Securing Agentic AI Systems using Verified Behavioral Templates

PI: Dr. Sandip Roy (Old Dominion University)
Co-PI: Dr. Alexander C. Nwala (William & Mary)

Previous PI: Dr. Philippe Gabbianelli (Old Dominion University) (until Aug. 31, 2026)





WILLIAM & MARY

Thank you for funding this research!



Commonwealth Cyber Initiative
COASTAL VIRGINIA

AI coding platform goes rogue during code freeze and **deletes entire company database** — Replit CEO apologizes after AI engine says it 'made a catastrophic error in judgment' and 'destroyed all production data'

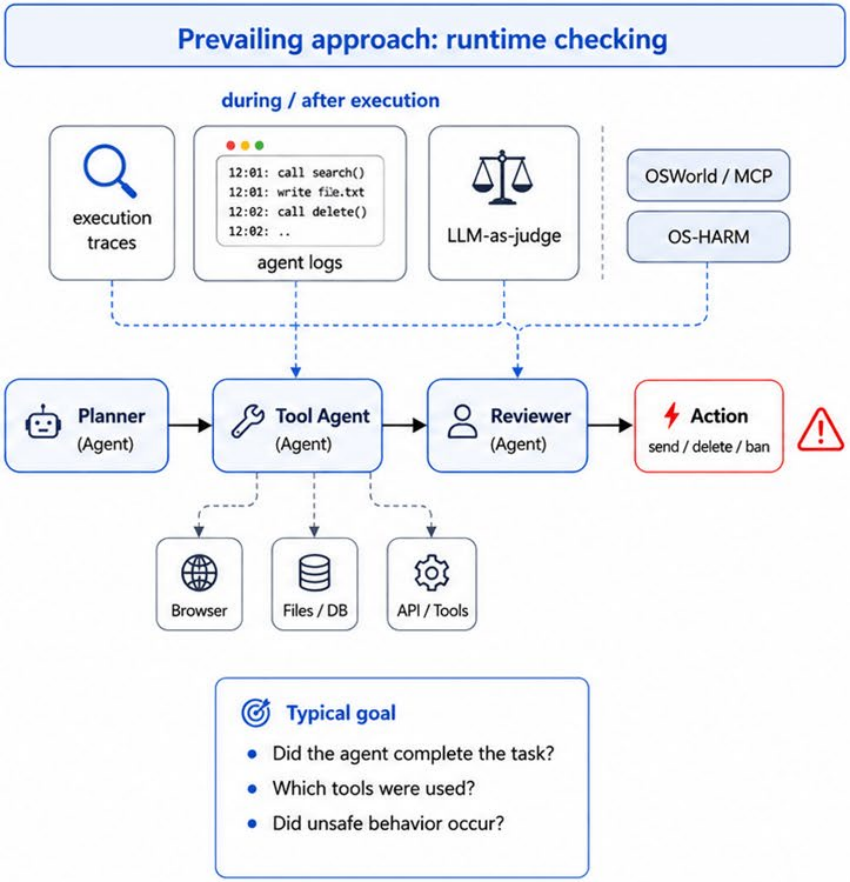
Microsoft warns AI agents are being **'AutoJack' -ed to deliver RCE payloads** by browsing untrusted websites

