International Journal of Core Engineering & Management
Volume-7, Issue-12, 2024 ISSN No: 2348-9510

# DETERMINISTIC GOVERNANCE HARNESSES FOR BOUNDARY-AWARE ORCHESTRATION OF ENTERPRISE AI AGENTS
## A SYSTEMATIC REVIEW AND REFERENCE ARCHITECTURE

*Shivakrishna Vangala*
*Principal Architect*
*T-Mobile USA Inc*
*Bellevue, USA*
*shivakrishna.vangala@gmail.com*

*Abstract*

***The advent of autonomous agents with the help of large language models has made it imperative to provide controls that are predictable even when the model isn't. The aim of this systematic review is to identify a core set of 30 sources from various domains related to access control, safety engineering, governance, auditing, and agent construction to arrive at a deterministic governance harness for boundary-aware orchestration. A deterministic governance harness is an enforcement layer outside of model reasoning that admits or denies every action proposed by a rule set that is fixed by the action, the acting identity, and the recorded trace. The review protocol sorts the sources into six themes, with governance the largest at 26.7 percent and agent architectures the next largest at 23.3 percent. Results from the corpus show that transparency is emphasized in 73 out of 84 analyzed guideline documents, that interleaved reasoning and acting increased absolute task success by 34 percent on one benchmark and that safety properties are a class of policies that are enforceable by runtime monitors. A five-layer reference architecture is suggested, which includes formulas for authorization, admission, composite coverage, latency, boundary compliance, and residual risk. At an automated overhead of 18 milliseconds per action, five layers give a composite coverage of 97.5 percent and a residual risk reduction of 92.8 percent, per illustrative parameters. The results show that governance principles do not come into existence unless they are reduced to the level of predicates that can be checked, and that orchestration roles, evidence of audit and escalation rules need to be articulated in addition to technical controls.***

***Index Terms—Deterministic governance, AI agents, boundary-aware orchestration, policy enforcement, role-based access control, runtime monitoring, prompt injection, AI auditing, systematic review.***

## I. INTRODUCTION

The development of reasoning, tool usage, and memory is driving the acceleration in enterprise adoption of autonomous agents based on Large Language Models (LLMs). With the delegation of control of records, payments, and communications to such agents comes the issue of how behavior will be regulated. The purpose of the deterministic governance harness is the control layer outside of the model reasoning cycle where all proposed actions are evaluated by rules whose outcome is completely determined by the action, the actor acting and the recorded trace. The classical distinction between safety and liveness properties is provided by the conceptual foundation.

Prompts cannot be used to certify probabilistic model behavior. Five problem classes of learning systems failure modes were identified, namely: unintended side effects, reward hacking, inadequate monitoring, unsafe exploration, and distributional shift [2]. It amplifies each class then there are agents connected to enterprise tools, because there are errors that are being transformed from text to operations. Boundary-aware orchestration is thus defined as the allocation of tasks to agents in explicit limits of authority, data and tools, which are verified outside the model.

The aim of this current systematic review is to pull together evidence from 30 sources and harmonise it within a product design for a harness. Three research questions are answered: What formal properties can be deterministic enforced, what governance frameworks can be operationalized as runtime controls and what are the residual risks? As in [3] the design is modular, this is encouraged by composable policy languages, such as a design that distributes independent policies into a single enforcement unit. The result is a layered reference architecture, with formulas, and a sample risk calculation included.

## II. REVIEW PROTOCOL AND CORPUS

A fixed corpus of 30 sources from 38 years of publication was chosen, and a structured review protocol was used. A screen was performed on each source to determine its relevance to enforcement, governance, safety or agent construction and only one primary theme was assigned to each source. Ethical, legal and technical opportunities were analyzed and provided as the anchor of the scoping [4]. The resulting distribution is summarized in Table I and the temporal profile is shown in Fig. 1.

TABLE I. CORPUS CLASSIFICATION BY PRIMARY THEME

| Primary theme | Sources (n) | Share (%) | Reference numbers |
|---|---|---|---|
| Governance, ethics, and principles | 8 | 26.7 | [4], [7], [8], [11], [14], [18], [25], [26] |
| Agent architectures and reasoning | 7 | 23.3 | [16], [22], [24], [27], [28], [29], [30] |
| Formal policy enforcement and access control | 5 | 16.7 | [1], [3], [6], [21], [23] |
| Auditing, documentation, and evaluation | 5 | 16.7 | [5], [13], [15], [17], [19] |
| AI and systems safety | 4 | 13.3 | [2], [10], [12], [20] |
| Security threats | 1 | 3.3 | [9] |
| **Total** | **30** | **100.0** | |

Note. Shares are computed as n divided by 30 and rounded to one decimal place.

International Journal of Core Engineering & Management
Volume-7, Issue-12, 2024 ISSN No: 2348-9510

