# Hanzo AI for Compliance: An OSS Cloud Built to Pass the Audit

Zach Kelling — Hanzo Team

Version 1.0 2026

**Abstract**

Compliance is, in operational terms, an evidence-production problem. An auditor for SOC 2 or ISO 27001, an examiner for FINRA/SEC, a HIPAA assessor, or—for a government system—an assessor for an Authority to Operate (ATO) does not primarily ask whether a system *could* be secure; they ask the operator to *produce the artifacts* that demonstrate a control is designed and operating—access reviews, key-rotation records, immutable logs, change tickets, configuration baselines. Most stacks bolt auditing on after the fact, so producing those artifacts becomes months of manual screenshot-gathering, log-stitching, and spreadsheet reconciliation repeated at every cycle. This paper argues that Hanzo's sovereign open-source stack inverts that cost structure by emitting the evidence auditors ask for *by construction*: a centralized identity and access layer (IAM) produces access-control evidence natively; a key management service (KMS) eliminates plaintext secrets and produces key-custody evidence; a cryptographically signed, on-chain command/audit log produces tamper-evident, non-repudiable audit records; reproducible builds and policy-as-code produce change-management evidence; and 50 machine-checked Lean 4 proofs plus NIST Known-Answer-Test validation of FIPS 203/204/205 produce machine-checked cryptographic assurance rather than "we tested it." We give the control-framework mapping from these native capabilities to the SOC 2 Trust Services Criteria, to FINRA/SEC recordkeeping, and to the NIST SP 800-53 control families (AC, AU, IA, SC, CM) that underpin RMF and the ATO process. We then report an anonymized case study in which this same stack carried a regulated securities platform through FINRA/SEC approval and SOC 2 with a small team—while a 200+-microservice to 3-binary consolidation shrank the very surface that had to be audited. Throughout, claims are kept careful: the stack *provides evidence for* and *maps to* controls; certification is an assessor's judgment, not a vendor's.

**Executive Summary.** Passing an audit is mostly about producing proof. The painful part of SOC 2, FINRA/SEC, or a defense ATO is not deciding to be secure—it is assembling the mountain of evidence that shows, control by control, that the system is secure and stays that way: who can access what, where the secrets live and how often they rotate, an unalterable record of every privileged action, proof that the code in production is the code that was reviewed. When auditing is an afterthought, that evidence has to be reconstructed by hand every cycle, which is why compliance is expensive and recurring. Hanzo's stack is built so the evidence is a byproduct of running the system. Identity is centralized, so access reviews and least-privilege are queries, not archaeology. Secrets live in a key vault and are never in plaintext, so key-custody evidence exists by default. Every privileged action is cryptographically signed and written to an immutable, tamper-evident log, so the audit trail cannot be quietly edited—and the same property satisfies financial-recordkeeping rules that demand write-once records and supports non-repudiation. Builds are reproducible and policy is code, so change-control evidence is generated automatically. And the cryptography is not merely tested but *proved*—machine-checked theorems re-verified on every commit. The net effect is that SOC 2, ISO 27001, FINRA/SEC, HIPAA, and—for government systems—the RMF/ATO process all become tractable: less code to assess, a smaller audit boundary, and an evidence pipeline that is collected continuously rather than re-earned from scratch each year. For any regulated enterprise or government program—defense included—modernizing onto this stack means inheriting the audit pipeline, not building it.

## Contents

# 1 Compliance Is an Evidence-Production Problem

A control framework—SOC 2, FINRA/SEC recordkeeping, NIST SP 800-53 under the Risk Management Framework (RMF)—is a catalog of assertions a system is required to satisfy. An audit or assessment is the exercise of *substantiating those assertions with artifacts*. The decisive distinction in practice is not between secure and insecure systems; it is between systems that can *produce the evidence* of their controls on demand and systems that cannot. An assessor's day is spent asking "show me": show me the list of who has administrative access, show me that it was reviewed last quarter, show me that this secret was rotated, show me an unalterable record that this privileged action happened and who performed it, show me that the binary running in production is the artifact your pipeline built from reviewed source.

When a stack is assembled from many independently-operated services with logging and identity bolted on afterward, answering "show me" is manual archaeology. Access lists are scattered across dozens of consoles; secrets are pasted into environment variables and config files; logs are mutable application logs spread across services with no integrity guarantee; the relationship between source, review, and the deployed artifact is reconstructed from memory. The result is the familiar compliance cost: a SOC 2 Type II observation window plus weeks of evidence collection; a FINRA/SEC examination that demands recordkeeping the architecture was never designed to produce; an ATO package whose body of evidence (the security controls assessment) is assembled by hand. And because nothing about the system *emits* this evidence, the cost recurs in full at every audit cycle and every re-authorization.

