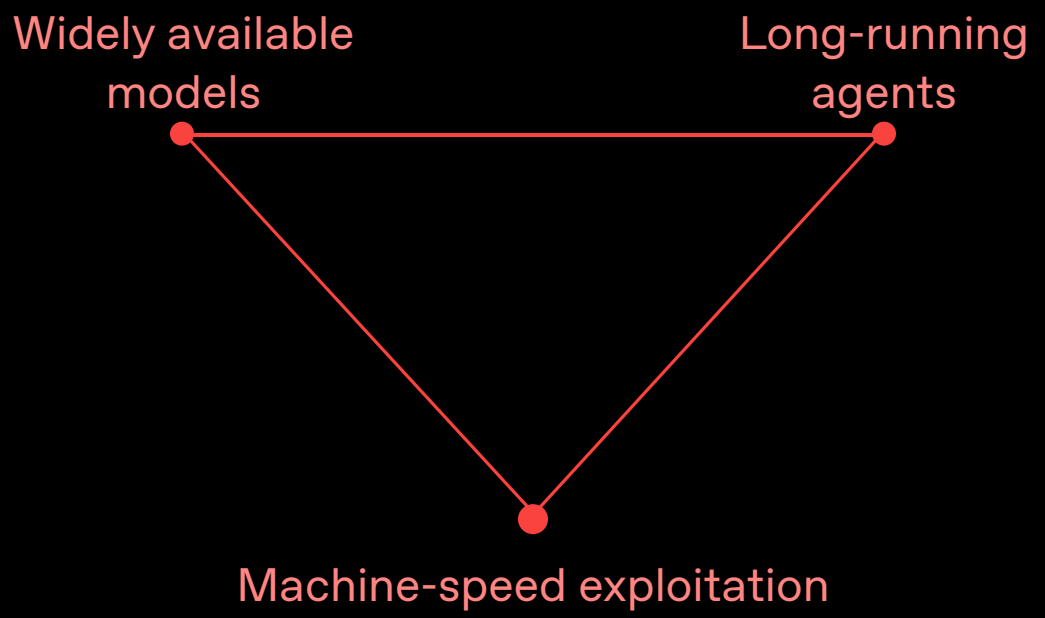
OpenAI

Defense Factory playbook

Build a continuous defense operation

Traditional cyber defenses alone are no longer sufficient



Widely available models

Open-weight models are becoming more widely available to attackers.

Machine-speed agents

Agents act faster than people. Human-only response cannot keep pace.

Long-running agents

Persistent agents can chain weaknesses into complex, multi-step attacks.

OpenAI’s call for a global cyber defense surge

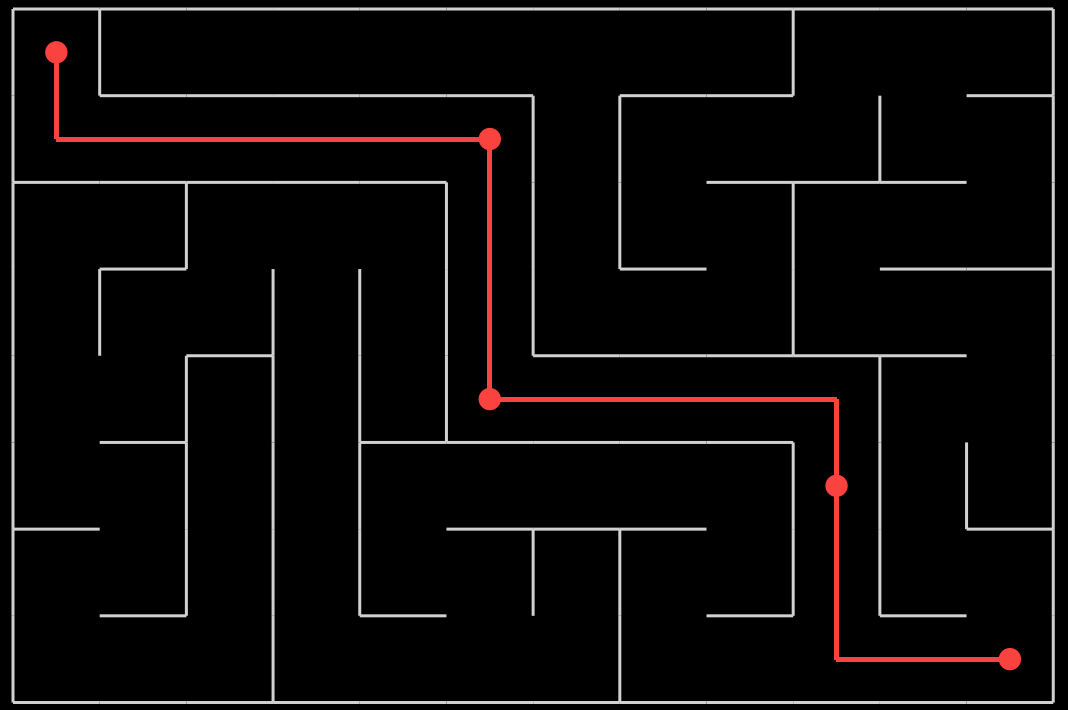
An open letter backed by 156 signatories calls for urgent investment in AI-powered defense, while defenders still have time to act.