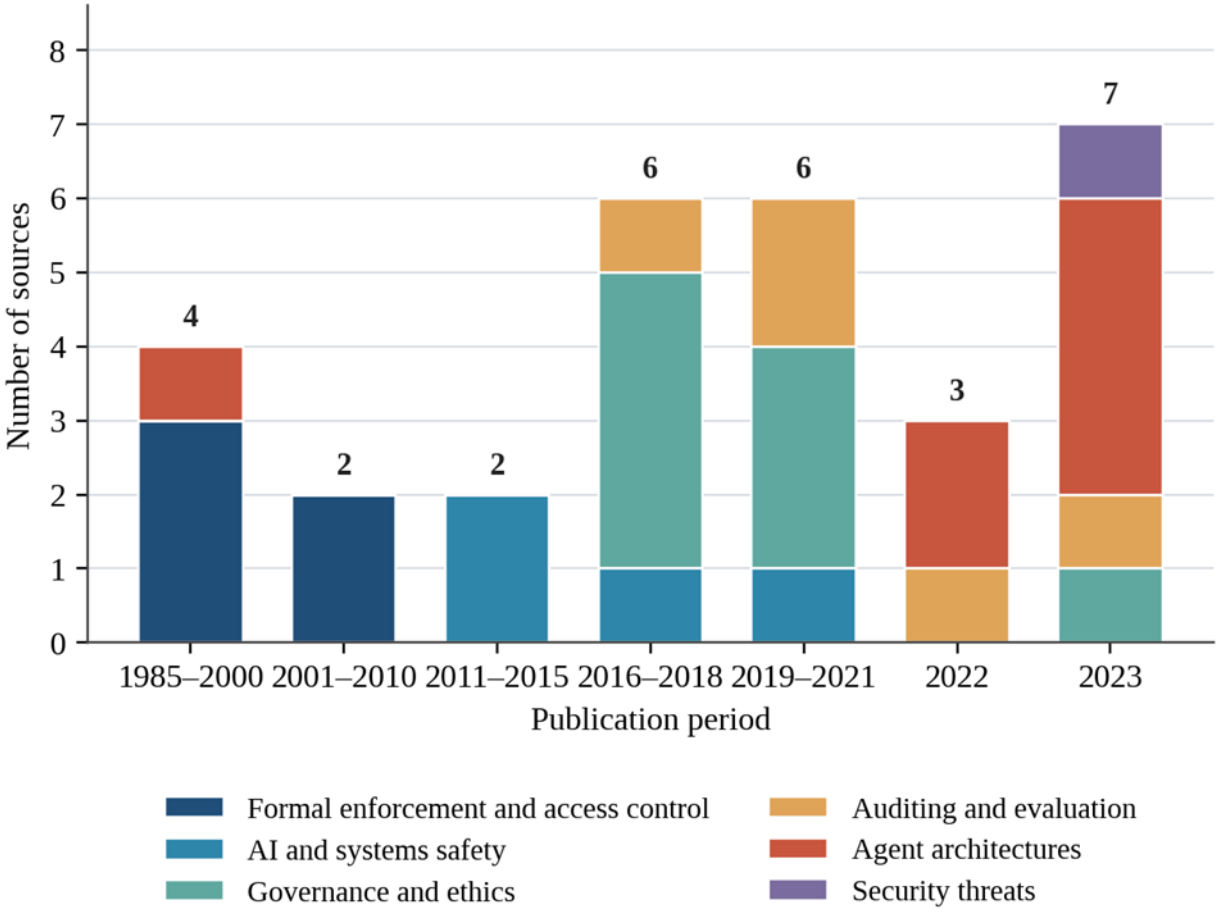



Fig. 1. Distribution of the reviewed sources by publication period and primary theme.

A three-level evaluation taxonomy that differentiates the response-based assessment, human-grounded assessment, and functionally grounded assessment [5] was used to grade the evidence. In this scheme, results of the benchmark were considered to be functionally based evidence; while audits and prescriptions in the framework were considered to be process evidence. Only those values which were explicitly stated in the sources were extracted in a quantitative way, and derived numerical values in later sections are explicitly called out as illustrative parameterizations. Equations were being used to make statements about relationships which had previously been made only qualitatively.

## III. GOVERNANCE AND ACCESS-CONTROL FOUNDATIONS

### A. Role-Based Boundaries

The key vocabulary with which the authority of agents is expressed is role-based access control (RBAC). The model was broken down into four components: core RBAC, hierarchical RBAC, static separation of duty, and dynamic separation of duty [6] based on a proposed national standard. Applied to agents, roles are task personas, permissions are tool operations on enterprise objects and separation-of-duty constraints are used to prevent a single agent from starting or approving a

International Journal of Core Engineering & Management
Volume-7, Issue-12, 2024 ISSN No: 2348-9510

sensitive transaction. The authority is thus limited, not to the text of the instructions per se, but by the declaration.

### B. Normative and Layered Governance

The principles have been reduced to five: beneficence, non-maleficence, autonomy, justice, and explicability, which are accompanied by 20 recommendations for a good AI society [7]. In a harness, these rules can't be enforced directly, and must be broken down into predicates that can be checked. Explicability, for example, is translated into a requirement for recording the rationale behind every decision, and a policy identifier; non-maleficence, for example, is translated into denial rules for high impact operations.

A governance model with layers is used that divides the social, legal, ethical and technical layers, and obligations are passed down, while evidence is worked up [8]. The top two layers are compiled to machine-evaluable rules in the technical layer. This placement helps establish responsibility: legislators establish aims, while engineering teams are responsible for rule sets and evidence stores. Misalignment between layers is thus considered as a design risk. The relationship of the foundational concepts to the harness components is presented in Table II.

TABLE II. MAPPING OF FOUNDATIONAL CONCEPTS TO HARNESS COMPONENTS

| Concept | Source | Key feature | Harness role |
|---|---|---|---|
| Safety property | [1], [23] | Violation detectable on a finite trace prefix | Runtime monitor rules |
| Liveness property | [1], [23] | Requires future events; not enforceable by monitors | Timeout conversion to safety check |
| Policy composition | [3] | Independent policies merged into one unit | Modular rule packs |
| NIST RBAC components | [6] | Four components (core, hierarchy, two separation-of-duty types) | Role, permission, and duty constraints |
| RBAC model family | [21] | Four reference models of increasing capability | Authorization core |
| Ethical principle set | [7] | Five principles; 20 recommended actions | Principle-to-predicate decomposition |
| Governance layers | [8] | Three layers: social and legal, ethical, technical | Placement of compiled rules |

Note. Source numbers refer to the reference list.

## IV. RISK LANDSCAPE AND SAFETY THEORY

Agents that fetch external content are vulnerable to indirect prompt injection, which involves injecting malicious instructions into the content that is then used as context, [9]. Demonstrations included data theft, tampering with tool calls, and polluting information ecosystems in integrated language model applications. The implication for the design of the “harness” is that retrieved text should not be used as an authority, but only as untrusted data, and that the tool invocation should be validated independently of the model-generated rationale.

A more general approach to safety was built around four problem areas: robustness, monitoring, alignment, and systemic safety [10]. There is a one-to-one correspondence between each area and a

International Journal of Core Engineering & Management
Volume-7, Issue-12, 2024 ISSN No: 2348-9510

harness function. Input validation and anomaly rejection are related to robustness; continuous trace inspection is related to monitoring; explicit objective constraints are related to alignment; and organizational controls, like escalation paths and change approval, are related to systemic safety.

