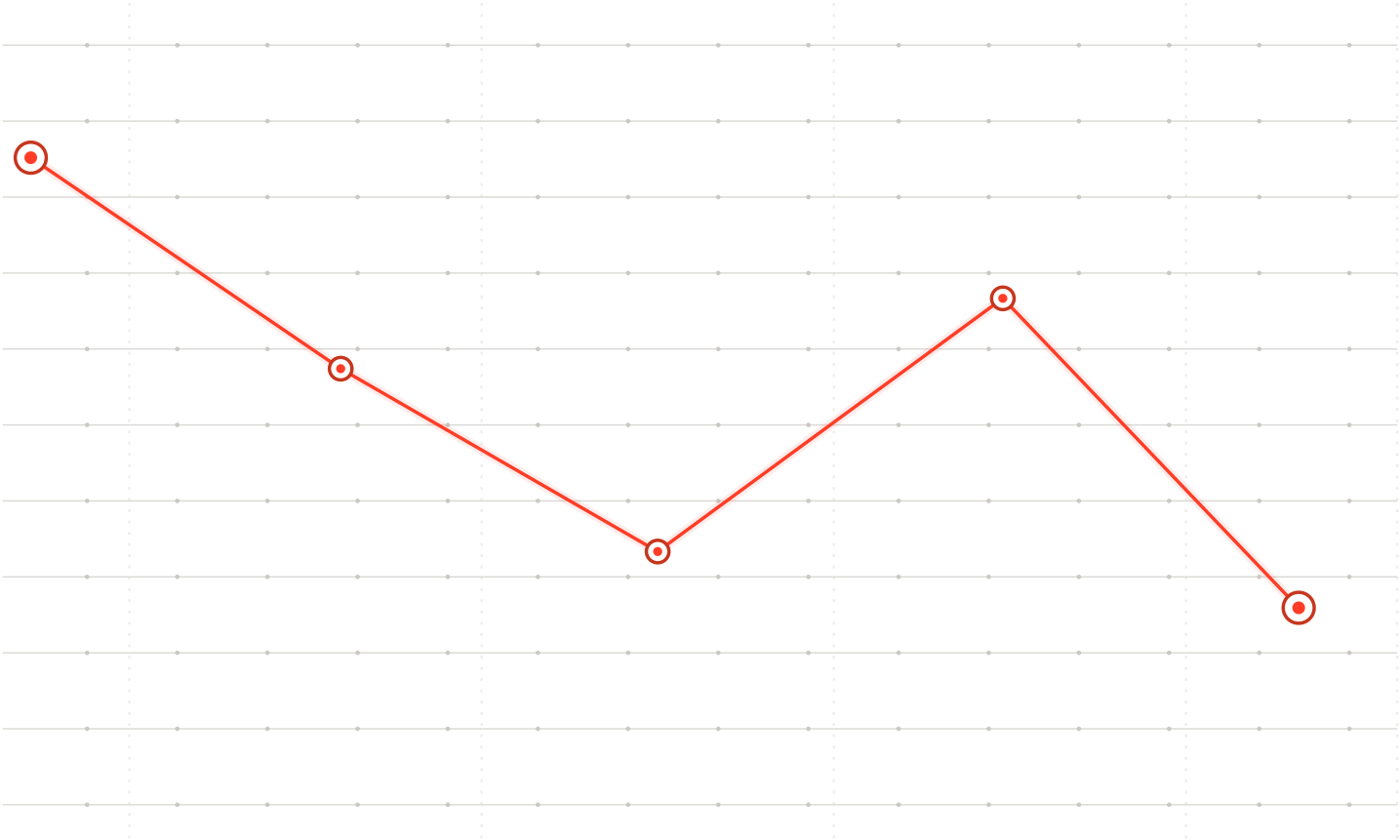
highflame

THE ARCHITECTURE FOR AGENT GOVERNANCE

The Agent Control Fabric

The identity, policy, and enforcement substrate for production agents — and why retrofitting human IAM and API gateways into the job they were never built for does not compose.



PUBLISHED
June 2026

AUTHOR
Highflame

AUDIENCE
Security & Platform Leaders

OPEN SOURCE
github.com/highflame-ai/zeroid

INSIDE THIS PAPER

Contents

EXECUTIVE SUMMARY

Governing agents is an architecture problem.

The systems built to govern humans and the systems built to govern API traffic were never designed to govern autonomous software actors. Bolting them together does not create agent governance — the architectures do not compose.