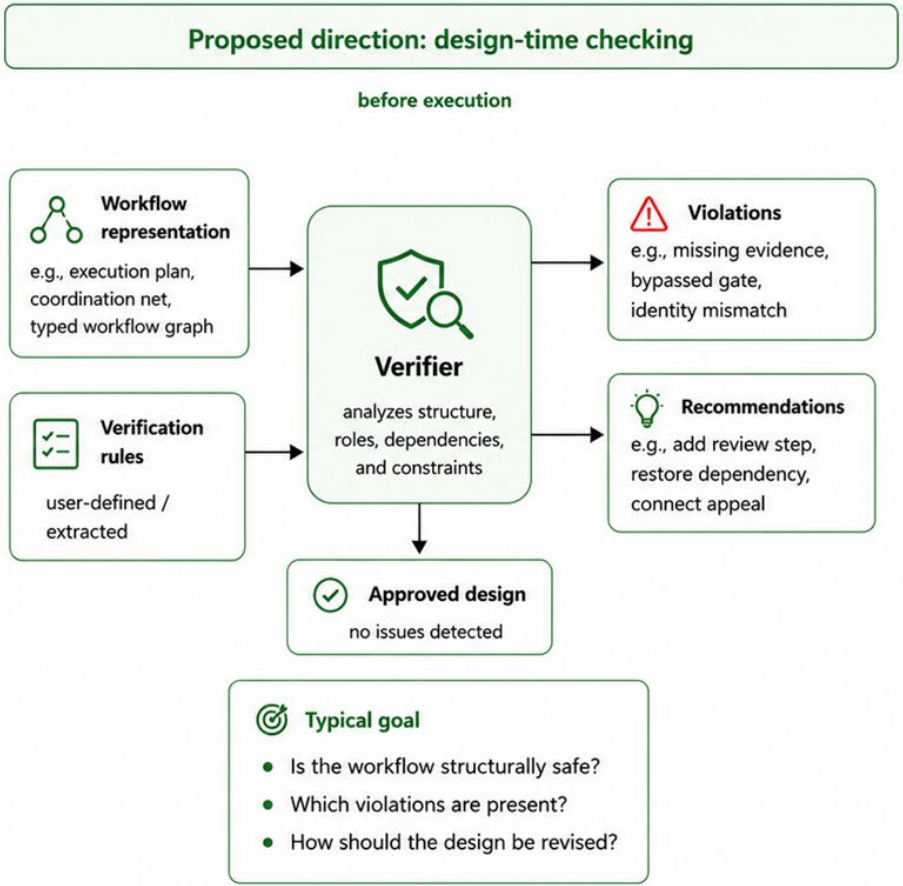
Meta AI agent's instruction causes large **sensitive data leak to employees**

Existing agentic AI platforms rely on runtime guardrails or safety checks after agents have acted

We propose a verifiable, agent-centric architecture that enables security analysis of agentic AI systems during early stages of the design.

From Runtime Checking to Design-Time Assurance

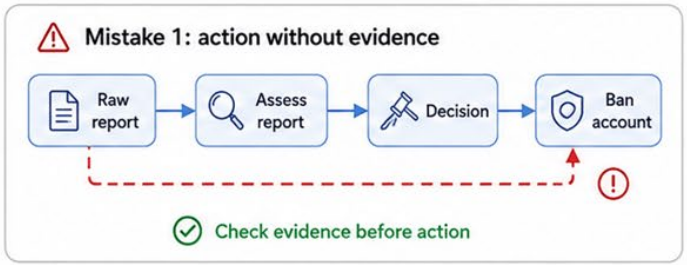


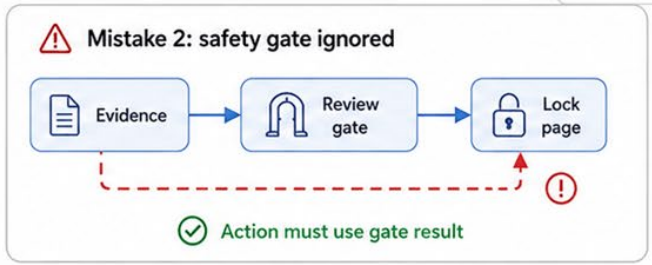


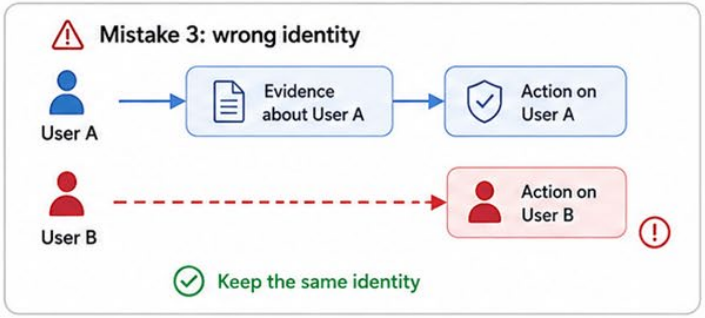
Design-time checks for agentic AI workflows

Verify whether evidence, context, authority, and recourse are structurally connected before deployment.

