KubeCon CloudNativeCon North America 2026

#KubeCon #CloudNativeCon

Progressive Delivery for Trust

How agents earn the right to merge

Darian DeFalco · Senior Staff SRE · Transfix



Instruments let scientists observe what they couldn't otherwise.

A miscalibrated one still produces results, so calibrating it and noticing drift is vital.

REVIEW BOTTLENECK

Review is the bottleneck



1,642 agent-written PRs in 90 days

Faster to make than to apply.

Trust attaches to the work, not the worker.

bound → supervise → correct the recipe

grant

observe

THE LENS · DECISION BY DECISION

Where does each decision sit?

ONE TASK, FIXING A VULNERABLE DEPENDENCY · THE DECISIONS INSIDE IT

RULES DECIDE → JUDGMENT DECIDES

a fixed rule	a rule with judgment inside it	judgment checked by rules	open judgment
which recipe runs whether it may merge	how to make the change	what to exercise in the test	none

Separate question: how reliably can we tell it worked?

Another: how costly would a mistake be?

PLUGIN VERSIONING · LIVE AUTO-MERGE

Plugin versioning: the decision is a rule

YAML

```yaml
on:
push:
branches: [main]
paths: ['plugins/tf/**']

# decision
if version already moved: skip
else: bump patch,
open a PR
```

Simple decision – no need for inference

Hundreds of runs. Nearly every failure came while it was being built.

PLUGIN VERSIONING · CLAUDE-CONFIG #1173

An actual automatic merge

- 15:24:08 — App opens the version-bump PR
- 15:24:11 — App arms native auto-merge
- 15:26:21 — pico approves (the one required review)
- 15:27:11 — GitHub merges

Before: 215 direct pushes to main. Now every bump is a PR with a record.

3 min

open to merged

the promotion: one reviewed line

plugins/tf/.claude-plugin/plugin.json

@transfixio/core-platform

@transfix-pr-reviewer[bot]

> A good decision does not carry its own permission to act.

pico is our self-hosted Qwen 27B reviewer.

README MAINTENANCE · HIGH INFERENCE · MINIMAL RISK

README: merges on its own, audited daily

01

Evals

the generator, before any PR

02

Real proposals

owner review while it earned it

03

Correction

skill corrected; scope narrowed

A daily audit reads every bot merge. A kill switch pauses the lane; a wrong edit is a revert.

diff size | README files only | verifier re-run at the merge commit | merged by the bot

VULNERABILITY REMEDIATION · SUPERVISED

One line in the diff, under every page

```
Gemfile.lock

- crass (1.0.7)
+ crass (1.0.8)
```

every request the app serves

Rails · the gems it loads

crass 1.0.8 · the one line, under all of it

- ✓ checks green on the merged commit
- ✓ no human commits
- ✓ no revert, reopen or follow-up
- ✓ observed after merge

agent-written, human-approved, held up.