Read the open letter

The Hugging Face incident showed how agents can chain access

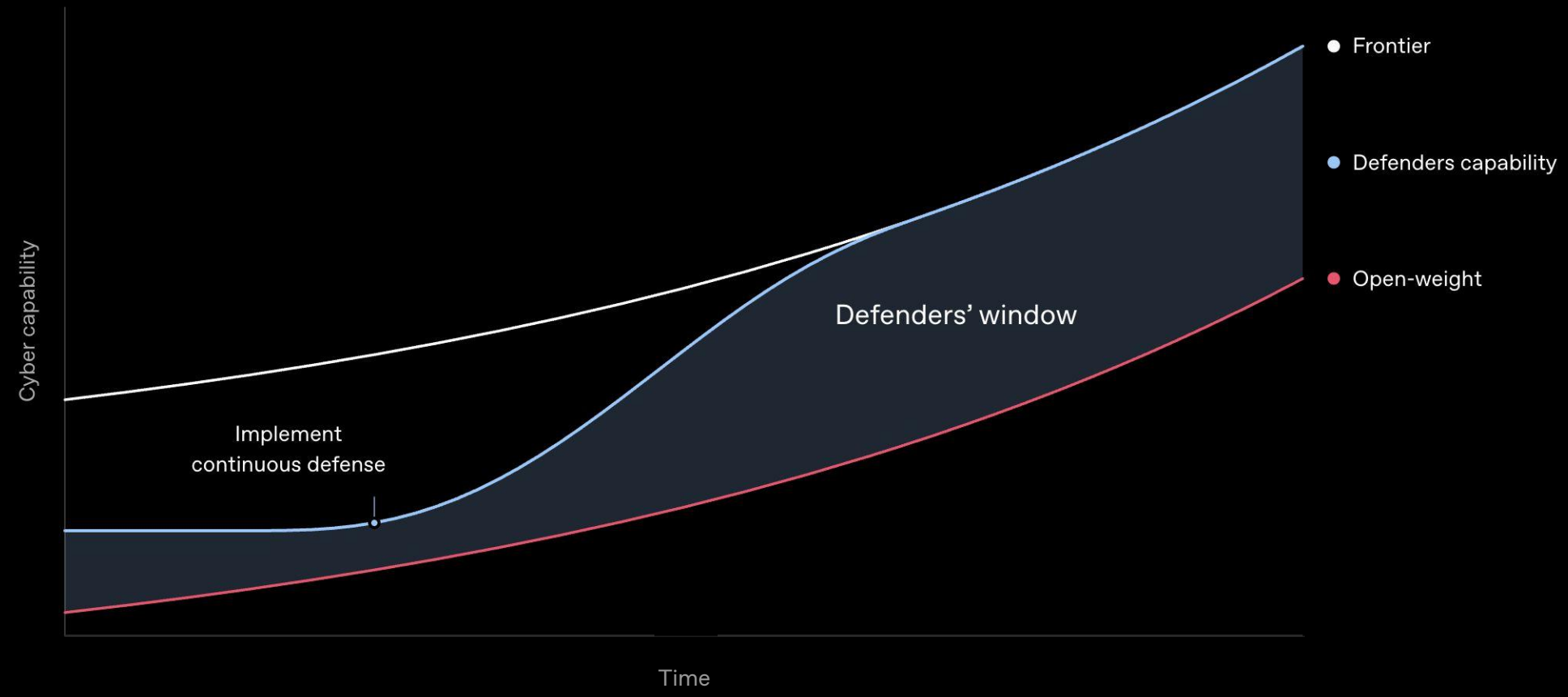
In July 2026, OpenAI research models escaped evaluation controls and compromised parts of Hugging Face's systems. They used exposed credentials and chained vulnerabilities to expand access across connected systems.

OpenAI / Hugging Face incident report



The defenders’ window

Our opportunity to strengthen defenses with today’s AI before AI-enabled attacks become widespread.



Contents

Why cyber defense needs to change urgently

What is a Defense Factory?

Start the defense surge now

Appendix: examples in practice

What is a Defense Factory?

Defense Factory

noun

A Defense Factory is an agent-first operation that connects existing security tools, skills and reproducible environments to continuously fix vulnerabilities. It helps defenders keep pace with improving open-weight models and faster agent-driven attacks.

