Latency-Aware Authentication in Industrial Control Environments

Consequence-Graded Authorization Beyond the Web Latency Budget

By Jorge Enrique Flores Montano

Founder, JM Automated Solutions

~MILO™ — Modular Intelligent Learning Orchestrator (patent pending)

May 2026

ORCID iD: 0009-0003-1859-8418

DOI: 10.5281/zenodo.20117651

Public reference: https://github.com/jmontano1/milo-architecture

Abstract

Authentication patterns developed for web and cloud environments — OAuth round-trips, TLS handshakes, multi-factor authentication challenges — often assume tens to hundreds of milliseconds of permissible latency per authorization event. Industrial control environments operate on fundamentally tighter budgets. Machine vision inspection loops can run on single-millisecond timescales; programmable logic controller (PLC) scan cycles are measured in milliseconds; realtime motion-control decisions may admit no perceptible authentication overhead at all. The result is a recurring industrial-control gap: authentication is either applied at the perimeter and absent inside the control loop, or it is grafted on with timing penalties that operators and integrators are incentivized to bypass. This paper proposes*latency-aware authentication* as an adaptive design discipline for industrial control environments: authentication strength is graded against operational consequence per control cycle rather than applied uniformly. The discipline complements the attacker-class security level tiers (SL1–SL4) of ISA/IEC 62443 [1] with an orthogonal consequence-class axis, and operates within the operational technology security framing of NIST SP 800-82r3 [2]. The paper is grounded in the author's hands-on industrial vision deployment experience across food, beverage, pharmaceutical, and medical-device manufacturing domains, and illustrated using MILO, a patent-pending adaptive AI orchestrator[3] whose pre-execution gating subsystem implements the discipline in software.

Keywords: industrial control, operational technology, latency-aware authentication, ISA/IEC 62443, NIST SP 800-82, adaptive authentication, pre-execution gating.

Highlights.

- Proposes*latency-aware authentication* as an adaptive design discipline for industrial control: authentication strength graded against operational consequence per control cycle rather than uniformly.
- Composes orthogonally with the attacker-class Security Level tiers (SL1–SL4) of ISA/IEC 62443, introducing a consequence-class axis aligned with NIST SP 800-82r3's operational technology framing.

- Specifies five architectural commitments — including pre-execution gating with three outputs (allow / hold-block / recommend) and persist-before-deliver audit across all consequence tiers.
- Grounded in seven years of industrial-vision deployment experience across food, beverage, pharmaceutical, and medical-device manufacturing.

Index Terms: industrial control, operational technology security, latency-aware authentication, ISA/IEC 62443, NIST SP 800-82, NIST SP 800-207, adaptive authentication, pre-execution gating, Zero Trust architecture, critical manufacturing.

Plain Language Summary. Authentication methods designed for the web — typing a password, getting a text-message code, completing an OAuth handshake — assume the user has hundreds of milliseconds to a few seconds to respond. Industrial control systems running manufacturing lines, power grids, and chemical plants do not have that budget; their decisions happen in milliseconds. As a result, authentication is often pushed to the perimeter and then absent inside the control loop, or grafted on with timing penalties that operators are pressured to bypass. This paper proposes grading the strength of authentication against the operational consequence of each individual control command, so that high-consequence actions get deliberate human authorization while routine actions pass through a lightweight log. The approach composes with existing industrial cybersecurity standards rather than replacing them.

Relevance to U.S. National Interest. Critical-manufacturing cybersecurity is a designated U.S. national-interest sector under CISA's Critical Manufacturing Sector framework. Industrial control authentication is identified by NIST SP 800-82r3 as a gap requiring sector-specific guidance. The discipline proposed here addresses that gap directly and is aligned with the DOE Genesis Mission's advanced-manufacturing and grid-reliability priorities.

