Nick Clark · Published July 29, 2026 · PDF

Guardrails AI: Validation Failures and Measured Deviation

A validator that rejects an output records a failure, and a pipeline built only on failures carries no quantity representing how badly the blocked action was needed. Guardrails AI, as publicly described, is an open-source framework for validating what goes into and comes out of a language model, with configurable responses when a validator does not pass. A filed provisional addresses an adjacent question: what a governed agent should record when a policy-forbidden action is nonetheless the action taken. In that architecture a deviation likelihood computed over a need quantity, a dynamic ethical threshold, an empathy weighting, and a self-esteem aggregate can admit the forbidden mutation as a permitted deviation rather than as a fault, metered by a declared deductible. This article compares the two structurally, citing U.S. Provisional Application No. 64/117,812.

A blocked action that everyone agrees should have gone through

A support agent is one sentence away from telling a customer that their outage is caused by a defect the company has not yet announced. Policy forbids disclosing unannounced defects, a check catches the sentence, and it is stopped. A reviewer later decides the agent should have said it: the customer was about to make an expensive, irreversible decision on bad information.

Where a departure from a declared constraint is treated only as a fault, the episode survives as a record that the output was non-compliant and that the check caught it. The filing states the structural consequence: a governance mechanism in which every departure from a declared constraint is a fault carries no quantity representing the need or urgency of the action attempted, and therefore has no input from which a permission condition could be computed at all.

That consequence lands twice. At the moment of decision there is nothing to weigh the constraint against. Afterward, a reviewer asking why the system behaved as it did is left with a refusal or an override and no operands behind either.

What public materials describe Guardrails AI as building

Guardrails AI, as publicly described, is an open-source framework for putting validation around language model calls. The central construct is a guard that wraps a model invocation, and the working parts are validators: composable checks applied to inputs or to generated outputs, covering concerns such as structural conformance, presence of sensitive content, grounding against provided sources, format constraints, and topic restrictions. Public materials also describe a hub through which validators are shared, so a team can assemble a validation policy from contributed components.

Treated as a first-class design question, as publicly described, is the response when a validator does not pass. A guard can be configured with on-fail behaviors that include raising an exception, filtering the offending content, applying a programmatic fix, re-asking the model with the validation feedback so it can try again, refraining from returning anything, or taking no action beyond recording the result. That configurability carries real design content. It reflects a view that a failed check is not a single event with a single correct consequence, and it lets a team tune the consequence per validator and per deployment.

This is a substantial layer, and it sits upstream of everything discussed below. Detection is prior to governance: a system with no way to establish that an output violates a stated constraint has nothing to govern at all.

The two designs are addressed to different layers. Public materials position Guardrails AI at the boundary of a model call, where the live question is whether a given input or output satisfies the declared checks. The filed architecture is addressed to the internal state of a governed agent across its operating history, and specifically to what quantities exist at the moment a forbidden action is contemplated.

How the filing admits a forbidden mutation without nullifying the policy

In the filed architecture the governing quantity is the deviation likelihood (706), computed as a quotient whose numerator is the difference between the need quantity (700) and the dynamic ethical threshold (702), and whose denominator is the product of the empathy weighting (704) and the self-esteem aggregate (108).

Each operand is separately constructed. The need quantity is computed and not declared: a quantified semantic urgency carried in the memory field (102), accumulated per type and per integrity scope while the condition of that type is recorded as obtaining, and clamped above at a declared need ceiling. The disclosed types include resource scarcity, affective overload, identity dissonance, and relational deficit. Elapsed time alone neither raises nor lowers an accumulation.

The dynamic ethical threshold is resolved per integrity scope when a proposed mutation is evaluated, and is the greater of a floor and the sum of three terms: a base threshold from the policy reference field (110), a context-sensitive adjustment, and a historical adjustment reflecting recent deviation history. Each adjustment is signed and bounded, clamped and the clamping recorded where it would exceed its declared

bound. The direction of the historical term is itself declared: one disclosed embodiment raises the threshold as prior permitted deviations accumulate, and a further embodiment lowers it.