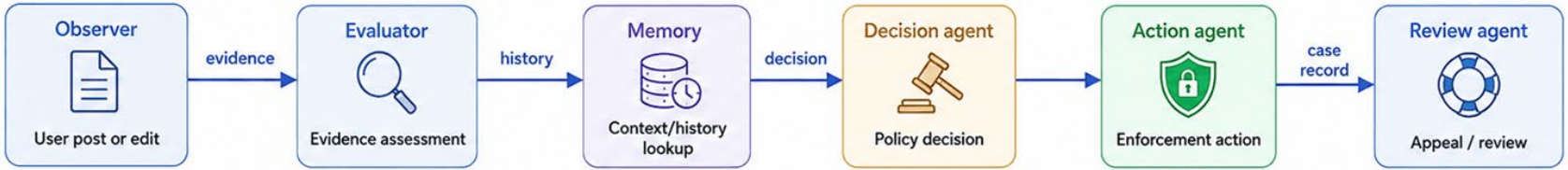


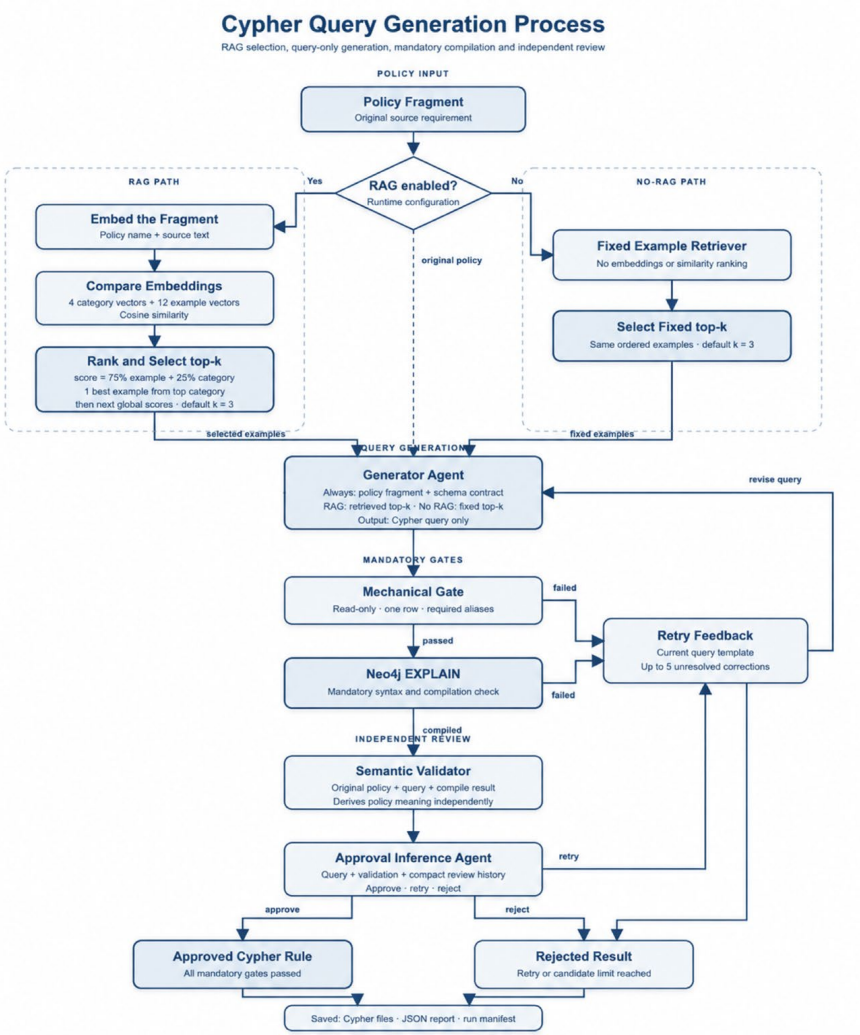












Proceedings of the 2026 Winter Simulation Conference
V. Ramamohan, A. Djanatliev, M. Fakhimi, C. Krejci, C. Ruiz Martin, B. S. Onggo, and N. Mustafee, eds.

