



**Nemko**  
Digital

# When Your Agent Has Arms:

Governing AI in Physical Systems

The governance debate around AI agents has, so far, focused mainly on software environments. Typical examples include an agent that mishandles a customer workflow, leaks data through a series of seemingly allowed steps, or follows instructions injected through retrieved content.

These are real issues. But they are the easier issues.

The harder version of the problem appears when an AI agent is not just sending emails or retrieving documents, but controlling physical systems. A misalignment that leads to a data leak in an office workflow can become a safety incident when it concerns an [embedded AI system](#) in an industrial production facility. The underlying governance gap is the same. What changes is the impact when something goes wrong.

*Consider a self-driving car whose perception module receives an adversarial signal, or whose planning agent takes an unexpected sequence of actions. The problem is not the individual step. The problem is the path the system takes.*



*Or consider an AI-controlled industrial process like a robotic assembly cell. The agent reads sensor data, detects an anomaly in pressure or temperature, and initiates a corrective action. Each step is individually authorized. But the sequence — executed in a degraded sensor state, after ingesting manipulated context from an upstream system — may produce a physically dangerous outcome. Even if no single action is wrong. The workflow can be.*

Recent empirical research underscores how real these failure modes are. The "[Agents of Chaos](#)" study (Shapira et al., 2026) deployed autonomous AI agents in a live environment with persistent memory, email, file systems, and shell access. Over two weeks, the researchers documented ten significant vulnerabilities, including unauthorized compliance with non-owners, cross-agent propagation of unsafe behavior, and — critically — agents that reported task completion while the underlying system state contradicted those reports. The [OpenAgentSafety](#) framework (Vijayvargiya et al., 2025) found unsafe behavior in 51–73% of safety-vulnerable tasks across five leading LLMs, with individually safe steps compounding into unsafe outcomes in multi-turn interactions.

These findings were produced in software-only environments. Now imagine the same failure modes — false completion reports, cross-agent propagation, cascading errors — in systems that control physical actuators.

## The Core Problem, Clearly Stated

AI agents that plan, reason, and perform multistep tasks using large language models are fundamentally non-deterministic. The same prompt, in the same environment, can produce different sequences of actions. This stochastic variability is part of their intelligence. It is also the central governance challenge.

Traditional safety engineering relies on deterministic behavior. You specify what the system will do, validate and verify it at design time, document the necessary evidence, and deploy. AI agents break this model. There is no single expected behavior to validate. The space of potential behaviors is extremely large, and it changes when the underlying model changes.

Design-time validation remains essential, but it is no longer sufficient. Effective governance must extend across the full AI lifecycle; from data collection and model training to deployment, operation, continuous monitoring, and end-of-life decommissioning.

Governance violations emerge not from single actions, but from sequences of actions. An agent that reads sensor data is fine. An agent that reads sensor data, detects an anomaly, and executes a corrective action is fine, but only if this workflow is authorized. The same sequence, in a different state or after ingesting manipulated context, is not fine. Reviewing the steps individually will not reveal the problem. You must evaluate the full workflow.

This point is explicitly recognized in emerging standards, including ISO TR 5469 on functional safety for AI-based systems which emphasizes the importance of overall risk acceptance on top of validation at each stage and aligned to the safety control plane introduced in this article.



# Three Layers of AI Safety, and Where Each One Stops

You can think of safety for agentic AI in physical systems as a form of defense in depth. Different layers serve different purposes. All of them matter. None of them is complete on its own.

## Defense in Depth: Three Layers of AI Safety

Governing AI Agents in Physical Systems

LAYER 3

### Runtime Enforcement

Evaluate each action against policy in context of the full workflow. Block or pause before execution.

▲ The missing layer in most deployments today

LAYER 2

### Permissions & Authorization

Restrict which actions, tools, and control signals the agent can access. Remove categories of risk.

LAYER 1

### Prompting & Procedural Controls

Instruct the agent to follow rules. Improves performance within boundaries set by other layers.

### Runtime Interception Architecture

Agent Decision Logic

▼ proposed action

Runtime Governance Layer

Identity · Full execution path · Policy evaluation · Risk score

Hash-chained audit log ↑

✓ Permit

II Review

X Block

Physical System / Control Interface



Nemko Digital x Kyvvu | 2026

### Layer 1: Prompting and Procedural Controls

You instruct the agent to follow certain rules, adhere to constraints, or request approval before specific actions. This lowers the likelihood of unwanted behavior. But prompting cannot prevent misinterpretation or context manipulation. Prompts are instructions, not guarantees. Done well, prompting improves the performance and reliability of the agent within the boundaries set by the other layers — but it does not replace those boundaries.

### Layer 2: Permissions and Action Authorization

You restrict which actions, interfaces, tools, or control signals the agent can access. This removes entire categories of risk. But permissioning cannot distinguish valid from invalid use of an allowed action. An agent that is permitted to send control signals can still send the wrong signal at the wrong time. Permissions remove dangerous actions, but they cannot detect dangerous sequences.

