MINIO

Compliance Without Compromise

“AIStor reduced the number of moving parts in our compliance stack—and that reduced risk immediately.”

— Director of Infrastructure, Defense Contractor

AIStor closes the gaps that auditors find in fragmented architectures.

One platform, not a multi-vendor flowchart: Encryption, retention, versioning, and audit in one place

Native at the storage layer: No external key management, no separate WORM appliance

Unified audit logs: No scattered trails to correlate across systems
One answer for auditors — point to one system, not an architecture diagram

Federally validated cryptography: FIPS-ready without separate binaries

Tamper-proof retention: Immutability enforced on the same code path as production data

Compliance built in, not bolted on: One operational surface replaces the sprawl

The Challenge: Compliance Sprawl Creates Risk

If you're responsible for compliance in a regulated environment, you know the architecture: a key management service here, a separate WORM appliance there, audit logs flowing to yet another platform. You're the one coordinating encryption key rotation across fragmented services, scheduling the downtime windows, and fielding the call when something doesn't sync.

You're watching retention policies enforced by external engines drift out of sync with actual storage state. You're correlating timestamps across scattered audit logs when someone needs incident reconstruction. Each integration point adds operational overhead, failure modes, and attack surface that you have to manage. Every additional service means another vendor relationship, another contract, another team that needs training, and another system to explain when auditors start asking questions. The fragmentation isn't just inefficient. It's a liability you carry. And when auditors ask how data is protected, you're the one standing in front of a flowchart instead of giving a straightforward answer.

The AIStor Solution: One Answer for Auditors.

There's a better way. AIStor doesn't delegate compliance to external services. Encryption, immutability, access governance, and audit trails all operate natively at the storage layer, using the same stateless architecture that makes AIStor resilient and performant. You meet retention and encryption objectives using a single platform instead of orchestrating multiple services, and because these capabilities are runtime-enforced, the path from evaluation to production compliance is dramatically shorter. What follows is how each capability works in practice: immutable retention that auditors trust, encryption that protects every object with its own key, audit logging that captures everything with nanosecond precision, and versioning that preserves complete history.

Immutable Records and WORM Retention

Object Lock implements write-once-read-many retention directly in the storage layer. Configure it on a bucket via S3 API using PutObjectLockConfiguration and enforcement happens on internal object metadata, not in an external policy engine that might fail independently or drift out of sync. You get two retention modes. Compliance mode prevents deletion or modification by any user, including root, until the retention period expires. No override. No bypass. The storage layer simply refuses the operation. Retention periods cannot be shortened once set, meeting the non-rewriteable, non-erasable standard that regulators require. Governance mode provides the same protection but allows users with explicit bypass permissions to modify or delete objects when business requirements demand flexibility, with enforcement logic checking retention mode and expiration dates before permitting any modification.

When litigation or investigation hits, Legal Hold gives you indefinite retention independent of standard retention periods. Place a hold on an object and it stays protected until you explicitly remove it, regardless of any configured expiration. This implementation has been independently assessed by Cohasset & Associates for SEC 17a-4(f) compliance, providing the third-party validation that auditors expect for broker-dealer records retention under FINRA and CFTC requirements. When an auditor asks how you guarantee immutability, you have a validated answer.

Object Lock protections apply per version, enabling retention of historical data even when current versions expire. This per-version enforcement extends protection across the complete object history, and it means you can satisfy both retention and deletion obligations on the same data over time.

Encryption Without Excuses

Most encryption implementations force tradeoffs you shouldn't have to make. Use a single key per bucket and a breach exposes everything in that bucket. Require an external KMS for every operation and you've added latency plus a dependency that can take down your storage when the KMS becomes unavailable. You end up choosing between blast radius and operational complexity. AIStor eliminates that choice. Every object you store is encrypted with a unique, randomly generated 256-bit key using AES-256-GCM authenticated encryption. Not one key per bucket. Not one key per tenant. One key per object. If a single key is compromised, a single object is exposed, not your entire dataset.

Your object keys are never stored in plaintext. They are sealed using key-encryption-keys derived via HMAC-SHA256, following the DAREv2 (Data At Rest Encryption v2) format. DAREv2 provides authenticated encryption that detects tampering. If someone modifies encrypted data without the key, decryption fails rather than returning corrupted plaintext. When an auditor asks how you protect data integrity, you can point to cryptographic authentication that catches corruption or tampering automatically.

One key per object. One object at risk.

When you activate FIPS 140-3 mode, every cryptographic operation restricts to approved algorithms. Data at rest encrypts with AES-GCM. TLS cipher suites for data in transit limit to AES-GCM, with TLS 1.3 permitting AES-128-GCM and AES-256-GCM, and TLS 1.2 permitting ECDHE key exchange with AES-GCM encryption. Clients attempting to negotiate non-approved ciphers simply fail to connect, meaning zero misconfiguration risk. If you're subject to FISMA, FedRAMP, or CMMC requirements, you get validated cryptography without the procurement complexity of maintaining separate binaries for compliant environments.

