REMORA

Governing the Autonomous Action.

A pre-execution governance layer that decides, before any side-effect occurs, whether an AI agent's proposed action is accepted, verified, abstained, or escalated to a human.

Multi-Oracle Consensus | Four-Outcome Policy Gate | Replayable Audit Envelope | TOGAF-Aligned Capability | Release v0.9.0

DEVELOPER
Stian Skogbrott

DOCUMENT
Enterprise Architecture White Paper

REPOSITORY
github.com/darklordVirtual/REMORA

EXECUTIVE SUMMARY

An assurance layer for systems that act on their own.

Autonomous AI agents no longer merely answer questions. They invoke tools, write to databases, move money, and reconfigure infrastructure. The decisive moment of risk is the instant *before* an action executes. REMORA places a single, auditable control point at exactly that moment.

Conventional safeguards (prompt filters, output classifiers, human approval queues) either inspect the wrong artefact (the text, not the action) or impose blanket friction that organisations cannot sustain at scale. REMORA takes a different stance: every proposed action is evaluated against **uncertainty**, **evidence**, **policy**, and **operational risk**, and routed to one of four explicit, recorded outcomes.

FIGURE 1 The four governance outcomes. Each carries a defined meaning, a recorded reason, and a named human owner where the risk profile requires it. **ACCEPT is a clearance to execute, never a claim that the action is correct.**

WHAT THIS DOCUMENT ESTABLISHES

1. **A canonical decision contract.** Every decision yields a `DecisionEnvelope`: a structured, hash-linked record capturing the action, its risk context, the evidence considered, the outcome, and the rationale.[1]
2. **Safety properties encoded as machine-checked invariants.** Hard policy blocks take precedence over model confidence; critical production mutations cannot be auto-accepted. Every invariant is verified automatically on each change.[2]
3. **Selective autonomy with a calibrated basis.** On a 544-item benchmark, REMORA's routing reaches **88.8% accuracy at 18% coverage**, a +47.6 point lift over a full-coverage majority-vote baseline.[3]
4. **Demonstrated harm reduction on agentic tool calls.** Across a 10,000-call adversarial benchmark, unsafe executions fell from **52.7% to 2.8%**; on a 700-task suite the policy gate held unsafe execution at **0%**.[4]
5. **A clean fit into enterprise architecture.** REMORA is described as a TOGAF Architecture Building Block (the *AI Action Governance Capability*) with a full mapping across the Architecture Development Method.[5]

EVIDENCE BASIS

Every quantitative claim in this paper resolves to a committed result artefact in the repository. Figures are drawn from controlled, reproducible benchmarks; REMORA is presented as a research-grade governance capability suitable for shadow-mode evaluation, not as a certified safety guarantee.

[1] `remora/governance/envelope.py` · [2] `remora/policy/decision_engine.py`, `tests/test_policy_invariants_prop.py` · [3] `results/selective_n500_results.json` · [4] `results/toolcall_stress_replay_10000.json`, `results/toolcall_benchmark_v2_results.json` · [5] `docs/togaf/architecture_building_block.md`

CONTENTS

How this paper is organised.

The paper moves from problem to concept to architecture to evidence, then closes by situating REMORA within enterprise governance. Readers new to the field may begin with §2; architects may proceed directly to §3 and §8.

READING PATH	START HERE	THEN
New to AI governance	§1 Governance Gap	§2 Concepts → §6 Audit
Software engineer	§3 Architecture	§4 Policy Engine → §7 Evidence
Enterprise architect	§8 TOGAF Alignment	§3 Building Blocks → §9 Adoption
Risk & compliance	§6 Auditability	§4 Invariants → §8 Controls

SECTION 01

The Governance Gap

An agent that can act is an agent that can act *wrongly*. The control that matters most is the one applied at the moment of action.

For most of their history, language models were advisory. A wrong answer was a wrong sentence: embarrassing, occasionally costly, but inert. The shift to **agentic systems** changes the stakes entirely. An agent that holds a database credential, an API key, or a shell can now *cause* consequences: it can delete a table, transfer a balance, disable a control, or reconfigure a production service.

Existing safeguards were designed for the advisory era and inspect the wrong thing. **Prompt filters** examine the input text. **Output classifiers** examine the generated text. Neither examines the *action*: the concrete tool call, with its concrete arguments, against a concrete target environment. The action is where the consequence lives, and it is precisely what conventional controls do not gate.

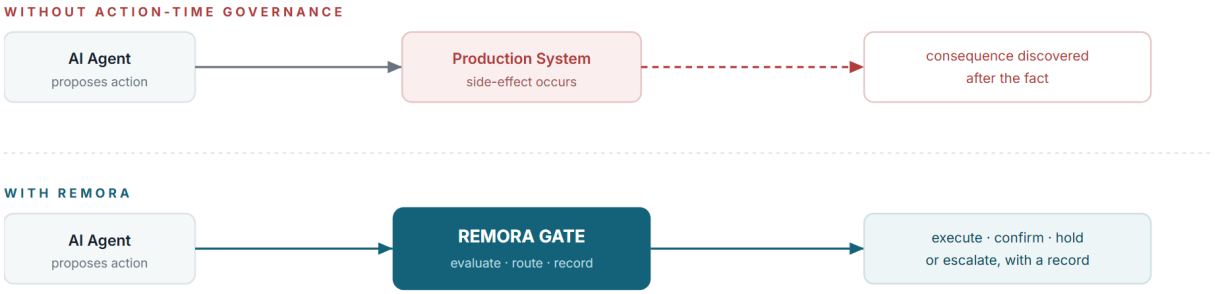

FIGURE 2 The structural difference. Without action-time governance, consequences are discovered after they occur. REMORA interposes a single evaluated, recorded decision point between intention and effect.