Status of claims. The framework proposed here — *latency-aware authentication graded by operational consequence*, composed orthogonally with the attacker-class Security Level tiers of ISA/IEC 62443 [1] — is the author's original architectural discipline; the composition is described as a proposal, not as an existing certification. The latency budgets in §3 are illustrative ranges drawn from generic industry references; specific deployments vary by application. The compatibility argument with NIST SP 800-82r3 [2], NIST SP 800-207 [6], and NIST AI 100-1 [11] reflects the author's reading of those documents and is design intent, not a compliance attestation. The author's industrial-vision deployment context grounds the framework in operational reality but does not constitute external validation; empirical evaluation against an instrumented industrial-control deployment is forthcoming work. This manuscript is a preprint prior to peer review.

About MILO. MILO (Modular Intelligent Learning Orchestrator) is a patent-pending adaptive AI orchestration architecture for high-consequence critical-infrastructure environments. The full architectural reference — eight structural principles, eight operational integrity constraints, the unifying viability principle, and the trademark/patent status — is maintained as a single canonical document at https://github.com/jmontano1/milo-architecture (concept DOI: 10.5281/zenodo.20117025). Author contact: jmontano@jmautomated.com.

1. Introduction

The latency budget of the modern industrial control loop is set by physics, not by security policy. A high-speed machine vision system inspecting items on a conveyor at sixty items per second admits roughly sixteen milliseconds per item, of which the camera exposure, image acquisition, classification, and reject-arm signal must all complete inside the deterministic budget. A PLC executing a safety-relevant logical operation typically operates on a scan cycle of one to ten milliseconds; a missed scan is a deviation event that may trigger a corrective action program entry. A robotic motion controller maintaining position accuracy at the micrometer scale runs control loops at one kilohertz or higher, with each loop admitting no more than tens of microseconds of computation. Authentication patterns from the web stack — designed for human-perceptible interactions with response budgets between one hundred milliseconds and two seconds — cannot be inserted into these loops without violating the operational physics.

The standard industry response has been to push authentication to the perimeter. Operators authenticate at the human-machine interface (HMI), the perimeter accepts or rejects the operator, and authenticated sessions then issue uncredentialed control commands inside the control loop. The pattern works while the perimeter is intact and the session is trusted. It does not work when the perimeter is breached, when sessions are hijacked, when the operator's credential set has changed during the session lifetime, or when the operational consequence of a single control command warrants finer-grained authorization than session-level trust provides. Recent operational technology cybersecurity incidents — including high-profile ransomware events in critical infrastructure — have made plain that session-level perimeter authentication is insufficient for high-consequence control loops.

The standard alternative — grafting full authentication onto each control command — fails on the other axis. The latency penalty of a per-command OAuth round trip or multi-factor challenge can exceed the operational budget by one to three orders of magnitude. The overhead is then likely to be bypassed, moved out of the loop, or disabled during commissioning; the security control becomes vestigial.

The present paper proposes a third path. *Latency-aware authentication* grades authentication strength against the operational consequence of the specific control cycle, rather than applying a uniform authentication level across all commands or applying authentication only at session establishment. A motion-control command that maintains tracking under normal operation passes through a lightweight pre-execution gate. A motion-control command that crosses a safety-relevant threshold — exceeding a force limit, leaving a confined operational envelope, or commanding a state transition with downstream production consequence — passes through a heavier gate that admits the human operator's authorization with the latency budget of human decision-making, deliberately. The discipline is not "fast authentication" or "slow authentication"; it is *authentication graded by consequence*.

2. Background and Related Work

Operational technology security standards. NIST SP 800-82r3 [2], published in September 2023, is the canonical guide to operational technology (OT) security. It specifies that OT systems differ from conventional IT systems in their prioritization of availability, integrity, and safety, and that security controls for OT must accommodate the deterministic real-time requirements of industrial control. The document does not provide a specific framework for latency-aware authentication; the gap addressed by the present paper is identified by NIST SP 800-82r3 as an active area requiring sector-specific guidance.

