CNCF PLATFORM ENGINEERING DAY · KUBECON NA

Compliance Drift on Kubernetes

ACRA · Agentic Control & Runtime Assurance

Four Kubernetes-native patterns, and the control plane that reads the composition.

01 · Kyverno gate
02 · OTel proof
03 · CRD controller
04 · Auditor evidence

Saideep Navakoti · Nov 9, 2026

A TRUE-TO-LIFE STORY

Every call returned 200. Every agent passed its own check. The approval at the end was one nobody authorised.

3 agents 2 clouds 1 document → counted as 3 sources

Combinative drift.

THE PROOF PROBLEM

Regulators ask you to prove it. For AI agents, CI/CD can't.

`same image · same manifest`

→ a different answer

`git diff · 0 changes`

→ behaviour moved anyway

No settled prescriptive rule for AI agents yet, so **the proof has to come from your own controls.**

RUNNING EXAMPLE · ONE TASK, FIVE LAYERS

1

Intake

A request and its documents

2

Cluster

Queues, services, agent workers

3

Agents

Extract, enrich, propose; a human approves

4

Models

Hosted, through one gateway

5

Records

Written back; downstream takes over

WHY SINGLE-LAYER CHECKS MISS IT

Every layer green. The composition wrong.

✓ Intake | ✓ Cluster | ✓ Agents | ✓ Models | ✓ Records = ✗ an approval nobody authorised

Agents
propose

→

ACRA GATE

reads the composition
admits the write — or holds it for a human

→

Records
only admitted writes land

WHAT YOU ALREADY RUN

You already run the tools. None of them read the *composition*.

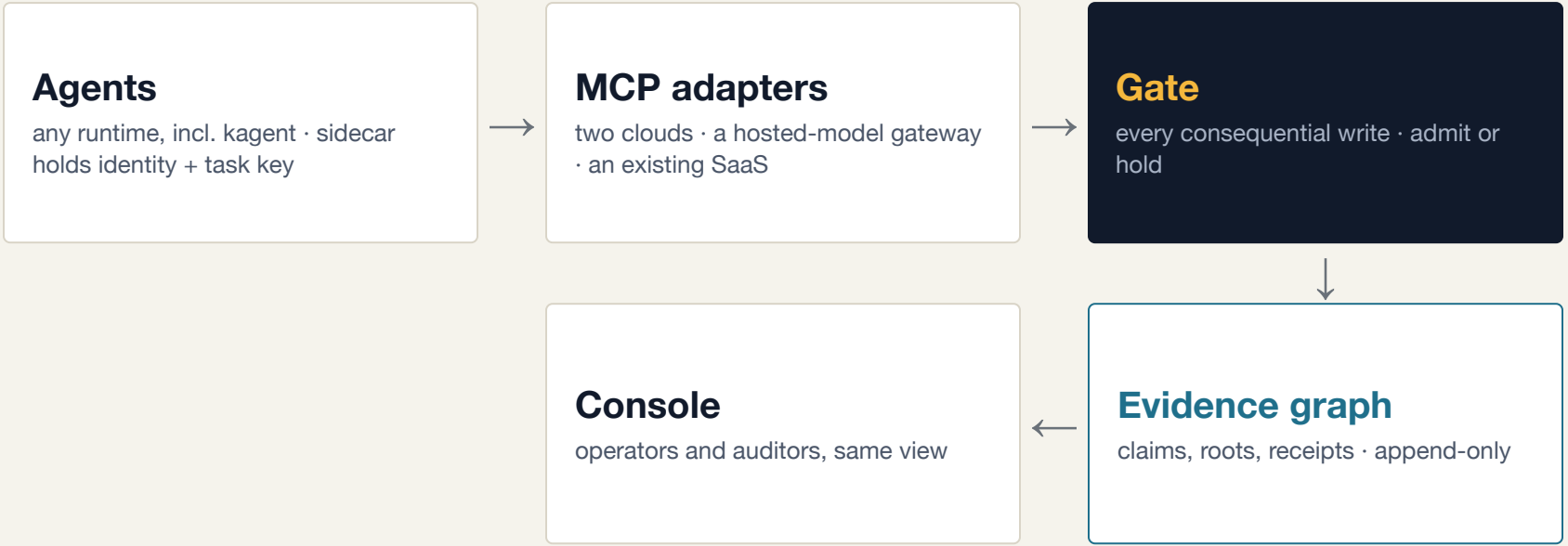
TOOL	SECURES	CAN'T TELL YOU, ALONE
Kyverno / Gatekeeper	`pod admission`	whether the *composition* was authorised
OpenTelemetry	`correlated traces`	that an effect was *authorised* — only that it happened
OPA / Rego	`policy decisions`	needs trusted business facts at the boundary
SPIFFE / SPIRE	`workload identity`	identity ≠ a current, task-scoped authorisation
AI gateway (Kong)	`model routing`	route success ≠ business approval

ACRA **reads the whole chain — did the composition stay authorised, and can you prove it?**

A green admission check is not business truth. A successful trace is not an authoritative receipt.