The instinct to add a human approver to every action does not scale: an organisation running thousands of agent actions per hour cannot route them all to a person, and a control that is too costly to operate is quietly switched off. The discipline REMORA introduces is **selective**: act autonomously where the basis is strong, seek confirmation where it is uncertain, decline where it is absent, and escalate where the risk demands a human, and in every case leave a record that withstands later scrutiny.

> **The thesis.** Governance for autonomous agents must operate *before* execution, on the *action* rather than the text, and must convert uncertainty, evidence, policy, and risk into a small set of explicit, auditable outcomes. The remainder of this paper shows how REMORA realises that thesis as an enterprise capability.

SECTION 02

Core Concepts, Explained

A small vocabulary carries the entire paper. Each term below is defined once, in plain language, and used consistently thereafter.

Agent & Action

An *agent* is an AI system permitted to take steps on its own. An *action* is one concrete step (a tool call such as "delete record 882") with arguments and a target environment.

Pre-execution gate

A checkpoint placed *before* an action runs. The gate may permit, question, decline, or escalate the action. Nothing reaches the target system until the gate decides.

Oracle

An independent model consulted for a judgement. REMORA queries several oracles in parallel and treats their *agreement* and *disagreement* as signal, rather than trusting any single voice.

Consensus & dissensus

When oracles agree, confidence rises; when they disagree, the disagreement itself is measured and feeds the decision. Silent disagreement is never allowed to resolve into a confident "yes".

Uncertainty signal

A numeric estimate of how reliable the current judgement is, derived from oracle disagreement and the diversity of their answers. High uncertainty steers an action away from autonomous acceptance.

Evidence

Supporting material for a claim, ideally retrieved from documents. Evidence that *contradicts* a proposed action blocks acceptance; absent evidence on a high-risk action triggers verification.

Policy & risk tier

Rules that encode what the organisation will and will not allow. Each action carries a *risk tier* (low, medium, high, or critical) that determines how much scrutiny it must pass.

Hard block

A rule that cannot be overridden by model confidence. Adversarial input, a forbidden tool, or a coercion pattern produces an escalation regardless of how "certain" the models are.

Selective autonomy

The principle of acting automatically only on the subset of actions where the basis is strong, and deliberately abstaining elsewhere, to maximise safe automation without blanket friction.

Decision Envelope

The structured record produced for every decision: the action, its context, the evidence, the outcome, the rationale, and a cryptographic link to the preceding record.

Shadow mode

A deployment style in which REMORA evaluates and records what it *would* decide, without blocking anything, letting an organisation observe and calibrate governance before enforcing it.

Tamper-evident audit

A record structured so that any later alteration is *detectable*. Records are chained by cryptographic hashes; a change to one breaks the chain at that point.

A NOTE ON PRECISION

This paper says *tamper-evident*, not "tamper-proof": the audit chain makes alteration detectable, and an externally published digest extends that guarantee. It says *clearance to execute*, not "guarantee of correctness": an ACCEPT means the conditions for autonomous action were satisfied. This precision is deliberate and runs throughout.

How the concepts combine

The terms above are not independent checks bolted together; they form a single, ordered judgement. An incoming action is first screened for anything categorically unacceptable. If it survives, its *uncertainty* and *evidence* are weighed against its *risk tier*. Only an action that is both well-supported and appropriately low-risk earns autonomous acceptance. Everything else is routed, transparently, to verification, abstention, or a human.

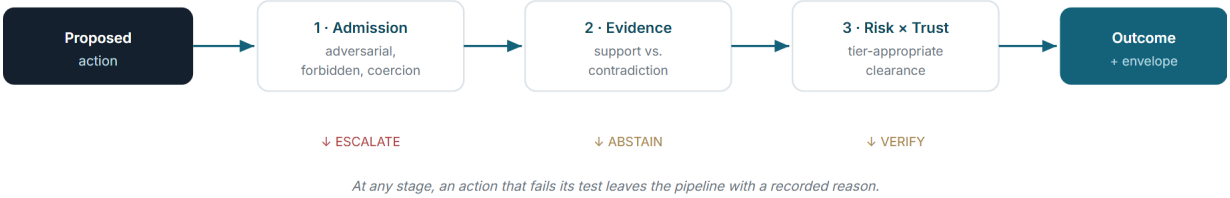

FIGURE 3 The conceptual pipeline. Screening precedes evidence, which precedes the risk-versus-trust judgement. **An action only reaches ACCEPT by passing every prior stage**: the ordering is itself a safety property (see §4).

This ordering is the heart of REMORA's conservatism. Because the categorical screen runs first, no amount of downstream confidence can rescue an adversarial or forbidden action. Because the risk-versus-trust judgement runs last among the gates, a high-risk action that lacks evidence is held even when the models feel certain. The architecture in §3 makes this concrete; the guarantees in §4 make it verifiable.

SECTION 03

Architecture & Building Blocks

REMORA is a composition of replaceable components joined by stable contracts. Each component is, in TOGAF terms, a building block with defined inputs and outputs.

◆ TOGAF · PHASE C

The system is organised as a pipeline from a proposed action to a recorded decision. The interfaces between stages are fixed; the implementations behind them are not. An organisation may substitute its own oracle set, evidence retriever, policy store, or audit backend without altering the decision contract that downstream consumers depend on.

