Agentic AI Security:
Why the Control Plane Must Change

Early enterprise AI projects began as LLM chatbots. Today, the center of gravity has shifted to AI agents—autonomous, tool-using systems that call APIs, query data, update records, and coordinate workflows. Increasingly, these systems operate as multi-agent networks where tasks are divided across specialized agents that plan, delegate, and compose results.

Adoption is accelerating but uneven. Emerging frameworks such as MCP and A2A are driving deeper integrations and real business value, while simultaneously exposing serious gaps in existing security and privacy protections. The constraint to scale is no longer business interest or technical capability; it is the absence of a trusted system designed for real-time agent execution rather than human-initiated sessions. [1]

The Problem: What breaks with agentic AI

AI doesn't break the pillars of security like cryptography, it breaks where, when, and how trust is enforced. Six critical trust problems arise from agentic AI and will drive security and privacy architectural changes. The following table maps each of these six problems to its corresponding solution:

Problem	Core Issue	Solution	Key Innovation
P1: Sessions → Actions	Risk occurs per operation, not per login	**S1: Edge Authorization**	Just-in-time credentials and blocking at the tool boundary
P2: Smarter Agents	Autonomy increases unauthorized action risk	**S2: Per-Agent NHI**	Non-human identity with autonomy-aware safeguards and mid-run revocation
P3: N² Complexity	Point-to-point topologies don't scale	**S3: Brokerless Pub/Sub**	O(N) complexity with distributed policy enforcement
P4: Centralized Trust	Hub-and-spoke creates SPOFs and latency	**S4: Decentralized Enforcement**	Trust Rules validated locally at every edge
P5: Machine-Speed Harm	Tokens have blast radius, delayed revocation	**S5: Point-of-Use Credentials**	Ephemeral, tool-scoped tokens with per-call permissions
P6: Weak Provenance	Fragmented logs, no signed evidence	**S6: Signed Audit Trails**	Cryptographic, tamper-evident lineage from edge to effect

Standards are crystallizing, creating a brief window before technical debt calcifies

Model Context Protocol (MCP) and A2A are moving from proposals to usable building blocks with clients, SDKs, and providers. This creates a narrow window to lock in agent-native architectures before retrofit paths become technical debt. Agent-native architectures require per-action edge decisions, signed intents, short-lived capability-scoped auth, and distributed revocable enforcement. Teams that bolt agents onto human-centric IAM/PAM/CASB inherit latency, blind spots, and single points of failure. In contrast, teams adopting agent-native trust primitives set the baseline others must match. Examples of agent-native primitives already in use include SPIFFE/SVID for workload identity, OPA/Rego for policy decisioning, proof-of-possession tokens, and hash-chained, tamper-evident logs.

Privileged Access Management (PAM) establishes a control plane, designed to govern human administrators (e.g., RDP/SSH into production) logging into applications. However, PAM alone is insufficient for autonomous, non-human actors. Securing agents requires a data-centric trust architecture that inherits PAM's building blocks but applies them at the agent/tool invocation level. This includes per-request authorization to specific resources, policy-driven brokerage of secrets and capabilities, provenance-rich tamper-evident telemetry, and guardrails against agent-specific threats like prompt injection and tool misuse. PAM patterns are necessary inputs, but a control plane for AI must be aligned to agents and data, not people and sessions.

Bottom line for solution providers

Agents don't just accelerate AI adoption; they change the trust problem. Security designed for human sessions and central gateways cannot govern multi-agent, tool-chaining, read-write workflows—especially under interconnect scaling pressure. Needed now is an agent-native IAM control plane that proves who did what, where, and why at the point of action, with short-lived authorization and tamper-evident evidence built in. Session-centric policy, human-oriented identities, static RBAC and perimeter monitoring consistently miss per-action, multi-tool, agent-to-agent flows. The next section explains how we deliver those primitives without introducing new single points of failure. [2]

Solution

Framing Identity and Access Management (IAM) for AI Agents

Zero Trust as a philosophy applies naturally to AI systems. Its two cornerstones, Identity and Access Management (IAM), remain the same in principle but must evolve in execution.

For humans, IAM relies on multi-factor authentication to prove identity, followed by policy engines that unlock access for a user's session. For AI, this model no longer works. Agents can't receive text codes or operate within session boundaries. Instead, identity and authorization must occur in real time, at the point of action; a per-call proof that an agent is who it claims to be and is authorized to perform the requested operation.

