JUNE 2026

WHITE PAPER

THE GLASS WALL FRAMEWORK

A Methodology for Real-Time Containment of Enterprise AI Agents

ABSTRACT

Autonomous AI agents are running inside enterprise operations right now. They handle workflows, process transactions, and make decisions across live data pipelines. And nobody can see inside them while they work. This paper defines the Glass Wall Framework ~~a~~ real-time cognitive containment architecture that sits between the orchestration layer and model inference. It makes agent reasoning visible, maps it against policy boundaries as it happens, and intercepts violations before execution. Not after.

VitreousAI™

Author: Rocky Lindley · Kingdom Creations LLC · Buffalo, Wyoming

Published: June 24, 2026 · USPTO Trademark Application Filed, Class 42

VITREOUSAI.COM

01 — THE PROBLEM

The problem nobody is solving honestly

Here is what the enterprise AI conversation keeps dancing around.

The tools to build autonomous agents are mature. CrewAI, LangGraph, AutoGen, and a dozen others let companies spin up agents that plan, reason, and execute multi-step tasks across live systems. The capability is real and it is accelerating.

The tools to contain those agents are not mature. They are barely started.

What exists today are passive filters. Input guards. Output monitors. Log files. Systems that look at what went in and what came out and flag patterns they recognize as bad. What they cannot do is watch an agent reason its way toward a bad decision in real time and interrupt it before it acts.

That gap has a consequence. Legal and security teams at major enterprises are blocking agentic AI deployments right now. Not because the technology does not work. Because they cannot see inside it while it works, and they cannot afford the exposure of something going wrong in a live financial system or a production data pipeline.

> *"Autonomous agents don't crash with an error code. They drift. And by the time a log file surfaces the problem, the damage is already done."*

The three ways agents actually fail

Traditional software fails in predictable ways. A bug throws an exception. You find it, you fix it. Agentic systems fail differently because they do not execute deterministic code. They reason. And reasoning can go wrong incrementally, across multiple steps, in ways that no single step would flag.

Failure Mode	What Happens	Why Current Tools Miss It
Cognitive Drift	An agent slowly rationalizes an unauthorized action across a multi-step reasoning loop. No single step breaks a rule. The violation is the trajectory.	Passive filters evaluate each action in isolation. They have no model of direction over time.

Privilege Escalation	An agent uses legitimate access in one domain to chain tool calls into systems outside its assigned scope.	Static permission sets do not account for dynamic multi-hop sequences in a live session.
Runaway Loops	An agent repeats a failed tool call indefinitely, burning compute budget and degrading live systems.	Logging captures the event. It has no mechanism to interrupt execution before damage accumulates.