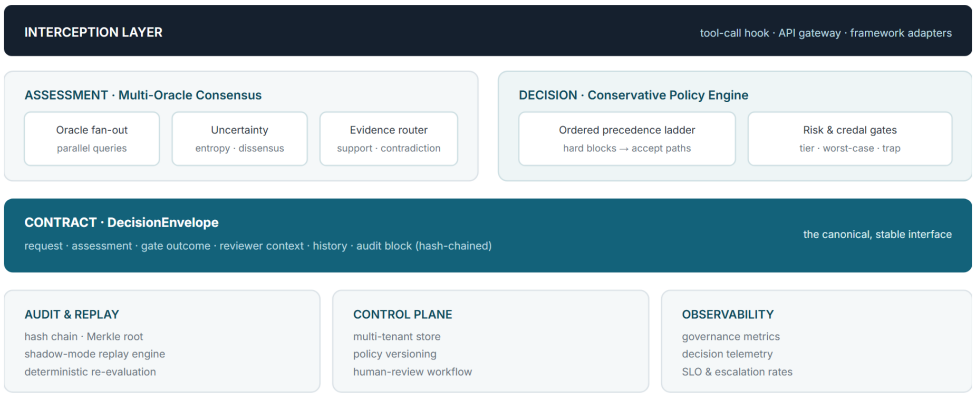

FIGURE 4 The layered architecture. Assessment and decision feed a single canonical contract, the **DecisionEnvelope**, which all persistence, replay, and observability components consume. Stable contracts at the centre let every surrounding implementation be replaced independently.

The building blocks

BUILDING BLOCK	RESPONSIBILITY	CONTRACT
Interception layer	Captures a proposed action before it executes, from a tool-call hook, REST gateway, or framework adapter.	`PolicyObservation`
Consensus engine	Queries multiple oracles, measures agreement and uncertainty, and routes evidence.	`AssessmentBlock`
Policy engine	Applies an ordered ladder of rules to produce one of four outcomes with a recorded rationale.	`DecisionReport`
Envelope & audit	Serialises the full decision and links it, by hash, to the preceding record.	`DecisionEnvelope`
Replay engine	Re-evaluates historical actions deterministically: the basis for shadow mode.	`replay()`

WHY CONTRACTS, NOT CODE, ARE THE ASSET

Enterprises do not adopt implementations; they adopt interfaces they can depend on. REMORA's value to an architecture is the `DecisionEnvelope`: a single, versioned governance record that every system, auditor, and dashboard can read the same way, regardless of which models or stores sit behind it.

Anatomy of a decision

A single action produces a single envelope. Its structure is intentionally flat and self-describing, so that it serialises cleanly to JSON, to an audit log, or to a compliance system without bespoke transformation.

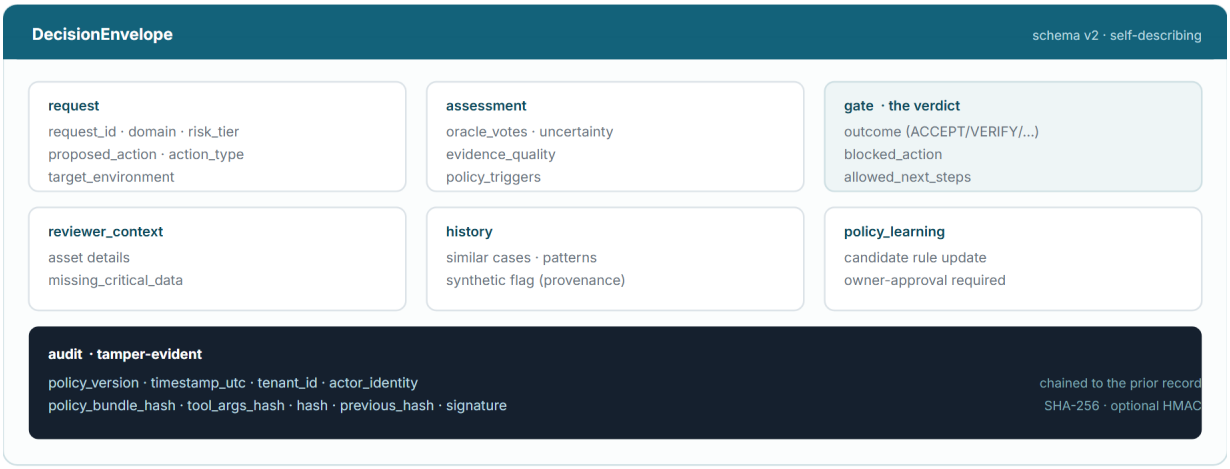

FIGURE 5 The DecisionEnvelope. Six descriptive blocks plus a cryptographic audit block. The **gate** block is the single source of truth for the outcome; the **audit** block links each record to its predecessor so that any later alteration is detectable.

Three properties of this contract matter to an enterprise. First, it is **complete**: a reviewer can reconstruct why a decision was made from the envelope alone. Second, it is **honest about provenance**: where case history is illustrative rather than drawn from live records, the envelope says so explicitly. Third, it is **stable**: the schema is versioned, so consumers can rely on it across releases.

SECTION 04

The Policy Engine: Properties That Hold

A governance layer is only as trustworthy as its worst case. REMORA's policy engine encodes its most important safety properties as *machine-checked invariants*, verified automatically on every change.

The engine evaluates an action against an **ordered ladder** of rules. Order is not a detail. It *is* the safety argument. Categorical prohibitions are tested first and exit immediately; the paths that can grant autonomous acceptance sit at the very bottom, reachable only by an action that has already survived every prohibition above it.

FIGURE 6 The precedence ladder. Acceptance is the **last** thing the engine considers. Because hard blocks and risk gates run first and exit immediately, no downstream confidence can lift a prohibited or critical-risk action into autonomous acceptance.

Properties verified as invariants

The most consequential properties are not merely intended; they are encoded as **property-based invariants** and checked automatically across thousands of generated input combinations on every change. An invariant that fails breaks the build.

INVARIANT	GUARANTEE
Critical-risk actions never reach ACCEPT	No combination of trust, phase, or evidence lifts a critical action to autonomous execution.
Hard blocks override maximum confidence	Adversarial input escalates even at the highest possible trust score.
Failed oracles cannot strengthen consensus	An unavailable model is excluded, never counted as agreement.
Evidence contradiction blocks ACCEPT	A conflicting evidence item forces verification or escalation.
Unknown risk context fails closed	An absent or misspelled risk tier on a mutating action routes to VERIFY, never silently through.
VERIFY and ESCALATE require human review	Both outcomes set the human-review flag consistently in the record.

