

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 transition 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 (Reflexio Agentic Governance Framework), 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 actions and 1,893 healthcare actions demonstrates operational viability: 41 unsafe actions prevented with zero false positives within validated operational scope, sub-30ms governance latency at p95, and fail-closed behavior across seven 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-adaptability tensions, ontology maintenance burden, and meta-governance requirements that technical architecture alone cannot resolve. RAGF provides enforcement infrastructure, not governance legitimacy.

Keywords

AI governance, agentic systems, boundary enforcement, safety certification, runtime verification, regulated domains

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 that scale poorly with deployment volume).

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 by traditional means) from deterministic boundary enforcement (certifiable through established methods). This paper demonstrates that boundary certification is technically viable while examining the governance trade-offs this approach introduces.

1.1 The Accountability Gap

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 through an *institutional checkpoint* where human-encoded values (expressed as ontologies and validation rules) constrain 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? 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 certification target, separating adaptive reasoning from deterministic authority mediation.
- (2) **Operational Taxonomy:** A three-level autonomy framework distinguishing advisory, supervised autonomous, and fully autonomous systems, clarifying where boundary enforcement applies.
- (3) **Governance Infrastructure:** Implementation of Validation Gate, semantic ontology integration, and cryptographic audit mechanisms with deployments across aviation and healthcare.
- (4) **Empirical Safety Evidence:** Demonstration of fail-closed enforcement across 14,740 production actions and 3,500 injected failures, with explicit formalization of guarantee scope and limitations.
- (5) **Governance Trade-off Analysis:** Systematic examination of authority centralization, encoding bias, rigidity-adaptability tensions, operational sustainability, and meta-governance requirements.

2 From Model-Centric 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 remains a model whose reasoning process is opaque and whose behavior cannot be guaranteed across all inputs.

2.1 The Governance Gap

We define the **Governance Gap** as the institutional deficit between adaptive AI capabilities and accountable execution. This gap is structural rather than merely technical: 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. The distributed nature of AI decision-making creates what legal scholars term “responsibility gaps”—situations where no single actor can be held clearly accountable [5].

Opacity Problem. Large language models 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 [2].

Certification Paradox. Regulators require demonstrable safety under failure conditions—a bedrock principle in aviation (DO-178C [18]), medical devices (IEC 62304 [10]), and industrial control (IEC 61508 [9]). Yet adaptive models exhibit non-deterministic behavior and distribution shift by design. Traditional certification assumes inspectable, stable logic operating within a defined envelope. Foundation models 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), and stable operating conditions (defined input space and failure modes). Large language models fundamentally violate these assumptions.

Non-Determinism. Temperature sampling, nucleus sampling, and beam search introduce variability. Even with temperature set to zero, floating-point arithmetic and hardware differences create non-reproducibility across runs and platforms.

Emergent Behavior. Capabilities not explicitly programmed emerge from scale and training data. Models exhibit sudden capability jumps and compositional generalization that were neither designed nor anticipated [20].

Distribution Shift. Models trained on historical data encounter novel inputs in deployment. Fine-tuning and reinforcement learning

from human feedback improve alignment but cannot guarantee safe behavior on out-of-distribution examples [3].

These properties are fundamental to how contemporary AI systems operate, not engineering limitations to be overcome through better tooling.

2.3 Autonomy Levels and Governance Requirements

To clarify where boundary enforcement applies, we distinguish three operational modes for agentic systems:

Table 1: Autonomy Levels and Governance Requirements

Level	Execution Model	Governance Need
Advisory	AI recommends; human decides and executes	Documentation, explainability
Supervised Autonomous	AI proposes and executes with validation	RAGF boundary enforcement
Fully Autonomous	AI decides and executes without intervention	Formal verification, stronger guarantees

RAGF targets the *supervised autonomous* level: systems where AI agents propose and initiate actions but where institutional accountability requires validation before commitment. This level represents the current frontier for regulated deployment—more capable than pure advisory systems, but with governance mechanisms that maintain human oversight over execution authority.

Advisory systems require documentation and explainability but not execution-time enforcement; the human decision-maker serves as the governance checkpoint. Fully autonomous systems require guarantees stronger than runtime validation can provide—formal verification of model properties, which remains an open research challenge for neural networks at scale [11].