The thesis of this paper is structural: if the stack is built so that the evidence auditors ask for is a *native output* of normal operation, the cost collapses. The audit stops being a discovery project and becomes a query against artifacts the system already produces. Hanzo's stack is built this way deliberately, and the rest of this paper walks the specific capabilities, maps them to the frameworks, and shows—with an anonymized production case—what it does to the audit.

# 2 Evidence by Construction

The stack is consolidated around a small number of owned, open-source components. Each one, as a side effect of doing its job, emits a class of evidence that an auditor asks for. We take them in turn and name the control objective each supports.

## 2.1 Identity and Access (IAM): access-control evidence

All authentication and authorization route through one identity layer (Hanzo IAM) rather than per-service credentials. Authentication is centralized (OIDC/OAuth 2.1 with PKCE; one canonical set of endpoints); authorization is role- and scope-based; and because identity is a single source of truth rather than a federation of ad-hoc accounts, least-privilege is enforceable and *auditable*. The evidence an assessor wants for access control—the complete roster of principals and their entitlements, the record of access grants and revocations, the periodic access review—are queries against one system rather than a reconciliation across many.

This supports the SOC 2 Security (Common Criteria) logical-access criteria and maps directly to the NIST SP 800-53 **AC** (Access Control) and **IA** (Identification and Authentication) families: centralized account management, enforced least privilege, separation of duties expressed as roles, and unique attributable identities for every actor. Because there is one identity plane, an access review is *produced*, not assembled.

## 2.2 Secrets and Keys (KMS): confidentiality and key-management evidence

Secrets never live in plaintext. A key management service (Hanzo KMS) holds credentials and cryptographic keys under centralized custody with rotation, and services obtain secrets at runtime rather than embedding them. There are no long-lived secrets in source, in container images, or in configuration. The evidence auditors request for confidentiality and key management—where keys live, who can reach them, how custody is separated, when each was rotated—is the KMS's normal bookkeeping.

This supports the SOC 2 Confidentiality criteria and maps to the SP 800-53 **SC** (System and Communications Protection) family, specifically the key-establishment and key-management controls (SC-12/SC-13) and protection of secrets at rest. The structural property—*no plaintext secret anywhere in the pipeline*—is itself the control, and the KMS's rotation log is the evidence that it operates. A verification gate enforces the property mechanically: the build fails if a plaintext-secret pattern reappears, so "no secrets in code" is checked rather than asserted.

## 2.3 Immutable audit trail: audit-logging, integrity, and non-repudiation

This is the keystone. Every privileged action is cryptographically signed by the identity that performed it and written to an *append-only, on-chain, tamper-evident* command/audit log. Two properties follow that ordinary application logging cannot provide:

- **Tamper-evidence.** The log is hash-chained and consensus-anchored, so any retroactive edit, deletion, or reordering is detectable. An auditor does not have to trust that the log was not altered; the log's own structure demonstrates it.
- **Non-repudiation.** Because each entry is signed by the acting identity (post-quantum signatures; §2.4), the actor cannot later deny the action, and a reviewer cannot forge one. The signature binds *who* to *what* and *when*.

This maps to the SP 800-53 **AU** (Audit and Accountability) family in full— audit event generation (AU-2), content of records including actor and outcome (AU-3), protection of audit information from modification (AU-9), and non-repudiation (AU-10)— and to the SOC 2 Security monitoring criteria and Processing Integrity. Crucially, the same immutability is exactly what financial recordkeeping rules require: the write-once, non-rewriteable, non-erasable ("WORM-like") retention demanded for broker-dealer books and records is an emergent property of an append-only, hash-chained ledger, not a bolt-on archival appliance. One mechanism satisfies the defense integrity control *and* the securities recordkeeping rule.