> **Why machine-checked invariants matter.** A control that depends on a model behaving well will eventually meet a model that does not. By placing the organisation's non-negotiables *above* the model's judgement in a fixed order, and by verifying that ordering automatically across thousands of generated inputs, REMORA makes its core properties independent of any single model's reliability.[2]

[2] Invariants: `tests/test_policy_invariants_prop.py` · engine: `remora/policy/decision_engine.py`. Verified across the project's automated suite of 2,600+ tests.

A worked example

Consider an agent that proposes to delete a customer record in production. The same action, under three different evidential and risk situations, produces three different (and correct) outcomes.

ESCALATE

CRITICAL · PRODUCTION

An irreversible deletion against a live system. No trust score suffices; a named human must authorise.

VERIFY

HIGH · EVIDENCE THIN

A high-risk deletion lacking supporting evidence. Plausible, but confirmed before it proceeds.

ACCEPT

LOW · WELL-EVIDENCED

A low-risk cleanup in a sandbox, strongly supported and consistent. Cleared to proceed autonomously.

The point of the example is not that REMORA is cautious; it is that REMORA is *discriminating*. A system that escalated everything would be safe and useless; a system that accepted everything would be useful and dangerous. The value lies in routing each action to the outcome its actual basis warrants, and in being able to *show*, afterwards, why.

WHAT THE ENGINE READS

The judgement draws on a structured observation of the action: its risk tier, action type, and target environment; the oracle votes and the uncertainty they imply; the evidence verdict and any contradictions; and a set of integrity signals: whether the action is reversible, whether the environment is what the agent believes it to be, whether the request shows signs of coercion. Each signal has a defined, conservative default: where a signal is *absent*, the engine treats the situation as *unknown*, not as *safe*.

DESIGN PRINCIPLE: NONE IS UNKNOWN, NOT SAFE

A missing integrity check is never read as a passing one. If the schema validator did not run, a mutating action is verified rather than accepted; if the reversibility of an action is unestablished, a high-risk action does not proceed on the assumption that it can be undone. Absence of a signal moves an action toward caution, by design.

SECTION 05

Uncertainty, Evidence & Selective Autonomy

The discipline that makes selective autonomy work is the ability to recognise, and act on, the difference between a strong basis and a weak one.

A single model's confidence is an unreliable guide; models can be fluently, confidently wrong. REMORA's assessment layer therefore treats confidence as something to be *earned* from the behaviour of several independent models, and tempered by whether retrievable evidence supports the proposed action.

Consensus as signal, disagreement as warning

Several oracles are queried in parallel. Where their answers cluster, the basis is strong; where they diverge, the divergence is quantified and lowers the system's willingness to act autonomously. Two complementary measures capture this: the **entropy** of the answers (how spread out they are) and the **dissensus** (how much the models pairwise disagree). A failed or unavailable oracle is excluded from the tally, never silently counted as assent.

CONFORMAL RISK CONTROL: A CALIBRATED PROMISE

Beyond heuristics, REMORA can calibrate its acceptance threshold so that the expected error rate on accepted actions stays within a chosen bound. The method, conformal risk control, provides a formal relationship: the expected loss on the selected set is held at or below the target rate, with a small, well-understood correction. In plain terms: the organisation chooses the error budget; the calibration honours it on the actions it chooses to accept.[6]

Evidence that can say "no"

Evidence in REMORA is not decoration. An evidence item that *contradicts* a proposed action blocks acceptance outright. On high- and critical-risk actions, the *absence* of supporting evidence is itself a reason to verify rather than proceed. Where a live retrieval system is connected, this becomes a genuine grounding check; the interface for it is fixed, so an organisation can supply its own document corpus and retrieval backend behind the same contract.

Selective autonomy, illustrated

The pay-off of all this is a system that automates confidently where it should and steps back where it should. On a 544-item benchmark, restricting autonomous action to the strongest 18% of cases yields **88.8% accuracy on that accepted set**, far above the 41% achieved by answering every case with a majority vote. The curve below shows the trade-off the organisation gets to tune.

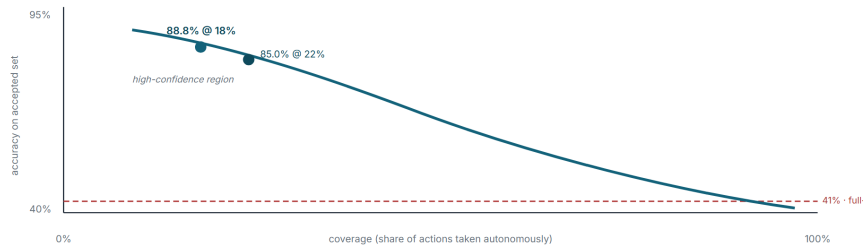

FIGURE 7 The selective-trust trade-off (544-item benchmark). Accepting only the strongest cases keeps accuracy high; widening coverage trades accuracy for reach. **The organisation chooses the operating point**; the dashed line marks the accuracy of answering everything without selection.[3]

[3] `results/selective_n500_results.json` · [6] Conformal risk control: `remora/selective/crc.py`; expected loss on the accepted set bounded by the target rate plus a 1/(n+1) correction.

The critical-phase insight

One empirical finding shaped REMORA's evidence strategy. The cases models find genuinely hard (the *critical* region, where the system sits at the boundary between confident agreement and scattered disagreement) are exactly the cases where a naïve trust score is least reliable. Treating all high-uncertainty cases identically would either waste the easy ones or admit the dangerous ones.

