

Drift Origin Model

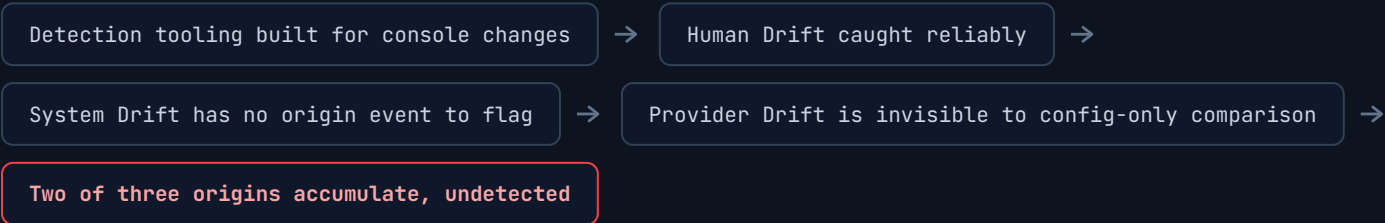
DEFINITION

A classification of where IaC drift originates — console changes, emergency fixes, automation side effects — so drift is designed for detection rather than assumed prevention.

FRAMEWORK FLOW

01 The Condition IaC drift emerges from three structurally different origins — human action, autonomous systems, and provider-side changes — yet is commonly collapsed into a single "someone changed something" category.	02 The Boundary Whether remediation is matched to the origin — enforcement and culture for Human Drift, detection architecture for System and Provider Drift — or every event gets the same response.	03 Failure State Single-Origin Blindness — a detection system built to catch console changes, solving for one origin while two others accumulate silently.	04 Consequence Enforcement gets tightened against a problem it was never capable of solving, while the other two origins compound until they surface as an incident.
---	--	---	---

Failure State: Single-Origin Blindness



THE DIAGNOSTIC SIGNAL

ORIGIN ATTRIBUTION

Ask whether a given drift event can be attributed to human, system, or provider origin within an hour. For most teams, only Human Drift reliably can.

COMPOSITE COVERAGE

Ask whether detection architecture covers all three origins simultaneously — usually no, because tooling was built to catch console changes, not autonomous reconciliation or provider-side mutation.

TRIGGER CONDITION

Any environment running autonomous controllers or provider-managed services introduces System and Provider drift possibilities by default — human-only detection tooling was never going to catch it.

REMEDIATION SHAPE

Detection-first architecture matched to origin: enforcement and culture for Human Drift; continuous reconciliation, behavioral baseline tracking, and origin-aware detection for System and Provider Drift.

THE THREE DRIFT ORIGINS

01 Human Drift Caused by people bypassing the declared control plane, intentionally or accidentally. The one category enforcement and culture can actually address.	02 System Drift Caused by systems with their own execution authority — controllers, autoscalers, autonomous remediation — operating concurrently with IaC. Pipeline enforcement cannot address it.	03 Provider Drift Caused by the provider changing something, with no action from the organization. Invisible to tools that only compare resource configuration state.
--	---	--

ARCHITECTURAL RELATIONSHIPS

#162

Security Entropy Accumulation

DEPENDENCY STRONG

Security Entropy Accumulation depends on this taxonomy to identify degradation where the originating drift event cannot be traced.

ANCHOR /iac-drift-detection/

FRAMEWORK HOME

The traceable-origin taxonomy this framework establishes is the baseline other drift and entropy frameworks build on.

"Every origin category can be correctly detected on its own terms. Nobody asks whether all three are covered at once — not because anyone neglected it, but because most detection architecture was only ever built to catch one."