## 2.4 Post-quantum identity and transport: forward-looking confidentiality

Identities, signatures on the audit log, and transport between services are post-quantum (ML-KEM for key establishment, ML-DSA for signatures; FIPS 203/204). The compliance-relevant point is durability of evidence: an audit record, a signed action, or an encrypted channel captured today must remain confidential and verifiable for the full retention horizon—which, for financial records and for classified material, is measured in years to decades. "Harvest now, decrypt later" makes classical-only confidentiality a liability for any record with a long required life. Post-quantum identity and transport make the audit trail's signatures and the data's confidentiality resistant to a future quantum adversary, which is the forward-looking posture that CNSA 2.0 mandates for national-security systems and that a prudent long-retention financial archive should adopt. This

supports the SC family's transmission-confidentiality and cryptographic-protection controls with algorithms chosen for the record's lifetime, not just today's threat.

### 2.5 Formal verification: machine-checked evidence, not "we tested it"

Most software is trusted because it was tested and nothing broke. The cryptographic core here goes further: its security properties are proved as theorems and re-checked by a machine on every code change. The suite comprises **50 machine-checked Lean 4 proofs**, including **ML-DSA EU-CMA** unforgeability (FIPS 204) and **ML-KEM IND-CCA2** security (FIPS 203), alongside consensus safety/liveness, threshold-signature, and protocol-correctness proofs; complementary **NIST Known-Answer-Test (KAT)** validation covers all three post-quantum standards (FIPS 203/204/205). For an assessor this is a different *kind* of evidence: not a test report asserting that some inputs produced expected outputs, but a machine-checked proof with an explicit, inspectable set of assumptions. A proof regression blocks the build, so the assurance is continuous rather than a point-in-time snapshot. This is the strongest available evidence for the SC family's cryptographic controls and for the "correctly implemented" burden in any cryptographic accreditation, and it is auditable end-to-end precisely because the stack is owned and open—there is no third-party black box whose internals are out of the prover's reach.

### 2.6 Reproducible builds, policy-as-code, and change control

The path from source to running artifact is itself evidence. Builds are reproducible and produced by a shared CI/CD pipeline; images are pinned to immutable, content-addressed tags (a semver tag or a commit-pinned `sha-` digest), never to a floating `:latest`; and deployment policy and infrastructure are expressed as code under version control. The consequence is that the relationship auditors probe for change management—reviewed source → built artifact → deployed instance—is a chain of immutable, attributable records rather than an after-the-fact reconstruction. Who changed what, who reviewed it, and exactly which artifact reached production are all recorded by the normal pipeline.

This maps to the SP 800-53 **CM** (Configuration Management) family—baseline configuration (CM-2), change control (CM-3), and configuration of the deployed system—and to the SOC 2 change-management criteria. Because policy is code, the configuration baseline is the repository state, and any drift is a diff.

## 3 Control-Framework Mapping

Table 1 maps each native capability to the SOC 2 Trust Services Criteria it provides evidence for, the NIST SP 800-53 control families it supports under RMF/ATO, and the corresponding FINRA/SEC recordkeeping relevance. The verbs are deliberate: a capability *provides evidence for* or *maps to* a control. Whether a control is satisfied is an assessor's determination on the specific deployment; the stack's contribution is to make the evidence native and the boundary small.

| **Native capability** | **SOC 2 TSC** | **800-53 family** | **FINRA/SEC & ATO relevance** |
|---|---|---|---|
| Centralized IAM (OIDC; RBAC; least privilege; access reviews) | Security (CC6 logical access) | AC, IA | Supervisory access controls; attributable identities for the ATO access-control assessment |
| KMS (no plaintext secrets; central custody; rotation) | Confidentiality | SC (SC-12/13) | Protection of customer/non-public data; key-management evidence in the SCA |
| Immutable, signed, on-chain audit log | Security (monitoring); Processing Integrity | AU (AU-2/3/9/10) | WORM-like books-and-records retention; non-repudiation of privileged actions |
| Post-quantum identity & transport (ML-KEM / ML-DSA) | Confidentiality | SC (SC-8/12/13) | Long-retention record durability; CNSA 2.0 forward posture |
| Formal verification (50 Lean 4 proofs; NIST KAT) | Security; Processing Integrity | SC, SA | Machine-checked "correctly implemented" evidence for the SCA |
| Reproducible builds; policy-as-code; pinned artifacts | Security; Availability (change mgmt) | CM (CM-2/3) | Change-control records; configuration baseline as code |
| Consolidation (fewer services / smaller boundary) | All five (reduced scope) | All families (smaller boundary) | Smaller examination surface; smaller authorization boundary |

Table 1: Native stack capabilities mapped to SOC 2 Trust Services Criteria, NIST SP 800-53 control families (RMF/ATO), and FINRA/SEC recordkeeping relevance. "Maps to / provides evidence for"—not a certification claim.

