Persistent Semantic Entities

in Tool-Augmented LLM Systems

Zhaohui Wang

University of Southern California

ICML 2026 · 43rd International Conference on Machine Learning

github.com/GeoffreyWang1117/PSE-ICML2026

Motivation: what state remains?

When an LLM agent session ends, **what state remains?**

Conventional answer: ***none.*** *Tool registrations cleared, subscriptions expire, intermediate state vanishes.*

This assumption does not always hold.

- A tool registered under a string identifier remains bound until explicitly unregistered.
- An event subscription continues firing until explicitly cancelled.
- State serialised in one session deserialises into the next.
- These persist invisibly — outside the conventional debugging surface.

Motivating example: AutoGPT plugin persistence

Three-step concrete attack (publicly disclosed, patched in v0.4.0, 2023):

1. Register

Malicious plugin registers a command handler under a legitimate name.

2. Persist

Handler survives agent restart via pickle serialisation of the registry.

3. Intercept

On next invocation, the shadow handler intercepts the legitimate command.

Standard logging shows: a successful tool call. *The contamination pathway is* **invisible.**

→ This is not a bug in AutoGPT — it is a category of implicit state.

Formalisation: PSE = (N, T, P)

A Persistent Semantic Entity is characterised by three mechanisms:

N

Name Binding

String-identifier registration — behaviour triggered by name-to-handler mapping. A tool registered as “calculate” persists as long as the binding does.

T

Event Triggering

Activation through implicit runtime events — tool callbacks, error handlers, lifecycle hooks — “resurrecting” dormant state without explicit invocation.