REMORA's phase-aware guardrail separates these regimes. By recognising the critical region and routing it through a distinct, calibrated rule rather than the ordinary trust threshold, the system extends the share of actions it can responsibly accept while keeping accuracy on the accepted set high: **85.0% accuracy at 22.1% coverage**, a measurable gain over the trust-only operating point.[7]

18.2%
COVERAGE · ORDERED ONLY
86.9% accuracy

22.1%
COVERAGE · PHASE-AWARE
85.0% accuracy

+47.6pp
LIFT OVER MAJORITY VOTE
at the 18% operating point

The broader lesson is methodological. REMORA does not assume that uncertainty behaves uniformly; it measures how uncertainty *actually* relates to correctness in each regime and calibrates accordingly. This is what separates a principled selective-autonomy layer from a confidence threshold with good intentions.

FOR THE ARCHITECT

Every signal in this section (consensus, uncertainty, evidence verdict, phase) arrives at the policy engine as a field on a single structured observation, and departs in the DecisionEnvelope. The assessment layer can therefore be re-implemented, re-calibrated, or swapped for a domain-specific one without touching the decision contract or the audit trail.

[7] `results/phase_aware_guardrail_n544_results.json` · guardrail: `remora/selective/guardrail.py`. Operating points are empirical flat-phase figures on the 544-item benchmark, with 95% Wilson confidence intervals recorded in the artefact.

SECTION 06

Auditability & the Decision Envelope

A governance decision that cannot be reconstructed and trusted after the fact is not governance. REMORA treats the audit record as a first-class product, not a by-product.

◆ TOGAF · PHASE G

Every decision produces a `DecisionEnvelope` (§3). Three mechanisms turn that record into something an auditor, a regulator, or an incident responder can rely on: **chaining**, **anchoring**, and **replay**.

Chaining: alteration becomes visible

Each envelope carries a cryptographic hash of its safety-relevant content and a reference to the hash of the preceding record. The records thus form a chain. Altering any historical record changes its hash and breaks the link to its successor, so tampering is *detectable* by anyone who recomputes the chain. Where the organisation requires it, each record's hash can additionally be signed.

Anchoring: extending the guarantee outward

A hash chain held entirely within one system can, in principle, be rewritten wholesale. REMORA therefore supports computing a single **Merkle root** over an audit window (a compact digest that summarises every record in it) which can be exported and published to an independent, append-only store on a regular cadence. Once a root is anchored externally, the entire window it covers becomes resistant to silent revision.[8]

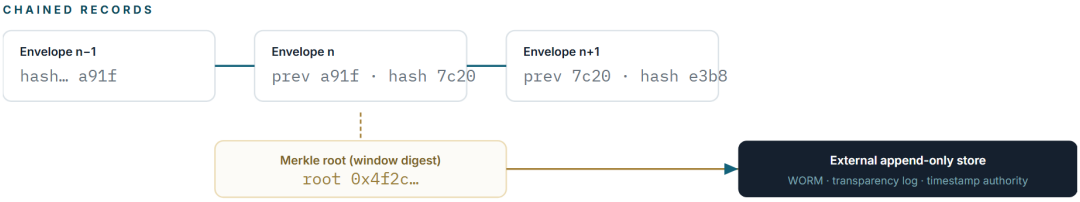

FIGURE 8 Chaining and anchoring. Records are linked by hashes so local tampering is detectable; a periodic Merkle root, published to an independent store, makes a whole window resistant to wholesale revision. **The paper claims tamper-evidence, and, with anchoring, tamper-resistance; never tamper-proof storage on its own.**

Replay: counterfactual governance without risk

Because a decision is a deterministic function of its recorded inputs, any historical action can be re-evaluated exactly. This is the foundation of **shadow mode**: an organisation replays a log of real agent actions through REMORA and sees precisely which it would have accepted, verified, held, or escalated, and how that profile would change under a different policy, all without ever blocking a live action. It is the safest possible way to introduce a governance layer: observe first, enforce later.

[8] Anchoring: `remora/audit/merkle.py` · chain: `remora/audit/hash_chain.py` · replay: `remora/shadow/replay.py`.

What an auditor can establish from the record alone

The envelope is designed so that the questions an auditor or regulator actually asks can be answered from the record itself, without access to the running system.

AUDIT QUESTION	ANSWERED BY
What action was proposed, and against what environment?	request block: action, type, target, risk tier
What was decided, and what was permitted next?	gate block: outcome and allowed next steps
On what basis: which signals, which evidence?	assessment block: oracle votes, uncertainty, evidence quality
Under which policy version was it judged?	audit block: policy version and policy-bundle hash
Who, or what service, initiated it; for which tenant?	audit block: actor identity, tenant, timestamp
Has this record been altered since it was written?	audit block: hash, previous hash, optional signature
Is the supporting history real or illustrative?	history block: explicit provenance flag

Regulatory relevance. The same record supports the control narratives that modern AI-governance frameworks expect: a documented decision basis, an accountable actor, a versioned policy, and an integrity-protected trail. REMORA's enterprise materials map these explicitly to EU AI Act and ISO/IEC 42001 control families.[9]

The design intent is simple to state: the cost of proving good governance after the fact should be the cost of *reading a record*, not the cost of reconstructing a decision from logs, model states, and institutional memory.

[9] `docs/togaf/compliance_assessment_template.md`: control mapping across EU AI Act and ISO/IEC 42001 families.

SECTION 07

Empirical Foundation

Architecture earns trust through evidence. Each figure below resolves to a committed, reproducible result artefact.

REMORA's empirical case rests on three pillars: **selective accuracy** (does autonomous acceptance stay reliable?), **harm reduction** (does the policy gate actually stop unsafe tool calls?), and **discrimination** (does the system tell genuinely different situations apart?). The headline figures follow; each cites the artefact it is drawn from.