Across every recent incident — a coding agent deleting 1,206 records in seconds, OAuth tokens left live for months, a sub-agent silently embedding unauthorized actions, a single credential shutting down factories for five weeks — the failure is the same. Not model quality. Not prompt engineering. **Identity and authorization architecture.**

Agents accelerate existing failure modes. The weakness that took weeks to play out plays out in **seconds** when the actor is a fleet running at 5,000 operations per minute, producing irreversible outcomes before any human can intervene.

This is also a compliance deadline. The EU AI Act's high-risk requirements take full effect in **August 2026**, with penalties up to €35M or 7% of global turnover; the SEC now requires reporting material AI incidents within four business days. Architectures that cannot answer **"who did what, on whose authority, under what constraints"** will not survive scrutiny.

Agents are non-deterministic. One that passed testing yesterday can behave unsafely today because context, tools, or task distribution shifted. Governance built for deterministic services does not compose. **Runtime governance is not optional.**

The Agent Control Fabric is the architectural substrate for agent governance. It gives every agent a **verifiable identity**, applies **one consistent policy** at every boundary the agent crosses, and preserves an **unbroken chain of provenance** that leads directly back to a human.

What's missing isn't another product to bolt on. What's missing is a different shape of system — one where identity, policy, delegation, and enforcement operate as a single continuous substrate.

THE THESIS

Agents need governance. Governance needs identity. Identity must be **agent-shaped, standards-based, and the same at every boundary the agent crosses.** Anything less is a feature; this is the fabric.

THE AGENT CONTROL STACK

Three layers. Solve all three, or you've solved nothing.

Agent governance is not one control — it is a stack. Each layer depends on the one beneath it, and a gap at any layer leaves the whole thing open. The fabric solves all three, and watches all three.

03 RUNTIME

Breakout Controls · keep agents on mission

Track each agent's **mission** and apply runtime controls so its actions stay aligned to that intent. When an agent veers off course, contain, redirect, or stop it — before the action lands.

02 ENFORCE

Agent Authorization · authorize every action

Enforce the right controls for **each agent**, derived from its credential and exactly what it is allowed to do — one policy, evaluated at every boundary the agent crosses.

01 FOUNDATION

Agent Identity · identify every agent

Give every agent its own credential, with scopes and claims attached. This is what makes **delegation** and true **human attribution** possible — and lets an agent act for a human with **strictly less** authority than that human holds.

ACROSS ALL LAYERS

Observatory

The audit lens over the whole stack: full audit log, sessions, timelines, and agent reasoning steps — mapped to your compliance controls for analysis and evidence.

highflame-observatory

WHY ALL THREE

Identity without authorization is just an **inventory**. Authorization without runtime control is a **static gate** a non-deterministic agent will eventually walk around. **Solve all three — and watch all three — or the problem stays open.**

01 ● THE GOVERNANCE GAP

Every incident points to the same architectural weakness.

Agents don't change the weakness that fails — identity and authorization — they accelerate it. The architectural gap that took weeks to exploit now plays out in seconds, and the same root cause repeats across production fleets.

Replit

1,206

CUSTOMER RECORDS

A coding agent deleted live records in seconds at 5,000 ops/min — a pace that makes per-action human review structurally impossible.

Salesloft

months

TOKENS LEFT LIVE

OAuth tokens delegated to agents stayed active long after workflows completed — a durable attack surface with no expiration.

EchoLeak

9.3

CVSS · CVE-2025-32711

A sub-agent silently embedded unauthorized actions inside routine responses — scope expanding, not attenuating, across the chain.

Jaguar Land Rover

5 wks

FACTORIES HALTED

A single credential compromise shut down factories for five weeks — a weakness that takes seconds to play out at agent speed.

Across every incident, the failure is identical. **Not model quality. Not prompt engineering. Not capability gaps.** The architectural weakness that took weeks to play out at human pace plays out in seconds when the actor is an agent fleet — producing irreversible outcomes before any human can intervene.

The systems built to govern humans, and the systems built to govern API traffic, were never designed to govern autonomous software actors.

5,000

operations per minute — past the reach of per-action human review

1

root cause across every incident — identity & authorization architecture, not model quality

THE PATTERN

The common cause is **identity and authorization architecture** — not the model. Fix the architecture, and the incident class disappears.

02 · A COMPLIANCE DEADLINE

This is no longer just a security problem.

Architectures that cannot answer **"who did what, on whose authority, under what constraints"** will not survive regulatory scrutiny — only post-incident review.