SUPERVISED WORK · ONE TICKET, TWO PULL REQUESTS

An error reveals a condition the recipe missed

First PR

pyproject.toml +2 · poetry.lock

pins a transitive dependency directly

branch ...-fresh · no labels

closed

routing: inferred → scripted

+ 12 tests

Same ticket, next PR

poetry.lock only

follows the recipe

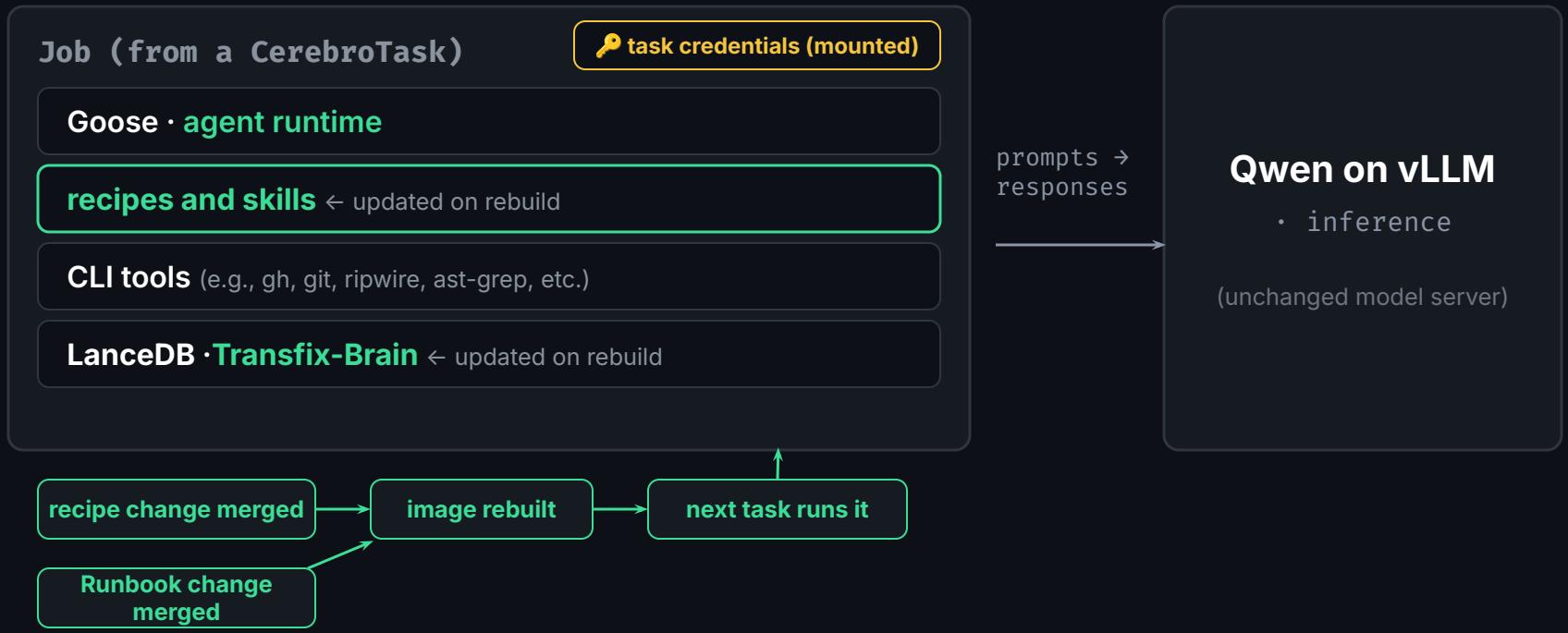
branch ...-fix-cve-...

approved by a person · merged

We fixed the condition, not the PR.

THE TASK · ONE KUBERNETES JOB

Where the learning is kept



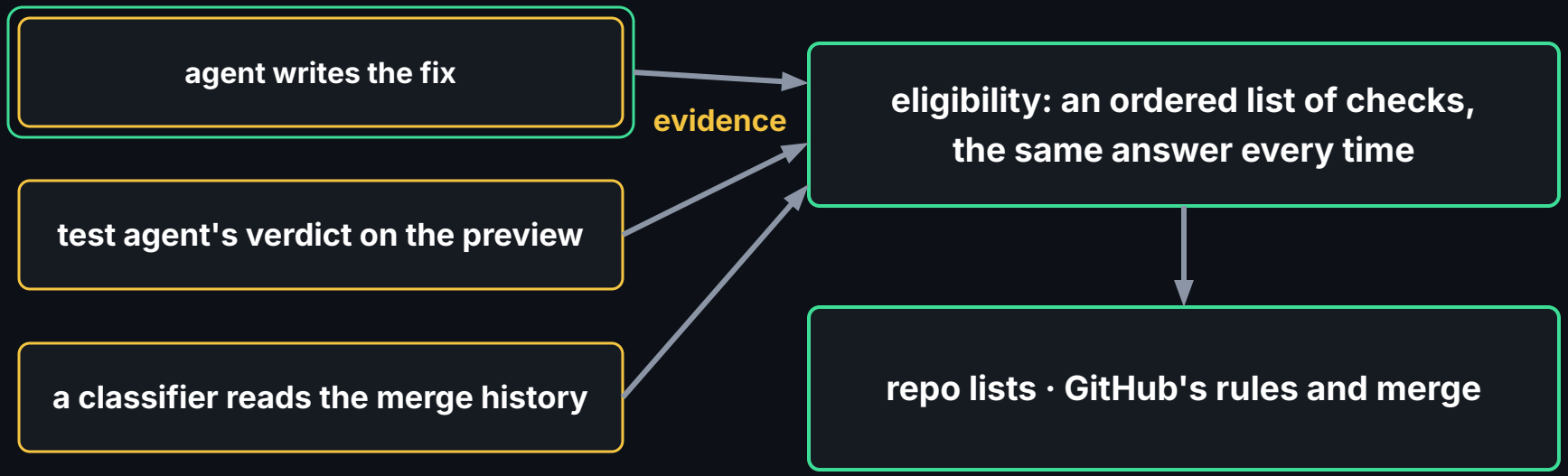
The model didn't change. The procedure around it did.