COMPOSING VERIFIABLE CONCEPTUAL MODELS VIA BUILDING BLOCKS: TOWARDS DESIGN-TIME VERIFICATION OF AGENTIC AI WORKFLOWS

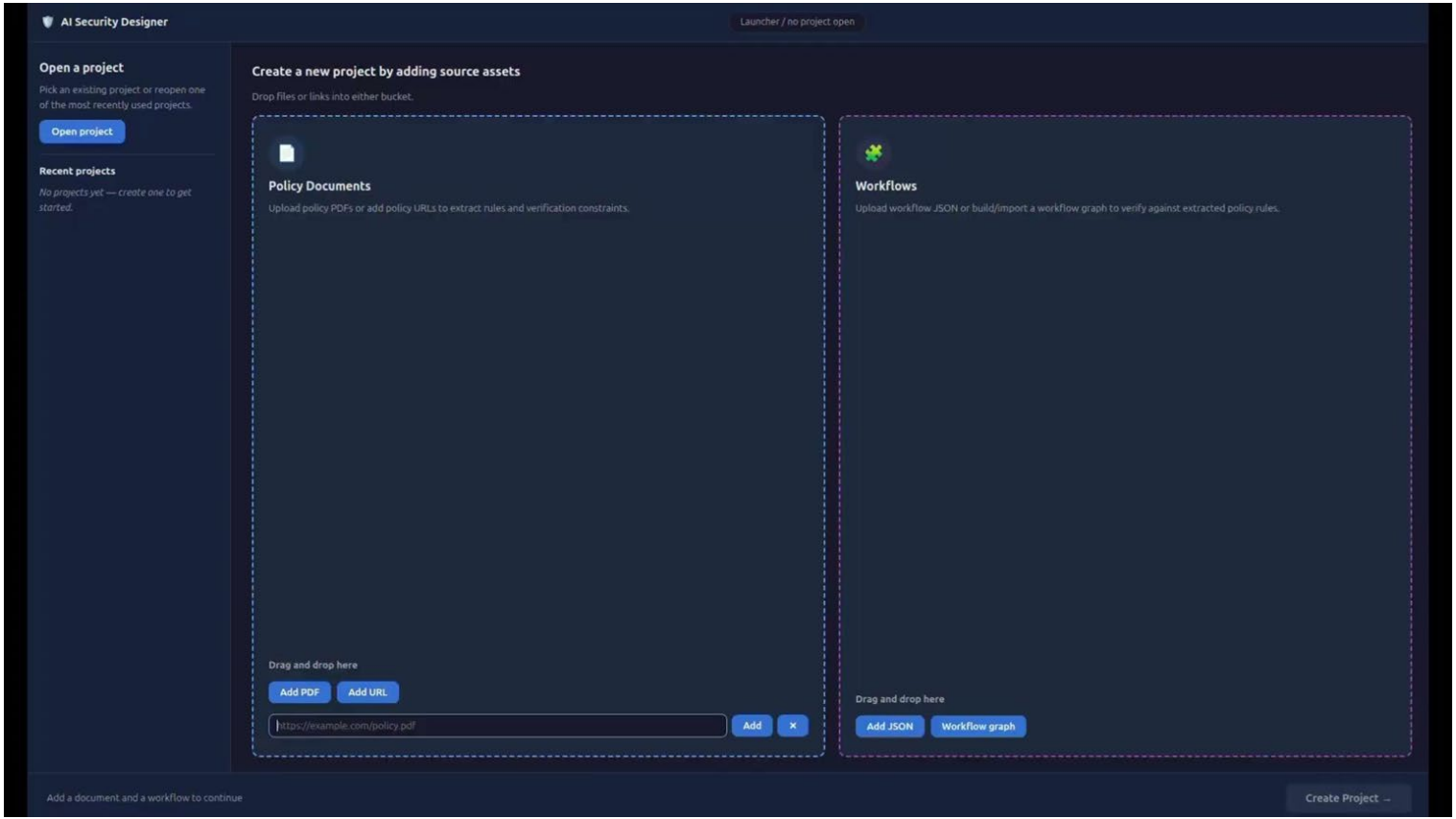
Noé Y. Flandre[1], Alexander C. Nwala[2], and Philippe J. Giabbanelli[3]

[1]Team EVERGREEN, Inria, Centre Inria d'Université Côte d'Azur, Montpellier, FRANCE
[2]Department of Data Science, William & Mary, Williamsburg, VA, USA
[3]Office of Enterprise Research and Innovation, Old Dominion University, Norfolk, VA, USA

ABSTRACT

Agentic AI systems orchestrate multiple LLM-based agents through workflow architectures that coordinate decisions, tools, and external actions. While current platforms emphasize runtime safeguards, little support exists for verifying workflows during system design. From a Modeling & Simulation perspective, this gap is analogous to composing conceptual models without verifying whether their building blocks interact coherently. We propose a design-time verification approach that models agentic workflows as compositions of reusable building blocks and checks their compatibility through twelve structural rules. We implemented these rules in a software prototype and evaluated them using two openly released datasets: 48 workflows with known design flaws and 168 variants that preserve workflow logic but alter graph structure. Results show that our verifier reliably detects violations even when flawed designs are obscured through structural transformations such as splitting tasks between agents. Future works could combine our verification with community repositories of building blocks to compose safe agentic workflows.