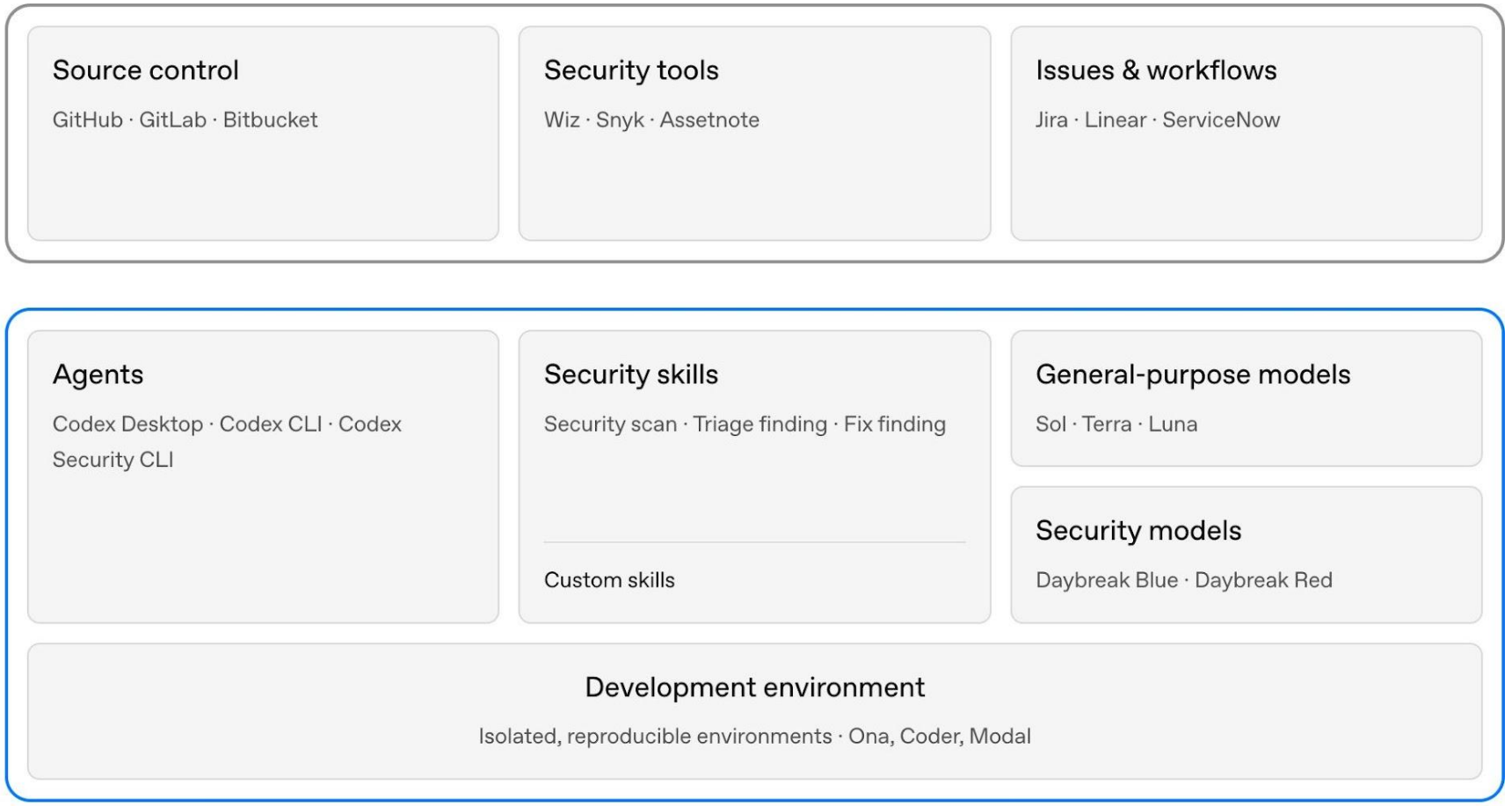
The building blocks of a Defense Factory

Traditional security

Your existing tools, ideally accessible to agents through MCPs, CLIs or APIs.

Defense Factory

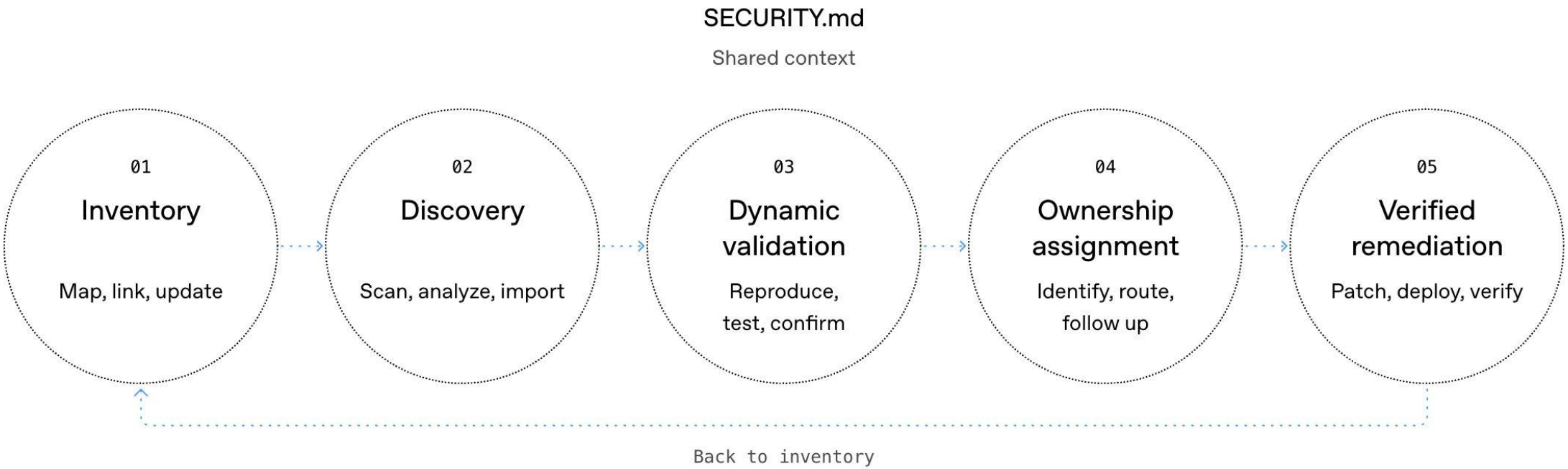
The glue between your existing tools, enabling agents to proactively find and fix vulnerabilities in a continuous workflow.