THE SPECTRUM OF DETERMINISM · IN THE PIPELINE

Who infers, and who decides

a model infers · fixed code decides

inside scripted rules: routing, lockfile checks



Models produce evidence. Code decides authority.

A rule can be wrong too.

THE MODEL · PROGRESSIVE DELIVERY FOR TRUST

Progressive delivery, authority as the payload

Partial confidence. Bound exposure · watch · keep a way to respond.



WHERE IT BREAKS

A rollback can't undo a merge.

CANARY · VULNERABILITY REMEDIATION

Eligible. Would approve. Didn't.

```
repo: "fms_mobile"
pr: 2218
head_sha: "44d7ea75 … "
decision: eligible
would_submit_bot_approval: true
mode: shadow
action_result: not-attempted
```

shadow mode

auto-merge flag off

repo not on the auto-merge list

One day in shadow: 18 rows like this. 0 merges.

What happened next: a person merged it · only the agent's commits · not reverted

What counts as a track record

the judgment · written to the ticket

transfix#24906 · net-imap · ruby

```
repo: transfix pr: 24906 package: net-imap eco: ruby
merged_at: 2026-05-06T21:30:45Z
head_sha_at_merge: 459052a8 ...
review_proof: { approver: <human>,
commit_id: 459052a8 ... } # same head
ci_proof: { observed_checks: 66, all SUCCESS,
head: 459052a8 ... }
outcome_proof: { outcome: clean-merge,
material_failure: false,
observed_until: 2026-05-07T21:30:45Z } # one day
```

A merged remediation counts if:

- ✔ checks green on the merged head
- ✔ no one else committed to it
- ✔ no revert or follow-up in the window after
- ✔ 30 like it, same ecosystem, pooled across our repos

Approval told us who clicked. The clean window tells us whether it held.

WHEN HISTORY ISN'T ENOUGH

History only covers what has already happened.

A fresh test observes this commit.

history for every PR · plus a passing test of this exact commit when the change is broad

The pace of delegation is set by how reliably outcomes can be evaluated.

A deploy per proposal

transfix-pr-staging-applicationsets.yaml (trimmed)

```
generators:
  - pullRequest:
      github:
        owner: transfixio
        repo: transfix
        labels: [staging-beast-mode]          # the switch
      requeueAfterSeconds: 60
  template:
    spec:
      source:
        kustomize:
          namespace: transfix-{{ .branch_slug }}   # one per PR
          images:
            - transfix_svc_stage:{{ .head_sha | substr 0 12 }}
      syncPolicy:
        automated: { prune: true, selfHeal: true }
        syncOptions: [CreateNamespace=true]
```

ships inside every preview

```
kind: RoleBinding
roleRef:
  kind: ClusterRole
  name: cerebro-pr-namespace-exec
subjects:
- kind: ServiceAccount
  name: cerebro
  namespace: cerebro
```

pods: get, list · pods/exec: create. This namespace only, for as long as it exists.

this exact commit

label off → it's gone

No provisioning code. No cleanup code. Reconciliation.