ACRA AT A GLANCE

Agents propose. Adapters wrap what exists. The gate admits or holds.



STABLE? OTel GenAI = Development kagent = Sandbox, alpha no settled US agent rule

PATTERN 1 · KYVERNO BEHAVIORAL-BASELINE GATE

Kyverno enforces the approved set. The gate decides drift.

```yaml
apiVersion: policies.kyverno.io/v1
kind: ValidatingPolicy
metadata: {name: agent-tuple-approved}
spec:
  validationActions: [Deny]
  validations:  # approved = signed list
  - expression: "object.metadata.labels
      ['acra.io/tuple'] in variables.approved"
```

KYVERNO · ADMISSION

Is this tuple on the signed list?

GATE + REGISTRY · RUNTIME

Does the live tuple still match what was agreed?

A TUPLE NOBODY APPROVED

prompt · manifest · rules · formula — each green; the combination never tested

A green admission check is not business truth — Kyverno enforces the approved set; the gate decides whether the composition drifted.

Legacy ClusterPolicy removed in Kyverno 1.20 → CEL types: ValidatingPolicy · GeneratingPolicy

PATTERN 2 · OPENTELEMETRY CONTINUOUS PROOF

Every span carries proof, not just timing.

`span · decision-agent.propose` → same IDs on every claim in the evidence graph

Operational correlation IDs

join records across systems

`task` `attempt` `step` `epoch` `trace` `SPIFFE caller`

ACRA proof attributes

what the model saw and said · custom

`model version` `prompt hash` `context sources` `response fingerprint`

A span proves it *happened* — not that it was *authorised*. The proof needs the gate's verdict, not just the trace.

OTel GenAI semantic conventions: emerging / experimental — not stable yet

PATTERN 2 · THE SPAN SCHEMA

Two ID sets on one span, kept apart.

```
span: decision-agent.propose
  # operational correlation IDs
  task_id=t-8c21  attempt=2  step=3  epoch=12
  trace_id=4bf9…  spiffe=…/ns/agents/sa/decision
  # ACRA proof attributes (custom)
  acra.model_version=<dated id>  acra.prompt_hash=7d1e…
  acra.context_sources=[doc-4411, e-7]
  acra.response_fp=sha256:c02a…
```

JOIN KEY

Kong log · kube event · gate record → one task

PROOF

Replay and diff what the model saw and said

PATTERN 3 · CRD CONTROLLER

Declared vs live: drift becomes a re-approval, not a silent fix.

```
apiVersion: acra.io/v1
kind: AgentPolicy
metadata: {name: decision-agent,
  namespace: agents}
spec:
  bundle: sha256:9f1c…   # approved tuple
  tools: [{server: lookup-tools, scopes: [lookup]}]
  budget: {tokens: 120000, calls: 6}
```

RECONCILE

labels · egress policy · Kong consumer

WATCH

restarts · eBPF connects · policy reports

STATUS · drift: 1 open

→ **re-approval requested**

PATTERN 3 · THE RESPONSE LOOP

Every drift walks the same loop. The agent never holds the keys.