**Five criteria, one stack.** The SOC 2 Trust Services Criteria are Security, Availability, Confidentiality, Processing Integrity, and Privacy. The capabilities above touch all five: *Security* via IAM, the audit trail, and reproducible builds; *Confidentiality* via KMS and post-quantum transport; *Processing Integrity* via the tamper-evident log and machine-checked proofs of the logic that moves records; *Availability* via policy-as-code change management and a consolidated, self-hostable footprint; and *Privacy* via centralized access control and confidentiality of non-public data. Because one set of mechanisms underlies all five criteria, the evidence collected once is reused across the report rather than re-gathered per criterion.

**The same evidence translates.** The reason one stack serves SOC 2, ISO 27001, FINRA/SEC, and HIPAA—and, for government systems, an ATO— simultaneously is that the underlying control objectives overlap heavily—access control, confidentiality, audit integrity, change control, cryptographic correctness—and differ mainly in the framework's vocabulary and the assessor's authority. An append-only signed ledger is "AU-9 protection of audit information" to an RMF

assessor and "non-rewriteable record retention" to a securities examiner; it is the same ledger. Centralized least-privilege identity is "AC-2/AC-6" to one and "supervisory and access controls" to another; it is the same IAM. Producing the evidence once and presenting it through each framework's lens is the efficiency the consolidated stack creates.

# 4 Case Study: One Stack Through FINRA/SEC and SOC 2

The argument is not theoretical. The same stack carried a regulated securities platform—an SEC-registered alternative trading system (ATS) with broker-dealer and transfer-agent functions—through FINRA/SEC approval and SOC 2, with a small team, in record time. Two facts from that engagement matter for the thesis of this paper.

**The evidence was native.** The regulated entity ran on this stack's centralized IAM, its KMS (no plaintext secrets), and its immutable signed audit log. The recordkeeping obligations that fall on a broker-dealer and transfer agent— non-rewriteable books and records, an attributable trail of who did what to which record, controlled and reviewed access—were satisfied by capabilities the platform already had, not by a separate compliance system layered on top. The audit and the regulatory approval drew on artifacts the running system produced as a matter of course.

**Consolidation shrank what had to be audited.** In parallel, the platform's architecture was consolidated from 200+ **microservices into 3 compiled Go binaries** (plus a small set of OSS platform services), collapsing roughly **3.07 million lines of code to about 1.06 million**—*∼3.5 million lines of dead code deleted*, around a 65–70% reduction. The consolidated rebuild also remediated the security-finding backlog of the legacy estate (on the order of 90+ findings) at the architectural level. This is the second, quieter half of why the audit was tractable: *every line of code and every service is attack surface that an auditor or accreditor must reason about.* Cutting the codebase by two-thirds and the runtime from a sprawl of services to three binaries shrinks the examination surface, the SOC 2 system description, and—for a government system—the authorization boundary. Less code plus centralized IAM/KMS/audit is a smaller, cheaper, faster audit. The consolidation and the evidence-by-construction property are the same lever viewed from two sides: one reduces *what* must be proven, the other makes the proof *native*.

*Anonymization.* The regulated entity is intentionally unnamed; the point is the stack's properties, not the client. What transfers to any future program is the mechanism: a small team reached a regulated, audited production posture quickly because the stack produced the evidence and the consolidation shrank the surface.

# 5 Continuous Compliance

A SOC 2 Type II report covers an observation window; an ATO is not a one-time event but a posture that must be *maintained* under continuous monitoring; a broker-dealer's recordkeeping is examined repeatedly. The expensive failure mode is treating each cycle as a fresh discovery project. Because this stack emits evidence continuously, the posture is maintained rather than re-earned:

- **Always-on collection.** The audit log is generated by operation, not by an end-of-period scramble; access reviews are queries against live IAM state; key-rotation records accrue from the KMS; build and deployment records accrue from the pipeline. The body of evidence is a continuous stream.

- **Continuous verification.** The Lean proofs and KAT vectors run on every commit, so cryptographic assurance is re-established automatically and a regression blocks deployment. Policy-as-code means configuration drift surfaces as a diff, not as an audit finding months later.
- **Agentic evidence assembly.** The same agentic engineering pipeline that built the consolidated system can collect, organize, and map evidence to controls continuously—turning the ongoing-authorization (cATO) ideal into routine output instead of a periodic mobilization.