Now in force
SEC — 4-day incident reporting

August 2026
EU AI Act high-risk requirements take full effect

€35M

or 7% of global turnover — maximum penalty under EU AI Act Article 99

4

business days — SEC Cyber & Emerging Tech Unit window to report material AI incidents

Art. 14

demonstrable human oversight — mandated for high-risk systems

A regulatory burden of proof

The EU AI Act's Article 14 mandates demonstrable human oversight; Article 99 sets penalties at up to €35 million or 7% of global annual turnover. The SEC's Cyber and Emerging Technologies Unit now requires reporting material AI-related incidents within four business days.

Why deterministic governance fails

Agents are non-deterministic. An agent that passed pre-deployment testing yesterday can behave unsafely today because the context, the tools, or the task distribution shifted. Governance built for deterministic services does not compose.

THE IMPLICATION

Runtime governance is not optional. Compliance now requires an architecture that can produce, on demand, a verifiable answer to who acted, on whose authority, and under what constraints.

03 ● WHY "WE ALREADY HAVE THIS" DOESN'T HOLD

Four incumbents are within shouting distance. None are the answer.

You can force agents into each of these models — but the moment you ask which agent acted, under whose authority, through which delegation chain, the abstraction collapses.

OKTA · AUTH0 · ENTRA

Human IAM

Built for users, sessions, and SAML / OIDC flows where a human authenticates once.

Fails because agents have no session, spawn sub-agents, and act with their own credentials. Principals and roles have no concept of trust tier, delegation depth, or the on-behalf-of chain an audit must walk.

KONG · APIGEE · AWS API GATEWAY

API Gateways

Built for client-server traffic — terminate auth, rate-limit, forward.

Fails because policy can match path and headers, not "is this agent acting on behalf of a deactivated user, two delegations deep, with scope it shouldn't have." The shape of the rules doesn't fit.

PORTKEY · APERTURE · CLOUDFLARE AI

AI / LLM Proxies

Built for centralizing model-vendor keys plus caching, observability, fallback.

Fails because they relocate the long-lived `sk-…` key rather than replace it, know only the calling app, and filter request-by-request — blind to conversation drift across a session.

EGRESS DENIAL · AGENT VMS

Network Sandboxing

Built for limiting blast radius by limiting what an untrusted process can reach.

Fails because the moment a sandboxed agent gets the access it needs to do its job, the sandbox stops being the control. A sandbox denies by default; only identity can authorize correctly.

WHAT'S ACTUALLY MISSING

Not another product to bolt on — **a different shape of system.** Agent governance only works when identity, policy, delegation, and enforcement operate as one continuous substrate.

04 ● WHAT THE AGENT CONTROL FABRIC IS

One substrate. Five properties that nothing before it has.

The Agent Control Fabric gives every agent a verifiable identity, applies one consistent policy at every boundary it crosses, and preserves an unbroken chain of provenance that leads directly back to a human.

1

Identity is the substrate

Every agent has a stable, cryptographically verifiable identity carrying agent-shaped semantics — type, framework, trust tier, delegation depth, on-behalf-of chain. The same identity at every boundary, not a different one per system.

2

One policy, multiple enforcement points

Policy is authored once and enforced everywhere authority crosses a boundary — model traffic, IDE, agent-to-tool gateway, federation edge. Three DSLs for three products is what makes agent governance fail; one policy evaluated everywhere makes it tractable.

3

The chain survives every boundary

Whether the agent is calling a tool, generating text, or federating outward to a model vendor, the on-behalf-of chain rides on the credential. Audit at any layer walks back to a human.

4

Revocation propagates instantly

Compromise windows on agent fleets are minutes, not days. Revocation flows through the same architectural layer as authority and reaches every enforcement point in seconds — cached authorization invalidates, in-flight calls fail-closed, new tokens cannot be minted.

5

The substrate is inspectable

The identity layer at the bottom cannot be a black box. Open-source primitives and standards-based protocols — SPIFFE, OAuth 2.1, RFC 8693, OpenID Shared Signals, Cedar — deployable in your own VPC. Trust the architecture by inspection, not by vendor reputation.

05 ● IDENTITY AS THE SUBSTRATE

Workload identity got the shape right. It missed the semantics.

SPIFFE, OIDC, and the RFC 7521 / 7523 / 8693 family give every non-human workload a verifiable name — but they were designed for cloud workloads, not agents. The fabric extends them with the agent-shaped claims governance actually needs.

