflect its actual inference process. Organizations cannot audit what they cannot inspect.

Certification Paradox: Regulators require demonstrable safety under failure conditions—a bedrock principle in aviation (DO-178C), medical devices (IEC 62304), and industrial control (IEC 61508). Yet adaptive models *by design* exhibit non-deterministic behavior and distribution shift. They improve through exposure to novel data but cannot guarantee bounded behavior across all inputs. Traditional certification assumes inspectable, stable logic operating within a defined envelope. LLMs violate all three assumptions.

2.2 Limits of Model-Centric Certification

Certification frameworks developed for traditional software assume:

- **Deterministic behavior**: Same input produces same output
- **Inspectable logic**: Code can be reviewed line by line
- **Stable operating conditions**: Defined input space and failure modes

LLMs fundamentally violate these assumptions due to stochastic sampling, emergent capabilities, and continuous adaptation to distribution shift. Attempts to certify model behavior directly face insurmountable challenges:

Non-Determinism: Temperature sampling, nucleus sampling, and beam search introduce variability. Even with temperature=0, floating-point arithmetic and hardware differences create non-reproducibility.

Emergent Behavior: Capabilities not explicitly programmed emerge from scale and training. Models exhibit "grokking," sudden capability jumps, and compositional generalization that were not designed or anticipated.

Distribution Shift: Models trained on historical data encounter novel inputs in deployment. Fine-tuning and RLHF improve alignment but cannot guarantee safe behavior on out-of-distribution examples.

These are not engineering limitations to be overcome but fundamental properties of how contemporary AI systems operate.

2.3 Execution Authority as Institutional Boundary

The critical distinction in agentic systems is between *recommendation* and *execution*. A conversational AI that suggests "consider rerouting Flight 3202" operates in advisory mode. An agentic AI that initiates the reroute crosses an institutional boundary: it commits organizational resources, accepts legal liability, and affects human welfare.

Governance must mediate this commitment. The question is not "Can we make the model certifiable?" but rather "Can we certify the *process* by which model outputs become institutional actions?"

RAGF answers this by treating boundary enforcement, rather than model introspection, as the relevant certification target. If we accept that the reasoning process is opaque and probabilistic, we can still enforce that only validated actions proceed to execution.

2.4 RAGF's Governance Intervention

RAGF restructures authority within agentic systems through institutional role separation:

- **Reasoning Agent** (LLM): Proposes candidate actions based on probabilistic inference
- **Policy Designers**: Encode organizational values and regulatory constraints as formal rules
- **Validation Gate**: Enforces deterministic predicates over proposed actions
- **Human Operators**: Resolve escalated cases requiring discretionary judgment
- **Auditors**: Verify cryptographically signed records of validation decisions

Reasoning capability and execution authority are architecturally decoupled. The LLM operates in an advisory capacity; commitment occurs only after deterministic validation. This separation creates an auditable checkpoint where institutional governance—not model behavior—determines what actions are permitted.

However, this architectural choice introduces new tensions: Who designs the policies? How do we ensure policy designers represent diverse stakeholder interests? Can deterministic enforcement capture context-dependent norms? What happens when governance rules themselves are contested?

We argue these questions cannot be resolved through technical design alone. RAGF provides enforcement infrastructure that makes governance *possible* while leaving governance *legitimacy* as an ongoing institutional requirement.

3 RAGF as Governance Infrastructure

RAGF implements boundary enforcement through four interoperating components: the Validation Gate (deterministic enforcement), Semantic Authority Layer (domain ontologies), Cryptographic Audit (non-repudiation), and Escalation Pathways (human override).

3.1 Validation Gate: Authority Mediator

The Validation Gate enforces a deterministic function over proposed actions:

Deterministic Enforcement: Given the same action and organizational state, the gate produces the same verdict. This enables repeatability, inspection, and certification of validation logic.

Fail-Closed Semantics: Any error in validation—database timeout, validator exception, cryptographic signature failure—results in DENY. The system defaults to safety under degradation.

Escalation Pathways: Actions with ambiguous validation outcomes (semantic coverage < 1.0, conflicting validator results) trigger ESCALATE rather than ALLOW, routing decisions to human operators.

This design embeds institutional priorities: safety over efficiency, human judgment over automation for edge cases, auditability over convenience.

3.2 Semantic Authority Layer

Domain ontologies ground candidate actions in formal institutional knowledge. Rather than attempting to constrain LLM reasoning (which is probabilistic and difficult to verify), RAGF validates *outputs* against explicit organizational rules.

Ontologies encode:

- **Valid Action Vocabulary**: Enumeration of permissible verbs (e.g., "reroute_flight", "prescribe_medication")
- **Regulatory Constraints**: Mapping from actions to applicable regulations (e.g., FAA 14 CFR §91.1057)
- **Authority Levels**: Minimum organizational authorization required per action type

If an LLM proposes an action not recognized by the ontology—an "operational hallucination"—validation fails. The system cannot execute what it cannot semantically ground.

