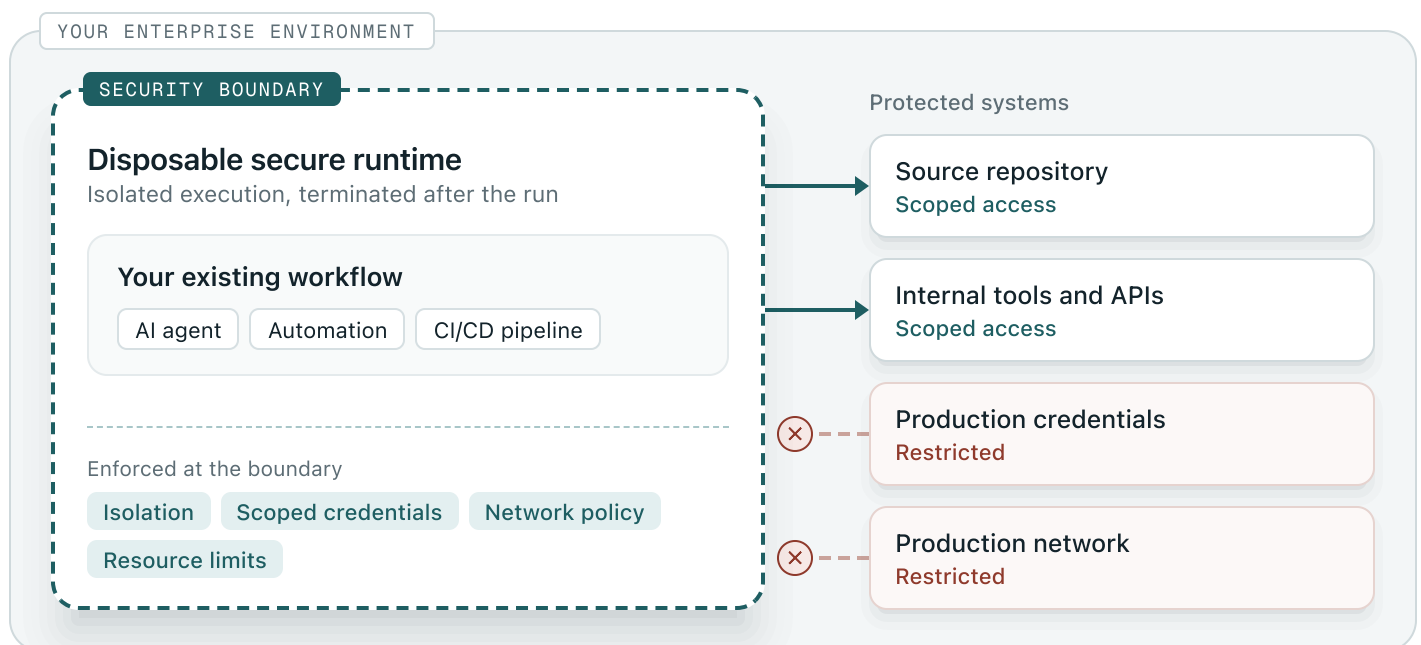


WHY SECURE EXECUTION MATTERS

Run autonomous workflows without unrestricted access.

Bring your AI agents, automation, and CI/CD pipelines into controlled execution environments. Give them the boundaries they need to operate, without unnecessary exposure to enterprise infrastructure.



Scoped access crosses the boundary. Everything else stays outside.

Your teams already have working agents and automation. Moving them into operational environments raises harder questions: what can they reach, which credentials do they hold, and who can account for what they did?

Smaller blast radius

Isolation limits what an unexpected action can affect.

Safer experiments

Test in disposable environments without maintaining persistent execution workloads.

Control over access

Credentials, networks, tools, and resources.

Visibility

Execution records and governance findings, together.