84 guideline documents were reviewed to map the normative landscape, and 11 different principles were identified within these [11]. As can be seen in Fig. 2, transparency was mentioned in 73 documents, while justice and fairness were mentioned in 68, non-maleficence in 60 and responsibility in 60, and privacy in 47. In this context, transparency and non-maleficence contributes to the importance of traceability and denial controls in a first harness release.

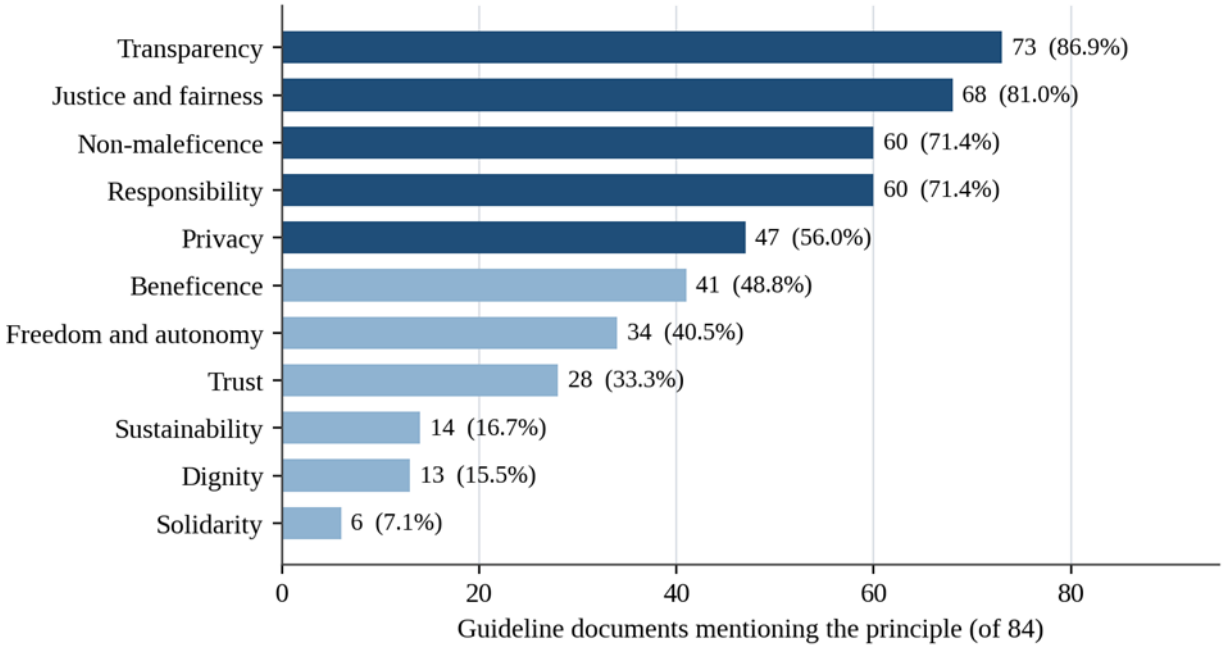



Fig. 2. Frequency of ethical principles across the 84 guideline documents analyzed in the corpus.

The systems-theoretic accident analysis (STA) approach to safety is a control problem where the constraints are introduced in a hierarchical control system with feedback [12]. According to this theory, accidents are caused by insufficient enforcement of constraints and not just component failures. The harness is thus simulated as a controller that observes the actions of agents, evaluates the observed state assuming safety constraints and issues commands to allow, modify, or deny. The loop is closed with feedback from the enterprise environment.

## V. ACCOUNTABILITY, DOCUMENTATION, AND AUDIT

Bounded deployment is supported by structured documentation. A report format, involving nine sections: intended use, evaluation factors, metrics, and caveats, was suggested for trained models [13]. The similar agent manifest is declared in the harness: recorded in a versioned document that is read by the orchestrator at start up and audited by auditors, the declared role, allowed tools, data classifications, escalation contacts and known limitations are documented.

The principles were claimed not to be enough, because four structural flaws set AI apart from traditional professions: They lack shared goals and obligations, lack professional standards, lack

International Journal of Core Engineering & Management
Volume-7, Issue-12, 2024 ISSN No: 2348-9510

methods that allow principles to be operationalized, and lack accountability mechanisms [14]. The third deficiency is resolved by the harness, which is capable of being implemented as a translation mechanism; the fourth is resolved by the immutability of the decision logs, which can then be used to make later claims of accountability.

The auditing structure of the language-model systems was organized in three levels: governance audits of the language-model providing organisation, language-model capability audits before language-model release and application audits when using the language-models downstream. [15]. The logs generated from enterprise agent harnesses are mainly used at the application layer, to create a log of each action sequence for auditing purposes. This structure is compared to other assurance frameworks in the corpus in Table III.

TABLE III. ASSURANCE FRAMEWORKS ALIGNED WITH HARNESS STAGES

| Framework | Source | Structure | Harness counterpart |
|---|---|---|---|
| AI risk management framework | [26] | Four functions: Govern, Map, Measure, Manage | Policy charter, boundary register, conformance metrics, incident procedure |
| Internal algorithmic audit | [19] | Five stages: scoping, mapping, artifact collection, testing, reflection | Boundary declaration, threat identification, log retention, adversarial testing, rule revision |
| Three-layered LLM audit | [15] | Three layers: governance, model, application | Application-layer evidence from decision logs |
| Layered AI governance | [8] | Three layers: social and legal, ethical, technical | Compilation of obligations into technical rules |
| Human-centered guidelines | [25] | 15 recommendations at team, organization, and industry levels | Rule sets, change control, certification evidence |
| Systems-theoretic safety | [12] | Control structure with constraints and feedback | Controller, constraint set, environment feedback |

Note. Source numbers refer to the reference list.