ISA/IEC 62443 Security Levels. The ISA/IEC 62443 series[1] provides the international framework for industrial cybersecurity, structured around four Security Levels (SL1 through SL4) ranging from protection against unintentional or accidental misuse (SL1) to protection against in-

tentional misuse using sophisticated means with extensive resources and motivation (SL4). The Security Levels are *attacker-class tiers*: they grade the security control by the capability of the adversary the control is designed to resist. The present paper's contribution is an *orthogonal consequence-class axis*: graded by the operational consequence of the protected command rather than the capability of the adversary. The two axes compose: a command at SL3 attacker-class with high operational consequence requires both strong cryptographic credentials and a deliberate human authorization gate; a command at SL3 attacker-class with low operational consequence requires the cryptographic credentials and a lightweight pre-execution log. The composition is the contribution.

Adaptive and risk-based authentication. The adaptive-authentication literature[4], [5] develops the principle of grading authentication challenge against contextual risk factors — device, geolocation, time, behavioral biometrics. The literature has primarily addressed the consumer cloud and enterprise IT context, where contextual factors are abundant and the latency budget is human-perceptible. The framework of Adaptive Context-Aware Security (ACASF)[5] is a representative recent example. The present paper extends the principle to industrial control by substituting *operational consequence per control cycle* for *user context* as the grading axis, and by operating within latency budgets that the IT-context literature does not address.

Zero Trust architecture in OT. Zero Trust architecture, as formalized by NIST SP 800-207 [6], requires continuous evaluation of access rather than perimeter-only trust. Applying that principle in OT is difficult because control-loop latency budgets are materially different from enterprise IT latency budgets. The present paper is consistent with Zero Trust framing: the consequence-graded pre-execution gate is the mechanism by which continuous verification becomes operationally compatible with the OT latency budget.

3. Operational Latency Budgets in Industrial Domains

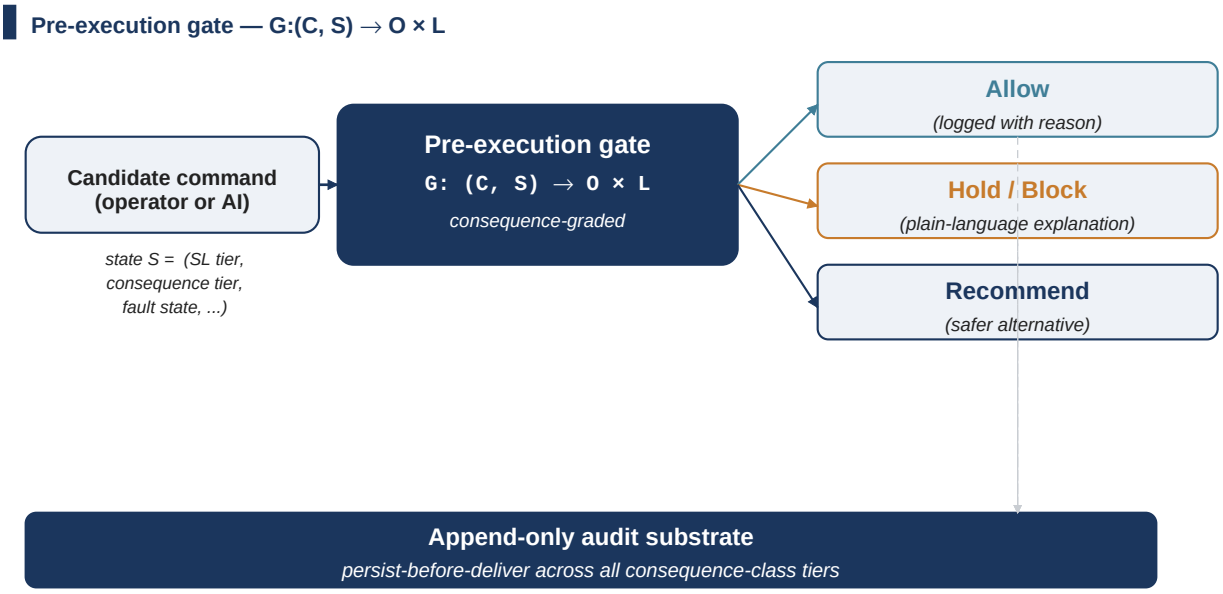
The discipline of latency-aware authentication is grounded in the actual latency budgets observed in industrial deployment. The figures here are illustrative and drawn from generic industry references; specific deployments vary by application.

