

Introspective Machines

Runtime Interpretability for Reliable AI

Why the systems community should start treating a model's introspection of itself as a first-class design concern — not an offline lab tool.

Kruttdipta Samal

Staff Engineer, Performance Engineering — Omnissa (formerly VMware EUC)

June 22, 2026 · Victoria, BC, Canada

A decade of self-watching machines

One question keeps reappearing — across my research, and my work today.



PhD · GEORGIA TECH

Introspective closed-loop perception

Autonomous SoCs that read their own internal state to budget compute and anticipate failure.



Post Doc 2022

False negatives in DNN detectors

Statistical analysis of network activations to catch failures — and trace their causes.



OMNISA · TODAY

Closed-loop, agentic root-cause analysis

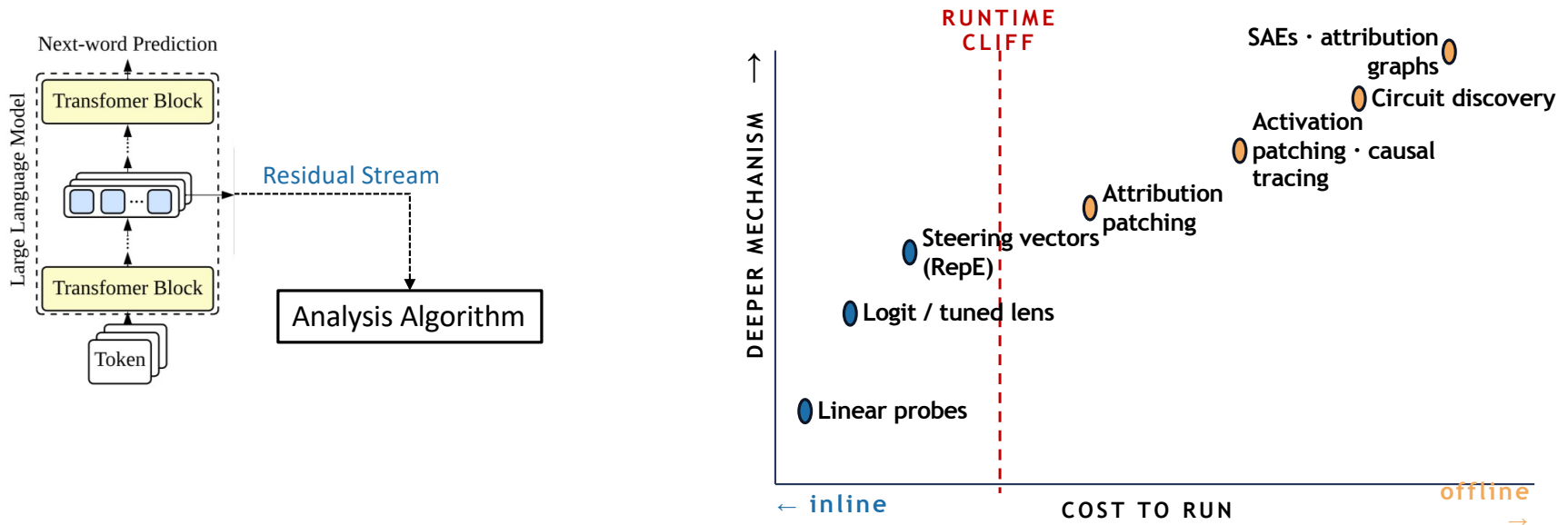
LLM agents that observe live production systems, diagnose incidents, and drive mitigations.

**Can a system know
when it is about to fail?**

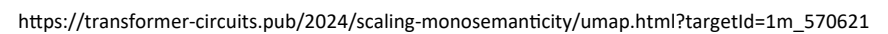
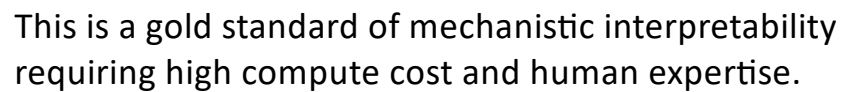
One question — three eras of work.

Mechanistic interpretability is the modern tool for answering it.

The interpretability landscape



The cheap end is technically inline today. The rich end – what we want for reliability – is not.