The effect on the ATO process specifically is that re-authorization draws on an evidence pipeline that never stopped running. The authorization boundary is smaller (consolidation), the controls produce their own evidence (by construction), and the evidence is current (continuous), which is precisely the posture continuous authorization is meant to reward.

## 6 Why It Matters Across Regulated Industries and Government

Any organization modernizing onto this stack—a healthcare provider, a financial firm, a critical-infrastructure operator, or a government contractor or agency (defense included)—inherits the evidence pipeline rather than building one. The same properties that make a commercial SOC 2, ISO 27001, FINRA/SEC, or HIPAA audit tractable are the properties a government RMF assessment and an ATO package require: centralized least-privilege identity (AC/IA), no plaintext secrets and managed keys (SC), a tamper-evident non-repudiable audit trail (AU), reproducible builds and configuration-as-code (CM), and machine-checked cryptographic correctness (SC/SA). Because the stack is sovereign, open-source, and self-hostable, it can be deployed in the environments these regimes demand—on-premises, air-gapped, at the edge—without renting from a closed vendor, and an assessor can inspect the entire path the evidence takes rather than trusting a black box. The cryptography is post-quantum, so the records and identities survive the retention horizon that financial, healthcare, and government archives all require.

This compliance posture is one facet of a broader sovereign stack built by Hanzo AI—an American-owned applied-AI lab and venture studio (Techstars '17, founded 2014). Hanzo owns both halves of the modern trust problem: the frontier AI models (the *Zen* family) and the production post-quantum cryptography that secures and signs the systems they run in. This matters for who a regulated or accredited operator can build on. U.S. frontier AI has consolidated into a closed commercial duopoly, while the broadly *open* frontier model weights available today are largely of foreign origin—an untenable basis for a system that must be inspected, accredited, and run on sovereign infrastructure. Hanzo is the American open-source alternative that owns the model and the cryptography together, deployable without renting from the closed duopoly or depending on foreign open weights. The evidence-by-construction property documented here is the auditable backing for that claim: not a vendor assurance an accreditor must take on faith, but artifacts—signed logs, key-custody records, reproducible builds, machine-checked proofs—an accreditor can read.

## 7 Honest Scope

This is a capability paper, and the credible version is explicit about what the stack does versus what an assessor does. No overstatement:

- **The stack provides evidence; it does not self-certify.** SOC 2 attestation is issued by a CPA firm, a FINRA/SEC approval is a regulator's decision, and an ATO is granted by an Authorizing Official. The stack's contribution is to make the evidence native, current,

and inspectable, and to shrink the boundary being judged—which is what makes those determinations faster and cheaper, not automatic.

- **Mapping is not equivalence.** A capability *provides evidence for* or *supports* a control family; closing a specific control still requires the operator to configure, document, and operate it for the assessed deployment. The tables here are a starting correspondence, not a completed controls matrix.
- **Formal verification has a stated boundary.** The 50 Lean proofs rest on explicit cryptographic hardness axioms (standard practice), and model checking and protocol analysis are on a published roadmap rather than complete. The assurance boundary is the true one, not a marketing one.

The strategic point holds regardless of where any one line sits: because Hanzo owns and open-sources every layer the evidence runs through, the artifacts an auditor or accreditor needs are produced by the system itself and can be inspected end-to-end— which is the difference between an audit that is a discovery project and one that is a query.

## 8 Conclusion

Compliance is an evidence-production problem, and most stacks make it expensive by producing no evidence until an auditor asks. Hanzo's sovereign open-source stack inverts that: centralized IAM emits access-control evidence, KMS emits key-custody evidence, an immutable signed on-chain log emits tamper-evident and non-repudiable audit evidence, reproducible builds and policy-as-code emit change-control evidence, and 50 machine-checked proofs plus NIST KAT validation emit cryptographic assurance that is proved rather than asserted. The same evidence maps across SOC 2's five Trust Services Criteria, ISO 27001, FINRA/SEC recordkeeping, HIPAA, and the NIST 800-53 families that underpin RMF and the ATO process—so one stack serves the commercial, financial, healthcare, and government frameworks (defense included) at once. A production case shows it carrying a regulated securities platform through FINRA/SEC and SOC 2 with a small team, while a 200+-microservice to 3-binary consolidation shrank the surface that had to be audited (and, for a government system, accredited). For any regulated enterprise or government program—defense included—the consequence is direct: modernize on this stack and inherit the audit pipeline—sovereign, auditable, post-quantum, and continuous—rather than rebuild it from scratch each cycle.