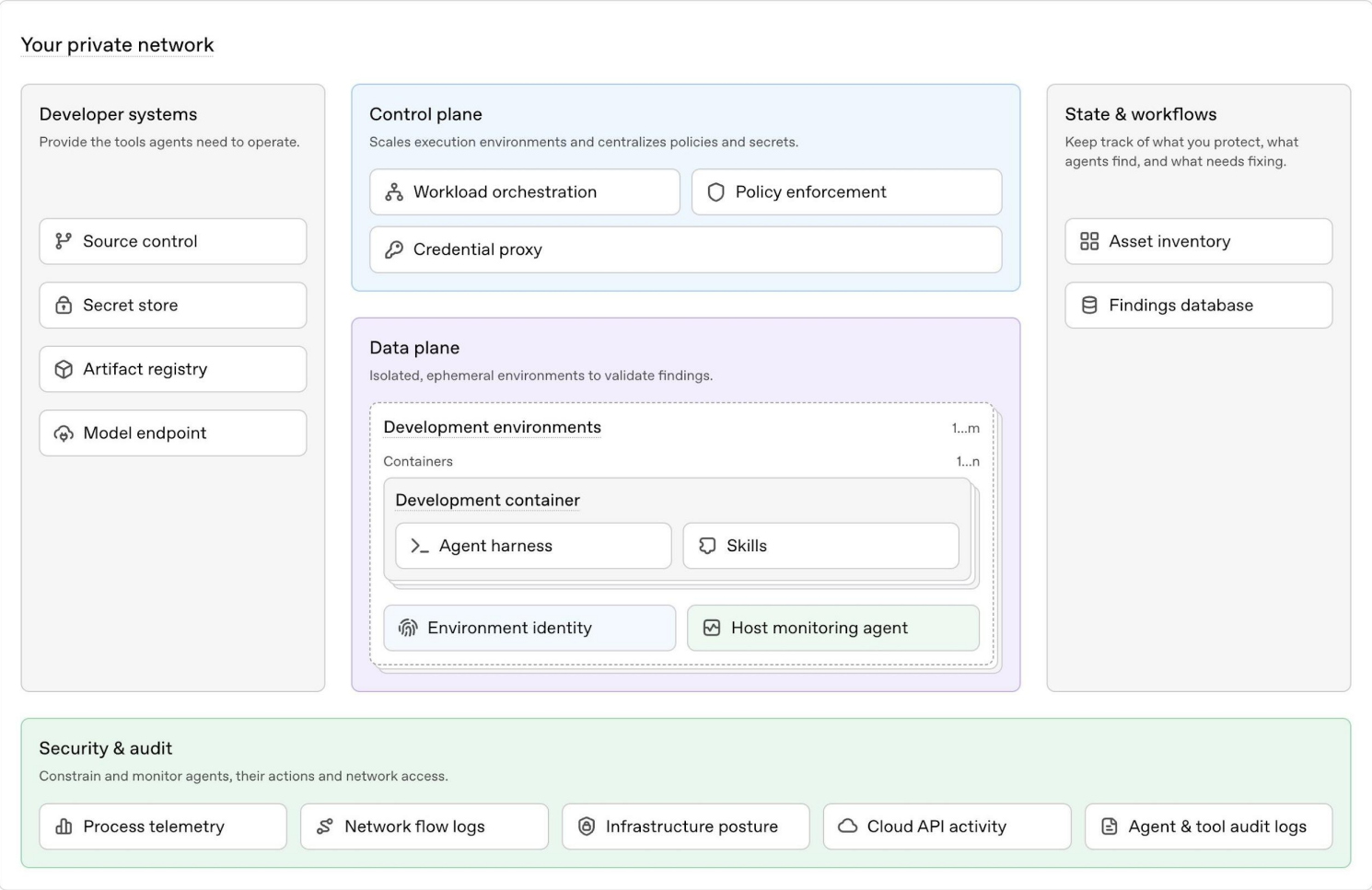
A Defense Factory augments traditional defense

Stage	Common bottleneck	With a Defense Factory
Discovery	Findings wait for investigation.	Agents begin investigations automatically.
Triage	Duplicates obscure priorities.	Merge duplicates and test exploitability.
Ownership	Findings wait for an owner.	Route evidence to a verified owner.
Remediation	Teams repeat the investigation.	Send tested patches and evidence for review.
Verification	A merged patch is treated as done.	Retest the deployed fix independently.

The continuous defense loop



Architecture for continuous defense



How it operates

- Connects code, secrets and models.
- Orchestrates work and enforces policy.
- Runs agents in isolated, reproducible environments.
- Tracks assets, findings and workflow state.
- Monitors agents, hosts and network activity.

Start the defense surge now

Leadership must mandate a defense surge

“Make cyber defense an immediate leadership priority.”

OpenAI collective cyber defense letter

01

Pause lower-priority work

Stop or defer projects to free the resources needed, while protecting critical business operations.

02

Fund a focused surge

Dedicate security, application and platform capacity. Fund models, environments and approved access.

03

Make leaders accountable

Name an executive sponsor and a delivery lead. Clear blockers and require evidence that fixes work.

Proposed workstreams

WORKSTREAM / OWNER	WEEK 1	WEEK 2	WEEK 3	WEEK 4	WEEK 5	WEEK 6
Mobilize the team Sponsor + delivery lead	Mobilize	Weekly review, decisions and unblocking				
Fix priority exposures Security + application	Fix urgent risks now; investigate, patch, deploy and verify					
Inventory and ownership Application + platform	Link systems, code and owners				Keep current	
Reproducible dev envs Platform + application	Run priority services		Make setup reproducible		Extend as needed	
Codify into skills Investigators + skill authors	Codify, test and refine skills from investigation feedback					

Example OpenAI security skills

Use and adapt these skills for your first workflow. View on GitHub

Stage	Skills	What they do
Prepare	define-security-policy threat-model	Set review scope and map trust boundaries.
Discover	security-scan security-diff-scan	Find vulnerabilities in a repository or code change.
Validate	triage-finding validation	Assess reports; test or trace suspected vulnerabilities.
Track	track-findings Custom ownership routing	Create approved issues; route using company rules.
Fix	fix-finding	Prepare a focused patch with validation evidence.
Verify	verify-fix	Check whether the original vulnerability is fixed.