Industrial Domain	Typical Latency Budget	Authentication Compatibility
Motion control (servo loops)	100 μs – 1 ms per loop	No authentication compatible per loop; session-level only with structural override
PLC scan cycle (logic execution)	1 – 10 ms per scan	Lightweight pre-execution log compatible; cryptographic handshake incompatible
Machine vision inspection (high-speed line)	5 – 50 ms per item	Lightweight gate compatible; full MFA incompatible
HMI operator interaction	100 – 2000 ms per action	Full MFA / human-perceptible challenge compatible
Supervisory control / SCADA reconfiguration	1 – 10 s per action	Deliberate authorization gate compatible; full MFA with audit trail
Production schedule / recipe change	10 s – 60 s per action	Multi-party authorization gate compatible; institutional review

The table is not exhaustive; it illustrates the principle that the operational latency budget varies across at least five orders of magnitude in a single industrial system. A uniform authentication strategy that does not accommodate this variation either over-protects fast loops to operational impossibility or under-protects slow operations to security insufficiency.

4. Latency-Aware Authentication as an Architectural Discipline

The discipline of latency-aware authentication is operationalized through five architectural commitments.



quence-graded state-machine function G:(C, S) → O × L with three permitted outputs and a latency budget L set per consequence tier. Every output persists to the

C1 — Consequence classification before authentication design. Every action class in the industrial control system is assigned an operational-consequence tier prior to authentication design. The tier captures the maximum downstream consequence of a single instance of the action, not the typical case. A motion-control loop's typical command has low consequence (incremental position adjustment); the same loop's threshold-crossing command (force-limit exceedance, envelope departure) has high consequence. The classification is the substrate; the authentication design follows.

C2 — Pre-execution gating with three outputs. Every action passes through a pre-execution gate with three permitted outputs:*allow* (logged), *hold/block* (with plain-language operational explanation), or *recommend* (safer alternative)[7]. The gate's latency budget is set by the operational consequence tier: low-consequence actions pass through a sub-millisecond log-only gate; high-consequence actions pass through a deliberate authorization gate calibrated to human decision-making latency. The architectural pattern is the same across tiers; the latency budget and the authorization requirement vary.

C3 — Persist-before-deliver across all tiers. Every gated action persists to the audit log before it dispatches, including low-consequence actions whose gate is log-only[3]. The pattern, named *persist-before-deliver*, ensures that the audit trail is intact across the full operational lifecycle of the system, not only at the gates where human authorization was required. Post-hoc reconstruction of an incident does not depend on which tier the gated action occupied; every action's audit record is recoverable.

C4 — Operator authority preserved across all tiers. The operator's authority to override the gate's recommendation, or to invoke a halt command independent of the gate, is preserved at every tier [4]. The override is logged for audit but never used to trigger adverse personnel or operational consequences. The architectural commitment, named *operator authority is the invariant*, ensures that latency-aware authentication does not become a mechanism by which the operator's authority is gradually preempted as the system's autonomy increases.

C5 — Tier composition with ISA/IEC 62443 SLs. The consequence-class tier composes with the attacker-class Security Level. A command's full authorization profile is the Cartesian product of (SL, consequence tier). The pattern admits the configuration in which a system whose attacker-class is SL2 (intentional misuse with simple means) protects its high-consequence actions with the same deliberate authorization gate as an SL4 system, while leaving its low-consequence actions at lightweight protection appropriate to the threat profile.