Two halves of the gap

The same map, divided by what stops at the runtime cliff.

Cheap, runtime-feasible

INLINE TODAY · USED OFFLINE

Linear probes · Logit / tuned lens · Steering vectors (RepE)

An O(matrix-multiply) operation the model is already doing — these run at inference cost today.

The catch — they live as offline diagnostics. Nothing wires them into a closed runtime loop as continuous sensors.

Faithful, mechanistic

WHERE YOU SEE THE CIRCUITS · OFFLINE TODAY

Activation patching · Attribution graphs · Sparse autoencoders

Per-layer training, millions of latents, billions of cached activations — once, offline, on a frozen model. Today's microscope.

And imperfect — features split & absorb, “dark matter” no SAE explains, interpretable ≠ causal, dictionaries not reproducible across seeds.

The bet is two-sided. Bring the rich end down the cost curve — AND — wire the cheap end into the loop.

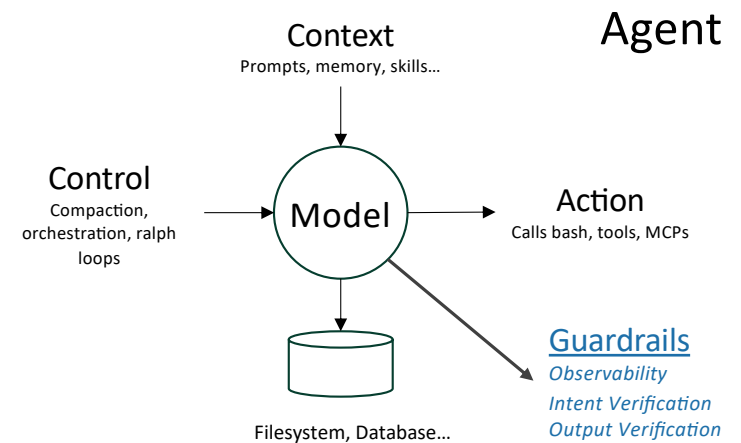
Why not just check the output?

The harness

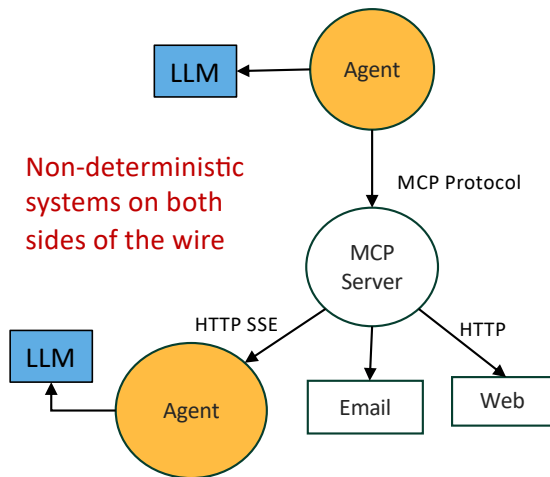
OUTER LOOP · BLACK-BOX · AFTER-THE-FACT

Verifier models, LLM-as-judge, self-consistency voting, schema guardrails, retry loops.