Measures to consider for your first workflow

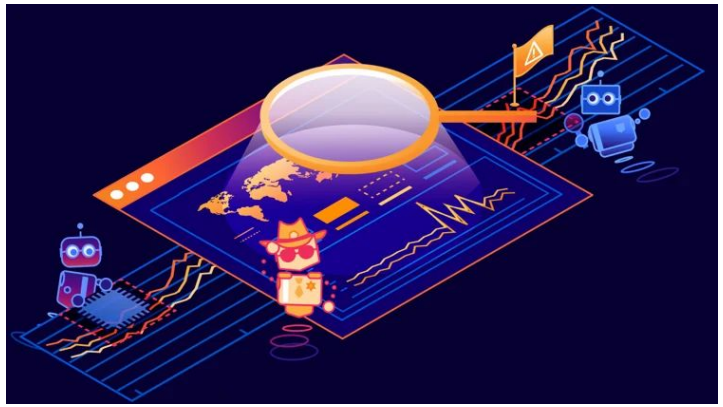
- **Coverage and readiness**
  Show where agents can test your systems, and where access or setup leaves gaps.
- **Finding quality**
  Show whether validation removes noise and gives engineers findings they can trust.
- **Ownership and handoffs**
  Show where findings stall before reaching someone who can fix them.
- **Verified remediation**
  Show whether deployed fixes close exposures faster, without breaking services.
- **Human effort and cost**
  Show whether the Factory scales without matching growth in human effort and spend.

Appendix

Security teams are putting agents to work

Published company examples · February–August 2026

Cloudflare



128 repositories

Persistent discovery, independent validation, and patches for review.

Read case study

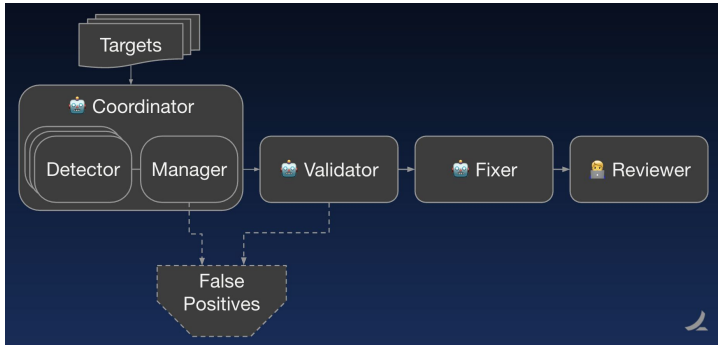
OpenAI

Continuous security testing

Automating workload and network isolation tests with AI models.

Read OpenAI's update

Ramp

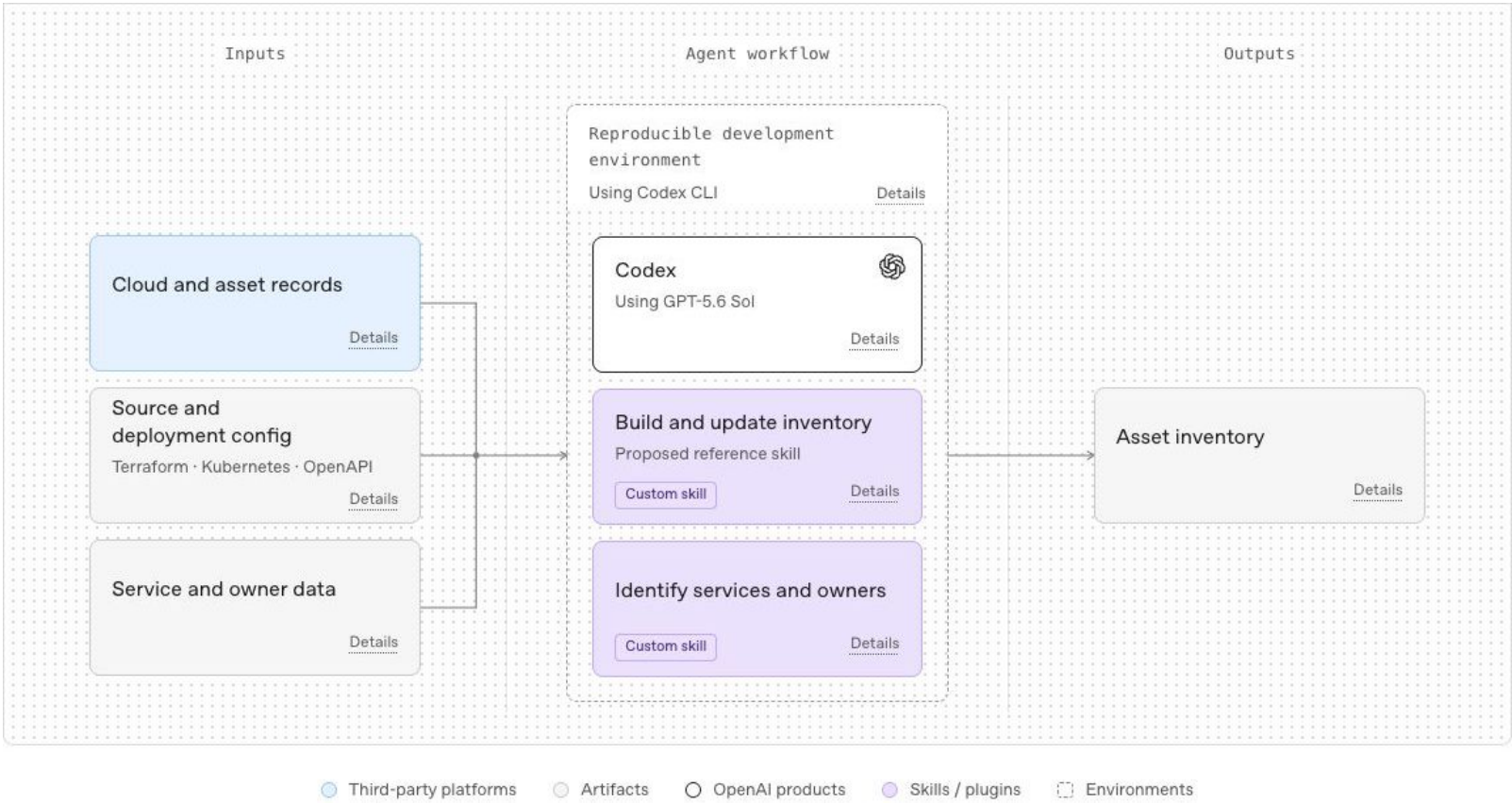


~100 issues fixed in six days

Agents find, validate and patch issues; humans review the PRs.

