CASE STUDY · AT&T

How AT&T Turns Signed Evidence into Deployment Decisions with Kyverno

AT&T uses a Kyverno ImageValidatingPolicy, written in CEL, to decide at admission whether an image may run in a given environment. It verifies the image's signed evidence, such as SLSA provenance and SBOM attestations from GitHub Actions, then checks those claims against the destination's rules.

990 → 96 ms	14 → 3	1.0 → 10.4	607 → 65 MiB
P50 ADMISSION LATENCY (KYVERNO BENCHMARK, ONE LAPTOP)	REGISTRY CALLS PER ADMISSION (KYVERNO BENCHMARK, ONE LAPTOP)	ADMISSIONS PER SECOND (KYVERNO BENCHMARK, ONE LAPTOP)	MEMORY PER REPLICA (KYVERNO BENCHMARK, ONE LAPTOP)

CUSTOMER	TEAM	EVIDENCE	POLICY	PRESENTED
AT&T	Kubernetes platform engineering	Sigstore signatures, GitHub Artifact Attestations (SLSA provenance, SBOM)	Kyverno ImageValidatingPolicy in CEL	KubeCon + CloudNativeCon North America 2026

§01 CHALLENGE

Why isn't a signature enough?

AT&T had added signing and attestations to its standardized, reusable build workflows. That raised the next question: now that we have cryptographically verifiable evidence, what do we do with it? The answer: **a valid signature is not deployment approval.** A signature says who built an image and that it hasn't changed. It doesn't say whether the image meets the rules of the place it is going.

ENVIRONMENT	REQUIRES	RESULT
Development	Verified provenance	Admit
Production	Provenance and a signed SBOM	Deny: SBOM missing

Checking at promotion isn't enough on its own. The destination may need extra evidence, requirements change, some attestations are time sensitive, and requests can come from GitOps, a pipeline or kubectl. Promotion approves an artifact; admission evaluates this request against the policy that applies now.

The platform team wanted shared controls that app teams don't each rebuild, support for more than one trust root, and enough retained evidence to explain any decision later, with measured latency, high availability and deliberate failure behavior.

§02 SOLUTION

One policy engine for evidence and destination rules

AT&T had used Kyverno for years and didn't want a second policy engine for one problem. The design has three building blocks:

BUILDING BLOCK	ROLE
Sigstore	Signing identity and trust material, public or GitHub's private instance
GitHub Artifact Attestations	SLSA build provenance, SBOM attestations and custom test and quality attestations
Kyverno and CEL	An ImageValidatingPolicy verifies the evidence and evaluates the claims against the destination's rules

The policy decides in three steps:

01. **Verify the evidence before reading it.** The payload is only read after the attestation's signature is verified against an approved attestor.
02. **Evaluate the verified claims.** The image must be built by AT&T's own GitHub organization, matched on its immutable numeric owner ID rather than its name.
03. **Apply the destination's rules.** Development requires provenance, an SBOM and a vulnerability scan. Production also requires at least two approving code reviews.

Internal builds and approved upstream images are handled by the same engine, each with an explicit image scope and approved attestors. Verifying images signed with GitHub's private Sigstore instance needed a Kyverno change: Chad raised it with Jim Bugwadia, who wrote the fix himself (kyverno/kyverno#16666). It shipped in Kyverno 1.19.

§03 RESULTS

Faster admission with each Kyverno release

These figures are a Kyverno benchmark, not AT&T production numbers. Sagar measured image verification on one laptop, from Kyverno 1.18.2 to the 1.20 development branch, changing only the Kyverno version: a kind cluster, one ImageValidatingPolicy, one signed image, a warm cache and a registry with a fixed 25 ms delay.

METRIC	KYVERNO 1.18.2	KYVERNO 1.19.1	MAIN (1.20-DEV)
Admission latency p50 / p99	990 / 1170 ms	312 / 333 ms	96 / 107 ms
Admissions per second	1.0	3.2	10.4
Registry calls per admission	14	10	3
CPU / memory per replica	44 mC / 607 MiB	67 mC / 463 MiB	34 mC / 65 MiB

The milliseconds depend on the machine, but the ratios travel. Fewer registry round trips help everyone. Measure with your own policies and your own registry.

§04 LESSONS

What AT&T would do again

01. **Make the right way easy.** Build evidence into reusable workflows, so no team has to become a signing expert before it can ship.
02. **Use the engine you know.** Make controls a shared platform capability, not a new compliance project for every application.
03. **Bring maintainers a real problem.** A concrete use case and a hands-on maintainer turned into upstream code.

Roll out in phases: audit, remediate shared workflows, enforce by tier, then expand coverage. Keep exceptions narrow, owned and time limited, decide up front how verification failures are handled, and retain every decision so “why was this image denied last Tuesday?” is a query, not a project.

nirmata

Nirmata created Kyverno, the CNCF graduated policy engine for Kubernetes. We help platform and security teams govern clusters, workloads, and AI agents with policy as code.

READ THE FULL STORY

How AT&T Turns Signed Evidence into Deployment Decisions with Kyverno

nirmata.com/2026/11/18/how-att-turns-signed-evidence-into-deployment-decisions-with-kyverno

SOURCE

Kyverno private Sigstore fix (kyverno/kyverno#16666)

github.com/kyverno/kyverno/pull/16666

TALK TO THE KYVERNO MAINTAINERS

nirmata.com/request-a-demo

© 2026 Nirmata, Inc. Kyverno is a CNCF graduated project. Kubernetes is a registered trademark of The Linux Foundation.