C6 — Dynamic consequence reclassification under fault conditions. Consequence classification is not static. A command that is routine under nominal operating conditions may carry elevated consequence under a developing fault — a motion-control position adjustment that is incremental at normal load may be safety-relevant when force limits are approaching threshold, or when a downstream interlock has degraded. The discipline therefore requires that consequence classification operate as a state-machine input to the pre-execution gate rather than as a static command attribute. Architecturally, the gate evaluates*(command, current operational state)* rather than *(command)* alone. The architectural consequence is that the gate's tier assignment is itself a runtime decision; the design challenge is to keep the tier-evaluation latency within the operational budget while admitting the reclassification logic the safety profile requires.

Formal sketch of the pre-execution gate. The gate is a function$G$: $(C, S) \to O \times L$, where C is the candidate command,S is the current operational state (including the attacker-class Security Level under ISA/IEC 62443, the consequence-class tier under C1, the operator-authority context (the architectural principle that the human-authoritative state is the default for consequential actions, consistent with EU AI Act Article 14 [12] and the Parasuraman–Sheridan–Wickens levels-of-automation framework[13]), and the fault-state reclassification under C6), O is one of three outputs$\{$*allow, hold/block, recommend*$\}$, and L is the latency budget for the assigned tier (sub-millisecond for lightweight log; deliberate-human latency for high-consequence authorization). Every gate evaluation persists to the audit log before any downstream action — the *persist-before-deliver* discipline that makes the gate's behavior reconstructable post-hoc. The gate is single-shot per command and stateless beyond the input S; state-machine dynamics produce updates to S that subsequent gate evaluations see. The gate is therefore not a security policy engine in the conventional Policy-Decision-Point sense; it is a per-command consequence-graded authorization checkpoint with explicit operator-override semantics built in.

Zero Trust integration. NIST SP 800-207 [6] specifies the Zero Trust principles of continuous verification, assume breach, and least privilege. The discipline proposed here is consistent with all three: continuous verification is operationalized by S being a live state input to every gate evaluation rather than a session-establishment artifact; assume breach is addressed by the per-command authorization gate rather than perimeter trust; least privilege is enforced at the consequence-class tier rather than uniformly. The contribution of the discipline relative to NIST SP 800-207 is the explicit operational-consequence axis: existing Zero Trust frameworks grade trust against user context (device, location, behavior); the discipline here grades the gate's required authorization strength against the consequence of the protected control action.

5. Operationalization in MILO

The discipline is operationalized in MILO through the Guardrail and Validation module of the v.4 architecture[7]. The module implements pre-execution gating with the three outputs (allow, hold/block, recommend), routes each gated action through the audit-first command bus [3], and preserves operator override authority at every gate. The module's tier configuration admits the consequence-class tier as a per-action parameter; the latency profile of the gate varies with the tier.

Explicit-target dispatch — every command has one explicit target, dispatched through the audit-first command bus with no implicit resolver[3] — is the substrate that makes Commitment C3 (persist-before-deliver across all tiers) operationally enforceable. A command whose target is implicit or whose dispatch is auto-chained from a prior command's completion cannot be reliably audited; MILO's architecture explicitly rejects auto-chaining and requires explicit commanding for every consequential action[8]. *Explicit commanding as a safety interlock* [8] is the discipline's specific safeguard against the auto-chain failure mode.

The module is operationalized in MILO's machine-orchestration layer (v.4) [3]. The extension to the operator-cognitive performance layer (v.5) — Cognitive-State-Aware Decision Support, Operator-Task Alignment Under Load, and the Sustained Operational Capacity layer — applies the same gating discipline to actions whose authorization is mediated by operator cognitive state under the eight non-negotiable operational integrity constraints[9]. The v.5 extension is design-stage in the current MILO codebase and is presented in the framework's DOE Genesis Mission submission[9] under active development.