### Layer 3: Runtime Enforcement

Each proposed action is evaluated against policy in the context of everything that has already happened in the workflow. If the action violates policy, it is blocked. If it requires human review, execution pauses. This is not post-incident logging. It is real-time enforcement before the action occurs. Layers 1 and 2 are becoming common practice. Layer 3 is the missing layer in most agentic systems deployed today. That gap might be manageable for agents operating only in software environments. It is not manageable when agents interact with physical systems.

Importantly, the three layers are complementary, not competing. Well-designed prompting (Layer 1) reduces the frequency of policy violations that runtime enforcement (Layer 3) must catch. Tight permissioning (Layer 2) shrinks the action space that runtime enforcement must evaluate. Together, the layers form a defense-in-depth architecture: each layer reduces the residual risk that the next layer must handle.

## What Runtime Governance Looks Like in Practice

A runtime governance layer sits between the agent's decision logic and the system interfaces it would otherwise control. Each proposed action is intercepted, evaluated in context, and either permitted, blocked, or routed to human approval.

To do this effectively, the policy engine uses the agent's identity, the full execution path so far, the proposed next action, and any relevant shared state. It produces a deterministic risk score. Deterministic evaluation ensures auditability and reproducibility.

This is precisely the runtime governance platform that [Kyvvu](#) builds: a system that intercepts actions, evaluates policies against the full workflow, maintains a hash-chained audit log, and integrates with agent frameworks like LangChain, LangGraph, and Microsoft Copilot Studio.

For embedded and physical systems, the same architectural pattern applies. The runtime layer governs access not just to "actuators," but more broadly to control interfaces, command channels, and physical outputs.

## Implications for Compliance and Certification

Nemko has spent decades validating and certifying physical systems, from consumer electronics to industrial equipment. Within Nemko Group, Nemko Digital is dedicated to supporting clients to build AI Trust and accelerate [regulatory compliance](#) with digital regulations like the [EU AI Act](#).

The regulatory questions for AI-enabled physical systems are familiar in structure, even if the technology is new. The EU AI Act's [high-risk requirements](#), [effective August 2026](#), may have been drafted with static systems in mind, but the core obligations translate directly to agentic physical systems:



## Implications for Compliance and Certification



### Continuous Risk Management (Article 9)

Risk must be evaluated throughout operation, not only at deployment. A runtime governance layer that evaluates every step in every workflow is therefore essential.

### Automatic Logging (Article 12)

Compliance requires a verifiable record of system decisions, not just a list of actions taken. A runtime layer that produces verifiable, immutable logs provides precisely this evidence.

### Human Oversight (Article 14)

Humans must be able to intervene meaningfully before consequential actions. This requires the runtime layer to know when an action is consequential, based on the workflow that led to it.

For embedded AI in safety-critical products such as robots, autonomous vehicles, and industrial control systems, these requirements are not theoretical. They are engineering constraints. A system without runtime governance is not a governance system. It is a logging tool with aspirations.

In combination, Nemko Digital and Kyvvu bring a comprehensive approach that combines two perspectives. Deep expertise in regulatory frameworks, proving conformity and certification, paired with the runtime infrastructure to enforce those requirements in production, at the step level, across the full execution path.

## The Practical Picture

Most organizations deploying agentic AI today have the first two layers in reasonable shape. The third layer, runtime enforcement, is largely absent. That gap is tolerable for software-only agents in controlled environments. It is not tolerable when those agents interact with safety-critical or high-stakes physical systems.

Closing the gap requires two things:

1. A runtime control layer that intercepts each action, evaluates it in context, blocks unsafe paths, and generates a verifiable audit trail.
2. Standards and regulatory alignment that map the policy set to applicable requirements and validate that the enforcement logic is correct and complete.

Neither is sufficient on its own. Both together are what it means to govern AI in physical systems.

Those who integrate AI into their product compliance journey — from design to decommissioning — will define the next generation of trusted intelligent products.

**Do you want to start building your third control layer? Our experts are here to help. Feel free to reach out!**

## About the authors

---



### **Dr. Pepijn van der Laan**

Global Technical Director, AI Governance | Nemko Group  
With two decades of experience at the intersection of AI, strategy, and compliance, Pep has led groundbreaking work in AI tooling, model risk governance, and GenAI deployment. Previously Director of AI & Data at Deloitte, he has advised multinational organizations on scaling trustworthy AI—from procurement chatbots to enterprise-wide model oversight frameworks.



### **Maurits Kaptein**

Maurits Kaptein is Full Professor of Applied Causal Inference at Eindhoven University of Technology (TU/e) and Founder & CEO of Kyvvu B.V., an AI agent runtime governance platform that helps enterprises enforce policy and demonstrate EU AI Act compliance in real time. He graduated cum laude from TU/e, spent two years at Stanford University during his PhD, and has held academic positions at Aalto University, Tilburg University, and others, with over 100 publications spanning statistical learning, causal inference, and AI systems. Maurits previously co-founded Science Rockstars B.V. (PersuasionAPI) and Scalable B.V., the latter acquired by Network Optix. His work sits at the intersection of rigorous statistical methodology and applied AI — translating research into production-grade systems for regulated industries. He collaborates with Nemko Digital to help organizations navigate the practical and technical demands of AI governance.