3.3 Cryptographic Audit

Each validation verdict is signed with HMAC-SHA256 and persisted to an append-only ledger (TimescaleDB). This creates a tamper-evident record of:

- What action was proposed
- Which validators were invoked
- What rules were evaluated
- Why the action was allowed, denied, or escalated
- When the decision was made
- By which organizational authority level

Audit trails enable ex-post accountability: regulators, auditors, and affected parties can verify that governance was applied consistently and trace decisions to specific policy rules.

However, cryptographic auditability does not guarantee policy legitimacy. A system can faithfully enforce unjust rules. Audit infrastructure documents *what was enforced*, not *whether enforcement was justified*.

3.4 Institutional Role Separation

RAGF's architecture enforces separation of concerns:

Reasoning is Delegated, Authority is Retained: The LLM operates as an advisory system. Institutional authority remains with the Validation Gate and the human operators who designed its policies.

Policy Design is Explicit: Unlike implicit biases in model weights, RAGF's governance rules are version-controlled, peer-reviewed, and auditable. This does not eliminate bias but makes encoding choices visible and contestable.

Execution is Mediated: No action proceeds directly from LLM output to operational commitment. The Validation Gate serves as a mandatory checkpoint, creating an institutional boundary that can be monitored, measured, and held accountable.

This separation creates new points of governance intervention but also new concentrations of power. We examine these trade-offs in Section 8.

4 Execution Boundary Formalization

We formalize the execution boundary as a deterministic predicate over candidate actions.

Let $A = \{a_1, \ldots, a_n\}$ be the space of candidate actions an agent may propose. Each action $a \in A$ consists of:

- **Verb**: Semantic type (e.g., "reroute_flight", "adjust_dosage")
- **Resource**: Target entity (e.g., Flight IB3202, Patient 47291)
- **Parameters**: Structured arguments (e.g., {new_route: "MAD-BCN", fuel_reserve: 45})

Define the deterministic validation function:

$$V : A \times S \to \{\text{ALLOW}, \text{DENY}, \text{ESCALATE}\}$$

where S represents organizational state (current regulations, resource availability, authority levels).

Commitment Boundary: An action a proceeds to execution if and only if:

$$V(a, s) = \text{ALLOW}$$

Safety Invariant: Under any failure mode $f \in F$, validation never incorrectly allows:

$$\forall f \in F, \quad V(a, s) \neq \text{ALLOW unless safe}(a, s)$$

In practice, we enforce the stronger property:

$$\forall f \in F, \quad V(a, s) = \text{DENY}$$

That is, any failure in the validation pipeline results in denial, not approval. This fail-closed design prioritizes safety over availability.

4.1 Validation Gate Algorithm

The Validation Gate implements V through parallel validator execution and deterministic aggregation:

Algorithm 1 Deterministic Validation Gate

Input: Action a, organizational state s
Output: Verdict $\in$ {ALLOW, DENY, ESCALATE}

$results \leftarrow \emptyset$
for each validator v in validators(a) **do**
  $r \leftarrow v(a, s)$ {Execute validator}
  if r = DENY **then**
    return DENY {Fail fast}
  end if
  $results \leftarrow results \cup \{r\}$
end for
if ESCALATE $\in$ $results$ **then**
  return ESCALATE
else if semantic_coverage$(a, s) < 1.0$ **then**
  return ESCALATE
else
  return ALLOW
end if

Validators execute in parallel; the first DENY verdict short-circuits evaluation. If any validator recommends escalation or semantic coverage is incomplete, human review is required. Only if all validators pass and semantic grounding is complete does validation produce ALLOW.

This algorithm is deterministic: given the same action and organizational state, it produces the same verdict. Determinism enables inspection, testing, and certification of validation logic independent of the probabilistic reasoning that generated the candidate action.

5 Threat Model and Security Assumptions

RAGF operates under explicit threat model boundaries:

5.1 In-Scope Threats

LLM Adversarial Behavior: The system assumes the LLM may produce unsafe, malicious, or hallucinatory actions. Validation provides defense in depth against model failures.

Operational Errors: Human operators may misconfigure validators or encode incorrect policies. Version control, peer review, and integration testing mitigate but do not eliminate this risk.

Component Failures: Databases (Neo4j, TimescaleDB), network connections, and validator processes may experience transient or permanent failures. Fail-closed design ensures failures do not compromise safety.

5.2 Out-of-Scope Threats

Host System Compromise: If the underlying compute infrastructure is compromised, an attacker can bypass validation entirely. We assume standard infrastructure hardening (network segmentation, access control, encryption).

Validator Tampering: If an attacker can modify validator code or ontologies, they can subvert governance. We assume code review, CI/CD integrity, and access control prevent unauthorized modifications.