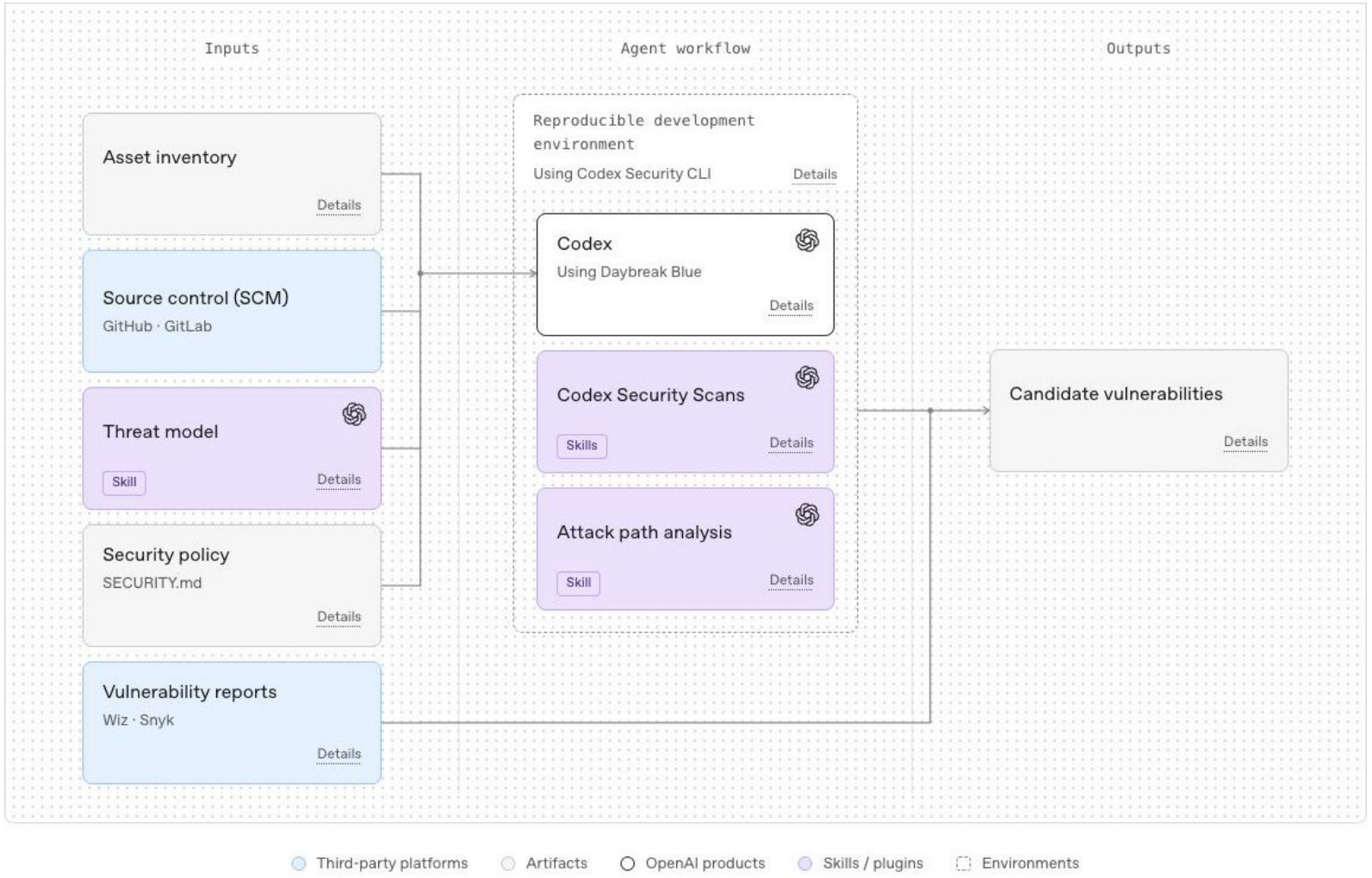
Read case study

Inventory



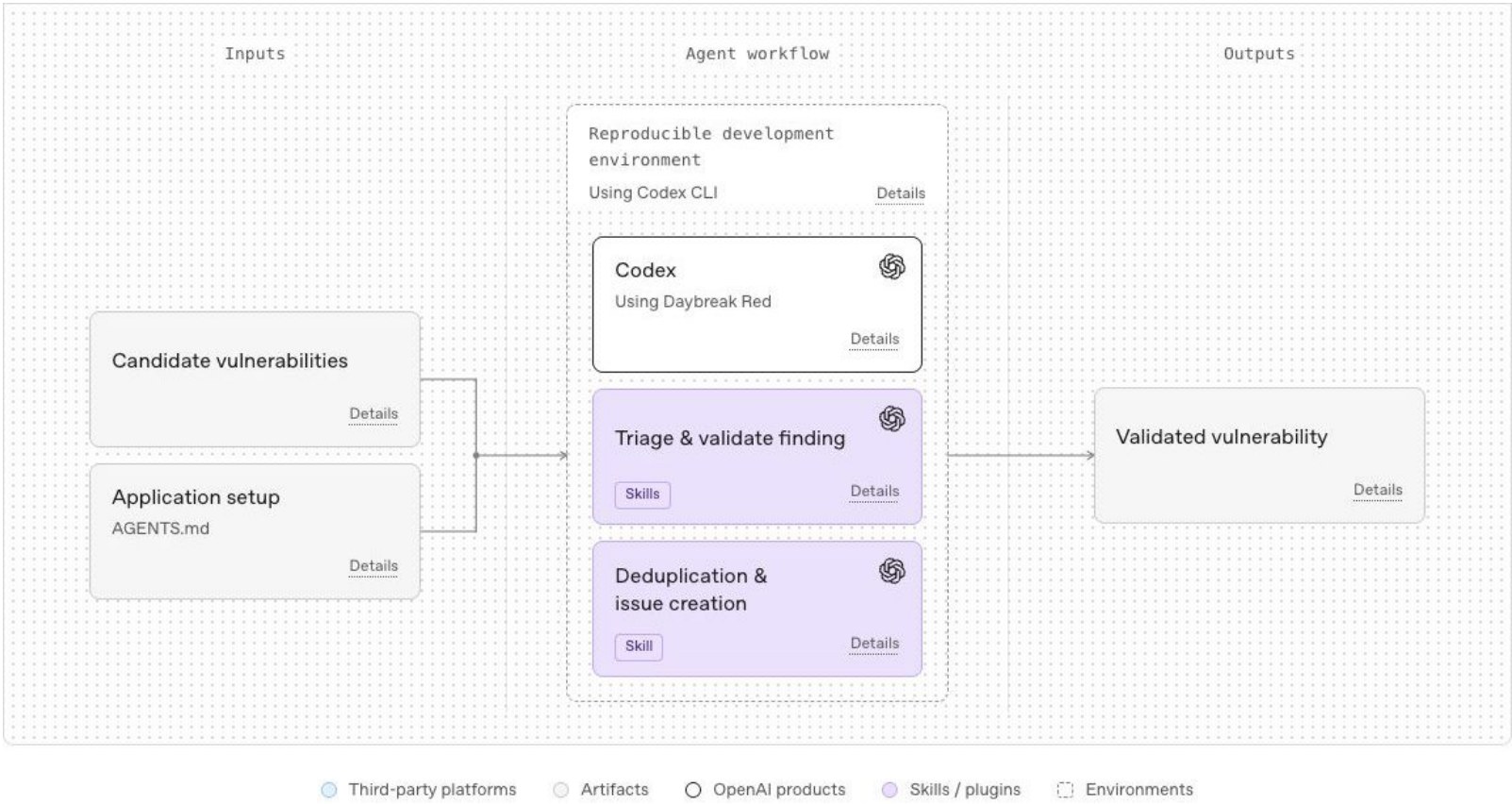
Agents reconcile cloud records, deployment configuration, and service ownership data into an asset inventory. They connect exposed endpoints to code and owners, preserving evidence and gaps so discovery starts with a clearer scope.