In Operant's architecture, identity and authentication are enforced at the transport boundary by proxies rather than inside agent data flow. Client proxies are cryptographically tied to each LLM or agent instance and mediate its connections; server proxies similarly front MCP/API services. All interactions use short-lived, mutually authenticated mTLS with X.509 certificates that can chain to an industrial certificate authority; the full mesh provisions, rotates, and can revoke these certificates over a brokerless publish/subscribe architecture. Private keys are hardware-bound (TPM/HSM), and frequent mTLS re-establishment makes agent or service spoofing significantly harder. Identity keys

reside with proxies and administrators, keeping credential control outside the data plane and invisible to the agents, preventing attacker administration escalation. Permissions are evaluated per connection against signed, programmatic trust rules authored by a designated administrator. The proxies also handle short-lived, agent- and tool-specific tokens injected at point of use by the Trust Fabric, as part of the PAM controls described next.

Privileged Access Management (PAM) offers a useful anchor to the agentic access control system: do not grant standing privilege; mint and inject only what's needed at the moment of use and prove it with audit-grade logs. That principal maps directly to agentic systems. The difference is scope and granularity enforcing trust per action, not per sessions. Every agent→tool call issues purpose-bounded credentials bound to the agent's identity, the specific tool/API, and the task, with a tight TTL (short expiration). Each step produces in-band, tamper-evident evidence.

Named Data Networking (NDN): A Protocol-Gapped Trust Fabric

NDN in brief: Over more than a decade of academic and applied work, the Named Data Networking (NDN) protocol has demonstrated a data-centric security model that is measurably resilient for distributed systems with iron-clad security that is enforced at the packet level. In NDN, every unit of data (a publication) is named, signed, and encrypted—so authenticity, confidentiality, and provenance travel with the data itself. Furthermore, all NDN packets are blocked unless explicitly allowed, true zero trust.

While NDN evolved through academic research, Operant Networks is responsible for NDN's first commercial deployments—securing control systems for critical energy infrastructure, representing ~2% of U.S. electrical generation, including on nuclear power plants. Energy facilities demand what session-based protocols cannot deliver: cryptographically signed commands, default-deny enforcement, and tamper-evident audit trails. Agentic AI now pushes enterprise security requirements into this same territory: autonomous actors making real-time decisions with write-side operations across multi-hop workflows—security demands that far exceed traditional chatbot or read-only AI deployments. Operant's NDN control plane applies this production-hardened critical infrastructure security to autonomous AI systems.

Trust Fabric mechanics: In Operant's NDN Trust Fabric, each NDN proxy holds a securely stored private key that establishes its Non-Human Identity (NHI). A central Administration Trust Portal authors Trust Rules for credential issuance and tool access; those rules are signed by a hardened Trust-Admin key and enforced at every node, for every interaction. Because authorization and provenance are bound to packets, AI agents sit outside the credential transport and cannot self-escalate. Even if an NDN proxy were compromised, its effective permissions remain bounded by the signed Trust Rules that each node independently validates.

Limited Domain deployment: Operant's NDN Trust Fabric aligns with the "limited domain" model—bounded membership, shared policy, and enforceable edges. It hardens agentic-AI workloads by isolating an enclave, enforcing identity and data policies inside it, and tightly mediating traffic at the domain boundary—without reworking the enterprise network. Unlike hyperscaler platform-native security suites that often assume uniform, organization-wide adoption from day one, Operant enables targeted rollout where agents run today, with a deliberate path to expand. Teams define an approved subset of the existing network and essential external services (e.g., MCP servers,

A2A endpoints); only vetted flows are permitted. All other paths are excluded under zero-trust enforcement, with the Trust Fabric providing ultimate authorization and change control.

Protocol-gapped by design: The overlay is logically and cryptographically separate ("protocol-gapped") from agent frameworks. It obtains short-lived OAuth/OIDC credentials (including OBO) and injects them at the point of use inside NDN Server Proxies, minimizing token exposure. It also publishes A2A "agent cards" over a brokerless pub/sub mesh, removing the single-point-of-failure common to centralized brokers. All agent↔agent and agent↔MCP-server connections are short-lived, mutually authenticated mTLS sessions with X.509 certificates managed locally and distributed over NDN, binding client/server proxies to their keys and preventing spoofing.

Protocol Proof Points: Operant's technical work with the U. S. Department of Defense (DoD), U. S. Department of Energy (DOE), Sandia National Lab, Idaho National Lab (INL), the National Renewable Energy Laboratory (NREL) and major energy providers have enabled multiple successful evaluations of the NDN transport for latency, reliability, and cybersecurity. NREL has completed a cybersecurity evaluation of nine key attack vectors in the NDN system summarized in a technical report and passing with no deficiencies. A large network scaling emulation project has now launched at NREL using their world-class 44-petaflop Aries Kestrel supercomputer.