1 **Detect**

2 **Hold**
reverses nothing

3 **Investigate**

4 **Authorise**

5 **Remediate**

6 **Verify**

HOW A FIX LANDS

Propose
investigator, exact change

→

Validate
Kyverno dry run

→

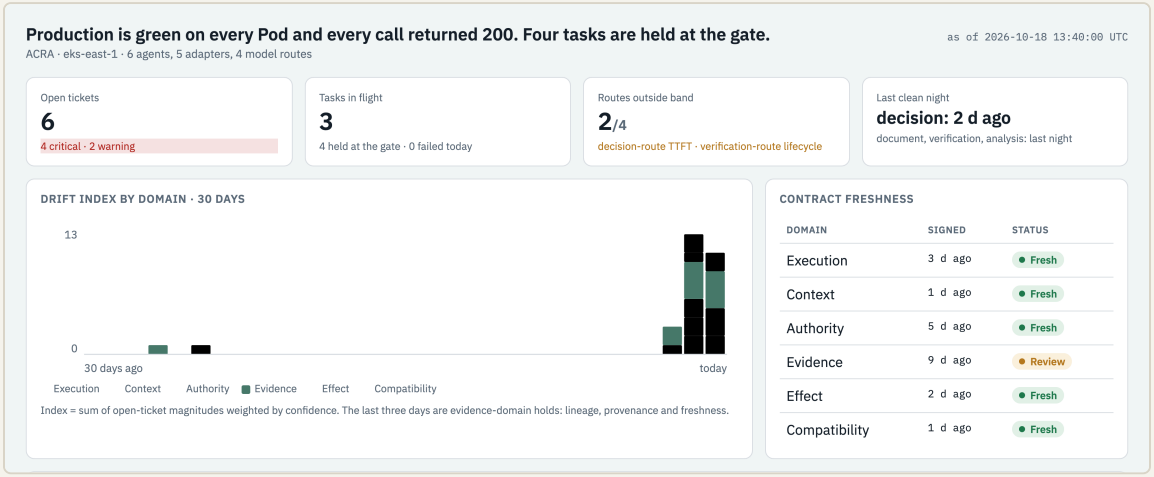
Approve
human signs the bytes

→

Apply
only what was signed

PATTERN 4 · AUTOMATED EVIDENCE

Evidence an auditor can query, not a spreadsheet.



Telemetry → graph → dashboard

no hand-built evidence

One graph for operators and auditors

same store, same answer

Writes nothing

except an approver's signature

PATTERN 4 · THE WEEKLY REPORT

What the auditor reads every week.



Evidence

Every hop for one task, kernel events inline, chain verified. The export is signed; nothing is assembled by hand.

as of 2026-10-18 13:40:00 UTC

trace t-3102 · chain intact · Signed export

task_id	4478
caller	decision-agent/dec-1f0
bundle	sha256:9f1c…

11:40:02	dispatch	task key minted · exp +40 m · budget 120k / 6 calls · fence 9
11:40:03	kong	POST decision-route · proof ok · reserved 14.2k · model pinned v1
11:40:08	model	200 · 13.9k settled · stop=end_turn
11:40:09	tool	lookup-tools.lookup-record · manifest 1.8.3 ok
11:41:50	kernel	connect() 10.4.2.17:8443 (lookup-tools) · allowed
11:51:40	agent	compaction event · context 118k → 61k
11:51:44	kong	POST decision-route · ruleset injected (e8d3…)
11:51:59	gate	claim c-9f2 roots=1 required=2 · hold
11:52:03	ticket	ACRA-3102 opened · independence 1 of 2 · radius 1 task held

Kernel rows come from the eBPF probe; they are what prove a worker did, or did not, reach a host directly.

Every control has evidence

Every row: task_id + epoch

Chain intact

168 / 168 verifications

Adverse effects 0 / 212

escalations 4 / 212

ONE DRIFT, END TO END · RECURSIVE EVIDENCE COLLAPSE

An agent's own summary came back as "independent" proof.



citations	root sources	required	→ HOLD
3	1	2	the drift no single-layer tool can see

THE GATE'S RULES

Seven invariants on every consequential write.

1

Derived is not independent

count roots, not citations

2

No status by summary

only a receipt moves a status

3

An epoch is a fence, not a proof

timeout = UNKNOWN, not NO

4

Authority is read against now

facts change → approval dies

5

The gate checks; the model proposes

loss class sets re-verification

6

Only approved tuples run

change one part → new bundle

7

Unknown stays unknown

a missing record never promotes

1 + 2 catch combinative drift.

WHAT DRIFT IS MEASURED AGAINST

A baseline is a signed snapshot of “what we agreed.”



Baselines

as of 2026-10-18 13:40:00 UTC

Signed snapshots of what we agreed, per layer. They change only by pull request with two approvers; the control plane signs the merge.