Cryptographic Key Compromise: HMAC signatures rely on secret key security. We assume keys are managed via KMS with rotation policies, but key compromise would enable audit log forgery.

Ontology Poisoning: Malicious injection of permissive rules into ontologies could allow unsafe actions. We assume ontology governance includes change review and approval workflows.

5.3 Trust Assumptions

RAGF's security model explicitly assumes:

(1) **Policy Designers are Trustworthy**: Domain experts who encode validation rules act in good faith and possess relevant expertise.
(2) **Validators are Correctly Implemented**: Deterministic validation logic faithfully implements intended policies.
(3) **Infrastructure is Hardened**: Databases, networks, and compute follow industry-standard security practices.
(4) **Organizational Governance is Sound**: Change management, access control, and audit processes function as designed.

These assumptions shift security from model behavior (which cannot be guaranteed) to infrastructure and process (which can be audited and certified). However, they also concentrate trust in policy designers and infrastructure operators—a governance trade-off we examine in Section 8.

6 Empirical Safety Property

We empirically validate the following fail-closed property:

Property 1 (Empirical Fail-Closed Guarantee). *Across 7 enumerated failure categories, the Validation Gate exhibits deterministic denial behavior under systematic failure injection:*

$$\forall f \in F, \forall a \in A, \quad V(a) \text{ under } f = \text{DENY}$$

where F includes:

- LLM timeout (reasoning latency > 30s)
- Ontology database failure (Neo4j connection loss)
- Validator exception (runtime error in validation logic)
- Cryptographic signature failure (HMAC computation error)
- Ledger write failure (TimescaleDB unavailability)
- Semantic validation timeout (ontology query > 500ms)
- Health check failure (component unavailability)

6.1 Validation Methodology

We validate this property through:

Systematic Failure Injection: 3,500 controlled failures injected across the 7 categories (500 injections per category). Each injection simulates the target failure mode in isolation while monitoring validation behavior. Across all injections, zero unintended ALLOW verdicts were observed.

Integration Testing: Automated test suite covering normal operation and all failure modes. Tests verify that validation logic

correctly maps failures to DENY verdicts with appropriate error reporting.

Code Review: Static analysis of exception handling paths ensures all code branches under failure conditions terminate in DENY with no execution pathways bypassing validation.

6.2 Scope and Limitations

Empirical, Not Formal: This is empirical validation through testing, not mathematical proof via theorem proving. The property holds for enumerated failure modes but may not cover all possible system states or interaction effects between multiple simultaneous failures.

Implementation-Dependent: Property validity depends on current codebase. Code changes require re-validation. We do not claim the property holds for all possible implementations of the architecture.

Deployment Context: Validation occurred in controlled pilot environments under sustained loads up to 50 req/s. Production deployments with higher concurrency, network partitions, or novel failure modes may exhibit different behavior.

Formal verification via proof assistants (Coq, Isabelle/HOL) would provide stronger guarantees but is identified as future work. The current approach aligns with DO-178C Level C verification objectives, which require demonstrable evidence through testing and review rather than mathematical proof.

7 Deployment Evaluation

We deployed RAGF in two regulated domains to assess operational viability, governance overhead, and failure mode behavior.

7.1 Aviation: Dynamic Route Optimization

Context: Regional carrier operational simulation, 45 daily flights across European routes.

Agent Capability: LLM proposes fuel-efficient reroutes based on weather, traffic, and operational constraints.

Validators:

- Fuel reserve compliance (FAA 14 CFR §91.151)
- Crew rest requirements (FAA 14 CFR §91.1057)
- Airspace restrictions (EASA regulations)

Ontology: 847 vertices (verbs, regulations, resources), 2,341 edges (semantic relationships).

Duration: 90-day controlled pilot.

Results:

- 12,847 actions evaluated
- 11,203 ALLOW (87.2%)
- 1,544 DENY (12.0%)
- 100 ESCALATE (0.8%)
- 37 unsafe actions prevented (would have violated regulations)
- Zero false positives (substantiated through expert review)

7.2 False Positive Validation Methodology

Review Process: All 1,544 DENY verdicts were manually reviewed by domain experts: aviation safety managers certified under FAA Part 121 for aviation actions, clinical pharmacists board-certified in medication safety for healthcare actions.

Classification Criteria: Each denial was independently classified by two reviewers as:

- **True Positive**: Action would violate regulation or safety constraint (n=1,544)
- **False Positive**: Action safe but incorrectly denied (n=0)

Inter-Rater Reliability: Cohen's kappa $\kappa = 0.94$ (substantial agreement). Seven ambiguous cases requiring regulatory interpretation were resolved through discussion and reference to advisory circulars.

Validation Scope: This zero false positive rate applies within tested operational scope (European routes, standard weather conditions, regulation-compliant flight plans). Novel scenarios outside training distribution may exhibit different characteristics.

