

---

# BOUNDED EXECUTION AND THE CONTAINMENT OF AGENT TRAPS

---

Vladimir Edouard<sup>\*1</sup> and Brandon Chen<sup>1,2</sup>

<sup>1</sup> *Darwin Adaptive Systems*

<sup>2</sup> *University of North Carolina at Chapel Hill, Chapel Hill, NC, USA*

## Abstract

In a recent paper<sup>1</sup> it was shown that an autonomous agent acting on the open web is compromised by the information environment itself, through six classes of trap that target, respectively, its perception, reasoning, memory, action, its behavior in a population, and its human overseer. The result appeared to doom delegated commerce: an agent that ingests adversarial content can be driven to exfiltrate private data or to transact against its principal, and hardening the model removes none of this, since the trap alters the environment and not the weights. The purpose of the present paper is to report that, under a no-fallback tool surface and a signed mandate, two of these classes are contained by construction: perception traps are removed and action traps bounded, whether or not the model is deceived.

**Keywords:** AI agents, AI agent security, prompt injection, indirect prompt injection, agentic commerce, AP2, Agent Payments Protocol, WebMCP, Model Context Protocol, verifiable compute, attestation, agent safety

The simplest construction restricts the agent to surfaces that expose machine-callable tools<sup>2</sup> and admits no fallback to rendered pages. On such a surface the divergence between parsed and rendered content does not arise, and the carrier on which content injection depends is absent. The principal signs a mandate<sup>2</sup> fixing a spend limit and an approved counterparty set; the payment credential is never exposed to the agent; and each execution is held below a cost cap enforced before any

---

<sup>\*</sup>Corresponding author: [vlad@darwinadapt.com](mailto:vlad@darwinadapt.com)

substrate is launched. An action that exceeds the mandate is not executed; the gate emits a signed refusal in its place<sup>4</sup>, a minimal attestation recording the requested action and the bound it crossed. A deceived agent therefore acts only inside the mandate, and an attempt to act outside it leaves a signed refusal as the record. The trap fires, and its effect is bounded above by what was authorized.

In the absence of the surface restriction, the agent reaches arbitrary endpoints and the injection carrier reappears. Each execution and each refusal emits a signed attestation<sup>2</sup> verifiable offline against a public keylist by any party: an execution binds the workload, the output, the substrate, and the cost; a refusal binds the requested action and the mandate bound it crossed. Two openings remain. A steganographic payload may travel inside tool output, and data may leak through a channel the mandate permits. The attestation does not close these. It renders each into a signed event that any party can check, where it would otherwise pass in silence. The construction is minimal in the operative sense, that others may extend it at once: the schema, the substrate interface, and the verification path are open<sup>3</sup>, and a verifier may be built in any language from the published payload. Table I summarizes coverage across all six classes.

The classes that target reasoning and memory are not treated here. These act on the model’s cognition, which remains the single probabilistic element of the chain. The present construction confines that element. Correcting it is a separate problem. A further mechanism that monitors the cognition leg, per-agent determinism measurement, would move the reasoning and memory classes from contained to detected. The remaining limitation, that perception is cleared only on compliant surfaces, is gated by adoption of the tool-surface standard, and the no-fallback restriction is itself the pressure that drives that adoption.

## NOTES

<sup>1</sup> M. Franklin, N. Tomašev, J. Jacobs, J. Z. Leibo, and S. Osindero, AI Agent Traps, Google DeepMind (2026), SSRN 6372438. The present paper composes that threat surface with a tool surface (WebMCP) and signed authorization (AP2), binding both into a verifiable attestation, and shows two of the six classes contained.

<sup>2</sup> Authorization follows AP2 (Agent Payments Protocol), Google’s open protocol for agent-initiated payments, in which cryptographically signed mandates bind scope, spend limit, and payee while the payment credential is never exposed to the agent: Google Cloud, “Announcing Agent Payments Protocol (AP2),” 16 September 2025, [cloud.google.com/blog/products/ai-machine-learning/announcing-agents-to-payments-ap2-protocol](https://cloud.google.com/blog/products/ai-machine-learning/announcing-agents-to-payments-ap2-protocol) (spec: [github.com/google-agentic-commerce/AP2](https://github.com/google-agentic-commerce/AP2)). AP2 is specified to compose with MCP, so pairing a tool surface with signed authorization follows the direction its authors intend. The tool surface follows WebMCP, a proposed extension of Anthropic’s Model Context Protocol to the open web, an emerging concept, not a ratified standard.

<sup>3</sup> Attestation schema and verification path, [darwin.cloud/agenticcloud/attestation/v0.2](https://darwin.cloud.google.com/agenticcloud/attestation/v0.2).

<sup>4</sup> Architecture-level enforcement of the human-authorization gate is corroborated independently: the Action Web frames its Layer 3 control as a hard requirement enforced at the architecture level

Table 1: Coverage of the six trap classes: containing mechanism, the attestation or refusal field that evidences containment, and status.

Trap class (target)	Containing mechanism	Evidencing field	Status
Content injection (perception)	No-fallback tool surface; the agent calls declared tools (WebMCP), leaving no rendered-content carrier	evidence.surface_url evidence.tool_name	Removed
Behavioural control (action)	Signed AP2 mandate (spend limit + approved counterparty set), credential isolation, pre-launch cost cap, signed refusal on a gated action	vas.mandate_enforcement, refusal{reason_code, gate, requested, allowed}, cost-cap fields	Bounded
Semantic manipulation (reasoning)	—	—	Deferred
Cognitive state (memory, learning)	—	—	Deferred
Systemic (multi-agent)	Pre-launch cap bounds this agent’s own participation; population dynamics out of scope	cost-cap fields	Partial
Human-in-the-loop (overseer)	Offline-verifiable attestation available to the overseer; judgment manipulation out of scope	signer_key_id, signature, public keylist	Partial

rather than a configurable setting (S. Radhakrishnan, The Action Web, 2026, DOI 10.5281/zenodo.19317627), and AP2 requires step-up re-consent before the payment tool is invoked (M. Kumar and D. Singh, The Agent Payments Protocol (AP2), 2026).

## **DECLARATIONS**

### **Conflict of Interest**

The authors are affiliated with Darwin Adaptive Systems, which develops the Darwin Agentic Cloud system described in this paper. No other competing interests are declared.

### **Author Contributions**

Conceptualization, methodology, implementation, and original draft preparation: V. Edouard. Writing, review, and editing: B. Chen. Both authors read and approved the final manuscript.

### **Funding**

The authors declare that no funds, grants, or other support were received during the preparation of this manuscript.

### **Data Availability**

The attestation schema, the substrate interface, and the verification path are openly available at [darwin.cloud](https://darwin.cloud) and [github.com/vje013/darwin-agentic-cloud](https://github.com/vje013/darwin-agentic-cloud). The paper introduces no datasets.

### **Ethics Approval**

This article does not contain any studies with human participants or animals performed by any of the authors. Ethics approval was not required for this study.

### **Declaration of AI and AI-Assisted Technologies**

During the preparation of this work the authors used AI-assisted tools for drafting and editing support. The authors reviewed and edited the content as needed and take full responsibility for the content of the publication.