2.4 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 certification target. If we accept that the reasoning process is opaque and probabilistic, we can still enforce that only validated actions proceed to execution.

3 RAGF Architecture

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

The Validation Gate enforces a deterministic function over proposed actions, serving as the architectural chokepoint through which all AI-proposed actions must pass before execution.

Deterministic Enforcement. Given the same action and organizational state, the gate produces the same verdict. This enables repeatability, inspection, and certification of validation logic independent of the probabilistic reasoning that generated the candidate action.

Fail-Closed Semantics. Any error in validation—database timeout, validator exception, cryptographic signature failure—results in DENY. The system defaults to safety under degradation, prioritizing prevention of unsafe actions over operational availability.

Escalation Pathways. Actions with ambiguous validation outcomes (incomplete semantic coverage, conflicting validator results, or novel action types not in ontology) trigger ESCALATE rather than ALLOW, routing decisions to human operators for discretionary judgment.

This design embeds institutional priorities: safety over efficiency, human judgment for edge cases, and 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 a valid action vocabulary (enumeration of permissible verbs such as “reroute_flight” or “adjust_dosage”), regulatory constraints (mapping from actions to applicable regulations), and 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. This provides defense against both model errors and adversarial prompt injection that attempts to induce unauthorized actions.

3.3 Cryptographic Audit

Each validation verdict is signed with HMAC-SHA256 and persisted to an append-only ledger. 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, and 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 restructures authority within agentic systems through explicit role separation:

The **Reasoning Agent** (LLM) proposes candidate actions based on probabilistic inference but holds no execution authority. **Policy Designers** encode organizational values and regulatory constraints as formal ontology rules. The **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.

4 Formal Specification

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

Let $\mathcal{A} = \{a_1, \dots, a_n\}$ be the space of candidate actions an agent may propose. Each action $a \in \mathcal{A}$ consists of a semantic verb v (action type), a target resource r (entity affected), and structured parameters θ (action-specific arguments).

Define the deterministic validation function:

$$V : \mathcal{A} \times \mathcal{S} \rightarrow \{\text{ALLOW}, \text{DENY}, \text{ESCALATE}\} \quad (1)$$

where \mathcal{S} represents organizational state (current regulations, resource availability, authority levels, temporal constraints).

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

$$V(a, s) = \text{ALLOW} \quad (2)$$

Safety Invariant. Under any failure mode $f \in \mathcal{F}$, validation defaults to denial:

$$\forall f \in \mathcal{F}, \forall a \in \mathcal{A} : V(a, s) \text{ under } f = \text{DENY} \quad (3)$$

This fail-closed property ensures that system degradation cannot result in unauthorized execution. The enumerated failure set \mathcal{F} includes LLM timeout, ontology database failure, validator exception, cryptographic signature failure, ledger write failure, semantic validation timeout, and health check failure.

4.1 Validation Algorithm

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

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 when all validators pass and semantic grounding is complete does validation produce ALLOW.

Algorithm 1 Deterministic Validation Gate**Require:** Action a , organizational state s **Ensure:** Verdict $\in \{\text{ALLOW}, \text{DENY}, \text{ESCALATE}\}$

```

1:  $results \leftarrow \emptyset$ 
2: for each validator  $v$  in  $\text{VALIDATORS}(a)$  do in parallel
3:    $r \leftarrow v(a, s)$ 
4:   if  $r = \text{DENY}$  then
5:     return DENY ▷ Fail fast on any denial
6:   end if
7:    $results \leftarrow results \cup \{r\}$ 
8: end for
9: if ESCALATE  $\in results$  then
10:  return ESCALATE
11: else if SEMANTICCOVERAGE( $a, s$ )  $< 1.0$  then
12:  return ESCALATE ▷ Incomplete grounding
13: else
14:  return ALLOW
15: end if

```

5 Threat Model

RAGF operates under explicit threat model boundaries that define the scope of security claims.

5.1 In-Scope Threats

LLM Adversarial Behavior. The system assumes the LLM may produce unsafe, malicious, or hallucinatory actions through model error, distribution shift, or adversarial prompt injection. 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, network connections, and validator processes may experience transient or permanent failures. Fail-closed design ensures failures result in denial rather than unsafe approval.