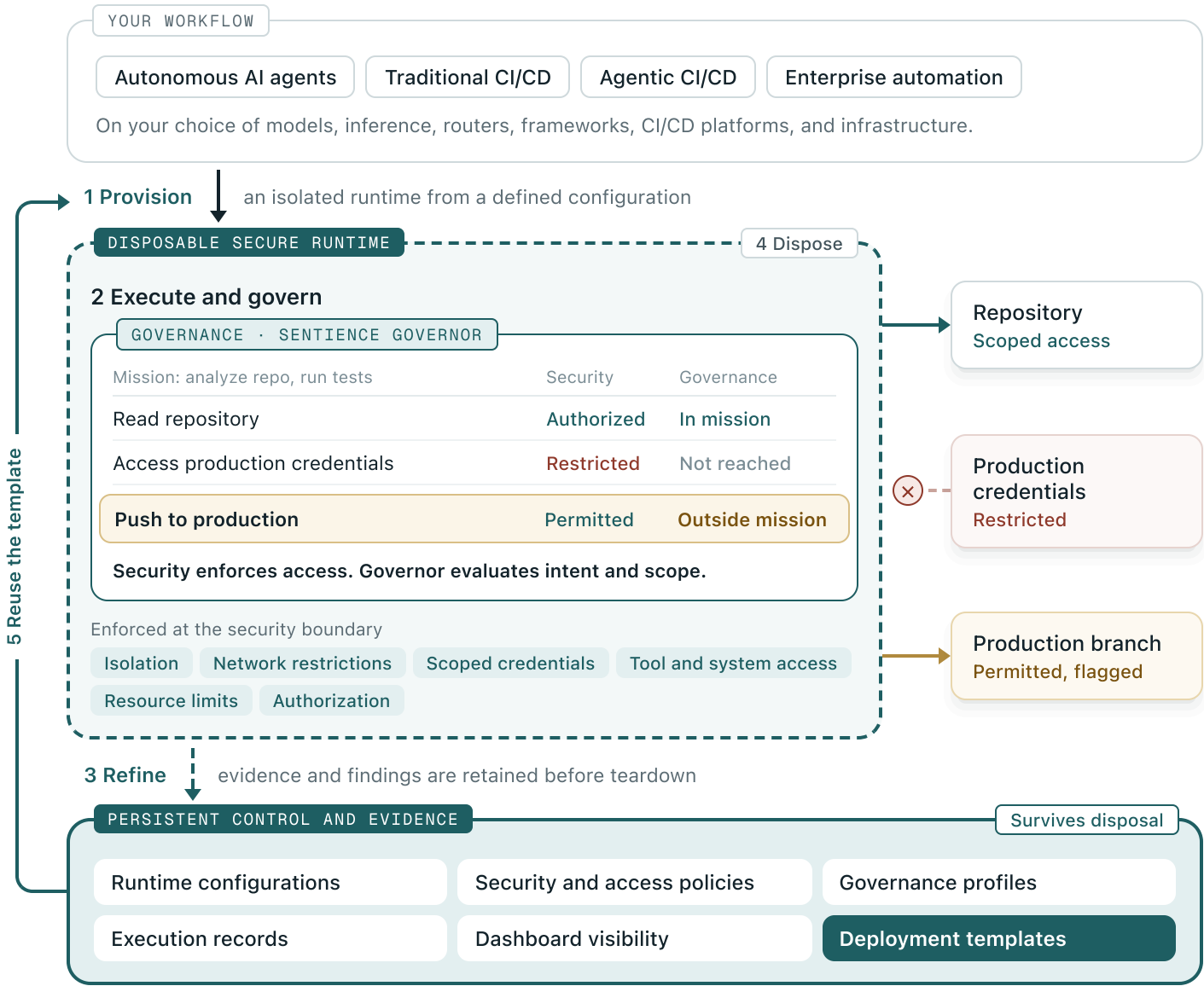
Have a workflow in mind? Talk it through with our architects.

[Talk to an architect →](#)

HOW THE ARCHITECTURE WORKS

Disposable execution. Persistent governance.

The runtime is disposable. The security knowledge and deployment pattern are reusable.



⬜ Disposable ⬜ Persistent — Execution - - Evidence — Reuse

Configurable commercial architecture. Integrations are confirmed during discovery.

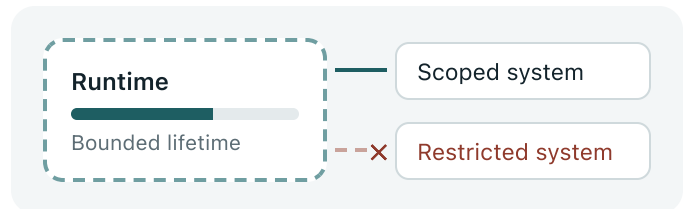
Want to map this architecture to your environment? [Talk to an architect →](#)

THE OPERATIONAL BENEFITS

Contain the risk. Learn from execution. Scale what works.

Contain

Isolated execution, restricted permissions, controlled credentials, and a bounded runtime lifetime reduce exposure of enterprise infrastructure.



Learn

Observe real behavior, identify the permissions a workflow needs, detect governance deviations, and refine policies from evidence.

Push to production

Permitted, but outside the declared mission

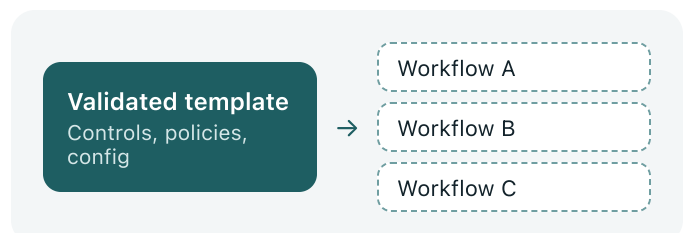
↓ finding reviewed by your team

Security policy updated

Production pushes require approval

Repeat

Keep the validated configuration, security controls, governance policies, and evidence as a reusable pattern for the next workflow.



Disposable is not reversible. Data transfers, API calls, repository changes, and infrastructure modifications can survive teardown, so access restrictions and approval controls still apply.

Want to find where your workflows need boundaries?

[Talk to an architect →](#)

FROM ONE WORKFLOW TO A REPEATABLE PATTERN

Start with one workflow. Build a pattern you can scale.

A focused proof of concept around a workflow you already run. No enterprise-wide migration.

- | | | | |
|---|--|---|--|
| 1
Bring your workflow
One agent, automation process, or CI/CD pipeline. | 2
Establish the boundaries
Access, restrictions, credentials, and governance policies. | 3
Run and validate
Execute, test the boundaries, review the evidence. | 4
Retain and repeat
Keep the validated controls and plan the next workflows. |
|---|--|---|--|

What you leave with

Not a demo. A foundation for repeatable, governed execution.

A validated workflow execution pattern

Governance policies for agentic workflows

A reusable runtime configuration or deployment template

Documented security and access boundaries

Execution and governance evidence

A defined path toward broader deployment

Exact deliverables and success criteria are agreed during discovery.

Bring your workflow. Secure its execution.

We'll help define the environment, controls, and outcomes for a focused POC.

[Build my POC](#)