Complete Audit Trails

When your audit logs live in separate systems for storage, identity, and application layers, reconstructing who did what means correlating timestamps across platforms that may not even use the same time source. AIStor eliminates that problem. Every S3 API operation generates a structured audit entry capturing user identity (AccessKey and ParentUser for assumed credentials), API name, bucket and object names, HTTP status codes, byte counts, and timestamps with nanosecond precision. Not sampled. Not summarized. Every operation. Entries ship to your configured targets (file, HTTP endpoint, or syslog) in JSON format, immediately parseable by SIEM platforms and compliance reporting tools without custom regex to maintain. When an auditor asks who accessed what data and when, you run a query. You don't reconstruct a timeline from five different systems.

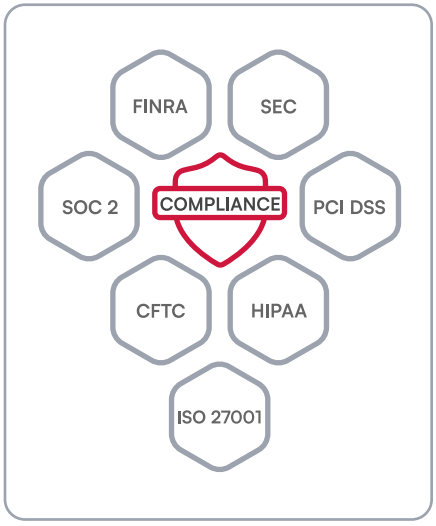
Immutable Versioning and Data Lifecycle

Versioning and lifecycle management complete the compliance picture by ensuring you can satisfy both retention and deletion obligations with full auditability. Each object version receives a 128-bit UUIDv4 identifier conforming to DCE 1.1 v4 UUID specification. Versions are immutable. Once created, they cannot be modified. All overwrites create new versions, preserving complete history. Delete operations create delete markers rather than removing data, enabling point-in-time recovery and maintaining audit trails that show exactly what data existed at any moment. To permanently remove a version, you must explicitly delete

it by version ID, and only users with appropriate permissions can purge versions. Combined with Object Lock, this creates defense in depth: even if credentials are compromised, attackers can't destroy historical data protected by retention policies. Automated retention policies execute transitions and expirations without external schedulers. Data residency controls and geo-fencing ensure compliance with regional sovereignty requirements. For GDPR and CCPA, verifiable data deletion meets right-to-erasure requirements with audit trails proving deletion occurred.

Regulatory Coverage

Because encryption, retention, audit logging, and versioning all operate at the storage layer, compliance evidence comes from one system rather than requiring correlation across multiple platforms. Object Lock Compliance mode has been independently assessed by Cohasset Associates for SEC 17a-4(f), FINRA 4511(c), and CFTC 1.31(c)-(d) broker-dealer records retention—third-party validation that gives financial services organizations the documentation auditors expect. The same native capabilities support control requirements across HIPAA, PCI DSS, SOC 2, ISO 27001, and CJIS. For GDPR and CCPA, data residency controls and verifiable deletion provide audit trails that document when data was removed. Organizations requiring FIPS-aligned cryptography enable it at runtime with an environment variable—no separate binary, no specialized procurement.



Traditional Compliance Architecture vs MinIO AIStor

	Traditional Architecture	MinIO AIStor
Encryption	External KMS required, key per bucket or tenant	Per-object 256-bit keys, AES-256-GCM, DAREv2 authenticated encryption
WORM Retention	Separate appliance or service, policy sync required	Native Object Lock with Compliance and Governance modes, storage-layer enforcement, Cohasset assessed
Audit Logging	Scattered across services, reconstruction required	Unified JSON logs, every operation captured, nanosecond timestamps
FIPS Compliance	Separate builds, complex procurement	Runtime activation via environment variable, single binary
Versioning	Application-managed or separate service	Native immutable versions with 128-bit UUIDv4 identifiers per DCE 1.1 spec
Retention Enforcement	External policy engines with potential sync lag	Direct enforcement on internal object metadata, no external dependencies

Why MinIO AIStor

The pattern across every capability is the same: what other architectures delegate to external services, AIStor handles natively at the storage layer. This isn't compliance bolted on after the fact. It's compliance built in from the start, operating on the same code path that handles your production data.

One platform to secure. One platform to audit. One platform to defend when regulators come asking questions.

Ready to see it in action?

Visit min.io to learn more. Download AIStor and test it yourself. Request a demo to see how these protections work in your environment.