5.2 Out-of-Scope Threats

Infrastructure Compromise. If the underlying compute infrastructure is compromised, an attacker can bypass validation entirely. We assume standard infrastructure hardening including network segmentation, access control, and encryption at rest and in transit.

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

Cryptographic Key Compromise. HMAC signatures rely on secret key security. We assume keys are managed via hardware security modules or key management services with rotation policies. 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 with separation of duties.

5.3 Trust Assumptions

RAGF’s security model explicitly requires that policy designers possess relevant domain expertise and act in organizational interest, that validator implementations faithfully encode intended policies, that infrastructure follows industry-standard security practices, and that organizational change management processes function as designed.

These assumptions shift the certification target from model behavior (which cannot be guaranteed) to infrastructure and process (which can be audited and certified through established methods). However, they also concentrate trust in policy designers and infrastructure operators—a governance trade-off examined in Section 7.

6 Empirical Evaluation

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

6.1 Aviation: Dynamic Route Optimization

Deployment Context. Regional carrier operational simulation with 45 daily flights across European routes over a 90-day controlled pilot period.

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

Validators. The deployment included validators for fuel reserve compliance (FAA 14 CFR §91.151 requiring 45-minute reserve for IFR flights), crew rest requirements (FAA 14 CFR §91.1057 flight time limitations), and airspace restrictions (EASA regulations for European operations).

Ontology Scope. The aviation ontology contained 847 vertices encoding action verbs, regulatory references, aircraft types, and route segments, with 2,341 edges representing semantic relationships and constraint dependencies.

Results. Across the evaluation period, the system processed 12,847 proposed actions. Of these, 11,203 received ALLOW verdicts (87.2%), 1,544 received DENY verdicts (12.0%), and 100 were routed to ESCALATE for human review (0.8%). Among denied actions, expert review identified 37 that would have violated regulatory requirements if executed.

6.2 Healthcare: Medication Recommendations

Deployment Context. 250-bed facility with internal medicine and cardiology departments over a 60-day controlled pilot in a production-parallel environment.

Agent Capability. LLM suggests medication adjustments based on patient history, laboratory results, vital signs, and clinical practice guidelines.

Validators. The deployment included validators for drug-drug interaction screening against a comprehensive interaction database, patient allergy contraindication checking, dosage range compliance with hospital formulary limits, and renal/hepatic adjustment requirements based on laboratory values.

Ontology Scope. The healthcare ontology contained 1,247 vertices encoding medications, diagnoses, laboratory parameters, and clinical pathways, with 4,892 edges representing interactions, contraindications, and guideline relationships.

Results. The system processed 1,893 proposed medication actions. Of these, 1,612 received ALLOW verdicts (85.2%), 243 received DENY verdicts (12.8%), and 38 were routed to ESCALATE (2.0%). Expert review identified 4 potentially harmful interactions that would have occurred without validation. The higher ESCALATE rate compared to aviation reflects the greater frequency of novel drug combinations not fully covered by the formulary ontology.

6.3 False Positive Validation

All DENY verdicts across both deployments underwent expert review to assess false positive rates.

Aviation Review. Denials were reviewed by aviation safety managers holding FAA Part 121 certification. Each denial was independently classified by two reviewers as either a true positive (action would violate regulation or safety constraint) or false positive (action safe but incorrectly denied).

Healthcare Review. Denials were reviewed by clinical pharmacists board-certified in medication safety. Classification followed the same protocol with independent dual review.

Results. Across 1,787 total denials (1,544 aviation plus 243 healthcare), reviewers classified all as true positives with zero false positives. Inter-rater reliability measured by Cohen’s κ was 0.94 for aviation and 0.91 for healthcare, indicating substantial agreement. Eleven cases across both domains requiring regulatory or clinical interpretation were resolved through discussion and reference to advisory circulars or clinical guidelines.

Scope Limitation. This zero false positive rate applies within the tested operational scope: European routes with standard weather conditions and regulation-compliant flight plans for aviation; formulary medications with complete patient records for healthcare. Novel scenarios outside these parameters may exhibit different characteristics.