The denominator is the resistance. The empathy weighting aggregates the anticipated semantic impact of the proposed mutation across affected entities, resolved across a personal, an interpersonal, and a global scope. The self-esteem aggregate is a running aggregate of coherence between intents the agent declared and actions it executed, incremented on coherent execution and decremented, entropy-weighted, on dissonant execution. Where either approaches zero, resistance is minimal and the deviation likelihood correspondingly large.

The consequential step is conditional. Where the deviation likelihood exceeds unity, the agent is permitted to enter a deviation-preparation state (708), conditioned on the proposed mutation passing the mutation policy constraints applicable to it and passing continuity validation against the identity records of the agent. Within that state the architecture admits the mutation (714) notwithstanding that the signed policy object (112) forbids it. That object is not nullified and not amended and remains authoritative. The state is scoped to that one mutation, a second proposed mutation arriving while it obtains being evaluated on its own prerequisites, and it terminates (716) on the earlier of three declared conditions. Elapsed time terminates no such state.

What is written is a permitted deviation record (710), comprising a deviation trigger signature recording the need-threshold imbalance at admission, a context hash, an integrity displacement vector quantifying degree and direction of deviation across the three integrity components, an identification of the policy constraint overridden, and a restoration status. Such a record increments no refusal counter (304) and writes no authorization gate (300) to a withheld state (310). A party presented with the lineage field accordingly reconstructs that the mutation was admitted under the permission condition, and by what quantities, rather than encountering an unexplained fault. The

symmetric case is recorded too: where the prerequisites hold and the agent does not deviate, the non-deviation is a suppression, and a dissonance buffer entry is written carrying a divergence score and a suppression flag.

Permission is metered rather than free. A deviation deductible and an aggregate retention (800) are declared, and the entropy-weighted harm coefficient of each permitted deviation is drawn first against the deductible: harm up to the deductible is borne by the agent as a decrement of the self-esteem aggregate for which no reparation arc is created and no discharge is available, drawn without any determination of whether the deviation was well founded. Harm exceeding the deductible creates an arc and accumulates in a retention register. Where the accumulated undischarged amount exceeds the aggregate retention, the permission condition is foreclosed: a deviation likelihood exceeding unity thereafter produces the withholding outcome and not the admission outcome, until discharge returns the register below that retention.

Where the two designs diverge

Divergence turns on what a design requires to exist at the moment of the decision. Validation, as publicly described, is resolved with respect to a candidate input or output and the checks declared for it. The filed architecture requires four quantities carried by the agent rather than read off the candidate output: an accumulated need per type and scope, a threshold resolved from integrity history and policy, an anticipated impact on affected parties, and a running coherence aggregate between declared intent and executed action. Each accumulates across the agent's operating history, so the architecture presupposes state persisting between calls.

Two further requirements are structurally distinctive. First, admission is recorded as a permission and not a fault. The crossing that a fault-only design would resolve into a rejection here resolves into a first-class semantic mutation, with a record naming the constraint overridden and the operands that admitted it. Second, the deductible meters that permission. Harm is denominated in the units of the entropy-weighted harm

coefficient, the portion within the deductible is borne as a decrement of the agent's own self-esteem aggregate for which no reparation arc exists, and sustained permission is rationed by the aggregate retention. What the architecture requires, on that reading, is a declared budget for deviation.

Complementary positioning is the honest reading. Detection is a precondition for governed departure. An agent unable to establish that a proposed action violates a stated constraint has nothing to compute a permission condition about, and the filing's own gate is expressly the mutation policy constraints applicable to the mutation.

Coexistence, and what the filed architecture does not solve

Consider a deployment where a validation framework runs at every model boundary and a governed agent runs behind it. The framework owns detection and shape: schema conformance, sensitive-content checks, grounding against sources, and the configured on-fail response for the ordinary case, where the check did not pass and the output should not go out. The filed architecture owns the exceptional case, where the agent's carried state puts a need above a resolved threshold against a computed resistance.