Operant's AI Trust Fabric via NDN: Detailed Solutions to the Problems

P1. From sessions to actions (risk is per operation, not per login)

Production stacks (e.g., OpenAI's AgentKit, Microsoft Agent Framework) treat agents as long-lived, tool-calling services that may perform dozens of write-side operations in one run. Login-time approval and perimeter controls "see" authentication but miss per-call intent, inputs, targets, and prior steps. Without pre-execution checks at tool boundaries—and a way to pause/kill mid-run—harmful actions can complete before monitors react.

Result: Approval at login, exposure at runtime—material gaps wherever decisions aren't made at each tool/API call.

S1. Per-action, edge-side authentication and authorization gating

We evaluate every tool/API invocation at the edge using an NDN server proxy co-located with the tool. For each call, the NDN overlay carries signed Trust Rules and delivers necessary credentials. The edge proxy performs multiple validations: it checks the call's intent, inputs, and target against Trust Rules; verifies the mTLS authentications; and obtains or refreshes the per-call token via NDN. Based on these checks, it either forwards or blocks the request before execution. Each decision and outcome are published as a signed NDN event so the call is both controlled and evidenced.

Result: Login approval no longer equals runtime exposure—every call is enabled with just-in-time creds or blocked at the edge, with signed NDN logging of who/what/why/when.

P2. Smarter agents, higher risk of unintended/unauthorized actions

As agents grow more capable, edge cases emerge within existing transport and IAM constraints. These include policy slip (actions outside intent), credential replay within short-lived windows

on an agent's behalf, and privilege creep across chained steps. Even carefully scoped starts can morph into unauthorized access to regulated data or high-impact systems. Verification must happen at the point of action. Agents are autonomous and persistent (scheduled runs, durable memory), so incidents can unfold without a person in the loop.

Result: Higher probability and blast radius of unintended changes until checks shift to the point of action.

S2. Per-agent Identity with autonomy-aware safeguards

We assign every agent its own unique identity through a securely stored private key. This identity binds to NDN-managed keys and certificates. All agent→tool paths use mutually authenticated mTLS. The NDN Trust Fabric keeps the trust plane protocol-gapped from the agent runtime through NDN cryptography with a separate root of trust. Agents cannot access this root of trust. The fabric delivers short-lived, point-of-use OAuth (OBO) tokens to the edge proxy only for specific tools and tasks. Trust Rules signed by a hardened administration private key define each agent's allowed tools, data scopes, and time bounds; the edge proxy enforces these on every call and can revoke access mid-run by refusing further token issuance or mTLS renewal if behavior or context drifts. Scheduled/durable agents are explicitly handled (timers, memory), so unattended runs can't continue once authorization changes.

Result: Higher agentic autonomy does not translate into higher unauthorized-action risk—identity, scope, and revocation are enforced per call, with short-lived creds and mTLS preventing replay or privilege creep.

P3. Interconnect standards create a multi-hop execution fabric—and N^2 complexity

MCP and A2A enable branching, low-latency workflows where agents invoke tool servers, exchange state, and spawn sub-agents across domains. Central gateways and human approvals cannot keep pace with agent-speed execution. Naïve point-to-point communication and security session topologies create $O(N^2)$ complexity for N agents as they scale. Each pairwise connection requires separate policy rules, keys, and monitoring—driving configuration complexity, latency, blind spots, and misconfiguration risk. Multiple vendor hubs can reduce interconnections, but each is a single point of failure (SPOF), a honeypot, and a source of added latency.

Result: Quadratic control/monitoring load with increasing chances of missed checks—productivity and blast radius scale together as hops increase.

S3. Brokerless pub/sub to tame multi-hop workflows and N^2 complexity

We replace brittle point-to-point links with an NDN brokerless publish/subscribe fabric. Agents and tools address named topics rather than specific peers. Agent capabilities, endpoints, and provenance are published as signed "agent cards." Subscriptions are authorized by signed Trust Rules. Edge proxies enforce policy at every hop and use mTLS to authenticate all publishers and subscribers. There is no central broker, policy and credentials are pushed to edge proxies so each agent maintains a local directory of authorized peers and agent cards, avoiding the known limits

of centralized brokers. Each hop's signed credentials, tokens, certificates, and Trust Rule decisions are logged as end-to-end, tamper-evident NDN events.