6.4 Governance Latency

Table 2: Governance Latency Under Sustained Load

Component	p50	p95	p99
Semantic Layer	4.2ms	6.8ms	9.1ms
Validation Gate	8.7ms	12.4ms	14.3ms
Cryptographic Signature	0.5ms	0.7ms	0.9ms
Ledger Write	4.9ms	8.2ms	8.9ms
Total Governance	18.3ms	28.1ms	33.2ms

Measurements from aviation deployment over 90-day period at sustained 50 requests per second.

Governance overhead at p95 of 28.1ms leaves approximately 170ms budget for LLM reasoning within typical 200ms total latency targets for interactive applications. This demonstrates that

deterministic validation is operationally viable without becoming a throughput bottleneck at moderate request rates.

6.5 Failure Injection Validation

To validate the fail-closed property, we conducted systematic failure injection across seven enumerated failure categories with 500 injections per category (3,500 total). Each injection simulated the target failure mode in isolation while monitoring validation behavior.

Table 3: Failure Injection Results

Failure Category	Injections	Unintended ALLOW
LLM Timeout (>30s)	500	0
Ontology DB Failure	500	0
Validator Exception	500	0
Signature Failure	500	0
Ledger Write Failure	500	0
Semantic Timeout (>500ms)	500	0
Health Check Failure	500	0
Total	3,500	0

Across all injections, zero unintended ALLOW verdicts were observed. All failures resulted in DENY with appropriate error classification in audit logs.

Methodology Limitation. This represents empirical validation through testing rather than formal proof via theorem proving. The property holds for enumerated failure modes tested in isolation; interaction effects between simultaneous failures or failure modes outside the enumerated set are not covered by this validation.

6.6 Comparative Context

To contextualize RAGF’s operational characteristics, we compared against human-in-the-loop validation from the same operational environment prior to RAGF deployment.

Table 4: Operational Comparison with Human-in-the-Loop Baseline

Metric	HITL	RAGF
Median Decision Time	4.2 min	28.1 ms
Actions per Hour	14	2,140
Denial Review (False Positive)	8.3%	0%*
Regulatory Violations	2	0
Audit Trail Completeness	67%	100%

validated operational scope. HITL baseline from 30-day period preceding RAGF deployment in aviation context.

Important Caveats. This comparison reflects acceleration of routine validation tasks that can be reduced to deterministic rule evaluation. Human operators provide contextual reasoning for cases RAGF routes to ESCALATE (0.8% aviation, 2.0% healthcare). The efficiency gain enables redeployment of domain expertise to higher-value activities including ontology refinement, escalation review,

and policy improvement rather than representing complete replacement of human judgment.

7 Governance Trade-offs

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

7.1 Centralization of Authority

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

Policy Designers as De Facto Policymakers. Domain experts who encode validation rules effectively exercise policymaking authority. 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 legitimacy questions: Should rules 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—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 and expertise.

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.

7.2 Policy Encoding Bias

Deterministic enforcement amplifies rather than mitigates encoded bias.

Codification Simplifies. Not all norms reduce to formal predicates. RAGF works well for bright-line rules ("crew rest < 10 hours → DENY") but struggles with context-dependent principles ("act in patient's best interests," "exercise reasonable care").

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

Systematic Rather Than Situational. While human operators exhibit bias, they can exercise discretionary judgment in edge cases. A human might recognize that a technically non-compliant action is warranted given unusual circumstances. Deterministic validators execute encoded rules without contextual flexibility, making encoding bias systematic rather than situational.

Position. We do not claim RAGF eliminates bias. Rather, 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 through institutional processes. However, this requires complementary governance mechanisms beyond RAGF's technical scope.

7.3 Rigidity Versus Adaptability

Strict boundary enforcement creates tension between safety and responsiveness.

Precautionary Default. RAGF's fail-closed design embeds a precautionary posture: when uncertain, 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.

Innovation Lag. 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. RAGF correctly escalated (per design), but the physician perceived this as bureaucratic impediment rather than safety mechanism.

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

We address this via ESCALATE pathways, but this reintroduces human bottlenecks. If ESCALATE is invoked frequently, the system converges toward supervised automation rather than governed autonomy.

7.4 Meta-Governance Requirements

RAGF introduces a governance layer, raising second-order questions.

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

Validator Accountability. Validators are trusted components in our threat model. But who audits the validators? External audit could provide oversight, but auditors face information asymmetry: policy designers understand domain constraints that auditors may not.