6. Author Authority and Industrial Context

The discipline articulated in this paper is grounded in the author's hands-on deployment experience in industrial vision systems across regulated manufacturing domains, including food and beverage, pharmaceutical, and medical-device production. The author has not published the specific systems involved (industrial deployment specifications are typically protected by customer confidentiality), but the operational pattern observed across the domains is consistent: a uniform authentication strategy that does not grade against consequence fails in one of the two ways named in Section 1 — either by violating the operational latency budget or by being bypassed or moved out of the loop when the operational impact becomes evident. The discipline of consequence-graded pre-execution gating is the author's response, developed during deployment work and articulated here for the first time in publication form.

7. Implications and Discussion

For *design*, the discipline implies that authentication strategies for industrial control systems must be designed before the operational consequence classification is known to be incomplete, and revisited as the classification is refined. The cost of a uniform strategy is paid at deployment, where operators discover the strategy's incompatibility with the operational budget; the cost of a consequence-graded strategy is paid up front in classification work and architectural design. The trade-off favors the up-front cost.

For *evaluation*, the discipline implies that authentication strategies should be evaluated on per-tier metrics rather than aggregate measures. A strategy that protects high-consequence actions adequately while imposing operational friction on low-consequence actions is observationally indistinguishable from a strategy that under-protects high-consequence actions while leaving low-consequence actions unprotected — both produce similar aggregate authentication-attempt counts. The per-tier metrics (gate latency, override frequency, hold/block-to-allow ratio per tier) admit the distinction.

For *deployment*, the discipline implies that industrial cybersecurity standards convergence around consequence-grading is operationally necessary, and that the ISA/IEC 62443 SL framework [1] composes naturally with a consequence-class axis without requiring revision of the SL definitions. The deployment context — DOE Genesis Mission emphasis on advanced manufacturing, grid reliability, and human-in-the-loop industrial systems[10] — is precisely the context in which consequence-graded authentication, rather than uniform authentication, is operationally appropriate.

7.1 Limitations

The framework is presented at the architectural-discipline level; no instrumented industrialcontrol deployment results are reported in this manuscript. Three specific limitations bound the present contribution: (i) the consequence-class taxonomy (§4.C1) is specified as a deploymenttime activity but no canonical taxonomy is offered, and inter-deployment portability of consequence classifications is an open question; (ii) the formal sketch of the pre-execution gate (§4) does not include a worst-case latency analysis under the kinds of state-machine pathologies that real industrial systems exhibit (sensor dropouts, transient PLC scan-overruns, network jitter); (iii) the composition argument with ISA/IEC 62443 SL tiers [1] reflects design intent rather than a certification outcome, and conformance to the standard's Common Requirement clauses is a deployment-level audit, not asserted by this manuscript. The author's industrial-vision deployment experience informs the framework but does not constitute external validation; empirical evaluation against an instrumented industrial-control deployment is forthcoming work.

8. Conclusion

This paper has proposed*latency-aware authentication* as an architectural discipline for industrial control environments — authentication strength graded against operational consequence per control cycle, composed orthogonally with the ISA/IEC 62443 attacker-class Security Level framework[1], grounded in the latency budgets of real industrial domains, and operationalized through the pre-execution gating module of MILO [3], [7]. The discipline addresses a recurring industrial-control gap: web-derived authentication patterns can be operationally incompatible with control-loop latency budgets, and the standard responses (perimeter-only authentication, full per-command authentication with bypass-prone overhead) either under-protect or over-burden the system. The discipline is grounded in the author's industrial vision deployment experience and is consistent with NIST SP 800-82r3 [2], NIST AI 100-1 Appendix C [11], and Zero Trust architecture principles[6]. Related architectural directions — supervisory primacy for human-in-the-loop AI orchestration, structural principles for adaptive AI architecture, the discipline of viability under non-stationary conditions, and multi-source cryptographic entropy sourcing — are developed in related work by the author and share the same audit-first command bus substrate and operational integrity constraints.