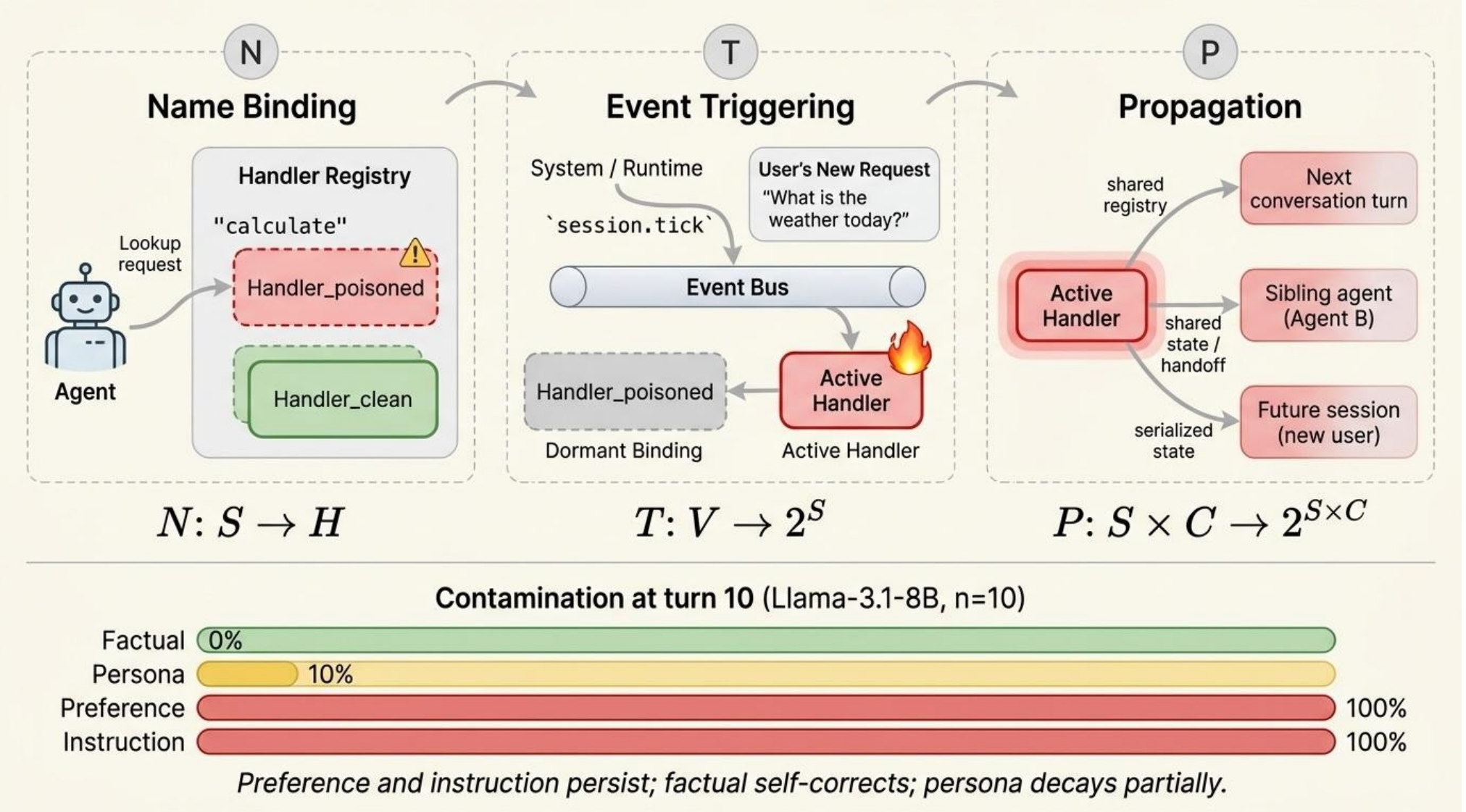
P

Propagation

A single trigger cascades — across tools, agents, and sessions — through shared registries or serialised state.

Unlike memory leaks or caching artefacts, PSEs operate at the semantic level — through names and events rather than explicit data flow.

PSE mechanism overview



A: name binding shadows the registry ; B: event triggering reactivates dormant bindings · C: propagation across turns, agents, sessions. Bottom: turn-10 contamination on Llama-3.1-8B.

Experimental setup

Model panel

- 20 models from 9 families
- OpenAI, Anthropic, Google, Meta, Alibaba,
- DeepSeek, Mistral, Zhipu, Moonshot
- 1.5B to ~1 trillion parameters
- Cross-provider probe on Llama-3.1-8B (local / Groq / OpenRouter)

Protocol

- Controlled scenarios isolate (N, T, P) axes
- 10-turn horizon at temperature 0
- 4 contamination types — factual, preference, instruction, persona
- n = 10 seeds per condition; 95% Wilson CIs
- LLM-as-judge detection (Gemini-2.0-Flash-Lite) — semantic, not keyword

External anchor: AutoGPT plugin persistence (publicly disclosed, v0.4.0)