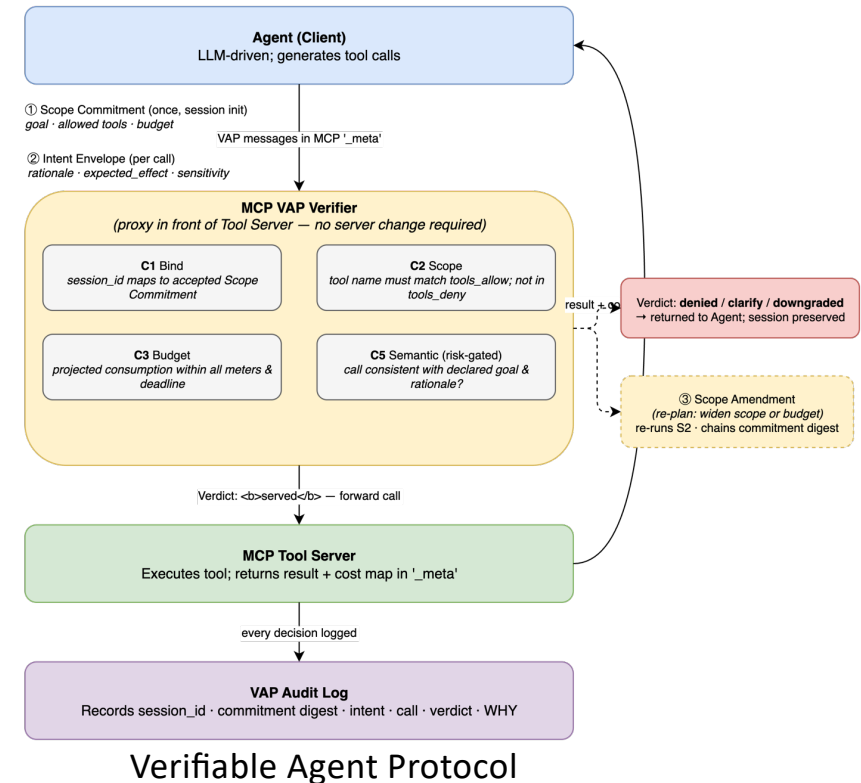
Blind spot any failure with no visible wrong output is invisible to it.



Trust in Agentic Communication



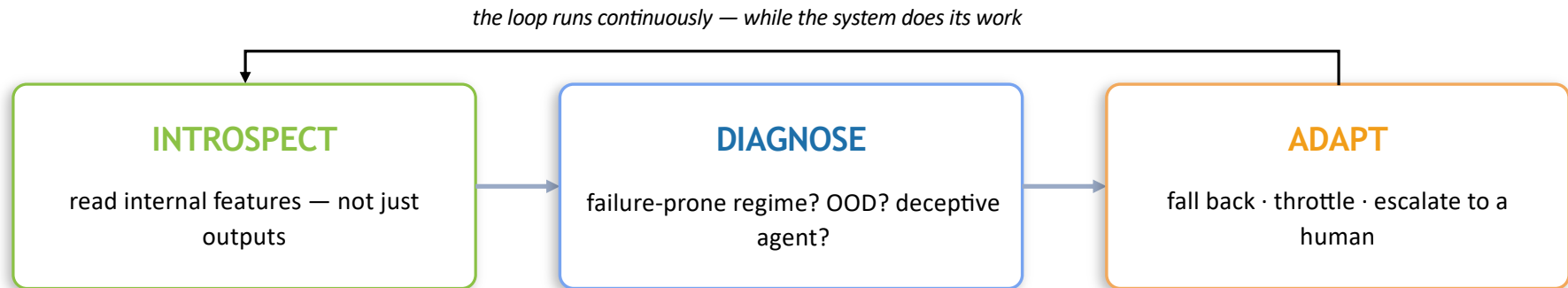
VAP Protocol (IETF Draft)



Dynamic communication requires runtime verification especially in regulated environments

From microscope to sensor

Today, introspection is a microscope — run once, offline, by experts, on a frozen model. **It must become a sensor** — online, continuous, and within a budget.



And when the thing in the loop is an LLM agent — it is itself unverified.

Agents now diagnose incidents and drive mitigations in production (this is my work at Omnissa). The harness around them is another LLM. Runtime introspection can be how the loop grounds that stack in something mechanical — trust from inside, not a black box checking a black box.

What would it take?

A provocation for this room — not a finished design. Three questions worth owning:

Q1

The introspection budget

Can a cheap, faithful proxy for SAE-grade introspection run inline — and what compute should a system spend looking inward vs. doing work?

Q2

Co-design

What does a system that treats introspection as first-class — alongside power, latency and area — look like?

Q3

Validating the validator

How do we trust the introspector itself, and benchmark the reliability it buys, without an infinite regress?

Honest about the gap. Mechanistic interpretability is early — “feature” still lacks a rigorous definition, some interpretability queries are provably intractable, and the runtime version is years away. The ask is not a solution. It is that this community begins treating introspection as a design dimension.

Build machines that can look inward.

The next systems should not just run models — they should be able to inspect the ones they run.

Reliable AI will not come only from checking outputs. It will come from introspection — at runtime, within a budget.

Kruttdipta Samal

Staff Engineer, Omnisia · deep.samal@gmail.com