Kilimanjaro

TECHNICAL WHITEPAPER

DATA PROTECTION & SOVEREIGNTY

One Stack. Many Frameworks.

How Kilimanjaro implements the technical controls behind 24 data-protection laws and security frameworks — and how it proves them.

June 2026 · v1.0 · 12 min read

01 The problem: frameworks specify outcomes, not products
02 One stack, mapped to every framework
03 Data protection: encryption at the column, not the disk
04 Provable erasure for structured personal data: destroy the key, not just the row
05 Audit accountability: a signed ledger, not assembled logs
06 Access control: identity-verified at the kernel
07 Data sovereignty: keys and data, in-region
08 Coverage: 24 frameworks, one posture

Executive summary

Compliance frameworks do not specify products. They specify outcomes: personal data must be protected at rest, access must be controlled and attributable, erasure must be provable, and — increasingly — data must stay within a jurisdiction. The work of compliance is mapping those outcomes onto controls, then evidencing them to an auditor.

Most teams assemble that mapping framework by framework, stitching together a database, a key manager, a logging system, and a residency story that were never designed to answer a regulator together. The evidence is reconstructed after the fact, and it is contestable.

Kilimanjaro takes the opposite approach. One technical stack — field-level encryption, cryptographic erasure, a signed audit ledger, identity-verified access, and regional key pinning — implements the technical controls that the major frameworks share, and produces provenance-valid evidence as a byproduct of running. This paper walks through each control, the primitive that implements it, the proof it emits, and the boundary of what the platform does and does not answer.

01 The problem: frameworks specify outcomes, not products

GDPR Article 32 requires encryption of personal data "by design." HIPAA's Security Rule requires technical safeguards for electronic Protected Health Information. PCI DSS 4.0 § 3.5.1 requires the Primary Account Number to be unreadable wherever it is stored. The NDPA, LGPD, POPIA, the DPDPA, and a dozen more each phrase the same handful of obligations in their own statutory language.

Read together, the frameworks converge on five technical outcomes: protect data at rest, control and attribute access, prove erasure, keep an accountable audit trail, and — for a growing set of jurisdictions — keep regulated data and its keys in-region. A control that genuinely implements one of these outcomes implements it for every framework that requires it.

The failure mode is not the controls themselves; it is the seams between them. Disk-level encryption leaves plaintext readable to anything with logical database access. Audit logs assembled from three systems are stale and contestable by the time an assessor reviews them. "We deleted the records" is a claim, not a proof — especially across backups. Each seam is where an audit finding lives.

Technical outcome	The seam where it breaks
Protect data at rest	Disk-level encryption leaves plaintext readable to anything with logical database access
Keep an accountable audit trail	Audit logs assembled from three systems are stale and contestable by the time an assessor reviews them
Prove erasure	"We deleted the records" is a claim, not a proof — especially across backups

Three of the outcomes frameworks converge on, and the seam — named above — where a stitched-together stack breaks each one.

02 One stack, mapped to every framework

Kilimanjaro implements the five outcomes once, at the infrastructure layer, so a single posture answers each framework rather than a per-framework integration. The same field-encrypted column that answers GDPR Article 32 answers HIPAA § 164.312, the GLBA Safeguards Rule, and PCI DSS § 3.5.1. The same signed ledger that evidences SOC 2 CC6.1 evidences SOX § 404 and ISO 27001 Annex A.8.15.

The mapping is not marketing alignment; it is the same primitive answering each citation. The table below shows the five controls and a representative framework citation each one implements.

