Where Does Your Policy Live?

SPEAKER

DADISI SANYIKA

CEO & FOUNDER, SOL DUARA

CONFERENCE

CDCON '26

MINNEAPOLIS, MN

Point to it.

§ 01 · THE SETUP

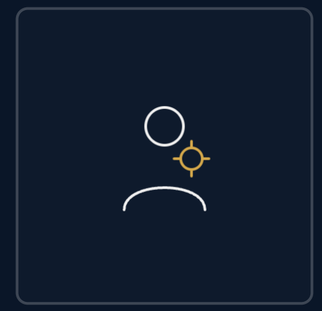
Two roles. One handoff that should be clean — but isn't.

Policy Maker

AUTHORITY

THE GAP

no connecting layer



Team

CONTROL

§ 02 · THE FRAGMENTATION

Same words. Different code.

POLICY · V1.0

"All deployments must pass security scanning."

TEAM A	Jenkins
scanner	Snyk
timeout	10 min
on-fail	FAIL

TEAM B	GitHub Actions
scanner	Trivy
timeout	5 min
on-fail	WARN

TEAM C	GitLab CI
scanner	???
timeout	none
on-fail	???

POLICY DIDN'T SAY WHICH TOOL · WHAT TIMEOUT · WHAT FAILURE MEANS

Same policy. Three implementations. Zero enforcement.

Jenkins · GitHub Actions · GitLab CI

§ 03 · DIAGNOSIS

The teams controlled the implementation, not the policy makers.

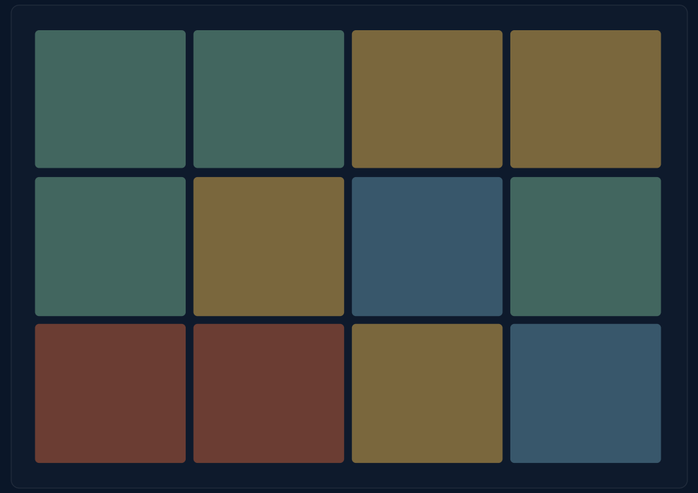
§ 04 · FIVE YEARS

Manageable becomes unauditable.

YEAR 1

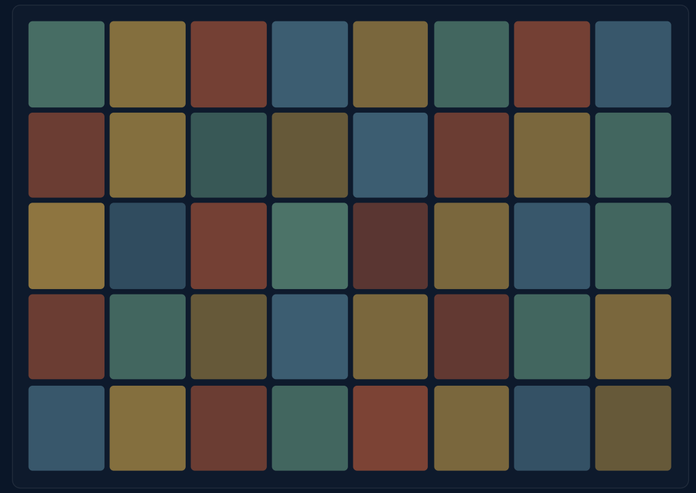
3 teams. 3 pipelines. One CI system. You could audit it by reading the YAML.