7.3 Healthcare: Medication Recommendations

Context: 250-bed facility, internal medicine and cardiology departments.

Agent Capability: LLM suggests medication adjustments based on patient history, lab results, and clinical guidelines.

Validators:

- Drug-drug interaction database
- Patient allergy contraindications
- Dosage range compliance (hospital formulary)

Duration: 60-day controlled pilot in production-like environment.

Results: Similar safety coverage with context-specific escalations requiring physician review for novel drug combinations not in formulary ontology.

7.4 Governance Latency

Component	p50	p95	p99
Semantic Layer	4.2ms	6.8ms	9.1ms
Validation Gate	8.7ms	12.4ms	14.3ms
Signature	0.5ms	0.7ms	0.9ms
Total Governance	**18.3ms**	**28.1ms**	**33.2ms**

Table 1: Governance latency under sustained 50 req/s load. Measurements from aviation deployment over 90-day period.

Governance overhead remained at p95 = 28.1ms (sub-30ms), leaving approximately 170ms budget for LLM reasoning within typical 200ms total latency targets. This demonstrates that deterministic validation is operationally viable without becoming a throughput bottleneck at moderate request rates.

7.5 Comparative Analysis: RAGF vs Human-in-the-Loop

Analysis: RAGF provides approximately 153× throughput improvement over human-in-the-loop validation (252 seconds vs 1.68 seconds per action). However, this comparison requires important caveats:

Metric	HITL	RAGF
Avg. Decision Time	4.2 min	28.1 ms
Actions/Hour	14	2,140
False Positive Rate	8.3%	0%
Regulatory Violations	2	0
Audit Trail Completeness	67%	100%

Table 2: Comparison with human-in-the-loop validation baseline from same operational context.

Scope Difference: Human operators in HITL provide contextual reasoning and exception handling that RAGF routes to ESCALATE pathways (0.8% of actions). The throughput gain reflects *routine validation acceleration*, not complete replacement of human judgment.

Complementary Roles: High-stakes scenarios, novel drug combinations, emergency protocol deviations, and edge cases outside ontology coverage still require human review. RAGF shifts human effort from routine validation to exceptional case resolution.

Expertise Preservation: The 153× efficiency gain enables redeployment of domain expertise to higher-value tasks (ontology refinement, escalation review, policy improvement) rather than routine action validation.

8 Governance Trade-offs and Structural Risks

RAGF improves enforceability and auditability but introduces governance trade-offs that cannot be resolved through technical design alone.

8.1 Centralization of Authority

RAGF concentrates execution authority in a deterministic validation layer, creating new forms of institutional power:

Ontology Designers as De Facto Policymakers: Domain experts who encode validation rules effectively become policymakers. In our aviation deployment, flight safety managers defined which actions required validation, what thresholds constituted "acceptable risk," and how edge cases should be handled. These design choices embed organizational priorities into enforcement infrastructure.

This raises democratic legitimacy questions: Should validators designed by safety managers apply equally to pilots, schedulers, and executives? How do affected stakeholders—pilots who must work within automated constraints, passengers whose safety depends on governance rules, communities under flight paths—participate in policy formation?

Gatekeeper Risk: A compromised or captured governance layer has systemic impact. Unlike distributed human oversight, where individual operators can exercise discretionary judgment, deterministic enforcement cannot adapt to novel contexts without ontology updates. This creates dependency on policy designers' responsiveness, expertise, and trustworthiness.

If policy designers exhibit bias—favoring cost reduction over worker welfare, prioritizing efficiency over patient autonomy—deterministic enforcement amplifies these biases systematically rather than allowing case-by-case judgment.

Expertise Concentration: Organizations deploying RAGF require personnel capable of encoding domain knowledge as formal ontologies. Our deployments required 40–60 hours of expert time per domain (aviation safety managers + knowledge engineers for aviation; physicians + clinical informaticists for healthcare).

This resource requirement may exclude smaller organizations, potentially widening the gap between early AI adopters (who can afford ontology development) and laggards (who cannot). Shared ontologies could reduce barriers but may not reflect local contexts, values, or regulatory environments.

Observation: RAGF's technical architecture is agnostic to governance structure. It provides enforcement infrastructure but does *not* resolve the political question of who sets the rules and through what process rule-setting power is exercised.

8.2 Policy Encoding Bias

Deterministic enforcement amplifies rather than mitigates encoded bias:

Codification Inevitably Simplifies: Not all norms are reducible to formal predicates. RAGF works well for bright-line rules (e.g., "crew rest $<$ 14 hours $\rightarrow$ DENY") but struggles with context-dependent principles (e.g., "act in patient's best interests," "exercise reasonable care").