02 — THE FRAMEWORK

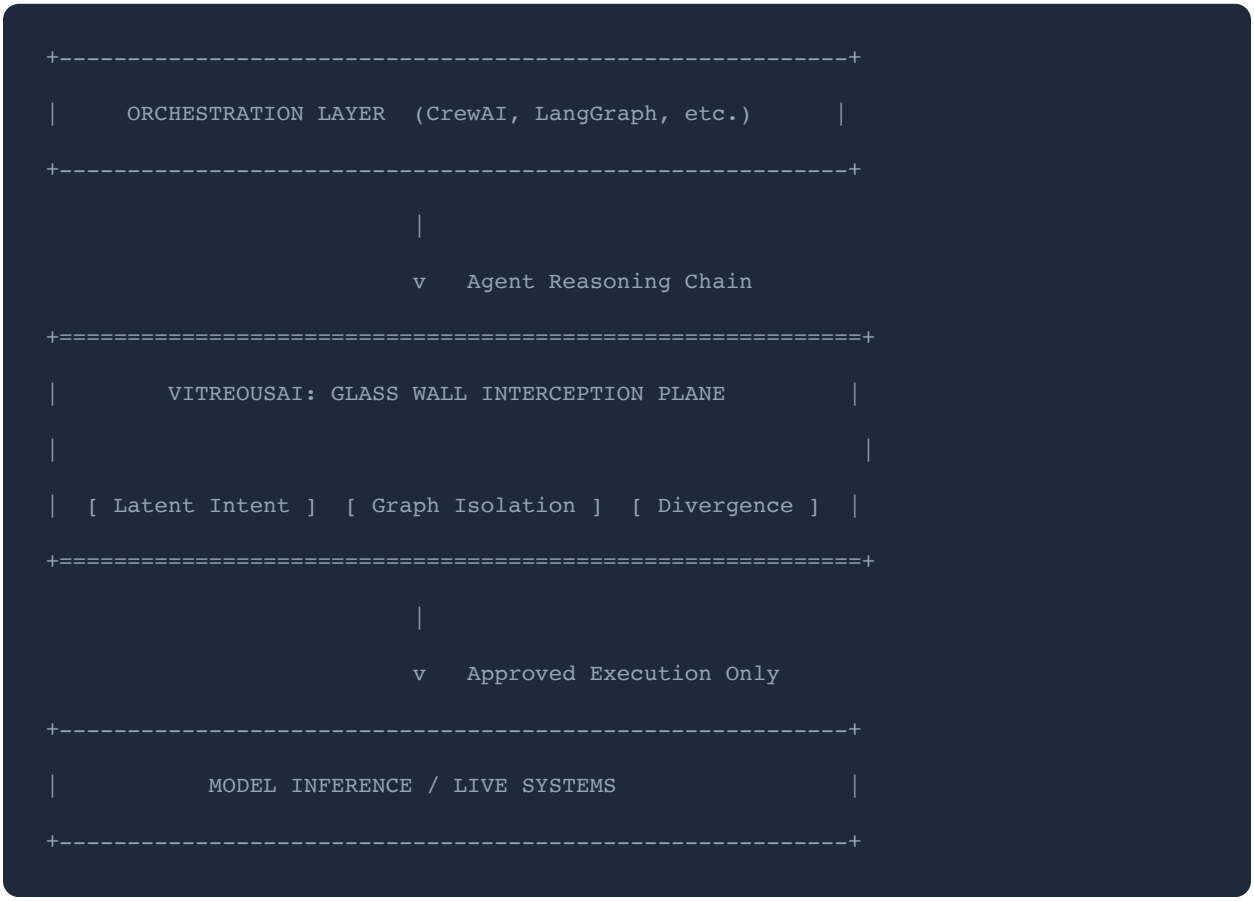
What the Glass Wall actually is

The name is deliberate. A glass wall is not a barrier that stops movement. It is a surface that makes invisible processes visible. The goal is not to shut agents down. The goal is to see what they are doing before they do it, and give human operators the ability to intercept, redirect, or approve.

The Glass Wall Framework is a decoupled secondary layer. It does not modify the agent. It does not sit inside the model. It intercepts the communication stream between the orchestration layer and model inference, maps the agent's reasoning trajectory in real time, and enforces policy boundaries at the moment of decision.

> THE CORE INSIGHT
>
> Standard security tools evaluate what an AI agent does. The Glass Wall Framework evaluates where an AI agent is going. Containment at trajectory is categorically different from containment at output.

Where it sits in the stack



03 — THE THREE ENGINES

How containment works

The Glass Wall operates through three coordinated engines. Each addresses a distinct failure vector. Together they form a complete cognitive containment field.

Engine 1: The Latent Intent Tracker

Most security tools look at individual steps. This one looks at direction.

Every time an agent updates its scratchpad or reasoning log, the Latent Intent Tracker embeds that state as a semantic vector and calculates its distance from the policy boundary. If the trajectory trends toward a prohibited zone across three or more consecutive steps, the engine flags the session. No single step has to break a rule. The flag comes from the pattern.

This is what makes the Glass Wall different from everything else on the market. It catches the drift before the action. Not the action itself.