YEAR 3



12 teams. 2 acquisitions brought their own tools, their own interpretations.

YEAR 5



40 teams. 7 CI systems. Policy updated three times. Implementations updated? Unknown.

§ 05 · ARCHAEOLOGY

Reconstructing intent from artifacts that only partially survive.

2025	github-actions.yml · trivy@v0.50 · opa-gatekeeper
2023	.gitlab-ci.yml · snyk-cli@1.1200 · ~~vault-sidecar~~
2021	Jenkinsfile · ~~whitesource~~ · ~~checkmarx-old~~
2019	~~build.gradle scan-task~~ · ~~bash-scanner.sh~~
2017	~~??? · author left in 2019~~

§ 06 · THE AUDIT

Show me your policy.
Now show me the implementation.

THE WIKI

Security Scanning Policy

/policy/security-scanning · last edited Feb 2024 · 1 source of truth

All deployments must pass security scanning before production release. Scans must cover known CVEs, license compliance, and container image vulnerabilities.

✓ clean · clear · organized

THE REALITY — WHICH ONE?

Jenkinsfile · team-a

```
stage('scan') {
sh 'snyk test'
timeout(10)
}
```

.github/workflows/ci.yml · team-b

```
- uses :
aquasecurity/trivy@v0
.20
with: severity: HIGH
continue-on-error:
true
```

.gitlab-ci.yml · team-c

```
# TODO: enable
scanner
scan_job:
when: manual
allow_failure: true
```

buildkite.yml · team-d

```
# inherits from team-
a (2021)
command: ./scan.sh
# script no longer
exists
```

§ 07 · THE CHALLENGE

"Prove these are equivalent."

Jenkins	GH Actions	GitLab CI	Buildkite	CircleCI	TeamCity	Argo
.xml + console	.yml + JSON	.yml + artifacts	.yml + annotations	.yml + workflows	kotlin DSL	CRDs + events

COST: WEEKS. PER AUDIT.

§ 08 · THE HONEST TRUTH

You're not proving consistency. You're arguing for it.

And the audit is the easy case. The auditor gives you time.

An incident doesn't ask questions.

§ 09 · WHAT IF

What if policy makers controlled policy?

Not as a wish on a wiki page. As the actual code that runs in every pipeline.

§ 10 · THE CHOICE

Two architectures. One produces drift. The other doesn't.

✗ Policy Inside Pipelines

WHAT YOU HAVE

Jenkins · team-a v1.0

policy: scan(snyk, 10m, FAIL)

GH Actions · team-b v?

policy: scan(trivy, 5m, WARN)