## VI. AGENT BEHAVIOR, RED TEAMING, AND OVERSIGHT

A memory stream, reflection, and planning were combined in a system of 25 generative agents which behaved believably in a sandbox community [16]. In multi-agent orchestration, setting clear boundaries is necessary because agents will coordinate in an emergent way, such as in information diffusion and event organization. This results in unconstrained memory and social exchange, and memory writes and messages are thus subject to policy limitations. Capability reported in the literature for the contemporary techniques are compared in Table IV and Fig. 3.

International Journal of Core Engineering & Management
Volume-7, Issue-12, 2024 ISSN No: 2348-9510

TABLE IV. REPORTED CAPABILITY METRICS OF AGENT TECHNIQUES AND GOVERNANCE IMPLICATIONS

| Technique | Source | Reported figure | Governance implication |
|---|---|---|---|
| Interleaved reasoning and acting | [30] | +34% (ALFWorld) and +10% (WebShop) absolute success | Higher action volume; every step gated |
| Verbal self-reflection | [24] | 91% pass@1 versus 80% baseline (HumanEval) | Regenerated actions re-evaluated |
| Self-taught tool use | [22] | 6.7 billion parameters; 5 tools | Gateway for all API calls |
| Chain-of-thought prompting | [28] | 540 billion parameters; 8 exemplars; leading GSM8K result | Traces stored as evidence only |
| Generative agents | [16] | 25 agents; memory, reflection, planning | Memory and message limits |
| Automated red teaming | [17] | 280 billion-parameter target; 4 generation methods | Regression suite for rule changes |

Note. Figures are those reported in the cited sources; benchmarks differ and are not directly comparable.

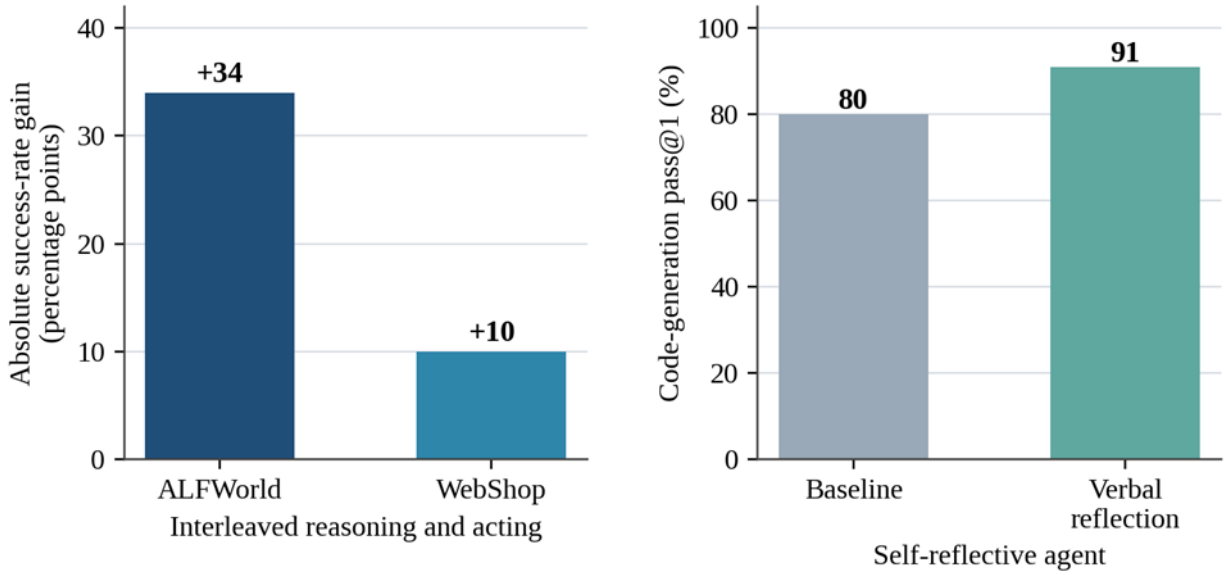



Fig. 3. Reported performance gains of interleaved reasoning and acting (left) and of verbal self-reflection (right).

Manual testing is not feasible in the input space of conversational systems. One language model was used to write the test cases for the other to elicit harmful behaviour, in both the zero-shot generation and few-shot generation approaches, and in supervised learning and reinforcement learning approaches, where the 280-billion-parameter dialogue model was used as the target [17]. Generated attack suites are added to the harness regression pipeline to ensure that any rule change is tested on the built up set of adversarial cases.

The idea of society-in-the-loop generalised human oversight, with the provision of an explicit, negotiated social contract among stakeholders, as opposed to a single operator to oversee the algorithmic systems. For enterprise agents the concept suggests that the interests of the customers, employees, and regulators form part of the escalation rules and approval thresholds as well as

International Journal of Core Engineering & Management
Volume-7, Issue-12, 2024 ISSN No: 2348-9510

those of the business unit deploying it. Human approval is thus treated as a rule result instead of an informal practice.

It was defined an end-to-end internal auditing framework consisting of five phases: scoping, mapping, artifact collection, testing, and reflection [19]. These phases are also the same for the harness life cycle. Each of these items relates to the scope, mapping, artifact collection, testing, and reflection, respectively.These are the scope, mapping, artifact collection, threat identification, and threat assessment, or conformance suites, and rule revision after incidents.

Focusing on the importance of robust and beneficial systems, research priorities were aggregated under the themes of verification, validity, security and control [20]. Verification is the closest to deterministic enforcement because it is formal checks that establish that a specified property holds. Validity and control broaden the field to the assurance of appropriate specifications and of human operator's ability to interrupt operation. An interruption primitive is therefore tested before each operation, and is called a kill switch.

## VII. SYNTHESIZED HARNESS ARCHITECTURE

### A. Authorization and Admission Rules

A family of four reference models of role-based access control (RBAC) was formalized—the base model, the model with role hierarchies, the model with constraints, and the consolidated model [21]. The consolidated model is used as the model of the authorization core of the harness. The authorization predicate used in Eq. 1 defines the relation: acting agent u, activated role set Ru, operation and object (op, obj), and permission set of role r, Pr.

$$\mathrm{Auth}(u, a) = 1 \Leftrightarrow (op, obj) \in \bigcup_{r \in R_u} P_r \tag{1}$$

The interfaces for question answering, calculation, search, translation and a calendar were a language model of 6.7 billion parameters that learns to determine which interface to call, when, and with which arguments [22]. With this approach the selection of the tool is thus effected by a probabilistic element, which creates the need for a gateway between the model and the tool. Before any enterprise operation is run, each call is validated by the gateway against a declared schema and the authorization predicate.