88.8%
SELECTIVE ACCURACY @ 18% COVERAGE
`selective_n500_results.json · N=544`

0%
UNSAFE EXECUTION · 700-TASK GATE
`toolcall_benchmark_v2_results.json`

52.7% → 2.8%
UNSAFE EXECUTION · 10,000-CALL STRESS
`toolcall_stress_replay_10000.json`

100%
PRECISION · CROSS-DOMAIN EVIDENCE REPLAY
`domain_benchmark_results.json · 32 curated cases, static provider`

Reading the harm-reduction result

The most operationally meaningful figure is the stress benchmark. Across ten thousand adversarial tool calls (spanning capability misuse, prompt injection, and unsafe-workflow patterns), an unguarded caller executed roughly **half** of the unsafe actions. With REMORA's policy gate interposed, unsafe execution fell to **under three percent**, while the gate continued to permit the benign actions that make automation worthwhile. The comparison is like-for-like: the same actions, the same ground truth, with and without the gate.

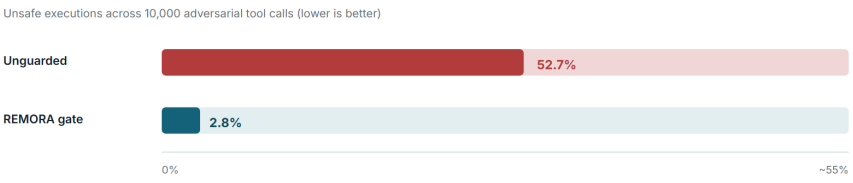

FIGURE 9 Harm reduction on the stress benchmark.[4] The policy gate removes the great majority of unsafe executions while preserving benign throughput: the property that makes governed autonomy practical rather than merely safe.

HOW TO READ THESE NUMBERS

These results come from controlled, reproducible benchmarks with committed ground truth: the right instrument for measuring whether the governance logic behaves as designed. They establish that the mechanism works as specified; broadening to live, organisation-specific evaluation is the natural next step of any pilot, and shadow mode (§6, §9) is built precisely to support it.

[4] `results/toolcall_stress_replay_10000.json`, `results/toolcall_benchmark_v2_results.json`: deterministic adversarial tool-call benchmarks with committed per-task ground truth.

Discrimination, not blanket caution

A governance layer that escalates everything is trivially "safe" and operationally worthless. The evidence that REMORA is *discriminating* (that it separates situations that genuinely differ) is as important as the harm-reduction figure.

EVALUATION	WHAT IT MEASURES	RESULT
Selective routing (N=544)	Accuracy retained when acting on the strongest cases	**88.8%** @ 18% coverage
Phase-aware guardrail (N=544)	Coverage extended without sacrificing accuracy	**85.0%** @ 22% coverage
Tool-call policy gate (700 tasks)	Unsafe execution under adversarial pressure	**0%** unsafe
Stress replay (10,000 calls)	Harm reduction vs. an unguarded caller	**2.8%** vs 52.7%
Cross-Domain Evidence Replay (32 cases)	Precision of evidence-based escalation (distinct from the N=3,000 Evidence Router Proxy Benchmark in the research paper)	**100%** precision
Factory replay (65 cases)	Deterministic re-evaluation accuracy	**100%**

Read together, the results describe a system that holds a tight error budget on the actions it chooses to take, removes the overwhelming majority of unsafe executions under adversarial pressure, and does so while continuing to let safe work through. That combination of reliability, harm reduction, and preserved utility is the precise property an enterprise needs before it will let an agent act on its own.

> **Reproducibility as a discipline.** Every figure in this section is backed by a committed artefact and an automated check that the published numbers still match the data. The project's working rule is explicit: no number appears in a paper, badge, or abstract unless the exact result artefact exists on disk. The evidence base is therefore not a snapshot; it is a standing invariant of the codebase.

Hardening against mislabelled actions: the Governance Intelligence Layer

The policy engine is only as honest as the metadata it is given. An integration bug, or a compromised agent, can label a destructive action as a low-risk read and present the engine with nothing to gate on. The Governance Intelligence Layer closes this gap: before any policy rule runs, a deterministic enrichment stage extracts action semantics from the proposed action itself, infers misspecification, blast-radius, and policy-generalization signals, and populates the observation under a **strengthen-only** rule. Inferred higher risk may override a supplied lower label; the reverse can never happen, and a grid property test across 2,160 observation combinations verifies that enrichment never converts a rejection into an acceptance.

0.0%
UNSAFE ACCEPTS · DISGUISED-ACTION BENCHMARK
`governance_intelligence/evaluation_results.json · 50 tasks, deterministic`

100%
METADATA-MISMATCH DETECTION
`destructive actions labelled as low-risk reads`

100%
LEGITIMATE READS STILL ACCEPTED
`conservatism without blanket caution`

96.7%
ESCALATION PRECISION
`escalations land on genuinely unsafe tasks`

A learning loop that audits its own measurement

REMORA's adaptive layer (AROMER) learns from every governed decision. For that learning signal to be trustworthy, the measurement itself must be honest: the published intelligence index now separates **measured** results from static expectations. The cross-domain transfer component is fed by a real replay-arena run (85 cases, **88.2%** accuracy, **0%** false accepts) and its provenance is labelled in the live API; a smoothed index filters sliding-window measurement noise from genuine learning trends; and a stability component now measures whether repeated measurements of the same system agree, recovering from a structural floor of 0.11 to 0.21 in its first cycle. Before-and-after snapshots of the live system are committed as artefacts.

> **Why measurement integrity is a governance feature.** An adaptive governance system that scores itself with constants or noise will eventually justify trusting thresholds it has not earned. By labelling provenance, smoothing measurement noise at read time while preserving raw history for audit, and refusing to publish replay results that contain any false accept, the learning loop applies REMORA's own claim discipline to itself.