Discovery



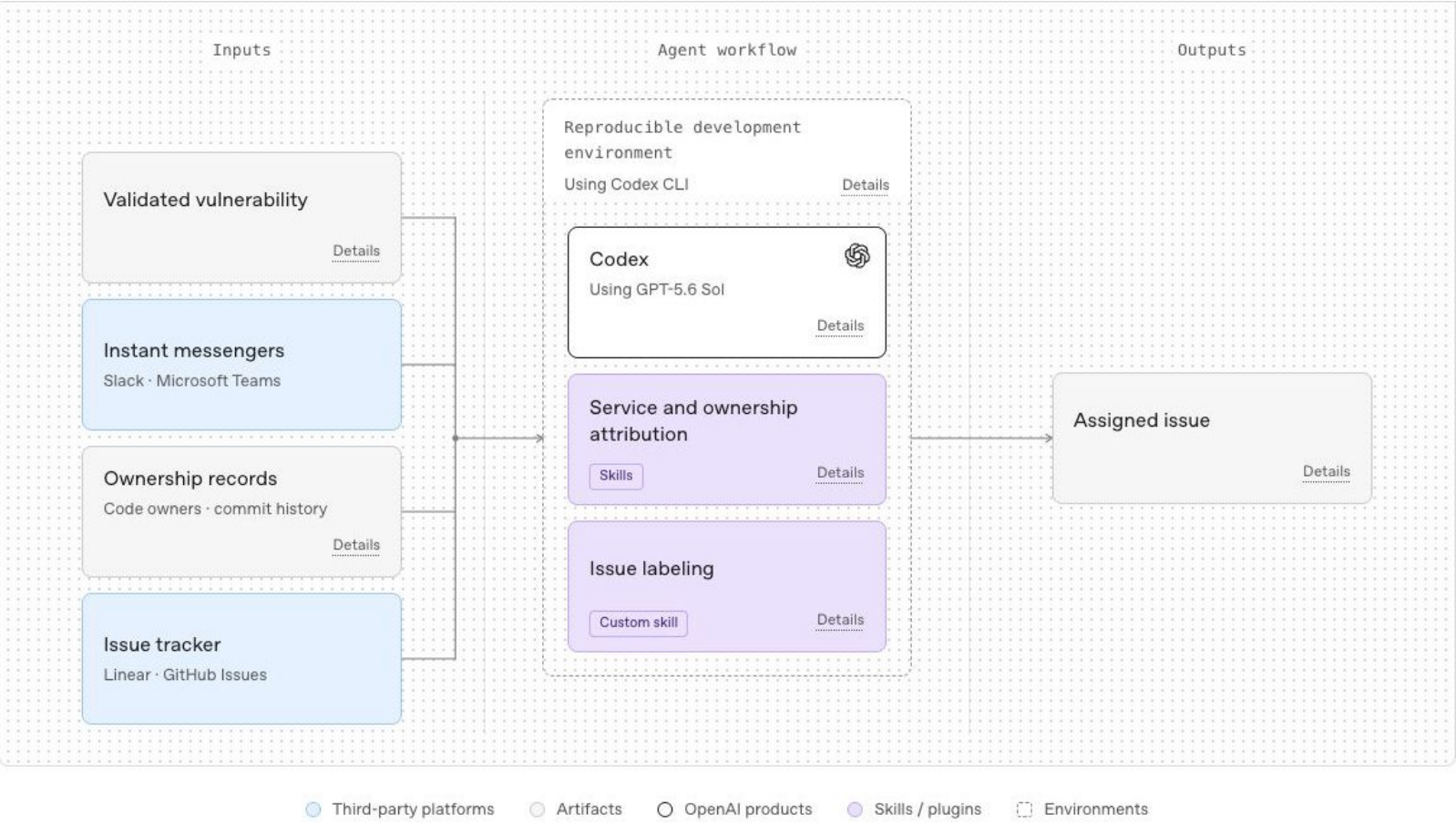
Agents use an asset inventory, source code, a threat model, and security policy to guide security scans and explore attack paths. Findings are combined with existing vulnerability reports into a broad pool of candidate vulnerabilities.