It was considered that policies which could be written as safety properties were enforceable policies, as they could be detected by the monitor that observes each action, sequentially, and is violated at a finite point of an execution.It was considered that Enforceable Policies are those that are expressible as safety properties, which are violated at a finite point of an execution and thus can be detected by a monitor that observes actions one by one [23]. A monitor cannot enforce properties that depend on future events, called liveness properties. Thus, an action is admitted if and only if every m constraint, represented by a binary predicate φi, evaluated on the trace σ extended by the action a, is satisfied (as in Eq. 2). The prefix-closure condition for a property P is defined in Eq. 3, where τ is any extension of the trace. Eventual approvals are replaced with safety checks via timeouts.

$$\Pi(\sigma, a) = \prod_{i=1}^{m} \varphi_i(\sigma \cdot a), \quad \mathrm{Admit}(\sigma, a) \Leftrightarrow \Pi(\sigma, a) = 1 \tag{2}$$

$$\forall \sigma: \quad \neg P(\sigma) \Rightarrow \forall \tau: \quad \neg P(\sigma\tau) \tag{3}$$

International Journal of Core Engineering & Management
Volume-7, Issue-12, 2024 ISSN No: 2348-9510

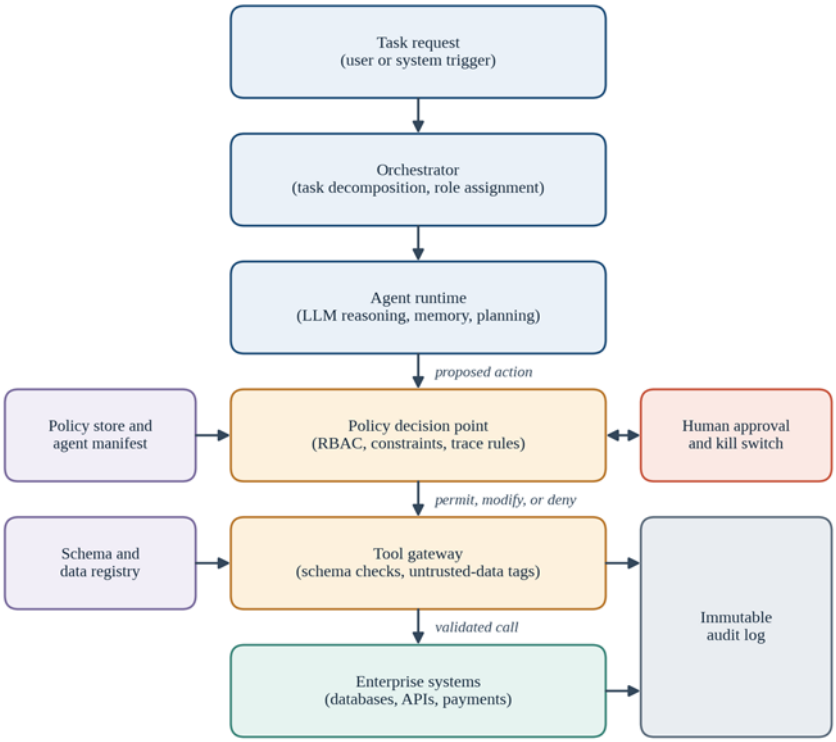



Fig. 4. Reference architecture of the deterministic governance harness, showing the enforcement path and supporting modules.

An approach that uses verbal self-reflection stored in memory for subsequent attempts was found to increase pass@1 accuracy on a code-generation benchmark from 80 percent to 91 percent, thus learning from failure [24]. Reflection loops enhance task quality but also create additional actions and every newly generated action is again passed to the same gate. The model can then be adapted freely whilst the boundary is held constant. The five control layers are summarized in Tab. V, composite coverage over L layers with individual coverage κj is expressed in Eq. (4), and cumulative automated latency with per-layer delay tj is expressed in Eq. (5).

$$\kappa_c = 1 - \prod_{j=1}^{L}(1 - \kappa_j) \quad (4)$$

$$T_h = \sum_{j=1}^{L} t_j \quad (5)$$

International Journal of Core Engineering & Management
Volume-7, Issue-12, 2024 ISSN No: 2348-9510

TABLE V. HARNESS CONTROL LAYERS WITH ILLUSTRATIVE PARAMETERS

| Layer | Control function | Property class | Assumed coverage κ | Assumed latency (ms) |
|---|---|---|---|---|
| L1 Identity and role authorization | RBAC core; separation of duty | Safety | 0.60 | 2 |
| L2 Schema and parameter validation | Typed arguments; value ranges | Safety | 0.50 | 3 |
| L3 Content isolation | Untrusted-data tagging; instruction and data separation | Safety | 0.40 | 5 |
| L4 Trace monitor | Finite-prefix rules; rate and step limits | Safety | 0.70 | 8 |
| L5 Escalation and interruption | Approval thresholds; kill switch | Liveness converted by timeout | 0.30 | Asynchronous |
| **Composite (Eq. 4 and Eq. 5)** | **All five layers** | | **0.975** | **18** |

Note. Coverage and latency values are assumed for demonstration and are not empirical measurements from the corpus.