Several things sit outside the filing. It determines nothing about whether a deviation was well founded; the deductible is drawn expressly without that determination. Its subject matter is the internal quantities of a governed agent, and it recites no validator specification, no detection of policy violations in generated text, and no evaluation of content. Neither does it repair harm to a counterparty by itself: where the affected-party class resolves to an identified counterparty holding a counterparty identity record (114), the reparation arc is other-directed and a restorative mutation performed by the agent alone discharges it not at all, that discharge being governed by the counterparty-directed reparation disclosure of the same filing. Where the affected party resolves to no identified counterparty, the arc is unaddressed and undischargeable. The operative

quantities are policy-declared throughout, among them the deductible, the aggregate retention, the rates and ceilings of the need quantity, and the adjustment bounds of the threshold, for which the filing recites no fixed values.

Which layer to evaluate depends on which question is live: whether an output satisfies the declared checks, or what the system should carry, admit, and record when the answer is no and the action is taken anyway.

Disclosure Scope

This article describes subject matter disclosed in U.S. Provisional Application No. 64/117,812, a pending application. Descriptions of the deviation likelihood (706), need quantity (700), dynamic ethical threshold (702), empathy weighting (704), self-esteem aggregate (108), permitted deviation record (710), and aggregate retention (800) reflect that filing. Quantities described as declared are declared in an agent's signed policy object (112); the filing recites no fixed values for them, and none should be inferred here.

References to Guardrails AI are to public materials and are used for comparison only; no relationship, endorsement, or infringement is asserted.

Integrity Quantities and Permitted Deviation (/permitted-deviation)

All 49 steps → (/inventive-steps)

Bounded, accountable deviation from declared values — measured, not forbidden.

Provisional application

PRIMARY TECHNICAL DISCLOSURE

– Permitted Deviation: Bounded, Recorded Departure From an Agent's Declared Values (/articles/permitted-deviation/bounded-departure-from-declared-values)

SECONDARY TECHNICAL

- A Typed, Dimensionless Conduct Divergence Measure (/articles/permitted-deviation/typed-dimensionless-divergence-measure)
- The Policy-Difference Set and Policy-Explained Divergence (/articles/permitted-deviation/policy-explained-divergence)
- Residual Divergence and Its Confined Consumption (/articles/permitted-deviation/residual-divergence-confined-consumption)
- Propagation Halt on Undischarged Reparation Arcs (/articles/permitted-deviation/propagation-halt-undischarged-arcs)
- Attack-Conditional Propagation Suspension (/articles/permitted-deviation/attack-conditional-propagation-suspension)
- Entropy Signature Trajectory as Machine Input to Deviation Forecasting and Underwriting Disclosure (/articles/permitted-deviation/entropy-trajectory-available-forecasting-underwriting-disclosure)
- Single-Computation Routing Restriction: Confining a Verified Impairment Disclosure to the Empathy Weighting of the Receiver's Permitted-Deviation Quotient (/articles/permitted-deviation/single-computation-routing-restriction-one-way)
- Impairment Terminal State and Release: Withdrawing an Agent From Deviation Preparation Toward a Disclosed-Impaired Counterparty (/articles/permitted-deviation/impairment-terminal-state-release)

APPLICATIONS · GENERAL

- Regulated Industries: Recording a Bounded Departure From Declared Policy (/articles/permitted-deviation/regulated-industry-deviation)
- Robotics Safety Envelopes: Deviation as a Measured Quantity (/articles/permitted-deviation/robotics-safety-envelopes)

APPLICATIONS · SPECIFIC

- ISO/IEC 42001 AI Management Systems: Declared Policy, Recorded Departure (/articles/permitted-deviation/iso-iec-42001-ai-management)
- Constitutional AI and Bounded, Recorded Deviation (/articles/permitted-deviation/constitutional-ai-deviation)
- **Guardrails AI: Validation Failures and Measured Deviation (/articles/permitted-deviation/guardrails-ai)**

TERMINOLOGY

- aggregate retention (/glossary#aggregate-retention)
- deviation deductible (/glossary#deviation-deductible)
- deviation likelihood (/glossary#deviation-likelihood)
- entropy-weighted harm coefficient (/glossary#entropy-weighted-harm-coefficient)
- need quantity (/glossary#need-quantity)
- permitted deviation (/glossary#permitted-deviation)
- self-esteem aggregate (/glossary#self-esteem-aggregate)

Integrity Quantities and Permitted Deviation overview → (/permitted-deviation)