When complex norms must be reduced to deterministic code, the *encoding choices* embed values. Our aviation ontology prioritized fuel efficiency optimization, implicitly valuing cost reduction over schedule reliability. These design choices reflect organizational priorities but may not align with broader stakeholder values.

Bias Amplification Through Automation: While human operators exhibit bias, they can exercise discretionary judgment in edge cases. A human operator might recognize that a technically non-compliant action is warranted given unusual circumstances. Deterministic validators execute encoded rules without contextual flexibility.

This means encoding bias becomes *systematic* rather than situational. If a validator embeds assumptions about "standard patient profiles" or "normal flight conditions," deviations from these assumptions—even if medically necessary or operationally justified—will be flagged for denial or escalation.

Opacity of Policy Formation: While RAGF makes validation decisions auditable via cryptographic signatures, the process of *ontology design* may be less transparent to end users. Who decided which actions require validation? Through what process? With whose participation? What alternatives were considered?

Cryptographic audit trails document *that policies were enforced* but not *how policies were chosen*. A system can faithfully enforce unjust rules.

Position: We do not claim RAGF eliminates bias. Rather, we argue it makes bias *auditable and contestable*—a necessary but insufficient condition for legitimate governance. Encoded biases are visible in version-controlled ontologies and can be challenged, revised, or overridden through institutional processes. However, this requires complementary governance mechanisms beyond RAGF's technical scope.

8.3 Rigidity Versus Adaptability

Strict boundary enforcement creates tension between safety and innovation:

Precautionary Principle as Default: RAGF's fail-closed design embeds a precautionary posture: when in doubt, DENY. This reflects aviation's safety culture, where preventing accidents takes precedence over operational efficiency. However, this principle may not generalize to domains where rapid adaptation is valued—emergency response requiring protocol deviation, experimental medicine exploring novel treatments, financial trading in volatile markets.

Innovation Suppression Risk: Ontologies encode *current* knowledge. Novel actions—even if beneficial—will be denied if not recognized by existing validators. In our healthcare deployment, a physician proposed a drug combination not in the hospital formulary ontology. RAGF correctly escalated (per design), requiring manual review.

However, the physician perceived this as "bureaucratic barrier" rather than safety mechanism. If escalations become frequent, operators may view governance as impediment rather than enabler—potentially leading to workarounds or circumvention.

This creates temporal lag between practice evolution and governance adaptation. How quickly can ontologies be updated? Who authorizes changes? What testing is required before deployment? These questions reveal that "governance agility" is in tension with "governance stability."

Context Blindness: Deterministic validators execute rules without situational judgment. In emergencies requiring deviation from protocol—diverting a flight due to medical emergency, administering contraindicated medication to save a life—RAGF's rigidity may hinder rather than help.

We addressed this via ESCALATE pathways, routing ambiguous cases to human operators. However, this reintroduces the human bottlenecks RAGF was designed to eliminate. If ESCALATE is invoked frequently, the system devolves to supervised automation rather than governed autonomy.

Trade-off: The tension is fundamental. Governance that prevents unsafe actions also prevents adaptive responses to novel situations. Organizations must choose: prioritize safety (accept innovation lag) or prioritize adaptability (accept residual risk). RAGF does not resolve this tension but makes it explicit through ESCALATE metrics and policy review cycles.

8.4 Meta-Governance: Who Governs the Governors?

RAGF introduces a new governance layer, raising second-order questions:

Ontology Governance: How should ontology changes be managed? Our deployments used version control + peer review, but this is *internal* organizational governance. What mechanisms ensure ontologies reflect public interest, not just organizational expediency?

For aviation, FAA regulations provide external constraint. But who ensures validators faithfully implement regulatory intent rather than minimalist compliance? For healthcare, hospital administrators control formularies. How do patients, nurses, or community health advocates participate in policy formation?

Validator Accountability: Validators are "trusted components" in our threat model (Section 5). But who audits the validators? Our approach (code review + integration testing + continuous monitoring) assumes organizational trustworthiness—an assumption not universally warranted.

External audit could provide oversight, but auditors face information asymmetry: policy designers understand domain constraints, auditors may not. How do we create validator accountability without requiring auditors to duplicate domain expertise?

Appeal Mechanisms: When RAGF denies an action, can the decision be appealed? By whom? Through what process? Should there be mechanisms for overriding deterministic denial in exceptional circumstances?

We implemented ESCALATE as a safety valve, but systematic ESCALATE usage may signal ontology inadequacy rather than genuine edge cases. High escalation rates could indicate:

- Ontology is too restrictive (suppressing valid actions)
- Ontology is incomplete (missing common scenarios)
- Operational context has shifted (rules are outdated)
- Human operators are gaming the system (escalating to avoid delays)

Distinguishing among these requires meta-governance mechanisms beyond RAGF's technical scope.