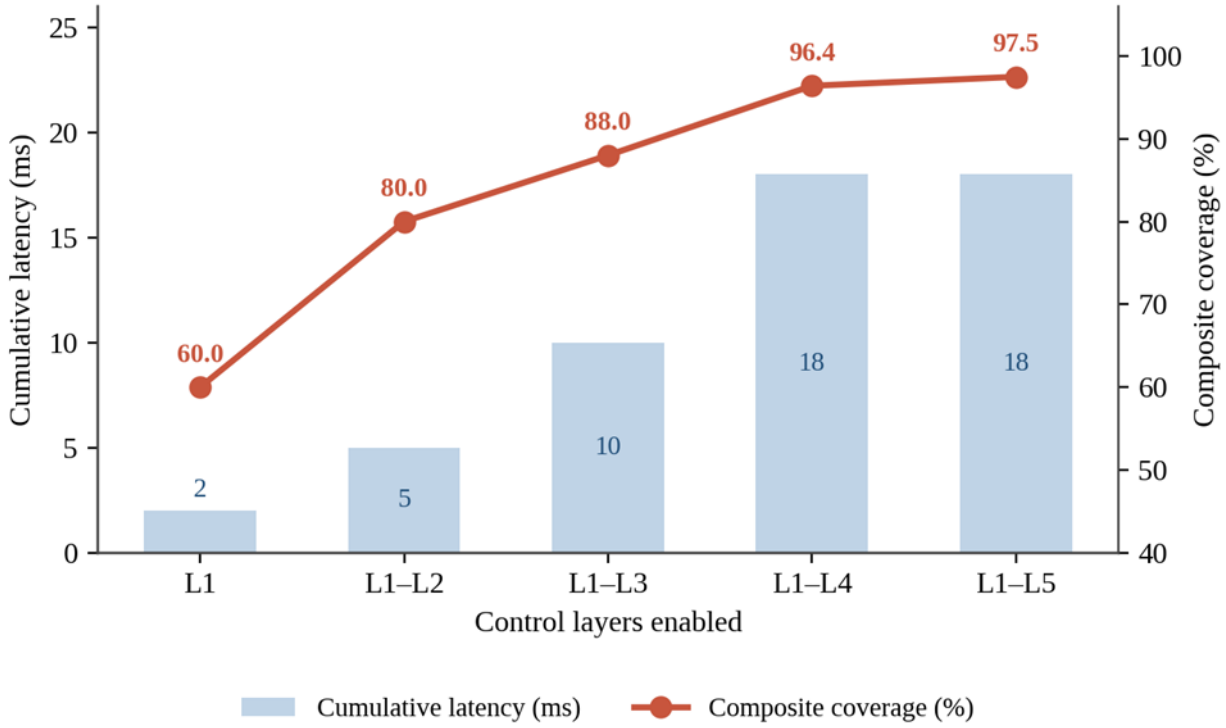



Fig. 5. Composite coverage and cumulative latency as control layers are added, under the assumed parameters of Table V.

International Journal of Core Engineering & Management
Volume-7, Issue-12, 2024 ISSN No: 2348-9510

## VIII. ORGANIZATIONAL ORCHESTRATION AND EVALUATION

The recommendations were grouped into three levels: effective engineering practice in teams, safety culture in organizations and trustworthy certification in industries [25]. The first is implemented through a testable set of rules in the harness, the second is facilitated through incident review, change control, and the third is drafted with evidence that will be acceptable for external inspection. There is a separation of the orchestration roles (policy owner, rule author, independent reviewer) to prevent a single person from having too much authority.

A national risk framework outlines activities under four functions: Govern, Map, Measure and Manage, and seven characteristics of trustworthy systems: valid and reliable, safe, secure and resilient, accountable and transparent, explainable and interpretable, privacy-enhanced, and fair with harmful bias managed [26]. A harness for each function is identified. The boundary compliance rate is used to quantify the conformance, as shown in Eq. 6, where V and N represent the number of boundary violations and the number of evaluated actions, respectively. The impact, frequency and coverage of an incident to a threat class k are represented by Ik, λk and κk, respectively, and are used to quantify the residual risk, as shown in Eq. 7.

$$\mathrm{BCR} = 1 - \frac{V}{N} \tag{6}$$

$$R_{\mathrm{res}} = \sum_{k=1}^{K} \lambda_k I_k (1 - \kappa_k) \tag{7}$$

A unified framework in autonomous agents with a language model based survey comprised four modules, profile, memory, planning, and action [27]. The modules establish natural points of enforcement. The role assignment is determined by the profile, memory is retained and classified according to the profile, planning outputs are filtered according to forbidden goals, and action outputs go through the gateway. When mapping controls to modules it is guaranteed that no capability is added without a corresponding constraint.

It was demonstrated that intermediate reasoning steps could be elicited to boost arithmetic and symbolic reasoning, with a model with only eight examples that, when prompted by the same examples, achieved a leading score on a grade-school mathematics benchmark [28] with 540 billion parameters. Reasoning traces are helpful when reviewing but not always faithful explanations. Traces are therefore recorded and stored as evidence and for triage, while authorization decisions are never made based on evidence from a trace.

Other classical definitions of an agent involve autonomy, social ability, reactivity and pro-activeness [29]. Each property comes with its own governance requirement. Message and delegation rules are necessary for social ability, explicit limits on unsupervised steps are necessary for autonomy, input validation is necessary for reactivity, and goal screening is necessary for pro-activeness. The simplest control for the first property is recommended to be a maximum on consecutive autonomous steps, set by risk tier.

Relative to imitation and reinforcement learning baselines, interleaving reasoning and acting resulted in absolute success rates increasing by 34 percent on an interactive household benchmark and 10 percent on a web-shopping benchmark [30]. These gains can lead to more actions that need reviewing and more impact. For five threat levels the illustrative residual risk values, calculated using Eq. 7, are reported in Table VI and plotted in Fig. 6, showing that the aggregated expected loss is reduced by 92.8 percent, with the assumed parameters.

International Journal of Core Engineering & Management
Volume-7, Issue-12, 2024 ISSN No: 2348-9510

TABLE VI. ILLUSTRATIVE RESIDUAL-RISK CALCULATION BY THREAT CLASS

| Threat class | λ (per 1,000 tasks) | I (1–10) | κ | Loss without harness | Loss with harness |
|---|---|---|---|---|---|
| Indirect prompt injection | 12 | 8 | 0.92 | 96.0 | 7.7 |
| Unauthorized tool invocation | 9 | 9 | 0.97 | 81.0 | 2.4 |
| Excessive autonomy | 7 | 6 | 0.90 | 42.0 | 4.2 |
| Sensitive data disclosure | 10 | 9 | 0.94 | 90.0 | 5.4 |
| Unsafe irreversible side effects | 5 | 10 | 0.88 | 50.0 | 6.0 |
| **Total** | | | | **359.0** | **25.7** |

Note. Parameters are assumed for demonstration. Loss is computed as λ × I without the harness and λ × I × (1 − κ) with the harness; the aggregate reduction is 92.8 percent.