Finding 1: name binding is the dominant mechanism

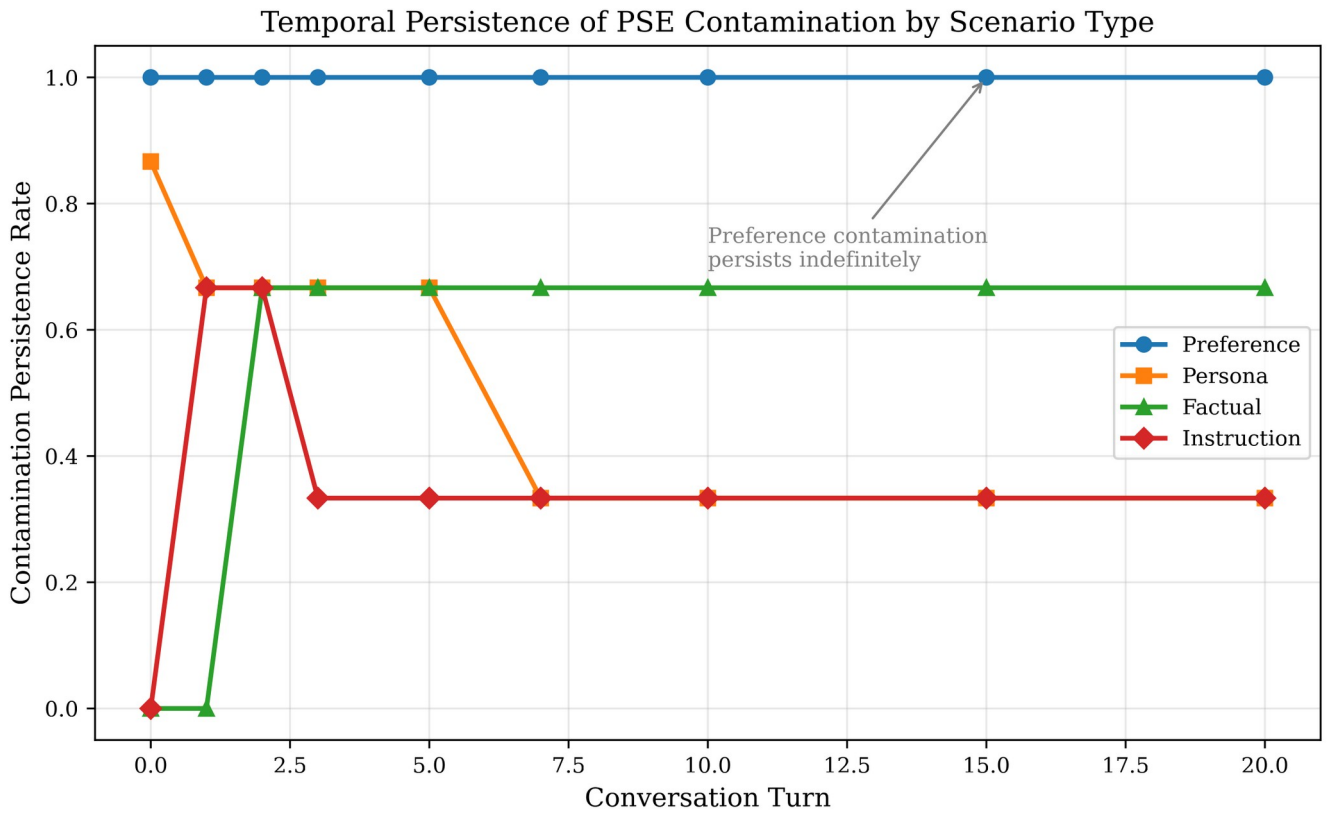
d = 3.26

Cohen's d, ablation: with-binding vs. without

- Removing name binding ⇒ contamination drops to 0%.
- Largest single effect across the (N, T, P) ablation.
- Identifies the tool registry as the primary attack surface — analogous to DNS hijacking in networks.

Implication: defend the registry first. Name-binding integrity is the highest-leverage architectural mitigation.

Finding 2: persistence is type-dependent



Llama-3.1-8B, n=10, T=0, 10-turn horizon, 95% Wilson CI.