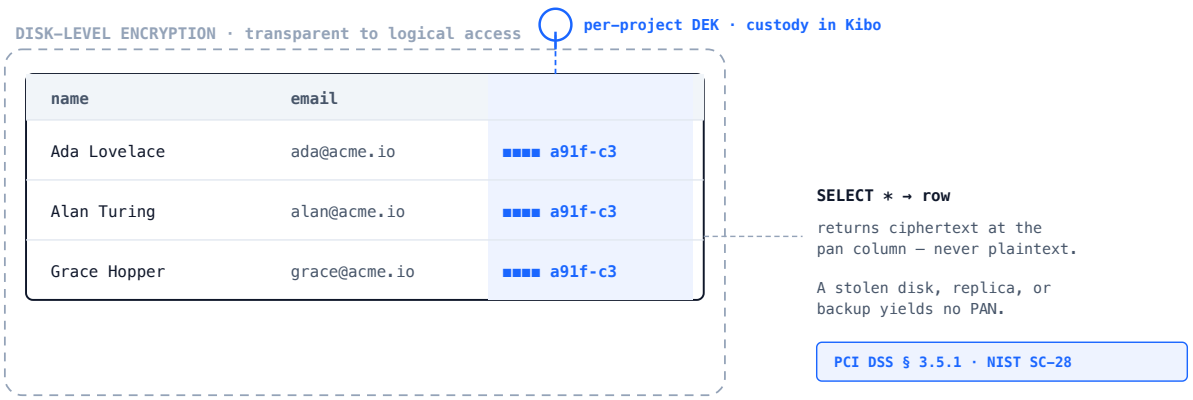
Control	Primitive	Representative citation
Data protection at rest	Field-level encryption under a per-project key	GDPR Art. 32 · HIPAA § 164.312(a)(2)(iv) · PCI DSS § 3.5.1
Provable erasure	Zero-Residue cryptographic key destruction	GDPR Art. 17 · CPRA § 1798.105 · APP 11.2
Audit accountability	Signed, hash-chained audit ledger	SOC 2 CC6.1 · SOX § 404 · ISO 27001 A.8.15
Access control	Gate identity-verified access	HIPAA § 164.312(a) · ISO 27001 A.5.15 · APPI Art. 23
Data sovereignty	Regional key pinning	Saudi PDPL · NDPA · CBN (Nigeria)

The five technical controls and a representative framework citation each implements. The full 24-framework mapping is published at kilimanjaro.dev/solutions/compliance.

03 Data protection: encryption at the column, not the disk

The control nearly every framework names first is encryption of sensitive data at rest. Kilimanjaro field-encrypts each sensitive column under a per-project Data Encryption Key (DEK) before the database write. Logical access to the database row returns ciphertext, not plaintext — so a compromised database, a misconfigured replica, or a backup that walks out the door yields nothing usable without the project key.

This is the distinction the frameworks increasingly make explicit. PCI DSS § 3.5.1.2 states that disk-level encryption alone is insufficient; the PAN must be unreadable at the column. NIST SP 800-53 SC-28 makes the same point for information at rest. Field-level encryption is the control that answers them — and it is the same control whether the data is a PAN, a PHI field, an SSN, or a row of personal data under the NDPA.



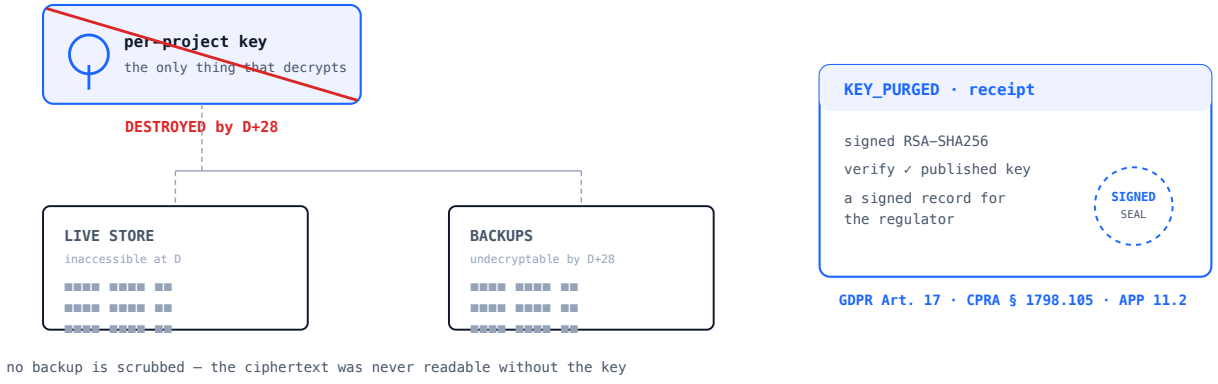
Field-level encryption: the PAN column is ciphertext at rest under the per-project key, while disk-level encryption stays transparent to anything with logical access.