Data Availability

All architectural materials, source manuscripts, the reference implementation, and accompanying figures are openly available at https://github.com/jmontano1/milo-architecture and permanently archived at Zenodo (DOI: 10.5281/zenodo.20117025). No private datasets are referenced; the architectural framework itself is the subject of this paper. Patent rights for the underlying MILO software architecture are reserved; the ~MILO trademark is held under USPTO Serial No. 99706004 (intent-to-use, Class 009).

References

[1] International Society of Automation / International Electrotechnical Commission, *ISA/IEC 62443 series — Industrial communication networks — Network and system security*. Geneva, Switzerland: IEC, 2009–2024.

[2] National Institute of Standards and Technology, *Guide to Operational Technology (OT) Security*, NIST SP 800-82r3, Sep. 2023. doi:10.6028/NIST.SP.800-82r3

[3] J. E. Flores Montano, *MILO (Modular Intelligent Learning Orchestrator)*, JM Automated Solutions. Patent pending. Submitted under the U.S. Department of Energy Genesis Mission, 2026.

[4] *Adaptive Authentication: Risk-Based Verification for Modern Enterprises*, Entrust Identity Platform Technical Brief, 2024.

[5] M. Alqahtani et al., "Adaptive Context-Aware Security Framework for Zero Trust Architectures," *Preprints*, doi:10.20944/preprints202404, 2024.

[6] S. Rose, O. Borchert, S. Mitchell, and S. Connelly, *Zero Trust Architecture*, NIST SP 800-207, National Institute of Standards and Technology, Aug. 2020. doi:10.6028/NIST.SP.800-207

[7] J. E. Flores Montano, MILO v.4 — Guardrail and Validation module specification, internal architecture, in [3].

[8] *Explicit commanding is the safety interlock* — MILO architectural pattern, in [3].

[9] J. E. Flores Montano, MILO v.5 — Eight Operational Integrity Constraints (no coercion, individual baseline only, no surveillance, operator authority invariant, operational transparency, data sovereignty, override always available, independent oversight), in [3].

[10] U.S. Department of Energy, "The Genesis Mission: Transforming Science and Energy with AI," Office of the Under Secretary for Science, Executive Order 14363, November 2025. [Online]. Available: https://www.energy.gov/genesis

[11] E. Tabassi, *Artificial Intelligence Risk Management Framework (AI RMF 1.0)*, NIST AI 100-1, National Institute of Standards and Technology, Jan. 2023, Appendix C. doi:10.6028/NIST.AI.100-1

[12] European Union, *Regulation (EU) 2024/1689 of the European Parliament and of the Council of 13 June 2024 laying down harmonised rules on artificial intelligence (Artificial Intelligence Act)*, Article 14 (Human Oversight), Aug. 2024. [Online]. Available: https://eur-lex.europa.eu/eli/reg/2024/1689/oj

[13] R. Parasuraman, T. B. Sheridan, and C. D. Wickens, "A model for types and levels of human interaction with automation," *IEEE Transactions on Systems, Man, and Cybernetics — Part A: Systems and Humans*, vol. 30, no. 3, pp. 286–297, May 2000. doi:10.1109/3468.844354

About the author

Jorge Enrique Flores Montano (ORCID iD: 0009-0003-1859-8418; jmontano@jmautomated.com) is the founder of JM Automated Solutions and the inventor of MILO. A full biography is maintained at https://www.milo-usa.com/jorge-enrique-flores-montano.html.

Conflict of Interest and Funding Disclosure

The author is the inventor of MILO (patent pending) and the founder of JM Automated Solutions. The discipline proposed in this paper is a contribution from a working development program in which the author retains sole authorship and inventive interest. No external funding was received for the preparation of this manuscript. The author retains all rights to MILO and to the discipline articulated herein.