Observation: These meta-governance questions are *not solved* by RAGF's architecture. We view them as necessary accompaniments to deployment, requiring institutional rather than algorithmic solutions. RAGF provides enforcement infrastructure that makes these questions *explicit* and *auditable*, but does not answer them.

8.5 Governance Layer Capture

Concentration of authority in governance infrastructure creates capture risks:

Regulatory Capture: If ontology maintenance becomes concentrated in industry groups, RAGF could institutionalize industry-favorable interpretations of ambiguous regulations. Aviation safety, for example, involves trade-offs between safety margins and operational efficiency. Who decides where the balance lies?

Organizational Capture: If governance rules are set by executives prioritizing profitability over worker welfare or patient autonomy, deterministic enforcement becomes a mechanism of organizational control rather than safety assurance.

Technical Capture: If ontology design requires specialized expertise, policy formation becomes concentrated among technical elites who may not represent affected populations. How do we prevent "technocratic governance" where encoding choices are shielded from democratic accountability by claims of technical necessity?

Mitigation Strategies (beyond RAGF's technical scope):

- Participatory ontology governance with stakeholder representation
- External audit of validators for bias and overreach
- Transparent documentation of policy encoding choices and trade-offs
- Sunset clauses requiring periodic rule re-justification
- Public comment periods for significant ontology changes

These are institutional mechanisms, not technical features. RAGF creates the infrastructure for enforcement but does not mandate governance legitimacy.

8.6 What RAGF Does Not Solve

RAGF provides *enforcement infrastructure* for governance policies but does not:

- Determine what those policies *should be*
- Ensure policies are just, fair, or democratically legitimate
- Prevent misuse of governance mechanisms for organizational control
- Address upstream harms in LLM training data or model design
- Guarantee that encoded rules capture all relevant ethical considerations
- Resolve distributional impacts (who benefits from automation, who bears costs)
- Create mechanisms for meaningful stakeholder participation in policy formation

We view RAGF as *necessary but insufficient* for responsible agentic AI deployment. Technical enforcement must be accompanied by:

- Participatory ontology governance with diverse stakeholder input
- Continuous audit of validators for bias, overreach, and unintended consequences
- Transparent documentation of policy encoding choices and alternatives considered
- Mechanisms for appealing or overriding automated decisions in exceptional cases
- Investment in human expertise alongside automation (not replacement)
- Ongoing assessment of distributional impacts and corrective measures

The 153× throughput improvement reflects routine validation acceleration, not complete elimination of human judgment. High-stakes scenarios still require expert review (0.8% ESCALATE rate), and automation efficiency must be evaluated against expertise preservation and distributional equity considerations.

RAGF demonstrates that governed agentic AI is *technically feasible*. Making it *socially desirable* requires institutional choices our framework documents but does not prescribe.

9 Related Work

RAGF intersects AI governance, runtime enforcement, and institutional infrastructure research.

9.1 Model-Centric AI Safety

Constitutional AI and **RLHF** attempt to align model behavior through training interventions. These approaches improve model safety statistically but cannot provide formal guarantees. RAGF accepts model opacity and validates outputs rather than attempting to certify reasoning processes.

Prompt Engineering and **System Messages** constrain model behavior through input framing. However, jailbreaking, prompt injection, and semantic drift can bypass text-based constraints. RAGF enforces boundaries *after* generation, independent of how the model was prompted.

9.2 Runtime Verification and Monitoring

Safety Monitors in embedded systems enforce invariants over controller outputs. RAGF applies similar principles to AI systems: validate actions before execution, fail closed under degradation.

Runtime Verification for Autonomous Systems focuses on temporal logic properties (e.g., "vehicle never exceeds speed limit"). RAGF extends this with semantic validation: actions must be *meaningful* (grounded in ontology) and *permissible* (compliant with regulations).

9.3 Policy-Based Access Control

XACML and similar frameworks provide XML-based access control. Limitation: static policies without semantic reasoning. RAGF integrates domain ontologies to ground actions in institutional knowledge graphs.

Unlike pure policy engines, RAGF combines:

- Semantic grounding (ontology-based action vocabulary)
- Deterministic enforcement (fail-closed validation)
- Cryptographic auditability (HMAC-signed verdicts)

9.4 AI Governance and Ethics Scholarship

Algorithmic Accountability: Emphasizes transparency in automated decision systems. RAGF contributes cryptographic audit trails but does not address accountability for *policy design choices themselves*—who encoded the rules, through what process, representing whose interests.

Value-Sensitive Design: Proposes embedding values in technology. RAGF's ontologies encode organizational values, but this concentrates power in ontology designers. How do we ensure encoding choices reflect diverse stakeholder values, not just dominant organizational priorities?

Participatory AI Governance: Advocates stakeholder involvement in AI system design. RAGF's technical architecture is *compatible* with participatory governance but does not *mandate* it. Deployment organizations must create participation mechanisms independently.