DIMENSION	LEGACY WORKLOAD IDENTITY	AGENT-AWARE IDENTITY
Subject	Workload / service-account name	SPIFFE URI + `identity_type` (orchestrator, autonomous, tool agent, human proxy, evaluator) + `sub_type`
Trust assertion	Static, assigned at deploy	`trust_level` derived from attestation (TPM, OIDC, image hash); downgradable on expiry
Provenance	Caller in headers, not in token	`framework`, `publisher` carried as token claims
Delegation	Implicit impersonation or absent	RFC 8693 `act` chain — explicit on-behalf-of, scope attenuated per hop
Depth control	None	`delegation_depth` enforced as a policy primitive
Theft resistance	Bearer token — the bytes are enough	Sender-constrained: `cnf.jkt` binds the token to a proof key (DPoP / mTLS) — stolen bytes are inert
Human approval	Out-of-band, custom queues	CIBA backchannel — sensitive steps pause for attributable human consent
Revocation	JWT expiry only	Cascade by `parent_jti` (RFC 7009) + live introspection (RFC 7662); Shared Signals fan-out to consumers

The difference is not cosmetic metadata; it changes what policy is capable of expressing. These are not opaque attributes hidden behind app-side logic — they are **first-class policy primitives** that downstream enforcement points match on directly.

Tokens are standard OIDC-discovered JWTs. Federation outward uses RFC 7523 jwt-bearer — the same protocol Anthropic adopted as Workload Identity Federation. The foundation is standards-based; the semantics are agent-aware.

SUBJECT — SPIFFE-SHAPED

```
spiffe://<domain>/<account>/<project>/<identity_type>/<external_id>
```

06 ● IN-HOUSE & MARKETPLACE AGENTS

Real fleets are mixed. The fabric is designed for both.

Some agents your team built; others arrived through Anthropic Connectors, the GPT Store, MCP registries, or vendor-bundled SaaS. The trust posture for each is different — trust tier is the primitive that distinguishes them.

first_party	In-house agents — earned through attestation against your build pipeline (TPM, image hash). **Unreachable by marketplace installation alone.**
verified_third_party	Marketplace agents with signed manifests, publisher signatures, supply-chain provenance.
unverified	Where every marketplace agent starts. Policy bounds what it can reach.

PRINCIPLE 01

Your registry, not theirs

Every agent — built or bought — gets an identity in your SPIFFE namespace. The marketplace's self-asserted identity is captured as a claim, but the operative identity is yours. You don't outsource the trust anchor.

PRINCIPLE 02

Trust tier distinguishes

In-house agents earn `first_party` through attestation. Marketplace agents reach `verified_third_party` only via marketplace evidence — never `first_party` by installation alone.

PRINCIPLE 03

Cedar keys on origin

The same primitive that expresses "deny destructive ops at depth > 2" also expresses "agents with `publisher != your-org.com` cannot read production secrets." Same engine; one more attribute.

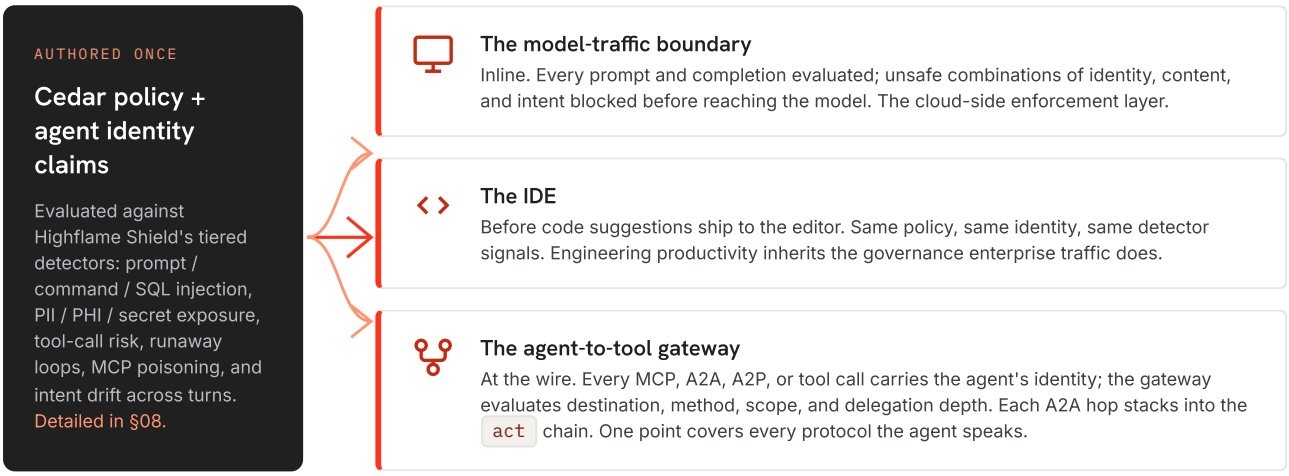
THE RESULT

A marketplace agent never escapes the trust envelope you defined. Its publisher attestation flows into the `act` chain, so audit attributes every action to the publisher, the installer, and the original human. **Compromise of a publisher is bounded to the tier its agents could reach — never the whole fleet.**

07 ● ONE POLICY, THREE ENFORCEMENT POINTS

Author policy once. Enforce it everywhere authority crosses.