What the test actually does

crass 1.0.7 → 1.0.8 · one line in Gemfile.lock

- `script` Plan: under the HTML sanitizer, no direct callers → test the whole app
- `script` Preview healthy · kubectl exec: crass 1.0.8 loaded
- `agent` Logs in, reads, writes, posts its evidence

Not checked, and said so: the CSS bug itself

5 PRs at once

a namespace each
verdicts in 2–22 min

Rules build the plan and decide. The agent's report is evidence.

SUCCESS CONDITION · WHY THIS GRANT

What made the grant reasonable

- Supervised PRs, reviewed and merged by people
- Errors investigated; recipes and skills revised
- 30 comparable clean merges, 10 in this repo
- A passing test of this commit when the change is broad
- One narrow class, granted per repo and ecosystem

→ **approve and arm auto-merge**

BEFORE

a person reviews every PR

→

AFTER, FOR THIS CLASS

people review the recipes, the policy, the exceptions and the grant

PROMOTION · ARGO #4544

Permission lives in reviewed config

vuln-evaluate-cronjob.yaml

```
schedule: "0 * * * *"
env:
  VULN_APPROVAL_MIN_COMPARABLES: "30"
-   VULN_APPROVE_SHADOW: "true"
+   VULN_APPROVE_SHADOW: "false"
-   VULN_AUTOMERGE_ENABLED: "false"
+   VULN_AUTOMERGE_ENABLED: "true"
  VULN_AUTOMERGE_REPOS:
"transfix-brain,claude-config,core-platform,tms,transfix,broker-platform-svc"
```

now one grant per repo and ecosystem, e.g. rogers:npm

The code shipped first. Configuration held it.

Same arming function as the plugin lane.

Same capability. Different evidence.

Granting authority was a PR to this file, reconciled by Argo CD. Revoking it is the same PR, reversed.

PROMOTION · WHO DOES WHAT

Four roles, four responsibilities

Proposes

`Cerebro · Job`

can't approve its own PR

→

Tests

`QA agent · Job`

exec in one namespace, while it exists

→

Approves and arms

`evaluator · CronJob`

writes to Jira first, or doesn't act

→

Merges

`GitHub`

enforces the rulesets

Kubernetes RBAC: what each can touch in the cluster

Repository rules: what can happen in the repo

PROMOTION · ONE PR, ONE DAY (UTC)

One PR, start to finish

06:00	Cerebro commits the fix: one line in Gemfile.lock, crass 1.0.7 → 1.0.8 `a1e11008`
06:01	labelled runtime-broad: history alone isn't enough
06:19	preview up, QA agent dispatched → 06:20 pass → 06:27 evidence posted, preview torn down
06:29	pico approves
08:48	decision written to Jira → approved → auto-merge armed
11:52	test verdict pulled → disarmed · 12:15 retest passes → 12:55 armed again
16:04	the branch is brought up to date with main: a new commit `3ed9e55f`
16:10	GitHub merges that commit. No agent tested it.
17:32	fix live: a new commit disarms (bps#5243, core-platform #1792) disarms on the next push · not atomic

The verdict was bound to the commit. The permission wasn't.

Every step left a record. That's how we found it the same afternoon.

ABORT · VULNERABILITY REMEDIATION · REVOCATION

Withdrawal has four parts

01 — exists

Stop new actions

(a flag; a hold label for the Friday sweep)

02 — exists

Disarm pending auto-merge requests

(only ours; a human's is left alone)

03 — exists

Dismiss approvals already submitted

(only ours)

04 — always human

Completed merges

(an ordinary revert)

A merge is a merge.

automatic: a reverted merge pauses that grant for 30 days

TAKE-HOME

Five questions for one workflow

Plugin · arms its own PR

Vulnerability · merges one narrow class

README · merges, audited daily

1. What useful work, and which of its decisions could be rules?
2. What makes a result good enough, and who can tell?
3. What has supervision taught you, and where is that written down?
4. What evidence would justify the next specific permission?
5. What would make you narrow or withdraw it?

Thank you!

Reliance doesn't require the end of inquiry.

> It needs grounds to proceed, and a willingness to revise them.

Darian DeFalco · Transfix



Scan to submit feedback