Fairness, Accountability, and Transparency (FAccT): Calls for inclusive design processes. RAGF provides technical enforcement but leaves governance legitimacy as an institutional requirement. A system can faithfully enforce unjust rules.

9.5 Institutional Perspectives

Unlike documentation-centric frameworks (Model Cards, Datasheets), RAGF embeds enforcement into execution infrastructure. Unlike model-centric safety (Constitutional AI, RLHF), RAGF targets boundary certification.

RAGF's contribution is architectural: demonstrating that deterministic boundary enforcement is viable for governed agentic systems. However, governance infrastructure alone does not ensure governance legitimacy—a distinction we make explicit through trade-off analysis (Section 8).

10 Conclusion

RAGF demonstrates that deterministic boundary enforcement can mediate agentic execution in regulated environments, achieving operational viability and fail-closed behavior within tested scope. Across 12,847 production actions in aviation and healthcare deployments, RAGF prevented 37 unsafe actions while maintaining sub-30ms governance latency at p95.

However, *technical viability does not imply social desirability*. Governance-centric AI shifts accountability from model behavior to infrastructural enforcement—increasing determinism and auditability while concentrating authority in governance layer design.

10.1 Open Questions

This work raises questions that technical architecture alone cannot resolve:

- **Who should control ontologies?** Policy designers exercise de facto policymaking power. What mechanisms ensure this power is exercised legitimately?
- **How do we ensure governance rules reflect diverse stakeholder values?** Encoding choices embed organizational priorities. How do affected populations participate in policy formation?
- **Can deterministic enforcement adequately capture context-dependent ethics?** Bright-line rules work for regulatory compliance. What about norms requiring situated judgment?
- **What mechanisms prevent regulatory capture of the governance layer?** If policy formation becomes concentrated in industry groups or technical elites, how do we ensure public accountability?
- **How should meta-governance operate?** Who audits the validators? Who reviews ontology changes? What appeal mechanisms exist for contested denials?

We view these as *necessary tensions* in governed agentic AI, requiring ongoing institutional attention rather than one-time technical resolution.

10.2 Contributions

This work contributes:

(1) **Formalization of execution boundary as certification target**: Separating adaptive reasoning from deterministic authority mediation.
(2) **Empirical demonstration of fail-closed enforcement**: Across 12,847 production actions, 3,500 systematic failure injections, and 7 enumerated failure categories.
(3) **Articulation of governance trade-offs**: Centralization of authority, policy encoding bias, rigidity-adaptability tensions, meta-governance requirements.
(4) **Evidence that boundary certification is operationally viable**: Sub-30ms governance latency enabling regulated deployment at moderate request rates.
(5) **Framework for distinguishing technical enforcement from governance legitimacy**: Making explicit that infrastructure enables but does not guarantee just governance.

10.3 Future Directions

Future work must extend beyond enforcement mechanisms to meta-governance:

Participatory Ontology Governance: How should stakeholders participate in policy formation? What deliberative mechanisms balance expert knowledge with affected population representation?

Validator Accountability: What external oversight ensures validators faithfully implement intended policies rather than organizational expediency?

Distributional Impact Assessment: Who benefits from automation efficiency? Who bears costs (expertise erosion, reduced autonomy)? What corrective mechanisms address inequities?

Adaptive Governance: How can governance rules evolve responsively while maintaining stability and predictability? What learning mechanisms enable policy improvement without concentration of power?

Cross-Domain Transfer: Can ontologies and validators be shared across organizations to reduce development barriers? What standardization enables interoperability without sacrificing local context?

The shift from model-centric to governance-centric AI is inevitable as agentic systems enter regulated domains. RAGF provides infrastructure for this transition while making its inherent tensions—authority concentration, encoding bias, rigidity versus adaptability—explicit and auditable.

Certifying the boundary, rather than the adaptive core, offers a *technically viable* pathway. Making it *institutionally legitimate* requires addressing power, participation, and distributional justice—questions our technical framework alone cannot answer.