Result: Pairwise interconnects, policy bindings, and monitors drop from $O(N^2)$ to roughly $O(N)$, while preserving per-hop visibility and enforcement without introducing a new bottleneck.

P4. Centralized trust can't scale to distributed agent flows (multi-cloud & edge)

Hub-and-spoke enforcement struggles across multi-cloud and edge deployments. It introduces single points of failure and relies on opaque code paths. Control-plane lag creates windows where agents run without enforcement. Meanwhile, strict central gating adds latency that breaks real-time workflows. In federated ecosystems, central brokers also become chokepoints and attractive targets. Perimeter-centric enforcement doesn't see agent-to-agent intent, so chains often complete before monitors fire.

Result: Brittle, SPOF-prone control that neither scales nor sustains assurance at the edges where actions execute.

S4. Decentralized, verifiable enforcement for multi-cloud and edge

We author a single set of Trust Rules centrally (signed by the admin key) and distribute them over NDN to every edge proxy, where they're validated and enforced locally—no central broker or gateway required. Each proxy makes allow/deny decisions where actions execute; agent↔agent intent is evaluated at the edges, not inferred at the perimeter. If the trust fabric is slow or unavailable, proxies continue with the last verified Trust Rules and refuse stale or unsigned updates. All paths are mTLS with NDN-managed certificates; every decision, credential issuance, and rule update is published as a signed NDN event for independent verification.

Result: Low-latency enforcement that spans clouds, regions, and edge sites—no single points of failure, no opaque choke points, no dependence on multiple cloud-specific security stacks—and verifiable evidence generated where actions occur.

P5. Write-side harm happens at machine speed (tokens, scopes, mid-run control)

Agents don't just read, they write and move records, files, money, and secrets. Many stacks use bearer tokens with replay-vulnerable lifetimes and coarse scopes that can't be tailored per step. SIEM and EDR tools alert only after execution completes. Static RBAC can't keep pace with evolving agent capabilities, leading to over-permissioning or brittle failures. Without pre-execution gating and mid-run revocation, the blast radius grows with every chained call.

Result: Token blast radius + delayed revocation = real impact—and loss of control mid-chain.

S5. Short-lived, point-of-use credentials with per-call permissions

We issue ephemeral tokens at the moment of use; preferably via tool-scoped OAuth (OBO). These tokens are delivered to the NDN server proxy via the NDN overlay. Each token is bound to four elements: the agent's identity, the specific tool/API, the task context, and a tight TTL. All paths use mTLS with NDN-managed X.509 certificates. The edge proxy enforces per-call permissions rather than static RBAC. Egress controls ride with content names and are applied at the tool edge

(e.g., allowlists/denylists for destinations, redaction rules). If behavior or context drifts, the proxy can terminate the mTLS session and refuse further token issuance, halting the chain mid-run. Issuance and use events are published as signed NDN records for replay detection and audit.

Result: Token blast radius collapses—replay and over-permissioning are neutralized by per-call, just-in-time creds, enforced egress, and mid-run revocation at the edge.

P6. Weak provenance and auditability (no enforced signed evidence)

Boards and regulators now expect tamper-evident lineage: intent → authorization → action → effect, bound cryptographically to the principals and policies in force. Today's logs and screenshots are fragmented across services and lack signed provenance. They rarely correlate prompts and reasoning with the exact credentials and tool calls used. This slows root-cause analysis and weakens regulatory defense.

Result: Disputable evidence and slow Root Cause Analysis, undermining compliance and incident response.

S6. Signed, in-band audit trails with correlated lineage

We record intent → authorization → action → effect as NDN publications from the edge proxy and tool servers during normal execution. Each trust-rule version, mTLS identity, credential issuance/use, decision, and tool/API result is a signed, timestamped, content-addressed NDN event; events are hash-linked for tamper-evidence. Auditors and incident responders can subscribe to evidence topics without touching runtime proxies; evidence bundles are exportable and independently verifiable against the signing keys establishing provenance. Log storage can be encrypted and access is governed by Trust Rules.
Result: Provable lineage and fast, defensible Root Cause Analysis —cryptographic evidence replaces screenshots, and any verifier can check it with the trust anchor.