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?

High ESCALATE rates could signal ontology inadequacy (too restrictive, incomplete coverage, outdated rules) or system gaming (operators escalating to avoid delays). Distinguishing among these requires meta-governance mechanisms beyond technical scope.

7.5 Governance Layer Capture

Concentration of authority creates capture risks.

Regulatory Capture. If ontology maintenance becomes concentrated in industry groups, RAGF could institutionalize industry-favorable interpretations of ambiguous regulations.

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.

Mitigation Strategies. These risks require institutional rather than technical responses: participatory ontology governance with stakeholder representation, external audit for bias and overreach, transparent documentation of encoding choices, sunset clauses requiring periodic rule re-justification, and public comment periods for significant ontology changes.

7.6 Operational Sustainability

Beyond governance trade-offs, RAGF introduces operational costs that affect long-term viability across different deployment contexts.

7.6.1 Ontology Maintenance Burden. The Innovation Lag discussed earlier manifests as an ongoing operational cost: ontologies require continuous maintenance to remain current with evolving regulations, organizational practices, and domain knowledge.

Maintenance Intensity Varies by Domain. Our aviation deployment benefited from regulatory stability—FAA and EASA regulations change through formal rulemaking processes with multi-year timelines. The ontology required only 23 updates across the 90-day pilot, primarily adding new route segments rather than modifying validation logic. Healthcare presented greater maintenance demands: formulary changes, new drug approvals, and updated clinical guidelines required 47 ontology modifications during the 60-day deployment.

Domains with rapid evolution—cybersecurity threat landscapes, financial instrument innovation, emerging therapeutic modalities—would impose substantially higher maintenance burdens. In such contexts, the cost of ontology maintenance may approach or exceed the operational savings from automated validation, reducing RAGF’s net value proposition.

Toward Sustainable Maintenance. Several strategies may reduce maintenance burden without compromising determinism. Hierarchical ontologies can separate stable foundational concepts (action verb taxonomies, regulatory frameworks) from volatile operational details (specific thresholds, temporary restrictions), allowing targeted updates. Ontology drift metrics can quantify the rate at which proposed actions fall outside current coverage, signaling when updates are needed before operational impact accumulates. Semi-automated proposal pipelines can surface candidate ontology extensions from patterns in ESCALATE decisions, with human review preserving deterministic guarantees while reducing manual monitoring effort.

We did not implement these strategies in our deployments; they remain directions for future work. Organizations considering RAGF adoption should assess domain volatility and budget for ongoing ontology governance as a recurring operational cost, not a one-time implementation expense.

7.6.2 State Complexity and Consistency. The formal specification presents organizational state S as an input to the validation function, abstracting what may be a significant integration challenge in practice.

Distributed State Sources. In operational environments, relevant state spans multiple authoritative systems: crew scheduling databases, patient electronic health records, inventory management systems, real-time sensor feeds. Capturing consistent state snapshots for validation requires integration with these systems, each with its own consistency model, latency characteristics, and failure modes.

Our aviation deployment integrated with three state sources (crew management, flight planning, maintenance tracking) via synchronous API calls with 50ms timeout. Healthcare integration proved more complex, requiring HL7 FHIR queries to the electronic health record system with eventual consistency guarantees that occasionally surfaced stale laboratory values.

Consistency Under Uncertainty. When state cannot be reliably determined—database timeouts, conflicting values across systems, stale cache entries—RAGF’s fail-closed design provides a conservative default: uncertain state produces DENY or ESCALATE rather than potentially unsafe ALLOW. This preserves safety but may increase operational friction in environments with unreliable state infrastructure.

Organizations deploying RAGF should map state dependencies during design, establish authoritative sources for each state element, and implement health monitoring for state integrations. Degraded state availability directly impacts validation availability, making state infrastructure reliability a prerequisite for RAGF operational effectiveness.

7.6.3 Single Point of Trust. The threat model acknowledges that Validation Gate compromise is out of scope, but this framing warrants elaboration. The Validation Gate is not merely a component that might fail; it is the *root of trust* for the entire governance architecture. Compromised validation logic or poisoned ontologies would subvert governance silently rather than causing observable failure.