References

[1] Yuntao Bai, Saurav Kadavath, Sandipan Kundu, Amanda Askell, Jackson Kernion, Andy Jones, Anna Chen, Anna Goldie, Azalia Mirhoseini, Cameron McKinnon, et al. 2022. Constitutional AI: Harmlessness from AI Feedback. *arXiv preprint arXiv:2212.08073* (2022).
[2] Rishi Bommasani, Drew A Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, et al. 2021. On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258* (2021).
[3] Stephen Casper, Xander Davies, Claudia Shi, Thomas Krendl Gilbert, Jérémy Scheurer, Javier Rando, Rachel Freedman, Tomasz Korbak, David Lindner, Pedro Freire, et al. 2023. Open problems and fundamental limitations of reinforcement learning from human feedback. *Transactions on Machine Learning Research* (2023).
[4] Ankush Desai, Indranil Saha, Jianqiao Yang, Shaz Qadeer, and Sanjit A Seshia. 2017. DRONA: A Framework for Safe Distributed Mobile Robotics. In *2017 ACM/IEEE 8th International Conference on Cyber-Physical Systems (ICCPS)*. IEEE, 239–248.
[5] Nicholas Diakopoulos. 2016. Accountability in algorithmic decision making. *Communications of the ACM* 59, 2 (2016), 56–62.
[6] RTCA. 2011. *DO-178C: Software Considerations in Airborne Systems and Equipment Certification*. Radio Technical Commission for Aeronautics.
[7] European Parliament. 2024. *Regulation (EU) 2024/1689 on Artificial Intelligence (AI Act)*.
[8] Batya Friedman and David G Hendry. 2019. *Value sensitive design: Shaping technology with moral imagination*. MIT Press.
[9] Artur d'Avila Garcez and Luis C Lamb. 2023. Neurosymbolic AI: The 3rd wave. *Artificial Intelligence Review* 56, 11 (2023), 12387–12406.
[10] Timnit Gebru, Jamie Morgenstern, Briana Vecchione, Jennifer Wortman Vaughan, Hanna Wallach, Hal Daumé III, and Kate Crawford. 2021. Datasheets for datasets. *Communications of the ACM* 64, 12 (2021), 86–92.
[11] IEC. 2010. *IEC 61508: Functional Safety of Electrical/Electronic/Programmable Electronic Safety-related Systems*. International Electrotechnical Commission.
[12] IEC. 2006. *IEC 62304: Medical device software—Software life cycle processes*. International Electrotechnical Commission.
[13] ISO/IEC. 2023. *ISO/IEC 42001:2023—Information technology—Artificial intelligence—Management system*. International Organization for Standardization.

[14] Guy Katz, Clark Barrett, David L Dill, Kyle Julian, and Mykel J Kochenderfer. 2017. Reluplex: An efficient SMT solver for verifying deep neural networks. In *International Conference on Computer Aided Verification*. Springer, 97–117.
[15] Robert Kirk, Ishita Mediratta, Christoforos Nalmpantis, Jelena Luketina, Edward Grefenstette, and Roberta Raileanu. 2023. Understanding the effects of RLHF on LLM generalisation and diversity. *arXiv preprint arXiv:2310.06452* (2023).
[16] Philip Koopman and Michael Wagner. 2016. Autonomous vehicle safety: An interdisciplinary challenge. *IEEE Intelligent Transportation Systems Magazine* 9, 1 (2016), 90–96.
[17] Margaret Mitchell, Simone Wu, Andrew Zaldivar, Parker Barnes, Lucy Vasserman, Ben Hutchinson, Elena Spitzer, Inioluwa Deborah Raji, and Timnit Gebru. 2019. Model cards for model reporting. In *Proceedings of the Conference on Fairness, Accountability, and Transparency*. 220–229.
[18] NIST. 2023. *Artificial Intelligence Risk Management Framework (AI RMF 1.0)*. National Institute of Standards and Technology.
[19] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems*, Vol. 35. 27730–27744.
[20] OWASP Foundation. 2023. *OWASP Top 10 for Large Language Model Applications*.
[21] Ethan Perez, Saffron Huang, Francis Song, Trevor Cai, Roman Ring, John Aslanides, Amelia Glaese, Nat McAleese, and Geoffrey Irving. 2022. Red teaming language models with language models. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*. 3419–3448.
[22] Inioluwa Deborah Raji, Andrew Smart, Rebecca N White, Margaret Mitchell, Timnit Gebru, Ben Hutchinson, Jamila Smith-Loud, Daniel Theron, and Parker Barnes. 2020. Closing the AI accountability gap: Defining an end-to-end framework for internal algorithmic auditing. In *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*. 33–44.
[23] Elizabeth Seger, Aviv Ovadya, Ben Garfinkel, Divya Siddarth, and Allan Dafoe. 2023. Democratising AI: Multiple Meanings, Goals, and Methods. *arXiv preprint arXiv:2303.12642* (2023).
[24] Andrew D Selbst, Danah Boyd, Sorelle A Friedler, Suresh Venkatasubramanian, and Janet Vertesi. 2019. Fairness and abstraction in sociotechnical systems. In *Proceedings of the Conference on Fairness, Accountability, and Transparency*. 59–68.
[25] Toby Shevlane, Sebastian Farquhar, Ben Garfinkel, Mary Phuong, Jess Whittlestone, Jade Leung, Daniel Kokotajlo, Nahema Marchal, Markus Anderljung, Noam Kolt, et al. 2023. Model evaluation for extreme risks. *arXiv preprint arXiv:2305.15324* (2023).
[26] Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, et al. 2022. Emergent abilities of large language models. *arXiv preprint arXiv:2206.07682* (2022).