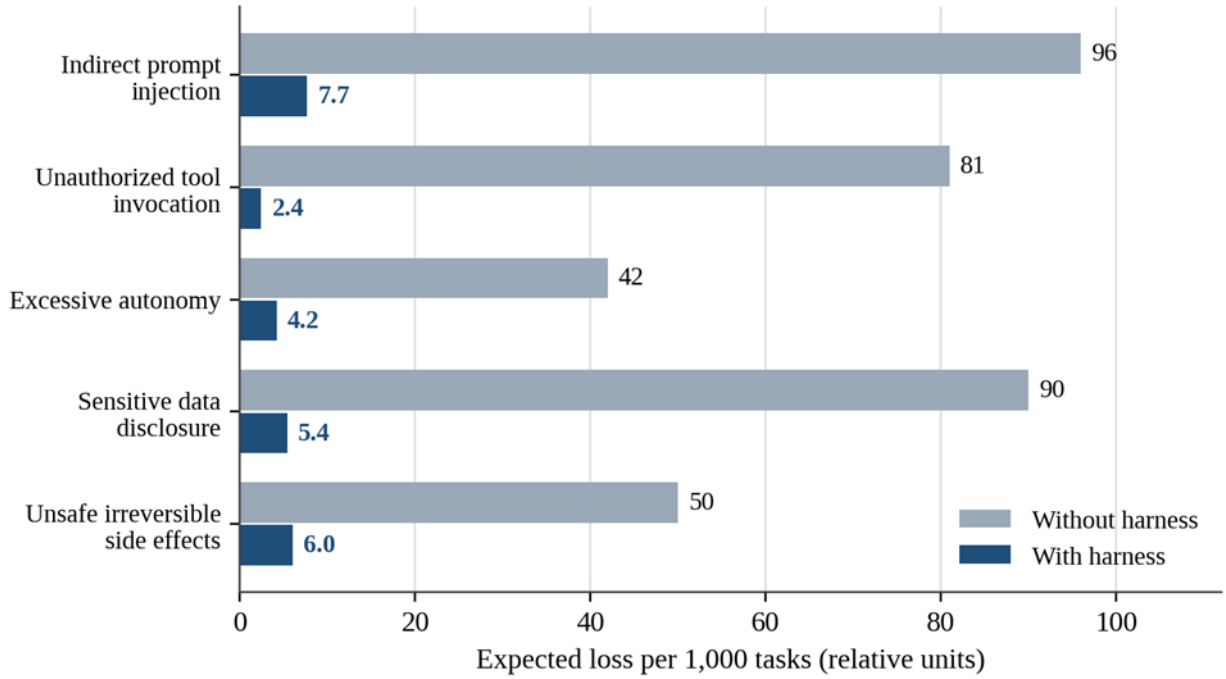



Fig. 6. Expected loss per 1,000 tasks by threat class with and without the harness, under the assumed parameters of Table VI.

## IX. CONCLUSION AND FUTURE DIRECTIONS

The review shows that a deterministic governance harness can be put together using time-tested access control and runtime enforcement principles, and some newer practices in auditing, documentation and risk management. There are three conclusions supported. First, only safety properties are enforceable via monitoring, and obligations have to be turned into prohibitions on finite traces. Secondly, operative principles can only be manifested through predicates, evidence stores and assigned owners. Third, their coverage layer multiple, and the illustrative parameters give a composite coverage of 97.5 percent at 18 milliseconds from an automated overhead.

A number of restrictions qualify these conclusions. The corpus is finite and heterogeneous, values of quantities are taken from different scales, and the worked example is based on assumed parameters and assumes independence of layers that may not exist in real-world situations. Empirical rule coverage and latency measurement in actual production orchestration is directed by future work, as is the formal handling of correlated layer failures, compositional verification of multi-agent delegation and the extension of documentation practice to tool-using systems with standardized agent manifests.

## REFERENCES

1. B. Alpern and F. B. Schneider, "Defining liveness," Information Processing Letters, vol. 21, no. 4, pp. 181–185, 1985. https://doi.org/10.1016/0020-0190(85)90056-0
2. D. Amodei, C. Olah, J. Steinhardt, P. Christiano, J. Schulman, and D. Mané, "Concrete problems in AI safety," arXiv, 2016. https://doi.org/10.48550/arXiv.1606.06565
3. L. Bauer, J. Ligatti, and D. Walker, "Composing security policies with polymer," International Journal of Information Security, vol. 4, no. 1–2, pp. 2–16, 2005. https://doi.org/10.1007/s10207-004-0046-8
4. C. Cath, "Governing artificial intelligence: Ethical, legal and technical opportunities and challenges," Philosophical Transactions of the Royal Society A, vol. 376, no. 2133, Article 20180080, 2018. https://doi.org/10.1098/rsta.2018.0080
5. F. Doshi-Velez and B. Kim, "Towards a rigorous science of interpretable machine learning," arXiv, 2017. https://doi.org/10.48550/arXiv.1702.08608
6. D. F. Ferraiolo, R. Sandhu, S. Gavrila, D. R. Kuhn, and R. Chandramouli, "Proposed NIST standard for role-based access control," ACM Transactions on Information and System Security, vol. 4, no. 3, pp. 224–274, 2001. https://doi.org/10.1145/501978.501980
7. L. Floridi, J. Cowls, M. Beltrametti, R. Chatila, P. Chazerand, V. Dignum, C. Luetge, R. Madelin, U. Pagallo, F. Rossi, B. Schafer, P. Valcke, and E. Vayena, "AI4People—An ethical framework for a good AI society: Opportunities, risks, principles, and recommendations," Minds and Machines, vol. 28, no. 4, pp. 689–707, 2018. https://doi.org/10.1007/s11023-018-9482-5
8. U. Gasser and V. A. F. Almeida, "A layered model for AI governance," IEEE Internet Computing, vol. 21, no. 6, pp. 58–62, 2017. https://doi.org/10.1109/MIC.2017.4180835
9. K. Greshake, S. Abdelnabi, S. Mishra, C. Endres, T. Holz, and M. Fritz, "Not what you've signed up for: Compromising real-world LLM-integrated applications with indirect prompt injection," in Proceedings of the 16th ACM Workshop on Artificial Intelligence and Security, pp. 79–90. Association for Computing Machinery, 2023. https://doi.org/10.1145/3605764.3623985
10. D. Hendrycks, N. Carlini, J. Schulman, and J. Steinhardt, "Unsolved problems in ML safety," arXiv, 2021. https://doi.org/10.48550/arXiv.2109.13916
11. A. Jobin, M. Ienca, and E. Vayena, "The global landscape of AI ethics guidelines," Nature Machine Intelligence, vol. 1, no. 9, pp. 389–399, 2019. https://doi.org/10.1038/s42256-019-0088-2