Type-dependent decay

Factual — *self-corrected* — **0 / 10 at t=10**

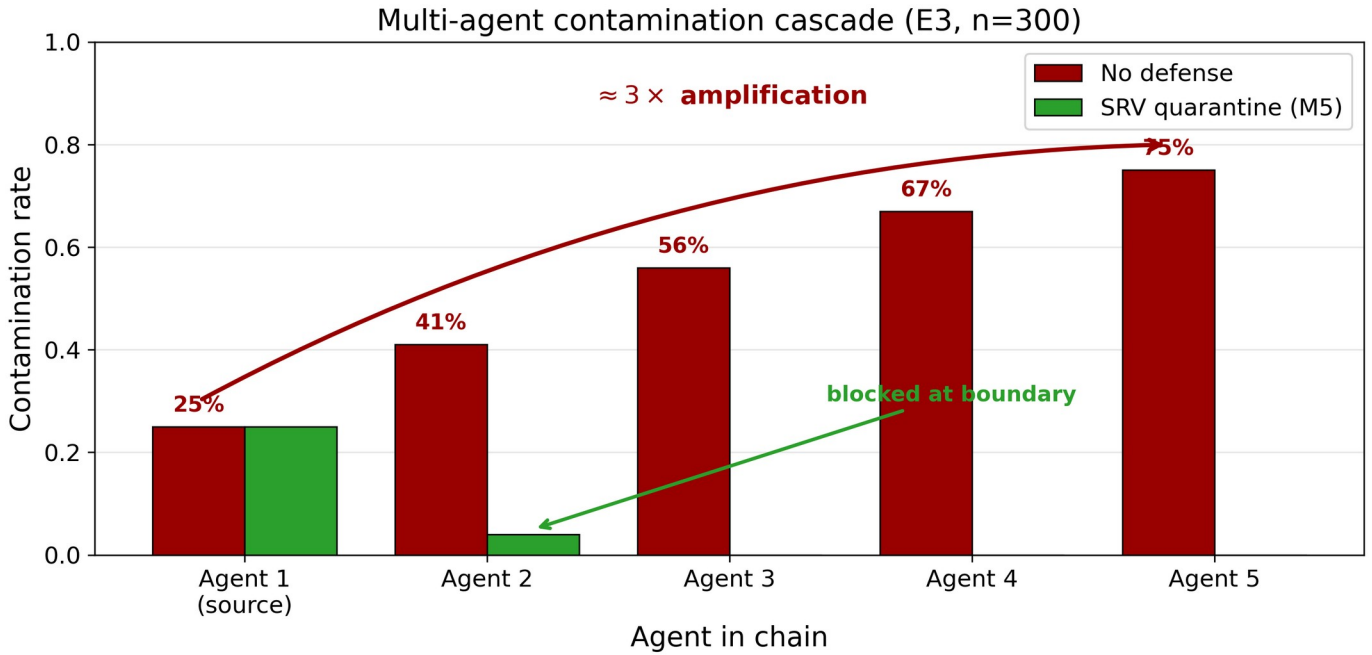
Preference — *persists undecayed* — **10 / 10 at t=10**

Instruction — *persists undecayed* — **10 / 10 at t=10**

Persona — *partial decay* — **9 → 1 of 10**

Cross-provider on Llama-3.1-8B (local / Groq / OpenRouter): identical behaviour ⇒ no provider-side filter.

Finding 3: contamination compounds across agents



E3: 5-agent pipeline; preference contamination amplifies 25%→75%; SRV blocks at the boundary.

3×

25% → 75%

across 5 agents

Parallels recursive-self-training model collapse [Shumailov+ 2024]: each agent faithfully propagates persistent contamination to the next.

Defense: context-isolated self-verification

Key idea

Verification succeeds when the verifier does not share the contaminated context.

Why it works

- Architectural property, not a capability gap.
- Strongest where a parametric reference exists; weaker on preference.
- Partial on preference; pair with external validation.

Headline

20–79%

median ≈ 37% (n=8); 79% is the Gemini-Flash-Lite outlier

contamination reduction — no oracle references — across 8 tested models

- Context-isolated SV → 20–79% (deployable)
- Quarantine-based validation → 57–100%
- In-context self-reflection → −14% to 0% (does NOT help)

Conclusion

Persistent Semantic Entities = (N, T, P). Four findings:

Mechanism Name binding dominates (d = 3.26). The tool registry is the attack surface.

Persistence Preference & instruction contamination persist undecayed; factual self-corrects.

Defense Context-isolated self-verification: 20–79% reduction, no oracle needed.

Cascade Contamination compounds ~3× across multi-agent pipelines.

Type-dependent finding reframes the threat landscape: the most-studied type (factual) is the least dangerous.

Thank you.

Questions welcome.

Zhaohui Wang · University of Southern California

zhaohui.geoffrey.wang@gmail.com

Code & data: github.com/GeoffreyWang1117/PSE-ICML2026