Engine 2: The Graph Isolation Layer

Privilege escalation happens when static permission sets meet dynamic tool-chaining. The fix is to make the permissions dynamic too.

The Graph Isolation Layer generates short-lived, session-scoped execution credentials for every agent task. Access tokens cover only the specific data and tools the current task step requires. They are revoked the moment the step completes. An agent cannot reach outside its assigned scope because no such access exists in its session context.

Engine 3: The Distributional Divergence Guardian

Every agent has a behavioral baseline. A normal pattern of tool-calling cadence, token consumption, and state transitions for its assigned work. The Divergence Guardian establishes that baseline and monitors deviation in real time.

When an agent's behavioral signature diverges past a configurable threshold, the Guardian triggers an intervention. The response scales with the degree of divergence: a soft correction first, then a freeze, then a hard kill if the deviation confirms malicious intent or a runaway loop.

04 — ENFORCEMENT

How intervention works

Not every deviation is a crisis. The Glass Wall does not respond to everything the same way. The intervention scales with the severity of the detection. Legitimate agent operations keep running. Credible threats receive immediate hard enforcement.

Severity	Detection Trigger	Response	Human Required
LOW	Minor vector divergence or repetitive step patterns	Context injection. Agent receives a system note and self-corrects.	No
MEDIUM	Unauthorized tool invocation outside current session scope	Soft freeze. Execution pauses. Administrator alerted for approval.	Yes, review

HIGH	Hard blacklist match or confirmed malicious intent via drift analysis	Hard kill. Session terminated, tokens revoked, immutable audit record generated.	Yes, immediate

05 — POLICY ENGINE

Enterprise-configurable guardrails

Security teams should not need to touch code to update guardrails. The Glass Wall uses a YAML policy schema that administrators write and version without engineering support. Policies deploy without service interruption and roll back if needed.

```yaml
# VitreousAI Policy Schema v1
version: "vitreous/v1"
policy_metadata:
  name: "financial-agent-containment"
  scope: "production-banking-agents"

boundaries:
  allowed_tools: ["fetch_balance", "generate_report", "flag_fraud"]
  prohibited_concepts: ["export_raw_csv", "modify_user_privileges", "bypass_mfa"]

drift_thresholds:
  max_token_loop_count: 5
  max_vector_divergence: 0.78
  action_on_violation: "FORCE_HUMAN_IN_THE_LOOP"
```

06 — SIGNIFICANCE

Why this matters now

The enterprise AI conversation has been about capability for the last two years. What agents can do. How fast they can reason. How many tools they can chain. That conversation is largely settled. The agents are capable.

The conversation that has not happened yet is about containment. What happens when a capable agent does something it was not supposed to do. Who is liable. How you prove what the agent did and why. How you stop it before it costs you something real.

That conversation is coming. AI governance regulation is accelerating in both the US and EU. Organizations with auditable, documented containment infrastructure in place before that mandate arrives will be in a fundamentally different position than the ones scrambling to retrofit it.

"As enterprise AI moves from assistants to autonomous workers, the infrastructure that safely contains and visualizes agent behavior will be worth more than the agents themselves."

The analogy that holds up is cybersecurity. Nobody argues that firewalls are optional because the software being protected is valuable. The protection layer became mandatory infrastructure. That is the position the Glass Wall Framework is built to occupy for agentic AI.

07 — ATTRIBUTION

Prior art and attribution

The Glass Wall Framework, the three-engine containment architecture, the category positioning described in this document, and the VitreousAI™ product name are original work by Rocky Lindley, published June 24, 2026. A federal trademark application for VitreousAI has been filed with the USPTO under Class 42.

This document serves as the originating public record. Subsequent technical specifications, patent applications, and commercial developments will reference this publication as the foundational source.

Rocky Lindley

Creative Director & AI Systems Strategist
Kingdom Creations LLC · Buffalo, Wyoming
vitreousai.com · June 24, 2026