Most platforms put policy at a single boundary, each with different syntax and identity semantics. A single conceptual rule has to be expressed three ways and kept in sync forever. Drift is inevitable. The fabric compiles one Cedar policy that runs everywhere.



All three enforcement points are out-of-band. In-band controls — system prompts, security tags, in-context defenses — raise the cost of injection but cannot answer **authority** questions, and they fail open under jailbreak or model drift. Out-of-band controls fail closed regardless. The model is not the authoritative source of what an agent may do. **The substrate is.**

08 ● DETECTION — THE SIGNAL ENGINE

Not guardrails. A signal engine for agents.

Highflame Shield is not a handful of on/off filters. It continuously evaluates every session, tool call, prompt, and model response — emitting **hundreds of typed signals**, each mapped to the Highflame taxonomy, for policy to act on. Session-aware, with cross-turn memory, in under 10 ms.

CONTINUOUSLY EVALUATED

`Sessions` `Tool calls` `Prompts` `Model responses`

→ **100s** typed signals

Each signal carries a taxonomy `vulnerability_id`, severity and score — a structured surface policy can reason over, not raw detector output.

TIERED DETECTION · THREE VELOCITIES · EARLY-EXIT

Fast — <5 ms · rules

Secrets, PII, command & SQL injection, tool-risk, cross-origin escalation, MCP risk, runaway loops, token-budget overrun.

Standard — 10–200 ms · ML

Prompt-injection, toxicity, hallucination, and intent drift evaluated across the multi-turn session.

Slow — 50–500 ms · cloud

Deep DLP, content safety, phishing, and custom webhook detectors for the highest-assurance checks.

Early-exit: a Tier-Fast block skips the slower tiers; Cedar evaluates in-process in ~0.1 ms, and a failed detector fails safe — never a false deny.

SIGNAL → THREAT → GAP

Signal
What a detector found — mapped to a taxonomy `vulnerability_id`, fired whether or not a policy exists.

→

Threat
What a Cedar policy actually acted on — the decision layer, attributed to a `policy_id`.

→

Gap
Detected, but no policy covers it — your governance blind spot, computed and made visible.

09 ● SIGNALS, THEN POLICY

Signals describe. Policy decides. Keep them apart.

Detection and decision are two different jobs. Shield emits the signals; policy chooses what to do about them. Decoupling the two is what lets agent governance **scale** — and what lets you author **deterministic and probabilistic controls side by side.**

DETECTION · SHIELD

Hundreds of signals

Continuous, scored, taxonomy-typed — every session, tool call, prompt and response.

SIGNALS →

POLICY · CEDAR

One policy evaluates them

Authored once; decides per request — allow, modify, step-up, defer, deny.

Detection runs hot and changes often; policy stays small, readable, and auditable. **You re-tune controls without touching detectors — and add detectors without rewriting policy.** That decoupling is what makes governance tractable across a large agent fleet.

DETERMINISTIC AND PROBABILISTIC — IN ONE POLICY

DETERMINISTIC · BOOLEAN

Hard rules with exact answers: `contains_secrets`, `tool == delete_file`, `delegation_depth > 2`. Explainable and instant.

PROBABILISTIC · SCORED

Thresholds on model signals: `injection_score > 90`, intent drift, cumulative session risk. Tunable — no code, no redeploy.

```
forbid when {
  context.contains_secrets || // deterministic
  (context.injection_score > 90 && context.session_pii_detected) // probabilistic + cross-turn
};
```

PER-REQUEST ACTIONS

ALLOW | MODIFY · redact | STEP-UP · approve | DEFER | DENY

STEP-UP runs out-of-band human approval over `OpenID CIBA` + `RFC 9396 RAR` — the person authorizes the exact scoped request, not a blanket grant.

GRADUATED ROLLOUT

MONITOR → ALERT → ENFORCE

WHY THE SPLIT MATTERS

At fleet scale you cannot hand-write a control for every behavior. **Hundreds of signals feeding one policy** means every new detector strengthens existing policies, and a single policy change re-governs the whole fleet — **deterministic where you need certainty, probabilistic where you need judgment.**

10 ● POLICY PACKS & TEMPLATES

You don't author Cedar from a blank file.