This concentration of trust is inherent to enforcement architectures, not a RAGF-specific limitation. Analogously, TLS security assumes private key integrity; operating system security assumes kernel integrity; access control assumes policy engine integrity. The alternative to concentrated trust is not distributed trust but absent enforcement.

Standard infrastructure hardening practices apply: the Validation Gate should execute in isolated environments with minimal attack surface, ontology modifications should require multi-party approval with cryptographic signing, validator binaries should undergo integrity verification at load time, and access to governance infrastructure should follow principle of least privilege with comprehensive audit logging.

These measures reduce compromise likelihood but cannot eliminate it. Organizations must accept residual risk of governance subversion as inherent to any enforcement architecture, while investing in defense-in-depth measures proportionate to the consequences of compromise.

7.7 Scope of RAGF

RAGF provides enforcement infrastructure for governance policies but does not determine what those policies should be, ensure policies are just or democratically legitimate, prevent misuse of governance mechanisms for organizational control, address upstream harms in training data or model design, guarantee that encoded rules capture all relevant ethical considerations, or resolve distributional questions of who benefits from automation and who bears costs.

Technical enforcement must be accompanied by participatory governance, continuous audit, transparent documentation, appeal mechanisms, investment in human expertise alongside automation, and ongoing assessment of distributional impacts.

8 Related Work

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

8.1 Model-Centric AI Safety

Constitutional AI [1] and reinforcement learning from human feedback [15] attempt to align model behavior through training interventions. These approaches improve model safety statistically but cannot provide formal guarantees required for regulated deployment. 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 [16]. RAGF enforces boundaries after generation, independent of prompting strategy.

8.2 Runtime Verification

Safety monitors in embedded systems enforce invariants over controller outputs [4]. 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 [12]. RAGF extends this with semantic validation: actions must be meaningful (grounded in ontology) and permissible (compliant with encoded constraints).

Neural network verification approaches [11] aim to prove properties about model behavior directly. While promising for specific architectures and properties, these techniques do not yet scale to foundation models. RAGF provides complementary runtime protection without requiring model-level verification.

8.3 Policy-Based Access Control

XACML and similar frameworks provide attribute-based access control. RAGF extends this paradigm with semantic grounding through domain ontologies, cryptographic auditability via signed verdicts, and escalation pathways for cases outside policy coverage.

8.4 AI Governance Scholarship

Algorithmic accountability research [5, 17] emphasizes transparency in automated decision systems. RAGF contributes cryptographic audit trails but does not address accountability for policy design choices themselves.

Value-sensitive design [7] proposes embedding values in technology. RAGF's ontologies encode organizational values, but this concentrates power in ontology designers. Ensuring encoding choices reflect diverse stakeholder values remains an institutional challenge.

Fairness, accountability, and transparency research [19] calls for inclusive design processes. RAGF provides technical enforcement but leaves governance legitimacy as an institutional requirement.

Model cards [13] and datasheets [8] document model characteristics and training data. These complement RAGF by providing transparency about the reasoning component while RAGF governs execution authority.

8.5 Regulatory Frameworks

The EU AI Act [6] establishes risk-based requirements for AI systems including transparency, human oversight, and conformity assessment. RAGF's architecture supports compliance through audit trails (transparency), ESCALATE pathways (human oversight), and deterministic validation logic (assessable conformity).

NIST's AI Risk Management Framework [14] emphasizes governance, mapping, measuring, and managing AI risks. RAGF provides technical infrastructure for governance and measurement dimensions while organizational processes must address mapping and management.

9 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 14,740 production actions in aviation and healthcare deployments, RAGF prevented 41 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.

9.1 Open Questions

This work raises questions that technical architecture alone cannot resolve.

Who should control ontologies, and through what process should rule-setting authority be exercised? How do we ensure governance rules reflect diverse stakeholder values rather than embedding only organizational priorities? Can deterministic enforcement adequately capture context-dependent ethics, or are some norms irreducibly situational? What mechanisms prevent regulatory capture of the governance layer? How should meta-governance operate—who audits the validators and reviews ontology changes?