12. N. G. Leveson, Engineering a safer world: Systems thinking applied to safety. MIT Press, 2011. https://doi.org/10.7551/mitpress/8179.001.0001
13. M. Mitchell, S. Wu, A. Zaldivar, P. Barnes, L. Vasserman, B. Hutchinson, E. Spitzer, I. D. Raji, and T. Gebru, "Model cards for model reporting," in Proceedings of the Conference on Fairness, Accountability, and Transparency, pp. 220–229. Association for Computing Machinery, 2019. https://doi.org/10.1145/3287560.3287596
14. B. Mittelstadt, "Principles alone cannot guarantee ethical AI," Nature Machine Intelligence, vol. 1, no. 11, pp. 501–507, 2019. https://doi.org/10.1038/s42256-019-0114-4
15. J. Mökander, J. Schuett, H. R. Kirk, and L. Floridi, "Auditing large language models: A three-layered approach," AI and Ethics. Advance online publication, 2023. https://doi.org/10.1007/s43681-023-00289-2
16. J. S. Park, J. C. O'Brien, C. J. Cai, M. R. Morris, P. Liang, and M. S. Bernstein, "Generative agents: Interactive simulacra of human behavior," in Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology, Article 2. Association for Computing Machinery, 2023. https://doi.org/10.1145/3586183.3606763
17. E. Perez, S. Huang, F. Song, T. Cai, R. Ring, J. Aslanides, A. Glaese, N. McAleese, and G. Irving, "Red teaming language models with language models," arXiv, 2022. https://doi.org/10.48550/arXiv.2202.03286
18. Rahwan, "Society-in-the-loop: Programming the algorithmic social contract," Ethics and Information Technology, vol. 20, no. 1, pp. 5–14, 2018. https://doi.org/10.1007/s10676-017-9430-8
19. D. Raji, A. Smart, R. N. White, M. Mitchell, T. Gebru, B. Hutchinson, J. Smith-Loud, D. Theron, and P. Barnes, "Closing the AI accountability gap: Defining an end-to-end framework for internal algorithmic auditing," in Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency, pp. 33–44. Association for Computing Machinery, 2020. https://doi.org/10.1145/3351095.3372873
20. S. Russell, D. Dewey, and M. Tegmark, "Research priorities for robust and beneficial artificial intelligence," AI Magazine, vol. 36, no. 4, pp. 105–114, 2015. https://doi.org/10.1609/aimag.v36i4.2577
21. R. S. Sandhu, E. J. Coyne, H. L. Feinstein, and C. E. Youman, "Role-based access control models," Computer, vol. 29, no. 2, pp. 38–47, 1996. https://doi.org/10.1109/2.485845
22. T. Schick, J. Dwivedi-Yu, R. Dessì, R. Raileanu, M. Lomeli, L. Zettlemoyer, N. Cancedda, and T. Scialom, "Toolformer: Language models can teach themselves to use tools," arXiv, 2023. https://doi.org/10.48550/arXiv.2302.04761
23. F. B. Schneider, "Enforceable security policies," ACM Transactions on Information and System Security, vol. 3, no. 1, pp. 30–50, 2000. https://doi.org/10.1145/353323.353382
24. N. Shinn, F. Cassano, E. Berman, A. Gopinath, K. Narasimhan, and S. Yao, "Reflexion: Language agents with verbal reinforcement learning," arXiv, 2023. https://doi.org/10.48550/arXiv.2303.11366
25. B. Shneiderman, "Bridging the gap between ethics and practice: Guidelines for reliable, safe, and trustworthy human-centered AI systems," ACM Transactions on Interactive Intelligent Systems, vol. 10, no. 4, Article 26, 2020. https://doi.org/10.1145/3419764

26. E. Tabassi, "Artificial intelligence risk management framework (AI RMF 1.0) (NIST AI 100-1)," National Institute of Standards and Technology, 2023. https://doi.org/10.6028/NIST.AI.100-1
27. L. Wang, C. Ma, X. Feng, Z. Zhang, H. Yang, J. Zhang, Z. Chen, J. Tang, X. Chen, Y. Lin, W. X. Zhao, Z. Wei, and J.-R. Wen, "A survey on large language model based autonomous agents," arXiv, 2023. https://doi.org/10.48550/arXiv.2308.11432
28. J. Wei, X. Wang, D. Schuurmans, M. Bosma, B. Ichter, F. Xia, E. Chi, Q. Le, and D. Zhou, "Chain-of-thought prompting elicits reasoning in large language models," arXiv, 2022. https://doi.org/10.48550/arXiv.2201.11903
29. M. Wooldridge and N. R. Jennings, "Intelligent agents: Theory and practice," The Knowledge Engineering Review, vol. 10, no. 2, pp. 115–152, 1995. https://doi.org/10.1017/S0269888900008122
30. S. Yao, J. Zhao, D. Yu, N. Du, I. Shafran, K. Narasimhan, and Y. Cao, "ReAct: Synergizing reasoning and acting in language models," arXiv, 2022. https://doi.org/10.48550/arXiv.2210.03629