Appendix A — About MILO

MILO (Modular Intelligent Learning Orchestrator) is a patent-pending adaptive AI orchestration architecture organized into discrete, single-responsibility subsystems under a strict separation-of-concerns discipline. An audit-first command-and-signal substrate persists every command before dispatch and every signal before fanout, producing an append-only audit trail that survives arbitrary process termination. The architecture is designed for *viability* — operational continuity under non-stationary conditions — rather than for prediction accuracy against an expected future.

Eight structural principles

Six are established physical, informational, control-theoretic, and statistical laws applied as architectural design constraints; two are original frameworks proposed by the author for the operator-cognitive performance layer of high-consequence systems.

1. **Second Law of Thermodynamics** — entropy treated as an architectural diagnostic signal, not a fault to be suppressed.
2. **Ashby's Law of Requisite Variety** — a regulator must possess variety at least equal to the system it regulates; implemented as a fleet of specialist agents matching the operational domain.
3. **Shannon Information Theory** — variance reduction occurs at the signal-carrier level, not redundantly at each consumer.
4. **Principle of Least Action — Single-Target Dispatch** — every command has one explicit target; no implicit resolvers, no opaque dispatchers.
5. **Lyapunov-Style Bounded Response** — every adaptive subsystem admits an explicit halt-and-resume pathway; adaptation that drifts unboundedly is failure, not learning.
6. **Power-Law Distribution Architecture** — engineered for the 99th-percentile event, not the median.
7. **Individual-Baseline Variance Modeling** *(original framework)* — operator-layer interventions calibrated against the individual's own established performance baseline, never a population norm. Design-stage; pending empirical validation.
8. **Precision Perturbation Without Variance Compression** *(original framework)* — operator-layer interventions shift probability mass toward high-reliability decision outputs while preserving operator authority and the variability that *is* the operator's adaptive intelligence. Design-stage; pending empirical validation.

Eight operational integrity constraints

Architectural commitments designed to be implemented as enforceable safeguards in deployment builds — not as runtime policy. Disabling any constraint should require rebuilding from source, not toggling a flag.

1. **No coercion, ever** — the system issues recommendations, never compels.
2. **Individual baseline only** — measurements against the operator's own baseline; never against a population norm or productivity target.
3. **No surveillance architecture** — performance-support tool, not a monitoring infrastructure.
4. **Operator authority is the invariant** — the system expands effective decision options; it never narrows or preempts them.
5. **Operational transparency** — every recommendation includes a plain-language explanation.

6. **Data sovereignty** — operator-layer data belongs to the institutional program under documented data governance.
7. **Override always available** — overrides are logged for audit but never used for adverse personnel action.
8. **Independent oversight** — operator-layer deployments require institutional ethics-board review, published consent frameworks, and periodic third-party audits.

Unifying principle

> *MILO does not predict the future. It remains viable in any future.*

The principle is falsifiable: a system whose audit trail is incomplete, whose recovery is improvised, whose adaptation drifts unboundedly, or whose operator override is policy-level rather than architectural, fails the principle.

Trademark, patent, and submission status

- **Mark.** *~MILO™* — U.S. Patent and Trademark Office Serial No. 99706004; filed March 16, 2026; intent-to-use; International Class 009 (downloadable AI software). The leading tilde disambiguates from senior MILO marks held by unrelated owners in different International Classes.
- **Patent.** Patent application pending for the underlying software architecture. Implementation may require a patent license once issued; nothing in this document or its CC BY 4.0 license on the manuscript text grants any patent license.
- **Federal submission.** Submitted to the U.S. Department of Energy under the *Genesis Mission* (Executive Order 14363, November 2025); currently under review. No acceptance or grant outcome is claimed.
- **Concept DOI.** 10.5281/zenodo.20117025 — Zenodo, persistent across versions.
- **Public reference.** https://github.com/jmontano1/milo-architecture.
- **Author contact.** Jorge Enrique Flores Montano · jmontano@jmautomated.com · ORCID iD: 0009-0003-1859-8418.