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

9.2 Contributions

This work contributes formalization of execution boundary as certification target, separating adaptive reasoning from deterministic authority mediation. It provides a three-level autonomy taxonomy clarifying where boundary enforcement applies. It offers empirical demonstration of fail-closed enforcement across 14,740 production actions and 3,500 systematic failure injections. It articulates governance trade-offs including centralization, encoding bias, rigidity-adaptability tensions, operational sustainability, and meta-governance requirements. It provides evidence that boundary certification is operationally viable with sub-30ms governance latency. Finally, it establishes a framework distinguishing technical enforcement from governance legitimacy.

9.3 Future Directions

Future work must extend beyond enforcement mechanisms to address meta-governance: how stakeholders should participate in policy formation, what external oversight ensures validators implement intended policies, how governance rules can evolve responsively while maintaining stability, and whether ontologies can be shared across organizations to reduce development barriers while preserving local context.

The shift from model-centric to governance-centric AI is necessary as agentic systems enter regulated domains. RAGF provides infrastructure for this transition while making its inherent tensions 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 justice—questions our technical framework alone cannot answer.

References

- [1] Yuntao Bai, Saurav Kadavath, Sandipan Kundu, et al. Constitutional AI: Harmlessness from AI Feedback. *arXiv preprint arXiv:2212.08073*, 2022.
- [2] Rishi Bommasani, Drew A Hudson, Ehsan Adeli, et al. On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*, 2021.
- [3] Stephen Casper, Xander Davies, Claudia Shi, et al. 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. DRONA: A Framework for Safe Distributed Mobile Robotics. In *ACM/IEEE International Conference on Cyber-Physical Systems*, pages 239–248, 2017.
- [5] Nicholas Diakopoulos. Accountability in algorithmic decision making. *Communications of the ACM*, 59(2):56–62, 2016.
- [6] European Parliament. Regulation (EU) 2024/1689 on Artificial Intelligence (AI Act), 2024.
- [7] Batya Friedman and David G Hendry. *Value sensitive design: Shaping technology with moral imagination*. MIT Press, 2019.
- [8] Timnit Gebru, Jamie Morgenstern, Briana Vecchione, et al. Datasheets for datasets. *Communications of the ACM*, 64(12):86–92, 2021.
- [9] IEC. IEC 61508: Functional Safety of Electrical/Electronic/Programmable Electronic Safety-related Systems. International Electrotechnical Commission, 2010.
- [10] IEC. IEC 62304: Medical device software—Software life cycle processes. International Electrotechnical Commission, 2006.
- [11] Guy Katz, Clark Barrett, David L Dill, Kyle Julian, and Mykel J Kochenderfer. Reluplex: An efficient SMT solver for verifying deep neural networks. In *International Conference on Computer Aided Verification*, pages 97–117, 2017.
- [12] Philip Koopman and Michael Wagner. Autonomous vehicle safety: An interdisciplinary challenge. *IEEE Intelligent Transportation Systems Magazine*, 9(1):90–96, 2016.
- [13] Margaret Mitchell, Simone Wu, Andrew Zaldivar, et al. Model cards for model reporting. In *Conference on Fairness, Accountability, and Transparency*, pages 220–229, 2019.
- [14] NIST. Artificial Intelligence Risk Management Framework (AI RMF 1.0). National Institute of Standards and Technology, 2023.
- [15] Long Ouyang, Jeffrey Wu, Xu Jiang, et al. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems*, volume 35, pages 27730–27744, 2022.
- [16] Ethan Perez, Saffron Huang, Francis Song, et al. Red teaming language models with language models. In *Conference on Empirical Methods in Natural Language Processing*, pages 3419–3448, 2022.
- [17] Inioluwa Deborah Raji, Andrew Smart, Rebecca N White, et al. Closing the AI accountability gap: Defining an end-to-end framework for internal algorithmic auditing. In *Conference on Fairness, Accountability, and Transparency*, pages 33–44, 2020.
- [18] RTCA. DO-178C: Software Considerations in Airborne Systems and Equipment Certification. Radio Technical Commission for Aeronautics, 2011.
- [19] Andrew D Selbst, Danah Boyd, Sorelle A Friedler, Suresh Venkatasubramanian, and Janet Vertesi. Fairness and abstraction in sociotechnical systems. In *Conference on Fairness, Accountability, and Transparency*, pages 59–68, 2019.
- [20] Jason Wei, Yi Tay, Rishi Bommasani, et al. Emergent abilities of large language models. *arXiv preprint arXiv:2206.07682*, 2022.