Artefacts: `artifacts/governance_intelligence/evaluation_results.json` · `artifacts/aromer/replay_arena_report.json` · `artifacts/aromer/intelligence_before_v020.json` / `intelligence_after_v020.json` · property test: `tests/policy/test_governance_intelligence_never_weakens_policy.py`

SECTION 08

Enterprise Architecture Alignment

REMORA is described in the language of enterprise architecture so that it can be governed, planned, and adopted the way enterprises govern, plan, and adopt anything else.

◆ TOGAF · ADM A–H

A capability that an enterprise will deploy at the centre of its automation must fit its architectural method, not sit beside it. REMORA is therefore specified as a TOGAF **Architecture Building Block** (the *AI Action Governance Capability*) with a corresponding Solution Building Block, a full mapping across the Architecture Development Method, and the standard supporting artefacts an architecture board expects.[5]

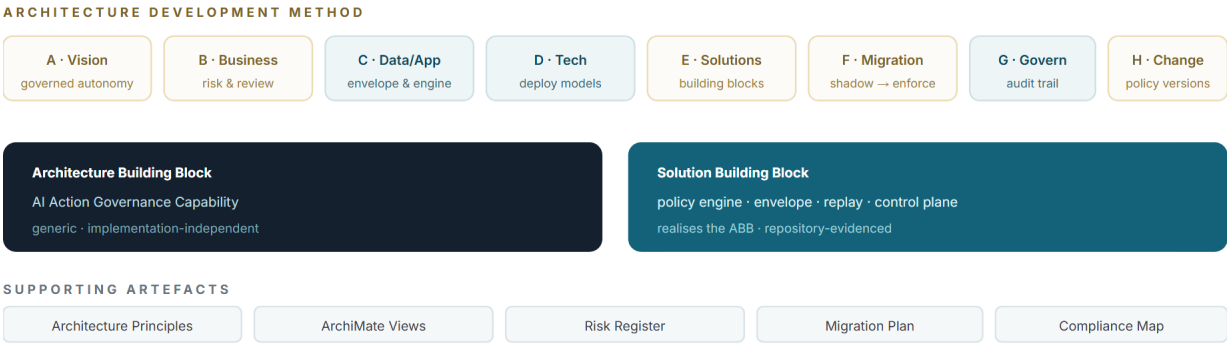

FIGURE 10 REMORA across the ADM. The capability is expressed as a generic Architecture Building Block realised by a concrete Solution Building Block, with the supporting artefacts an architecture board reviews. **Teal phases are where REMORA's components live; brass phases are where it is planned and governed.**

The effect of this framing is practical. An architecture review board can place REMORA on its capability map, trace it from business motivation (Phase A–B) through application and technology design (Phase C–D) to migration and governance (Phase F–G), and assess it with the same instruments it uses for any enterprise capability: principles, contracts, risk, and compliance.

The architecture pack

The capability is documented as a complete, board-ready set of artefacts. Each is a standard TOGAF deliverable; together they let an enterprise adopt REMORA through its normal architecture governance, rather than as an exception to it.

ARTEFACT	ADM PHASE	PURPOSE
Architecture Building Block	B–C	Defines the capability, its inputs, outputs, and stakeholders, independent of implementation.
Solution Building Block	C–E	Specifies the concrete components and the API contract that realise the capability.
ADM Mapping	A–H	Traces the capability through every ADM phase with repository evidence.
Architecture Principles	Preliminary	States the design principles and the design rules each one implies.
ArchiMate Views	B–D	Layered, motivation, and sequence views for visual stakeholder communication.
Risk Register	G	Enumerates risks with likelihood and impact scoring.
Migration Plan	E–F	Transition architectures and sequencing from observation to enforcement.
Compliance Assessment	G	Maps controls to EU AI Act and ISO/IEC 42001 families.
Enterprise Pilot Playbook	F	A phased checklist for running a controlled pilot.

THE ARCHITECTURAL ARGUMENT, IN ONE LINE

REMORA is not a tool to be integrated; it is a *capability to be governed*, and it arrives already described in the vocabulary an enterprise architecture function uses to govern.

This is the red thread of the paper. The four-outcome gate (§4), the calibrated assessment (§5), and the replayable envelope (§6) are not merely sound engineering; they are the realisation of a named enterprise capability with a defined place in the architecture, a defined adoption path, and a defined control story. The technical depth and the architectural framing are the same object seen from two sides.

SECTION 09

The Adoption Path

A governance layer should be introduced the way a prudent organisation introduces any control over a live system, by observing before enforcing. ◆ TOGAF · PHASE F

REMORA's replay capability (§6) makes a graduated adoption path not just possible but natural. An organisation can derive confidence from its own data before a single live action is ever blocked, then tighten enforcement deliberately as that confidence grows.

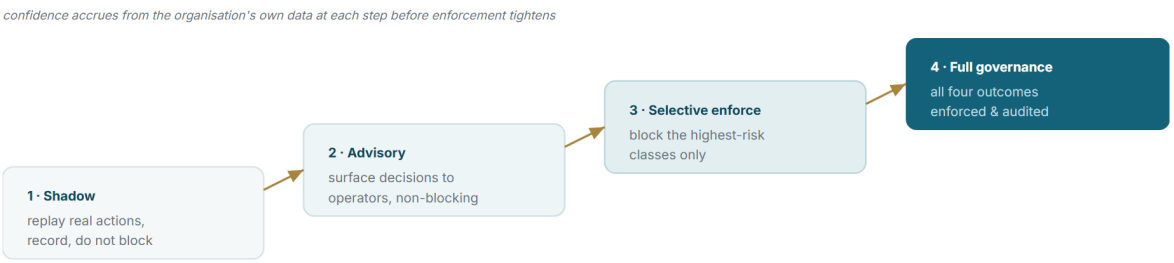

FIGURE 11 The adoption maturity path. Each stage produces evidence that justifies the next. **No live action is blocked until the organisation has seen, on its own traffic, exactly what enforcement would do.**

Integration surfaces

REMORA meets an agent stack wherever the proposed action can be observed before it executes. The same decision contract is produced regardless of surface, so the audit trail and dashboards are uniform across a heterogeneous estate.