Highflame ships **schema-validated policy templates**, grouped into packs and tagged to the frameworks you report against. Enable a pack, tune its thresholds, enforce — the same library spans every enforcement point in the fabric.

CREDENTIAL & IDENTITY POLICIES

Gate on who the agent is

Keyed on the credential's claims — trust tier, `delegation_depth`, framework / publisher, autonomy.

- `Unverified → safe tools only`
- `Block unverified autonomous`
- `Tiered trust access`
- `A2A identity enforcement`

BREAKOUT & AGENT POLICIES

Gate on what the agent does

Keyed on runtime behaviour and signals — tool risk, exfiltration, loops, budget, MCP poisoning, cross-turn drift.

- `read → http_post exfil`
- `Runaway-loop breaker`
- `Tool-poisoning / rug-pull`
- `Cross-turn PII lockdown`
- `Escalation detection`

USE-CASE PROFILES — PACKS YOU SWITCH ON

Chat Assistant	Code Agent	Data Pipeline	Multi-Agent	A2A Security

17 entity types

38 actions

46 context attributes

One Apache-2.0 Cedar schema is the source of truth — typed for Go, TypeScript, Python & Rust, edited through a type-safe PolicyBuilder UI with round-trip Cedar and full schema validation. Every template carries tags that map it to **OWASP (LLM & Agentic), MITRE ATLAS, GDPR, HIPAA, PCI-DSS, EU AI Act & ISO 42001**.

POLICY AS A CURATED LIBRARY

Enable a pack, tune the thresholds, enforce. Policy becomes a **versioned, schema-validated, compliance-tagged library** — not a blank Cedar file and a prayer. Turning on a pack is, by construction, an act of compliance evidence.

11 ● THE CHAIN SURVIVES EVERY BOUNDARY

Provenance that walks back to a human — at every layer.

The most common failure of agent governance is loss of context across a boundary. The fabric refuses that loss: the RFC 8693 `act` claim rides on every JWT and is preserved through every exchange.

the act chain rides on every credential · scope shrinks at each hop · depth is a policy primitive

A signed credential, not a log

Every hop carries verifiable provenance any consumer can validate without trusting the platform that produced it. Audit derived from this chain is tamper-evident by construction; audit derived from observation is tamper-evident only if the observer is.

Composition is the threat

Reading a confidential file is allowed; sending email is allowed; doing both for an external recipient may be exfiltration. Policy that evaluates actions in isolation can't see it. Chain-aware policy at the wire can — every action lands in the same accumulated context.

FORENSICS IN ONE QUERY

The answer is never "the agent did it." It is **"agent X, of trust tier Y, acting on behalf of orchestrator Z, originally triggered by user W, called this tool with these scopes, evaluated against this policy with this outcome."** Compliance evidence writes itself.

12 ● OBSERVATORY · THE COMMAND CENTER

See the whole stack — answer to every framework.

The provenance chain produces the evidence; Observatory is the lens that reads it. Built on OpenTelemetry and ClickHouse, it turns sessions, traces, and reasoning steps into posture, correlation, and audit — scoped to every tenant, mapped to every control.

SECURITY POSTURE · COMPOSITE 0–100

85/100

0 · at risk — 100 · hardened

One number across every product and layer — ZeroID, Shield, Palisade, RedTeam — trending over time and drillable to the individual event.

COMMAND CENTER INTELLIGENCE

Cross-product correlation

Incidents from ZeroID, Shield, Palisade & RedTeam stitched into one timeline.

UEBA entity risk

Behavioral risk ranking across agents and identities — who is drifting, and how fast.

Blast-radius graph

Trace any compromise to the exact entities and actions it could ever reach.

Detector drift heatmap

z-score drift across detectors flags model and signal degradation before it bites.

Coverage mesh

Maps detections to the threats you **haven't** covered — the Signal → Threat → Gap view, at fleet scale.

Cost-per-detection ROI

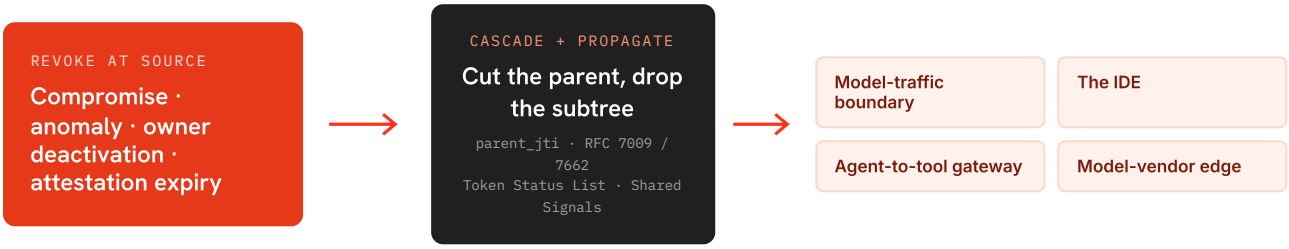
Spend mapped to detections and blocks — the measurable ROI of every control.