Summary

Operant's NDN-backed, protocol-gapped Trust Fabric delivers what the market now requires: per-call policy at the edge, cryptographic Identity authentication, short-lived, tool-scoped credentials, decentralized enforcement, and signed, in-band evidence. It's vendor-agnostic, brokerless, and fault-tolerant—a control layer that matches how multi-agent MCP/A2A systems actually run, not how legacy perimeter solutions hoped they would. [3]

Glossary

A2A (Agent-to-Agent Protocol) – A communication standard that enables AI agents to discover, coordinate with, and delegate tasks to other agents across organizational boundaries.

Agentic AI – Autonomous AI systems that can take actions, use tools, call APIs, and make decisions without continuous human oversight. Unlike chatbots, agents can write data, execute commands, and coordinate multi-step workflows.

Bearer Token – A credential that grants access to resources based solely on possession, without requiring additional proof of identity. Vulnerable to replay attacks if intercepted.

CASB (Cloud Access Security Broker) – Security enforcement point positioned between cloud service users and cloud applications, typically focused on visibility, compliance, and threat protection.

Content-Addressed – A naming scheme where data is identified by its cryptographic hash rather than its location, enabling verifiable retrieval and tamper detection inherent to NDN.

Credential Replay – An attack where a valid authentication token or credential is captured and reused by an unauthorized party within its validity window.

Edge Proxy – A security component deployed at the point of action (co-located with tools/APIs) that enforces authorization decisions, manages credentials, and logs activity locally.

IAM (Identity and Access Management) – Systems and policies that manage digital identities and control access to resources, traditionally designed for human users.

JIT/JEA (Just-in-Time / Just-Enough-Access) – Security principle of granting elevated privileges only when needed, for the minimum scope required, and for the shortest duration possible.

Limited Domain – An IETF-defined network model with bounded membership, shared policy, and enforceable edges—enabling strict security controls within a defined perimeter while coexisting with broader enterprise networks.

MCP (Model Context Protocol) – An open standard developed by Anthropic that enables AI systems to securely connect to external data sources and tools through a standardized interface.

mTLS (Mutual TLS) – A form of Transport Layer Security where both client and server authenticate each other using certificates, ensuring bidirectional trust.

NDN (Named Data Networking) – A network architecture where data objects (publications) are named, signed, and routable by content rather than by location, enabling inherent security and verifiability.

NHI (Non-Human Identity) – A cryptographically-backed identity assigned to autonomous systems, services, or agents rather than human users. Critical for agent-to-agent authentication.

OAuth/OIDC (Open Authorization / OpenID Connect) – Industry-standard protocols for authorization and authentication, enabling secure delegated access to resources without sharing passwords.

OBO (On-Behalf-Of) – An OAuth extension that allows a service to request access tokens on behalf of a user or another service, maintaining the original security context through delegation chains.

OPA/Rego (Open Policy Agent / Rego Policy Language) – A declarative policy engine and language for expressing fine-grained access control rules that can be evaluated at enforcement points.

PAM (Privileged Access Management) – Security framework for controlling, monitoring, and auditing access to high-risk systems and credentials, traditionally focused on human administrators.

Perimeter Security – Security model that focuses on protecting the network boundary, assuming trust within the perimeter. Ineffective for distributed, multi-cloud, and agent-based architectures.

Policy Slip – When an agent's actions drift outside the intended scope or purpose defined in its authorization policy, often due to chained operations or context changes.

Privilege Creep – Gradual accumulation of access rights beyond what's needed, often occurring as agents chain operations or traverse multiple services.

Protocol-Gapped -- An architectural separation where the trust/control plane operates on a different protocol layer than the application plane, preventing agents from manipulating their own authorization mechanisms.

Pub/Sub (Publish/Subscribe) – A messaging pattern where publishers send messages to named topics without knowing subscribers, and subscribers receive messages from topics.

RBAC (Role-Based Access Control) – Access control model that assigns permissions based on predefined roles. Static RBAC struggles with dynamic, context-dependent agent operations.

SIEM/EDR (Security Information and Event Management / Endpoint Detection and Response) – Security tools that collect, analyze, and respond to security events, typically operating on logged data after events occur.

SPIFFE/SVID (Secure Production Identity Framework For Everyone / SPIFFE Verifiable Identity Document) – Standard for workload identity that provides short-lived X.509 certificates or JWT tokens for service-to-service authentication.

SPOF (Single Point of Failure) – A component whose failure would cause an entire system to fail. Central brokers and gateways often introduce SPOFs in distributed architectures.

Trust Rules – Cryptographically-signed policy documents that define which identities can access which resources under what conditions, distributed and enforced at edge proxies.