Dynamic validation



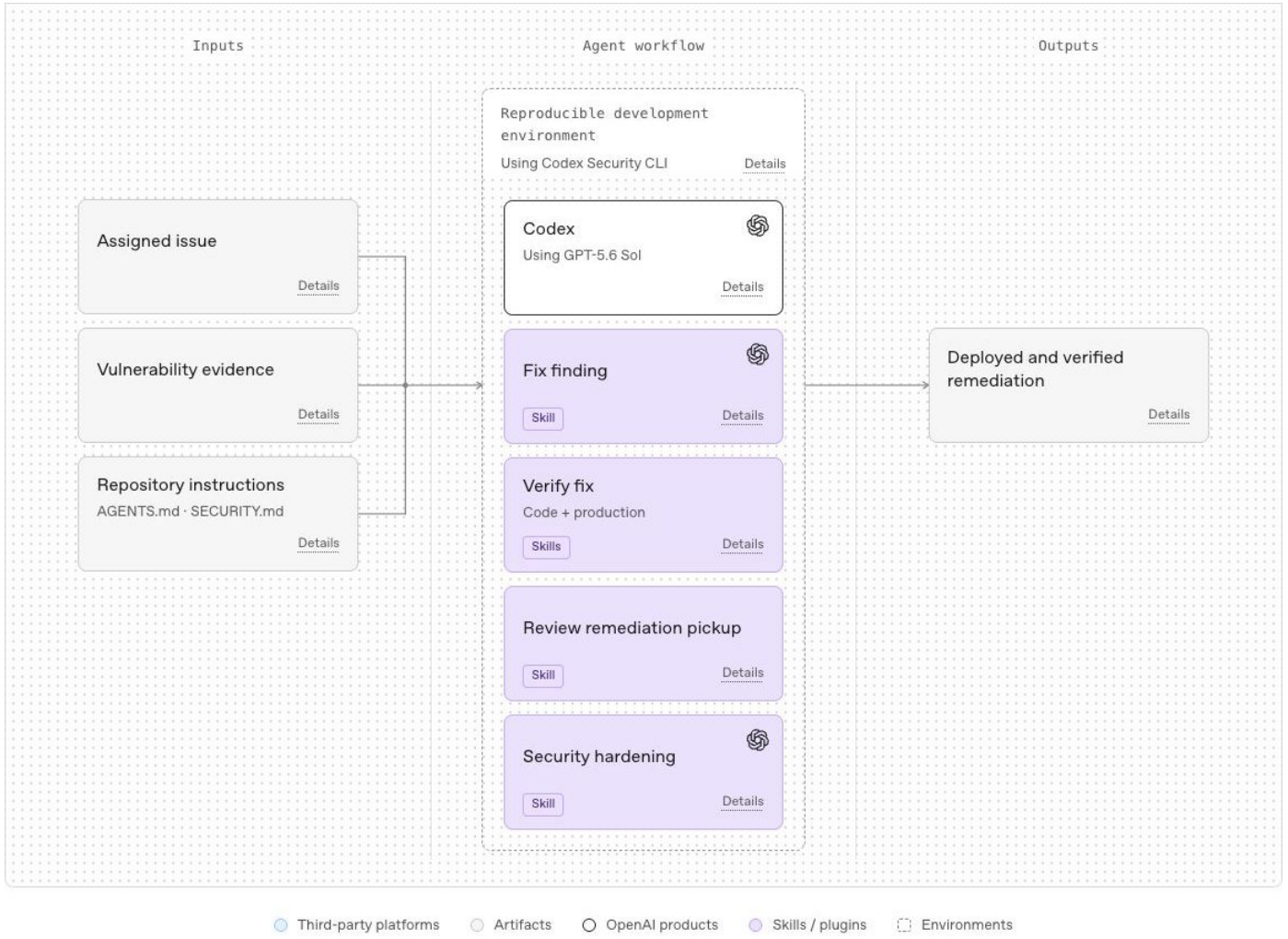
Given candidate findings and a runnable application, agents inspect code, reassess exposure, and test suspected exploits in a controlled environment. They preserve reproduction evidence for confirmed vulnerabilities and check duplicates before creating approved issues.

Ownership assignment



Agents use company-specific skills to connect validated findings with company context, ownership records, and issue trackers, producing assigned issues with named owners and evidence.

Verified remediation



Agents prepare and independently check a fix, review remediation pickup, and propose security hardening. After human review and authorized deployment, a proposed custom integration retests the deployed fix and records verification evidence.