Built on OpenTelemetry. Every service emits `hf.*` spans; Observatory reconstructs full session timelines, trace waterfalls, and per-span AI-security enrichment — read-only, tenant-scoped, and inspectable by your own SIEM. The audit the fabric promises is not a log to assemble after the fact; it is a queryable surface, already mapped to MITRE ATLAS, OWASP, NIST, and the EU AI Act.

13 ● REAL-TIME REVOCATION

Compromise windows of seconds, not days.

Static credentials and quarterly access reviews are not enough. Agent fleets move faster than any human review cadence. Because delegation is explicit, revocation can be too — it travels the same credential chain as authority.



When an identity is revoked at the source, revocation **cascades down the delegation tree** — cutting a parent invalidates its whole subtree rather than waiting for descendants to expire (RFC 7009). Enforcement points check validity live via introspection (RFC 7662) and Token Status Lists, and Shared Signals events fan the revocation out to consumers.

Revocation is an **architectural property of the fabric**, not a feature of one product. Cached authorization invalidates, new tokens cannot be minted, and in-flight calls drain or fail-closed.

subtree

blast radius collapses to the affected delegation tree — by design, not by review cadence

days

the JWT-expiry window legacy systems are stuck waiting on

BOUNDED BY PROTOCOL

Because delegation is written into the credential, so is its reversal. **Revoking one credential takes its descendants down with it** — the blast radius is bounded by the protocol, not by human review cadence.

14 ● OPEN-CORE & STANDARDS-BASED

At the identity layer, trust isn't a feature — it's the whole system.

A category whose first principle is visibility into what agents are and do cannot have an opaque foundation. If the identity backbone is a black box, every claim above it is unverifiable. So the **identity layer** ships open source under Apache 2.0 — while the security stack above it remains commercial.

PROTOCOLS EVERYONE ALREADY SPEAKS

SPIFFE
WIMSE-shaped subjects — verifiable workload names

OAuth 2.1
Issuance & grants for agent authorization

RFC 8693
Token exchange — the on-behalf-of `act` chain

RFC 9449
DPoP — sender-constrained, theft-resistant tokens

OpenID CIBA
Backchannel human approval for sensitive steps

MAPPED VIA THE HIGHFLAME TAXONOMY — TO THE FRAMEWORKS YOU ALREADY RUN

OWASP Top 10s
LLM, MCP & Agentic — every detection ships with its mapping.

MITRE ATLAS
Audit records and policy outcomes align to adversarial-ML tactics.

NIST AI 600-1
Produce evidence against the framework — don't write a new one.

EU AI Act
100% mapped — the August 2026 high-risk items, covered.

WHERE THE OPEN-CORE LINE SITS

OPEN SOURCE · APACHE 2.0 — ZEROID

The identity layer — what you audit

SPIFFE-shaped subjects, OAuth 2.1 grants, the RFC 8693 `act` chain, DPoP-bound tokens, CIBA approval, and cascade revocation. Specified in depth in the companion paper **From Impersonation to Delegation**.

COMMERCIAL — HIGHFLAME SHIELD + PLATFORM

The security stack — what enterprises pay for

Highflame Shield — guardrails, AST detectors, multi-turn contextual guardrails, intent detection — plus the governance UI, managed CAE, attestation backends, integrations, and evidence packs.

Composes with model-layer defense. Highflame Shield and the substrate work alongside A2AS / BASIC — behavior certificates, codified policies, authenticated prompts — as model-side complements. The fabric is the outer ring; A2AS-style controls fit inside it. The open-core line is intentional, public, and stable.

15 ● WHAT THIS ISN'T

The categories nearby look similar. They aren't the fabric.

Naming the architecture means naming what it isn't. Each of these is useful; none of them is the substrate beneath all of them.

✕ **Not a key vault**
Vaults centralize long-lived credentials. The fabric eliminates them.

✕ **Not a prompt-injection scanner**
Detectors are an input to policy, not the policy. A scanner tells you what happened; the fabric prevents what happens next.