Cluster — c-0147 · signed 2026-10-15 · m.chen, r.okafor

RuntimeClass on agent Pods	baseline gvisor	live gvisor	match
Approved bundle hashes	baseline 4	live 4	match
Mutating webhooks	baseline 2	live 2	match
Egress policy present (agents-*)	baseline yes	live yes	match
Pod Identity associations	baseline 6	live 6	match

Kong — k-0092 · signed 2026-10-17 · r.okafor, s.iyer

decision-route pin	baseline dated v1	live dated v1	match
decision-route token budget	baseline 120000	live 120000	match
Admission semaphore (decision)	baseline 24	live 12	drift
JWKS rediscovery	baseline 30 s	live 30 s	match
Allowed callers	baseline 3 SAs	live 3 SAs	match

Endpoints — e-0061 · signed 2026-10-17 · s.iyer, m.chen

Allowed regions (US geographic)	baseline 3	live 3	match
verification-route lifecycle	baseline Active	live Legacy	drift
Quota headroom floor	baseline 30 %	live 29 %	drift
Canary suite band	baseline 0.0–0.1	live 0.04	match

Agents — a-0210 · signed 2026-10-09 · m.chen, p.sato

decision rule set hash	baseline e8d3…	live e8d3…	match
decision calls band	baseline 3 ± 1	live 7 (task 4471)	drift
lookup-tools manifest	baseline 1.8.3	live 1.8.3	match
Key re-issue rate	baseline ≤ 0.5 %	live 1.9 %	drift

Evaluation — v-0033 · signed 2026-10-17 · p.sato, s.iyer

Golden set hash	baseline 3d4c…	live 3d4c…	match
Judge pin	baseline Foundry gpt-… NoAutoUpgrade	live same	match
decision band	baseline 92 ± 2	live 89.3 last night	drift

Evidence — x-0018 · signed 2026-10-06 · r.okafor, p.sato

Chain head	baseline 9b7e…	live 9b7e…	match
Object Lock mode	baseline compliance	live compliance	match
Spans rejected (24 h)	baseline 0	live 0	match

One per layer

cluster · gateway · model · agent · logic · evidence

Drift = live ≠ agreed

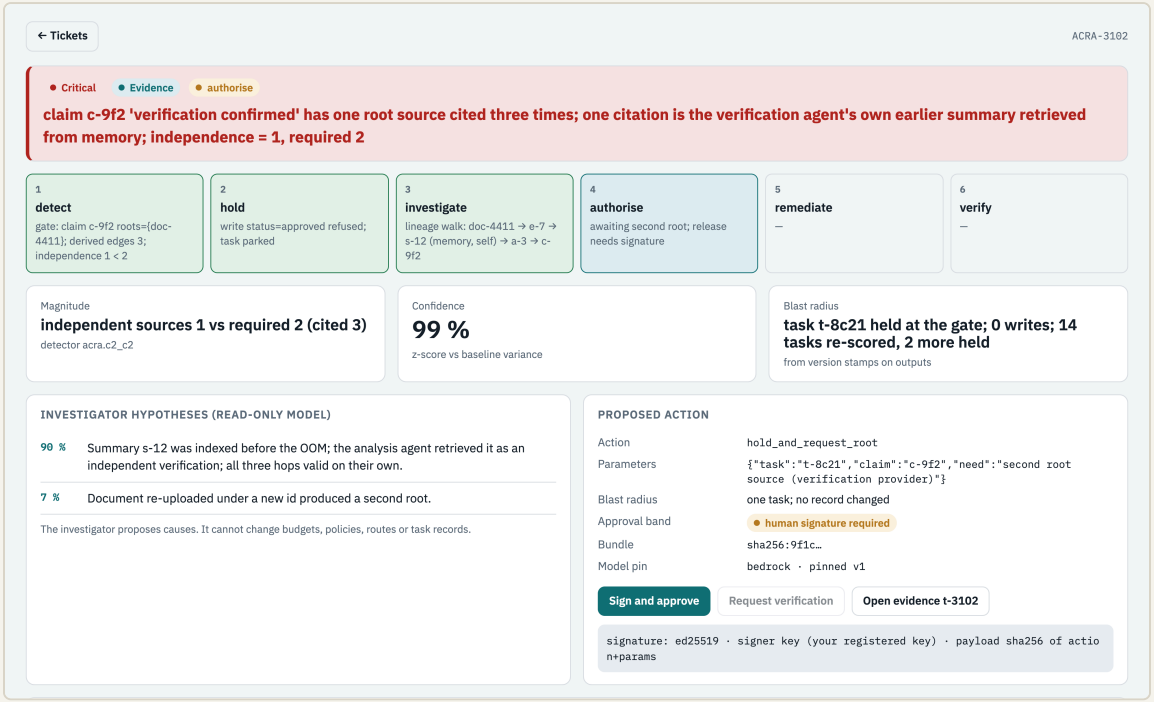
not “different from yesterday”

Changed only by a 2-approver PR

control plane signs the merge

DEMO

Break it on purpose.



