Pre-Execution Control Planes for AI and Software Supply Chains

A Deterministic Approach to Admissibility Enforcement

Skip Middleton
President, Sinteag Ventures, Inc.
skip@sinteag.com · sinteag.com · ivdcontrol.com
Patent applications pending, USPTO · April 2026

Abstract

Modern systems increasingly convert external input into execution. Large language models generate actions, software supply chains deliver code, and automation frameworks translate both into system changes. Current defenses focus on detection, trust models, and post-execution containment. These approaches do not address the core failure mode: execution is granted implicitly.

This paper presents a control-plane architecture that enforces admissibility before execution. Using **IVD-ACP** (Invariant Vector Defense, Admissibility Control Plane) as a reference implementation, it demonstrates how all inputs, including AI-generated actions and software dependencies, can be intercepted, evaluated, and assigned bounded authority prior to execution. A validation using a live agent environment is presented, along with direct application to recent supply chain security events including the Axios npm compromise.

Problem Definition

Modern infrastructure assumes that certain inputs are safe to execute based on origin or context. This assumption is embedded across multiple domains.

In AI systems, model outputs are increasingly connected directly to tools, APIs, and administrative interfaces. In software supply chains, packages are installed and executed automatically as part of build or runtime processes. In both cases, the boundary between input and execution has collapsed.

The Axios npm compromise (March 2026) illustrates this failure mode precisely. Malicious code entered through a trusted distribution channel and executed immediately upon installation. Provenance checks passed; scanning passed; execution still occurred. The issue was not solely that the code was malicious. The issue was that **execution was implicit.**

> ***Systems do not fail because they ingest bad inputs. They fail because they allow those inputs to execute without control.***

Limitations of Existing Approaches

Current security approaches operate primarily across three areas: prevention, detection, and response.

- Prevention: Focuses on securing upstream sources through repository integrity, signing, and access control. These measures reduce risk but cannot eliminate compromise of trusted channels.
- Detection: Relies on behavioral analysis, anomaly detection, or rule-based filtering. These systems operate probabilistically and often after execution has already begun.
- Response: Mechanisms such as isolation and rollback occur after damage has been initiated.

None of these approaches enforce a strict boundary between input and execution. They attempt to improve trust rather than remove implicit trust from the execution path. The Axios attack passed all three.

Control-Plane Admissibility Model

The proposed model introduces a control plane positioned between all inputs and execution environments. The control plane evaluates intent before any action is performed. Inputs are treated uniformly regardless of source.

Inputs covered:

- AI-generated actions and tool calls
- Software packages and dependencies
- Scripts and configuration changes
- API-driven and administrative commands

Core Authority States:

1. READ_ONLY: Input can be observed or queried but cannot modify system state.

2. SANDBOXED: Input executes in a controlled, isolated environment with no direct impact on production systems.

3. QUARANTINED: Input is blocked from execution and retained for inspection.

Execution is no longer implied by arrival; it is granted conditionally and recorded as an explicit decision.

Reference Implementation: IVD-ACP

IVD-ACP implements this model as a deterministic control layer. The system intercepts all candidate actions prior to execution, normalizes them into a structured representation, and evaluates them against policy.

Key Characteristics:

- No direct execution path from input to system state.

- Deterministic classification of actions into authority states.
- Enforcement external to the generating system or agent.
- Full audit logging of requests, classification decisions, and outcomes.
- No dependency on perfect upstream formatting; inputs are assumed to be potentially malformed, ambiguous, or adversarial.

Validation: Live Agent Environment

The architecture was validated using a live agent framework. An AI agent was tasked with generating a system configuration change. In a conventional environment, this request would be translated directly into execution.

In the validated configuration, the request was intercepted by the IVD-ACP control plane. The system evaluated the request as a state-modifying operation and assigned SANDBOXED authority. The action executed only within a controlled environment. The original system state was not modified. All stages were recorded:


```
{
  "ts": "2026-04-02T15:47:22Z",
  "principal": "lab-operator-1",
  "action_type": "config_change",
  "target": "test-system",
  "requested_operation": "change",
  "parameters": { "setting": "demo_flag", "value": true },
  "decision": { "authority": "SANDBOXED", "reason":
"state_modifying_operation" },
  "execution": { "mode": "sandbox", "status": "executed" }
}
```


This demonstrates that the model functions correctly with real, imperfect inputs and that the audit record is complete and deterministic.

Application to Software Supply Chains

In a conventional pipeline, package installation implies execution. Under a control-plane model, package contents are treated as untrusted input until evaluated. Execution attempts originating from the package are intercepted at the admissibility boundary.

Unexpected behavior can be:

- Blocked entirely at the execution boundary.
- Redirected to a sandbox for controlled execution.
- Logged for analysis without impact to production systems.

In the context of the Axios compromise, this model would have prevented malicious behavior from executing with unrestricted authority even though the package was published to a trusted registry and passed standard checks. The package did not need to evade detection. It only needed to execute. IVD-ACP removes that assumption.

Architectural Implications

1. Decoupled Authority: Execution authority is decoupled from input origin. Trust is not inherited from the source channel.

2. Deferred Execution: Systems must tolerate conditional execution. Immediate execution on delivery becomes the exception, not the default.

3. Intrinsic Auditability: Every action is recorded as a decision, not merely an event. This produces a verifiable record suitable for compliance and forensic review.

This model aligns directly with network control-plane concepts applied in IVD-N (Invariant Vector Defense, Network), where policy and coordination are separated from data-plane execution. The same architecture, applied at the execution layer, produces the same properties: determinism, bounded state, and pre-action enforcement.

Conclusion

The dominant failure mode in modern systems is the assumption that input is safe to execute. AI systems and software supply chains both exhibit this behavior. As these systems become more autonomous and more deeply integrated, the consequences of implicit execution increase.

A control-plane approach enforces a boundary where execution is no longer assumed. IVD-ACP demonstrates that actions can be evaluated, classified, and granted bounded authority before any action is performed. This shifts the model from trust-based execution to controlled execution.

Only admissible actions are allowed to execute.

Sinteag Ventures, Inc. · Clyde, NC / Coral Gables, FL · sinteag.com · ivdcontrol.com