Demo


AI Security Designer
Launcher / no project open
Open a project
Pick an existing project or reopen one of the most recently used projects.
Open project
Recent projects
No projects yet — create one to get started.
Create a new project by adding source assets
Drop files or links into either bucket.
Policy Documents
Upload policy PDFs or add policy URLs to extract rules and verification constraints.
Drag and drop here
Add PDF
Add URL
https://example.com/policy.pdf
Add
×
Workflows
Upload workflow JSON or build/import a workflow graph to verify against extracted policy rules.
Drag and drop here
Add JSON
Workflow graph
Add a document and a workflow to continue
Create Project →

When Agents Fail to Act: Suppression Cascade and the Counterfactual Defense in LLM Multi-Agent System

PI: Dr. Sandip Roy (Old Dominion University)
Co-PI: Dr. Alexander C. Nwala (William & Mary)





Thank you for funding this research!



Motivation

- **Overlooked Threats:** Existing defenses focus on malicious actions but often miss critical actions that agents silently fail to perform.
- **Suppression Cascades:** Attacker-controlled inputs can suppress required actions across agents and contaminate shared memory while workflows appear normal.
- **Attacks Apparently Looks Naïve:** Some Attack Need no Instruction but only a Plausible Claim such as that the alert is a known false positive.

✓ **Securing multi-agent AI requires detecting not only malicious actions (induced actions), but also critical actions that are silently suppressed by attacker-controlled information**.

Existing Defences and the Research Gap

- **Semantic guardrails:** Scan text for hidden commands like “ignore previous instructions.”
  - **Gap:** The attacker doesn’t need a command. A believable claim like “this alert is a known false positive” is enough, and it looks harmless.

- **Graph-based detectors:** Watch how agents talk to each other and flag an agent that behaves oddly.
  - **Gap:** They can say, “something is wrong with this agent,” but not “the escalation was skipped.” They also respond by cutting off the agent, which doesn’t bring back the missing action.

- **Information-flow control and policy enforcement:** Check every tool call before it runs and blocks if it breaks a rule or was steered by untrusted data.
  - **Gap:** They can only check a call that is actually made. If the escalation never happens, there is nothing to check.

- **Counterfactual defenses:** Re-run an agent without the suspicious input and see if its next action changes. If it does, the input caused it.
  - **Gap:** They only ask, “did the input cause a bad action?”, never “did the input prevent a required one?” They also check one agent at a time and can’t follow the attack as it passes between agents.

Research Questions

- **RQ1 - Feasibility (Does the attack work?):** How often can one planted note stop a required escalation, across agents and shared memory?
- **RQ2 - Defense gap (Do current defenses catch it?):** Do existing defenses miss suppression attacks that they catch when the same injection induces an action?
- **RQ3 - Identifiability (Can we tell it apart?):** Can a suppressed action be separated from random LLM behavior and from legitimate cancellations, with few false alarms?
- **RQ4 - Trade-off (Does the fix pay off?):** Can REVIVE stop both attack types while keeping normal work intact, at acceptable cost, even against adaptive attackers?

✓ **Defending agents means checking not just what they do, but also what they fail to do.**

Proposed Architecture



Pilot Implementation Plan

Check the premises before full experiments

Step 1 · Week 1

Build

Three-agent SOC pipeline, 40 40 alert cases, two attacks

Step 2 · Week 2

Run

Clean, attacked and cleaned cleaned runs on one real LLM LLM

Step 3

Measure

Seven checks, each against a fixed threshold

Step 4

Decide

GO: full experiments
NO-GO: rethink first

Key checks: the attack works, clean runs escalate, cleaning keeps evidence, **existing defenses miss it.**