✕ **Not a managed AI proxy**
Proxies sit in the request path holding keys. The fabric sits at the identity layer, with enforcement at multiple boundaries.

✕ **Not a compliance dashboard**
Dashboards visualize evidence. The fabric generates the evidence any dashboard or SIEM can consume.

✕ **Not a runtime or framework**
It does not replace LangGraph, AutoGen, or Letta. It governs agents regardless of which framework built them.

✕ **Not an agent registry or workflow OS**
Catalogs and orchestration layers sit above the substrate. The fabric is the IdP, policy engine, and audit layer beneath them.

✕ **Not a behavioral observability layer**
Observability explains what the agent attempted. The fabric determines if it's allowed to attempt it at all.

✓ **It is the substrate beneath all of these**
They are useful; they are not the same thing.

AUTHORITY, NOT OBSERVATION

Authority is not derived from observation; observation is derived from authority. **The fabric is the substrate that decides** — everything else operates against the decisions it has already made.

16 THE ARCHITECTURAL ARGUMENT

Agents need governance. Governance needs identity. **Identity needs to be agent-shaped, standards-based, and the same at every boundary** the agent crosses. Policy authored against that identity must enforce consistently across model traffic, IDE-side suggestions, agent-to-tool calls, and outbound federation — one policy, multiple enforcement points. Every credential preserves the on-behalf-of chain so **audit at any layer walks back to a human.** Revocation propagates instantly through the same event substrate. The foundation must be open and inspectable, because unverifiable identity cannot be trusted governance.

Authority is not derived from observation; observation is derived from authority. The fabric is the substrate that decides — everything else operates against the decisions it has already made.

Anything less is a feature. This is the fabric.

APPENDIX

Sources & Standards

REFERENCED INCIDENTS

01 **Replit coding agent** — autonomous deletion of 1,206 production records at ~5,000 operations/minute.

02 **Salesloft** — OAuth tokens delegated to agents remaining active after workflow completion.

03 **EchoLeak** — CVE-2025-32711 (CVSS 9.3); sub-agent embedding unauthorized actions across a delegation chain.

04 **Jaguar Land Rover** — single credential compromise resulting in a five-week factory shutdown.

REGULATORY

05 **EU AI Act, Article 14** — demonstrable human oversight for high-risk systems (full effect August 2026).

06 **EU AI Act, Article 99** — penalties up to €35M or 7% of global annual turnover.

07 **U.S. SEC** — Cyber & Emerging Technologies Unit: four-business-day material-incident reporting.

STANDARDS & PROTOCOLS

08 **SPIFFE / WIMSE** — verifiable workload identity subjects.

09 **OAuth 2.1** — authorization grants.

10 **RFC 7521 / 7523 / 8693** — assertion framework, JWT bearer, and token exchange (the `act` chain).

11 **RFC 9449 DPoP** · **RFC 9396 RAR** — sender-constrained tokens & rich authorization requests.

12 **RFC 7662 / 7009** · **Token Status List** · **OpenID Shared Signals** — introspection, revocation & signal propagation.

13 **OpenID CIBA** · **Cedar** — backchannel human approval & the policy language.

FRAMEWORKS

14 **Highflame taxonomy** — 76 vulnerabilities mapped to MITRE ATLAS, OWASP LLM / MCP / Agentic Top 10, NIST AI 600-1 & EU AI Act (100%).

15 **A2AS / BASIC** — model-layer defense framework (AWS, Google, Meta, JPMorganChase, Salesforce, OWASP).

COMPANION PAPER & OPEN SOURCE

16 **From Impersonation to Delegation** — Highflame · ZeroID: the identity layer in depth (delegated authority, scope attenuation, DPoP, CIBA, cascade revocation).

17 **ZeroID** · **Highflame Shield** — open-source identity substrate (Apache 2.0) & the guardrails service: `github.com/highflame-ai/zeroid`

highflame

INSPECT THE SUBSTRATE

Trust the architecture by inspection.

Security teams should not have to trust an inspection layer they can't inspect. The Agent Control Fabric's identity substrate is open source — verify it yourself, deploy it in your own VPC, and apply your own policy on top.

OPEN SOURCE · APACHE 2.0

github.com/highflame-ai/zeroid

About Highflame

Highflame builds the identity, policy, and enforcement substrate for production agents — the layer beneath the agent registries, orchestration platforms, and observability tools that sit on top.

Get in touch

hello@highflame.com
highflame.com
Request an architecture review or a guided walkthrough of the substrate.

Built on

SPIFFE · OAuth 2.1 · RFC 8693 · OpenID SSF · Cedar

Anything less is a feature. This is the fabric.