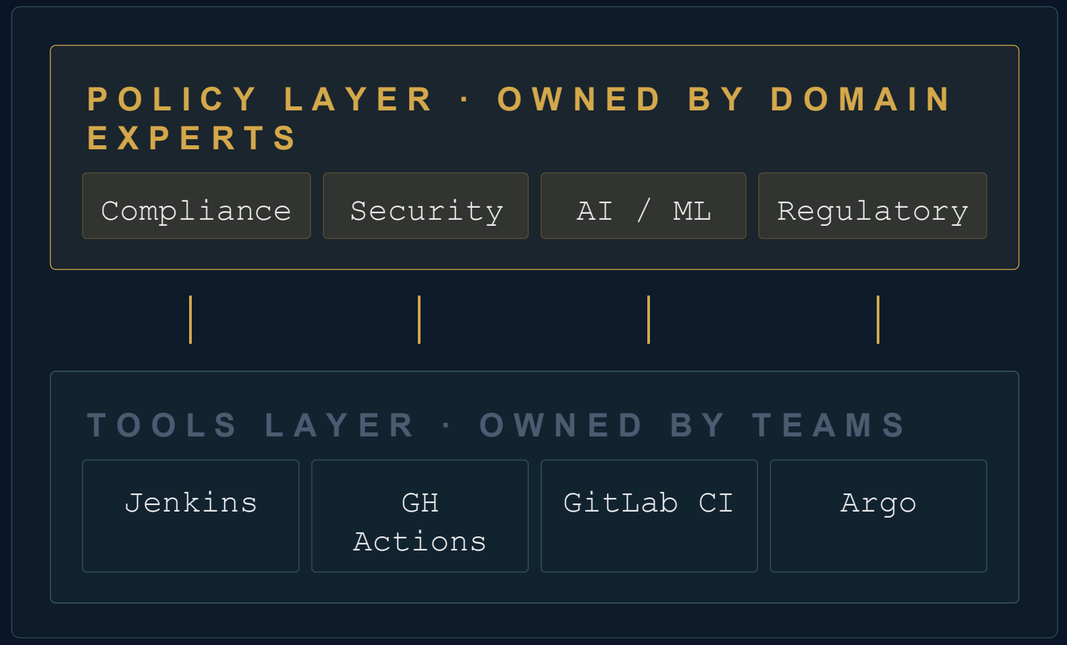
GitLab CI · team-c —

policy: ??? (manual)

Teams control implementation.

✓ Policy Separated

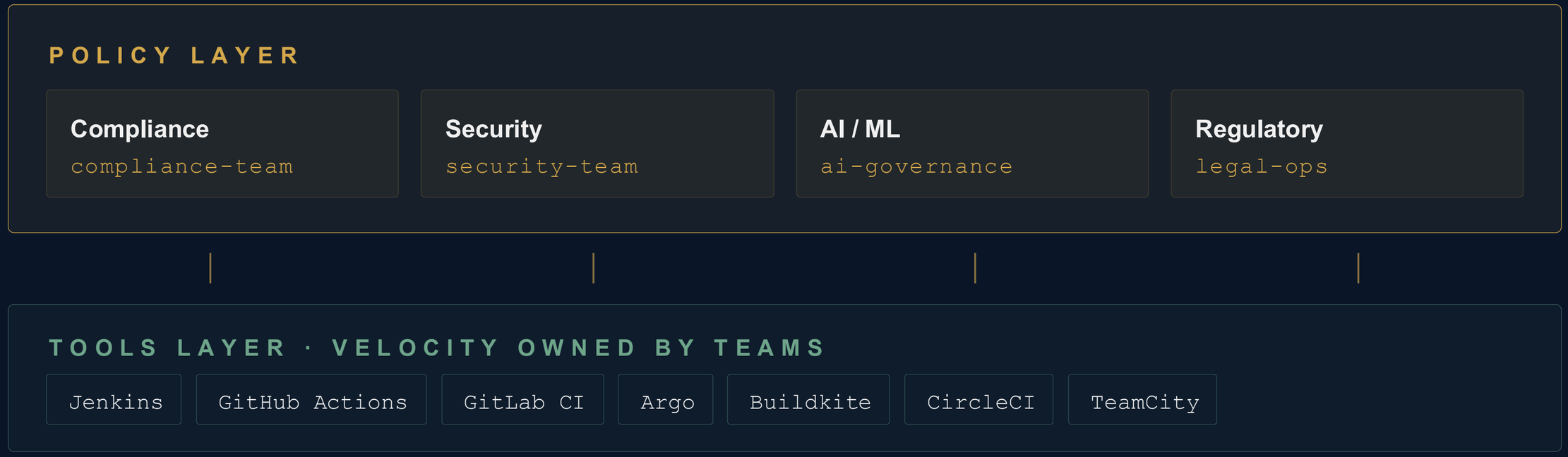
WHAT'S POSSIBLE



Pipelines source policy.

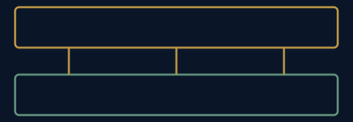
§ 11 · THE RE-ALIGNMENT

Domain experts own domain policy.



Where does your policy live?

INSIDE
PIPELINES



SEPARATED

LINKEDIN

CONTACT US

I want you to be able to *point*.