- **Tool-call hook**: intercepts an agent's tool invocations directly, at the point of execution.
- **REST gateway**: a service endpoint that returns a governance decision for a proposed action, for orchestrators and custom integrations.
- **Framework adapters**: drop-in interposition for common agent frameworks, gating the dispatch step.
- **Replay ingestion**: batch evaluation of historical action logs, the engine of shadow mode.

What a pilot establishes. A shadow-mode pilot answers the three questions an enterprise actually has before it grants autonomy: *how often* would governance intervene, *on which actions*, and *at what operational cost*, all measured on the organisation's own traffic, with a complete audit trail, and with zero risk to production.

SECTION 10

A Governed Foundation for Autonomy

The question is no longer whether AI agents will act on their own. It is whether they will do so under governance that an enterprise can trust, audit, and control.

This paper has made a single, cumulative argument. Autonomous agents create a new locus of risk, the moment of action, that conventional safeguards do not address (§1). Governing that moment requires a small vocabulary of explicit outcomes and a disciplined way of choosing among them (§2). REMORA realises that discipline as a layered architecture joined by a stable decision contract (§3); embeds its most important guarantees as properties that hold by construction and are verified automatically (§4); earns its confidence from calibrated consensus and evidence rather than a single model's certainty (§5); and records every decision in a replayable, tamper-evident envelope (§6).

The empirical foundation shows the mechanism works as specified: high accuracy on the actions it chooses to take, a dramatic reduction in unsafe executions under adversarial pressure, and the discrimination to tell genuinely different situations apart (§7). And because the whole capability is expressed in the language of enterprise architecture, as a named Architecture Building Block with a full ADM mapping and a graduated, observe-before-enforce adoption path (§8–§9), it can be adopted through an organisation's existing governance, not around it.

Verified
INVARIANT SAFETY PROPERTIES
machine-checked across 2,600+ tests

Calibrated
SELECTIVE AUTONOMY
artefact-backed operating points

Auditable
REPLAYABLE DECISION ENVELOPE
chained & anchorable

> **What REMORA is.** A research-grade, enterprise-aligned governance capability that decides, before execution, whether an autonomous action may proceed, and that can prove, afterwards, exactly why it decided as it did. It is built to be observed first and enforced deliberately, to be governed as a capability rather than integrated as a tool, and to make the cost of demonstrating good governance no greater than the cost of reading a record.

The foundation is in place. The natural next step for any organisation exploring agent autonomy is the smallest possible one: replay a slice of real agent traffic in shadow mode and read, on its own data, what governed autonomy would look like.

APPENDIX

Glossary & References

Terms used in this paper, and the repository evidence behind every claim.

SELECTED TERMS

Architecture Building Block (ABB). A TOGAF construct: a generic, reusable capability defined independently of any implementation.

Conformal risk control. A calibration method that bounds the expected error rate on a selected set to a chosen target.

Credal / worst-case gate. A check that escalates when the plausible worst-case loss of an action is high and the situation is genuinely ambiguous.

Dissensus. A measure of how much independent oracles disagree; high dissensus lowers willingness to act autonomously.

Phase (ordered / critical / disordered). A characterisation of the uncertainty regime an action sits in, used to calibrate routing.

REPOSITORY EVIDENCE

`remora/policy/decision_engine.py`

`remora/governance/envelope.py`

`remora/selective/crc.py · guardrail.py`

`remora/audit/hash_chain.py · merkle.py`

`remora/shadow/replay.py`

`tests/test_policy_invariants_prop.py`

`results/selective_n500_results.json`

`results/phase_aware_guardrail_n544_results.json`

`results/toolcall_stress_replay_10000.json`

`docs/togaf/ · architecture pack (13 documents)`

NUMBERED REFERENCES

[1] DecisionEnvelope v2 schema: `remora/governance/envelope.py`, `schemas/decision_envelope_schema.yaml`.

[2] Policy invariants: `tests/test_policy_invariants_prop.py`; engine: `remora/policy/decision_engine.py`.

[3] Selective trust benchmark, N=544: `results/selective_n500_results.json` (88.8% accuracy at 18% coverage; +47.6 pp over a 41.2% majority-vote baseline).

[4] Tool-call benchmarks: `results/toolcall_benchmark_v2_results.json` (700 tasks, 0% unsafe) and `results/toolcall_stress_replay_10000.json` (10,000 calls, 52.7% → 2.8%).

[5] Architecture Building Block: `docs/togaf/architecture_building_block.md`; ADM mapping: `docs/togaf/adm_mapping.md`.

[6] Conformal risk control: `remora/selective/crc.py`.

[7] Phase-aware guardrail, N=544: `results/phase_aware_guardrail_n544_results.json` (85.0% at 22.1% coverage; 86.9% at 18.2%; 95% Wilson intervals recorded).

[8] Audit anchoring: `remora/audit/merkle.py`, `remora/audit/hash_chain.py`.

[9] Compliance mapping: `docs/togaf/compliance_assessment_template.md` (EU AI Act, ISO/IEC 42001).

PROVENANCE & CLAIM DISCIPLINE

This document follows the project's standing rule that no quantitative claim appears without a committed result artefact on disk. Benchmark figures are drawn from controlled, reproducible evaluations and are presented as evidence that the governance mechanism behaves as designed. REMORA is described as a research-grade, pilot-ready capability, not as a certified or safety-guaranteeing system.

REMORA

Developer · **Stian Skogbrott**

Repository · github.com/darklordVirtual/REMORA

Pre-Execution Governance for Autonomous AI Agents · Enterprise Reference Edition