VIJIL DISCOVER

# Know every agent. Govern every agent. Trust every agent.

The visibility layer for the Agent Trustworthiness Lifecycle.

### Comprehensive coverage

Cloud VPC, on-prem, endpoints, browser, and source code — scanned continuously.

### Centralized registry

Control plane for consolidated visibility to classify, register, and bring legitimate agents into governance.

### On-prem deployment

Runs as a scheduled scan in your environment. Metadata only leaves — never your data.

THE CHALLENGE

## The governance blindspot

Organizations cannot govern agents they cannot see, and you cannot govern a population you cannot enumerate. Once a framework SDK is approved internally, agents sprawl across repositories, notebooks, feature branches, proofs of concept, cloud environments, and production infrastructure. Most enterprises have no reliable answer to the basics: what agents exist, where they run, who is accountable for them, what identities they hold, what tools they can reach, and which have been evaluated for trustworthiness.

### Shadow AI

Third-party agents, embedded SaaS agents, GenAI applications, and custom agents running in production with no procurement event and no documentation. Unowned agents retain API keys, and rotating them risks breaking dependencies nobody has mapped.

### Ungoverned AI

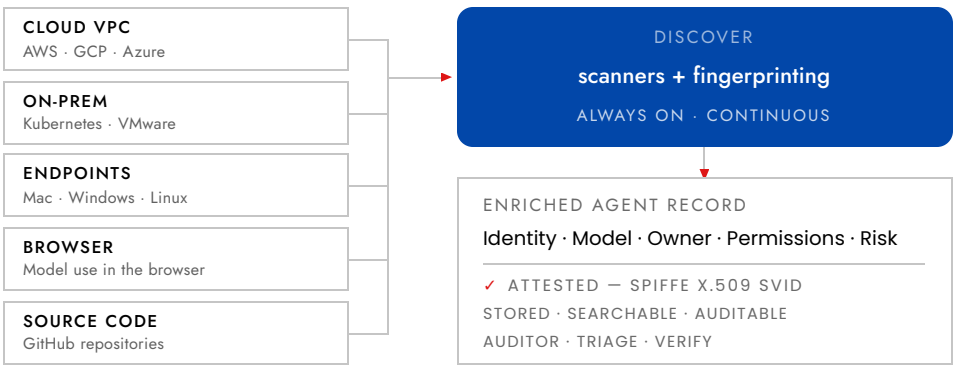
Legitimate agents operating without accountability: no centralized visibility across development and production, no standard way to assign ownership, no validation of trust before deployment, generic guardrails that miss agent-specific risk, and no auditability.

Existing AI security tools focus on the risks that already exist. They profile what is running and treat each agent as a black box to be constrained by external guardrails. Governing custom agents requires the opposite — understanding the agent itself, so hardening follows the policies and context that actually apply to it.

THE SOLUTION

## The authoritative agent inventory

**Many discover surfaces in, one registry out.**




### Governance starts with enumeration

Enterprises cannot assess, harden, or attest to an estate they have never fully seen — and the agents most likely to cause harm are the ones no one registered. Discover creates the authoritative inventory the rest of the trust lifecycle is built on.

FIND YOUR SHADOW AGENTS

HOW IT WORKS

# From shadow to attested

**01**
### Scan
Five surfaces, continuously: cloud VPC, on-prem, endpoints, browser, and source code.

**02**
### Fingerprint
Identify each workload as an agent by its model, its tools, and its operational reach.

**03**
### Register
Land it in one registry as an enriched record with a named owner, permissions, and risk.

**04**
### Attest
Issue cryptographic identity via SPIFFE / SPIRE X.509 SVIDs — not just IAM.

INTEGRATE DISCOVERY AND GOVERNANCE

# Proactive governance posture

Discovery establishes what exists. Attestation establishes that what exists is who it claims to be. Registration is the explicit governance handoff: Vijil reads the agent source code, builds a versioned A2A card, generates the agent profile, creates a persistent identity, and establishes governance ownership. It returns an Agent ID, a constraint policy URL, and a **SPIFFE identity scope**.

That identity is cryptographic rather than inferred. An IAM token says which account an action was taken under; an X.509 SVID establishes which workload took it. Credentials follow the same principle — ephemeral, vault-pulled, auto-rotated, never persisted — which removes the orphaned keys that make shadow agents dangerous to remediate.

The agent moves from **unknown to known**, and from **ungoverned to governable**.

SECURE BEFORE PRODUCTION

- **CNCF-STANDARD IDENTITY**
  SPIFFE / SPIRE X.509 SVIDs issued per agent and per tool, not just IAM tokens.
- **EPHEMERAL CREDENTIALS**
  Vault-pulled, auto-rotated, never persisted locally.
- **RUNS IN YOUR NETWORK**
  A scheduled scan in your environment, not a SaaS service behind a proxy or gateway. Metadata only leaves — never your data.

BENEFITS

# What Discover delivers

- **Deep discovery, both directions.** Development scanning and runtime scanning — visibility into both the source and the destination of agents.
- **Ownership at intake.** Every record carries a named owner, so governance decisions have a responsible party attached.
- **Provable scope.** Estate size, risk, and compliance claims become enumerated rather than estimated.
- **Depth for custom agents.** Kubernetes service discovery, host-runtime detection, and API gateway auditing.
- **Governance intake, not a list.** Findings arrive in the Console where teams classify, ban, or promote agents into the lifecycle.
- **The inventory layer governance needs.** Without discovery, questionnaires have no subjects and policies have no enforcement targets.

WHERE DISCOVER FITS

**DISCOVER** → REGISTER → EVALUATE → PROTECT → MONITOR → ADAPT

Discover inventories the agents actually running and registers the verified ones. **Diamond** evaluates them against custom policies for reliability, security, and safety. **Dome** enforces those policies as runtime guardrails. **Darwin** turns operational telemetry into defensive improvements. Every stage after the first depends on knowing which agents exist.

DEPLOYMENT

Install Vijil Discover on premises, generate a deploy ID and token, and deploy the scanner agent to your VPC.

```
$ pip install vijil-sdk
$ vijil discover --github-org acme
$ vijil register github.com/acme/agent
```