Trust Fabric -- A logically separate network layer that handles authentication, authorization, and audit functions independently from the underlying application protocols, enabling consistent security enforcement across heterogeneous systems.

TTL (Time To Live) – The lifespan of a credential, token, or cache entry. Short TTLs reduce the window for credential replay but require more frequent renewal.

Zero Trust – Security model that assumes no implicit trust based on network location, requiring continuous verification of all users, devices, and connections regardless of location.

Appendix: End Notes

[1] Introduction — adoption, risk signals, and standards baseline

- Inovia Capital (2025), *Cybersecurity and Agentic AI—A Race to Contain this Emerging Exposure* — agents vs. chat apps; early incidents; buyer demand for behavioral and context-aware protections (public blog).
- MIT Media Lab (2025), Project NANDA. *The GenAI Divide: State of AI in Business 2025*; 95 % of enterprise GenAI efforts show no measurable P&L impact.
- Cisco (2024), *Data Privacy Benchmark Study* — 27% of organizations temporarily banned GenAI tools over privacy/security risks.
- McKinsey (2024), *The State of AI in early 2024* — 65% using GenAI; 44% experienced at least one negative consequence (including cybersecurity events).
- Sequoia Capital (2025), *AI 50 — AI agents move beyond chat* — VC validation of agentic systems as the AI market's next adoption curve.
- CB Insights (2025), *The AI Agent Market Map* — analyst landscape mapping 170+ agentic startups across 26 categories.

[2] Problems — incidents, complexity, regulatory timelines

- *Windows + MCP* (2025) — controlled rollout with registry and consent signals maturing but incomplete plumbing.
- *Google A2A protocol + codelab* (2025) — practical guidance for multi-agent execution across cloud services; patterns are converging but still evolving.
- Tenable (2025), *FAQ: Cursor CurXecute and MCPoison vulnerabilities* — concrete MCP trust-path / RCE exploit example cited in P2–P3.
- Check Point Research (2025), *MCPoison: malicious tool injection via MCP* — demonstrates "agent-tool chaining exposure".
- NVD (2025), *CVE-2025-54135: Cursor IDE RCE via MCP* — authoritative record for exploit risk.
- Business Insider (2025), *AI agent in Replit deletes live production database* — real-world write-side harm cited in P4.
- Tom's Hardware (2025), *AI coding assistant goes rogue during freeze and wipes prod data* — timeline detail for the same Replit incident.
- EU AI Office (2025), *Implementation timeline for the EU AI Act* — cited for formal timeline pressure shaping enterprise controls.

[3] Solutions — PAM/JIT foundations, NDN, and agent standards

- ACM SIGCOMM CCR (2014), *Named Data Networking (NDN)* — data-centric routing and trust model underpinning the overlay; supports S3–S5.

- Anthropic (2024), *Introducing the Model Context Protocol (MCP)* — initial spec and vision; ecosystem adoption through 2025.
- OpenAI (2025), *ChatGPT Developer Mode MCP client* — full read/write MCP support; practical client-side adoption milestone.
- OpenAI (2025), *Introducing AgentKit* — toolkit for building, deploying, and operating agents.
- Microsoft (2025), *Agent Framework — overview (public preview docs)* — unified, open-source multi-agent orchestration.
- NIST NCCoE (2024), *SP 1800-18: Privileged Account Management (PAM) Practice Guide* — foundations for vaulting, brokering/injection, session governance.
- CyberArk (2024), *What is Privileged Access Management (PAM)?* — concise definition including human and non-human privileged activities.
- BeyondTrust (2024), *Just-in-Time Access: What It Is & Why You Need It* — eliminates standing privilege; supports S2.
- Open Policy Agent (OPA) (docs), *Policy Language (Rego)* — declarative edge policy engine basis.
- SPIFFE (docs), *SPIFFE Concepts: SVID (X.509/JWT) and trust domains* — workload identity; short-lived X.509/JWT.
- W3C (2025), *Verifiable Credentials Data Model v2.0* — portable, cryptographically verifiable claims for non-human identity/attestation; supports S1–S3.
- IETF (RFC 9449), *OAuth 2.0 Demonstrating Proof-of-Possession (DPoP)* — sender-constraining tokens to keys for per-call authorization; supports S2.
- W3C CCG (2025), *Authorization Capabilities for Linked Data (ZCAP-LD)* — object-capability style delegation with scoped caveats; supports S2 capability-based auth.
- Sigstore (docs), *Rekor transparency log overview* — append-only, auditable Merkle log for signed events/evidence; supports S5–S6.