1

Inject one composition

2

Single-layer detectors stay quiet

200s · green Pods · valid keys

3

ACRA holds

roots 1 < required 2

HOW THE CONTROL PLANE DECIDES

Three numbers on every ticket.

Δ

Magnitude

how far from baseline

`roots 1 vs required 2`

%

Confidence

from the baseline's own variance

`±2 pts normal → page at 85`

$

Blast radius

tasks, records, users, since when

`every output stamped with the drifted version`

Rules compute them. The LLM reads them — it never changes a number.

THE WEEK IN ONE TABLE · SAMPLE ENVIRONMENT

12 compositions. Every component green. None reached a record.

6 verified closed **2 verifying** **4 held for a human**

ACRA-3101 · held **phantom continuation**	ACRA-3102 · held · today's story **recursive evidence collapse**	ACRA-3103 · closed **causal identity fracture**	ACRA-3104 · closed **policy-model-tool skew**
ACRA-3105 · verifying **observability causal mirage**	ACRA-3106 · closed **autonomous recovery resonance**	ACRA-3107 · closed **multi-cloud evidence translation**	ACRA-3108 · held **cross-provider context rebuild**
ACRA-3109 · closed **recommendation promoted to status**	ACRA-3110 · verifying **dual-path double counting**	ACRA-3111 · closed **stale analytics vs live state**	ACRA-3112 · held **multi-cloud partial execution**

caught by: 6 counting · 4 refusing absence as proof · 2 binding state to authority

THANK YOU

A green component is not a correct system.

Build the layer that reads the composition.

01 · Kyverno gate 02 · OTel proof 03 · CRD controller 04 · Auditor evidence → embodied in ACRA

https://github.com/deepnavakoti/acra · Saideep Navakoti · Nov 9, 2026

BACKUP · DRIFT TAXONOMY

72 drift checks. 12 domains. We built the 12 that carry the story.

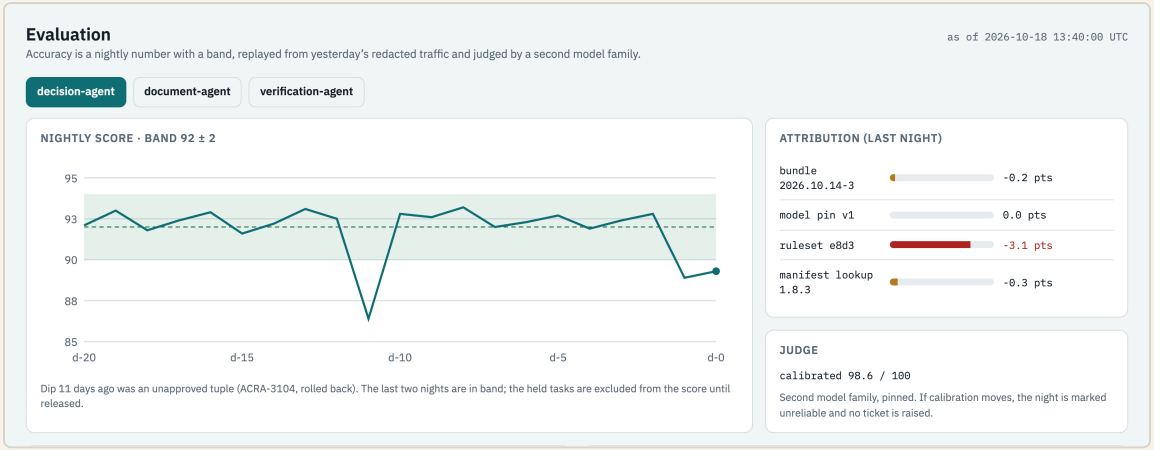
■ touched by today's 12 detectors (8) ☐ plug-ins on the same gate (4)

01 · 6 checks Cognition / goals	02 · 6 checks Prompts / context	03 · 6 checks RAG / memory	04 · 6 checks Caching / cost
05 · 6 checks Tools (MCP)	06 · 6 checks A2A delegation	07 · 6 checks Gateway / DNS / TLS	08 · 6 checks K8s scheduling
09 · 6 checks GPU / inference	10 · 6 checks Identity (SPIFFE)	11 · 6 checks Business integrity	12 · 6 checks Orchestration / recovery

Combinative drift lives in the *compositions across* these domains. The taxonomy is in the repo.

BACKUP · EVALUATION AS INFRASTRUCTURE

The evaluator never stops running.



Nightly replay
real gateway, side effects stubbed

Hourly canaries

A sampled judge
on the other provider