Before Execution:
A Typed, Calibrated Gate for Agent Actions

Ricardo Martínez
Cloud Security & Applied AI — NullGhost Research

September 2026

Abstract

An AI agent that decides its next action through free text produces, alongside that decision, no reliable signal of how confident the model actually is in having chosen well — and asking a second conversational model to audit that output is too slow and too expensive to sit in front of every step an agent takes. This text describes an architecture pattern — a typed output gate with calibrated confidence, evaluated here through the case of TypeSafe AI and its Jev model [1, 2, 3] — and carefully separates what is verifiable from what the vendor self-reports. It closes with design principles for adopting the pattern independent of any vendor, and a minimal illustrative code file that implements it.

1 A decision with no confidence signal

An agent built on a conversational language model decides its next action by generating text: a tool choice, a classification, a verdict on a finding. That output can be wrong, it can be manipulated by adversarial content the agent processed as part of its context, and — this is the point that matters — it carries no reliable measure of how much it should be trusted.

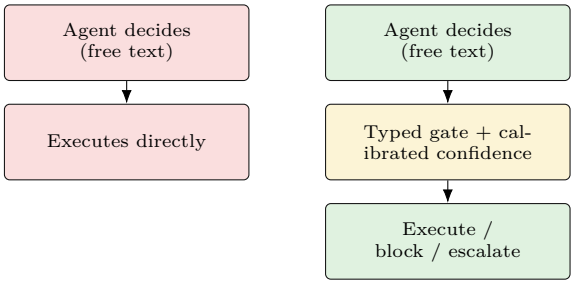

Figure 1: Without a gate, the agent's decision executes directly (left). With a typed, calibrated gate in between, execution depends on a second structural check, independent of the model that decided (right).

Asking a second conversational LLM to audit the first one's output is an obvious but expensive fix: 3 to 30 seconds of latency per check, at a token cost comparable to the original decision — unworkable if an agent runs dozens of steps per task.

2 A case study: "System One" models

TypeSafe AI, a lab founded in 2024 that exited stealth on September 15, 2026 with a $40M seed round led by DCVC [3], proposes a distinct category: models that return **typed values from a predefined schema** — classify, route, score — instead of free text, with a calibrated confidence score per field. Its first product, Jev, is trained with a proprietary method (RLCD, *Reinforcement Learning for Calibrated Decisions*) that optimizes for reported confidence matching real accuracy, rather than for human preference.

Verifiable	**Self-reported**
Published pricing, measurable latency, schema conformance (a constructive guarantee)	"193.6× faster, 444.6× cheaper" — vendor's own evals, no independent reproduction

Figure 2: How verifiable each type of claim about Jev is. TrueFoundry's technical analysis [2] explicitly flags the performance comparison as biased by the vendor itself.

The distinction in Figure 2 is not a minor detail: adopting an architecture pattern for its structural properties (bounded output, calibrated confidence, low latency) is an engineering decision; adopting it for a vendor's marketing figure is betting on a number nobody else has measured.

3 The pattern: a gate before execution

Three properties make this pattern viable as a security layer for agents, independent of which specific model implements them:

- **Schema conformance.** A gate that can only return a value from a predefined set cannot "hallucinate" an action outside that set — a structural constraint, not a heuristic.
- **Calibrated confidence.** A confidence threshold enables a simple operational rule: below a certain level, the action gets blocked or escalated to human review instead of executing blindly.
- **Low latency.** A 70–500ms gate can sit in front of every action an agent takes without breaking its latency budget; a 3–30s one cannot.

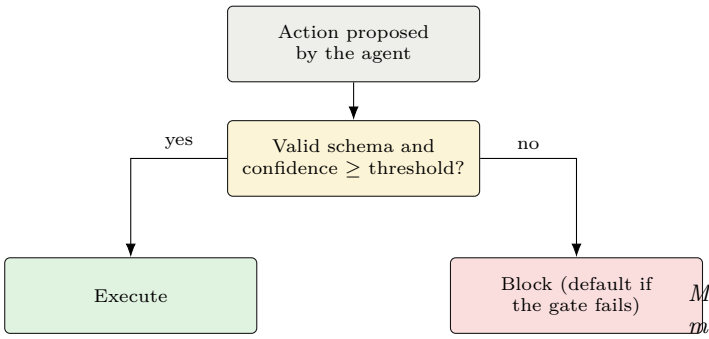

Figure 3: The pattern implemented in `patterns/01_typed_calibrated_gate.py`: if the gate itself fails (exception, unknown schema), the default outcome is to block, never to execute silently.

The file `patterns/01_typed_calibrated_gate.py` in this repository implements exactly Figure 3 in minimal, runnable form, with no external dependencies — written from scratch for this text, not extracted from any production system.

4 Where it doesn't help

- **It does not replace a deterministic scope control.** Validating that an action falls within the allowed schema is different from validating that the action's target is authorized; both controls are necessary, neither replaces the other.
- **Limited cardinality.** A decision schema with hundreds of possible categories may require a slower two-stage approach than the simple case.
- **No independent benchmark for adversarial-intent detection.** A vendor listing this as a use case is not the same as a third party measuring an accuracy rate.
- **Early-vendor risk.** A two-year-old provider in early access, with no self-hosted option yet, is a dependency to evaluate with the same scrutiny as any new external vendor.

5 Design principles

1. A gate that fails open is not a gate — the default on any error must be to block, never to execute.
2. Confidence without calibration is theater: a number that doesn't track real accuracy informs no decision.
3. Typed output is a structural guarantee, not a promise of correctness — it eliminates the wrong format, not the wrong category.
4. Don't adopt the vendor's benchmark as your own; measure calibration against your own known cases before trusting it.
5. This complements an existing deterministic scope control; it does not replace it.

Methodological note: this text evaluates a third-party commercial product (TypeSafe AI / Jev) based on the public sources cited below. It does not describe or depend on any production system belonging to NullGhost or KIRUX; the file under `patterns/` is original illustrative code, not extracted from a real system.

References

[1] TypeSafe AI. *Introducing System One Models & Jev.* typesafe.ai/blog, September 2026.

[2] TrueFoundry. *TypeSafe AI's Jev: What "System One Models" Actually Are.* truefoundry.com/blog, 2026.

[3] SiliconANGLE. *TypeSafe AI exits stealth with $40M to build AI for use by software.* siliconangle.com, September 16, 2026.