04 Provable erasure for structured personal data: destroy the key, not just the row

The right to erasure — GDPR Article 17, the CPRA's right to delete, Australia's APP 11.2 destruction obligation, Saudi PDPL Article 18 — asks for something harder than deleting live rows. It asks for proof that the data is gone, including from backups, where deleting rows is impractical or impossible.

Kilimanjaro answers with Zero-Residue: cryptographic erasure by key destruction. When an erasure request is ratified, that person's field key is destroyed. Their encrypted data is unreadable in live storage at once, and backup copies age out within 28 days. The platform issues signed ledger entries as the record for the regulator. No backup is scrubbed.

The platform provides cryptographically verifiable erasure of structured per-subject personal data through irreversible cryptographic key destruction, including protected backup copies. Certain derived artefacts generated from personal data may not presently support independent per-data-subject erasure where those artefacts contain information relating to multiple individuals. Customers should assess this limitation when determining whether the platform is appropriate for particular processing activities.



Zero-Residue: destroying the per-project key renders the live store and every backup permanently undecryptable, with a signed KEY_PURGED receipt as the record.

05 Audit accountability: a signed ledger, not assembled logs

Every framework with an accountability obligation — SOC 2, SOX, ISO 27001, and every data-protection statute with a regulator behind it — wants the same thing: a trustworthy record of who accessed or changed what, when, and under what authority. The weak point is almost never the absence of logs; it is that the logs are scattered, mutable, and reconciled by hand the week before the audit.

Kilimanjaro writes every access and change to a single audit ledger, attributed to a Gate-verified identity. The ledger is SHA-256 hash-chained and RSA-SHA256-signed by the platform: any tampering is evident, and any party can verify a record against the published key. A compliance pass compiles the per-control evidence from the ledger into a cited mapping — generated, not assembled. The auditor receives provenance-valid evidence rather than a spreadsheet.

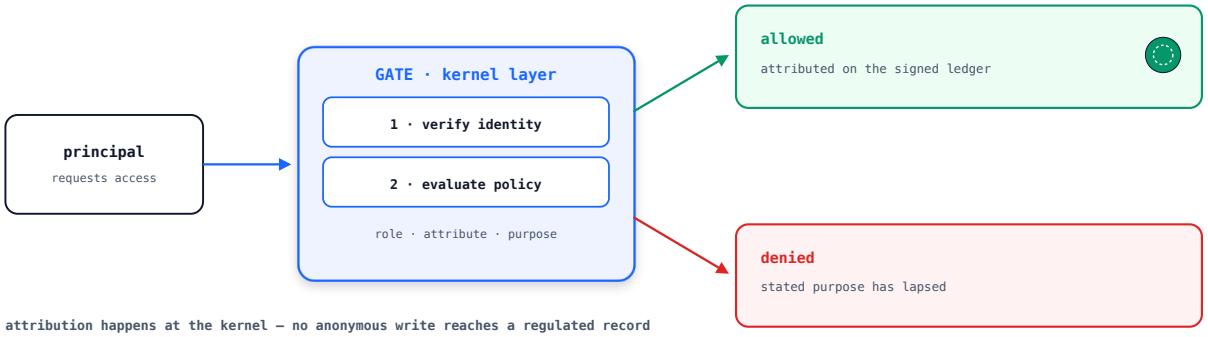


Every access and change is written to a SHA-256 hash-chained, RSA-SHA256-signed ledger — break any link and the chain stops verifying.

06 Access control: identity-verified at the kernel

Controlling access is only half the obligation; attributing it is the other half. Kilimanjaro routes access through Gate, which verifies the accessing principal's identity and evaluates the access against policy — role, attribute, and, where a framework requires it, purpose. A sensitive-data read can be denied when the stated purpose has lapsed, and every authorized access is attributed on the signed ledger.

Because attribution happens at the kernel layer rather than in application code, no anonymous write touches a regulated record, and the access record an assessor asks for already exists. This is the control behind ISO 27001 A.5.15, HIPAA's access-control safeguards, and APPI's purpose-gated handling of sensitive personal information.

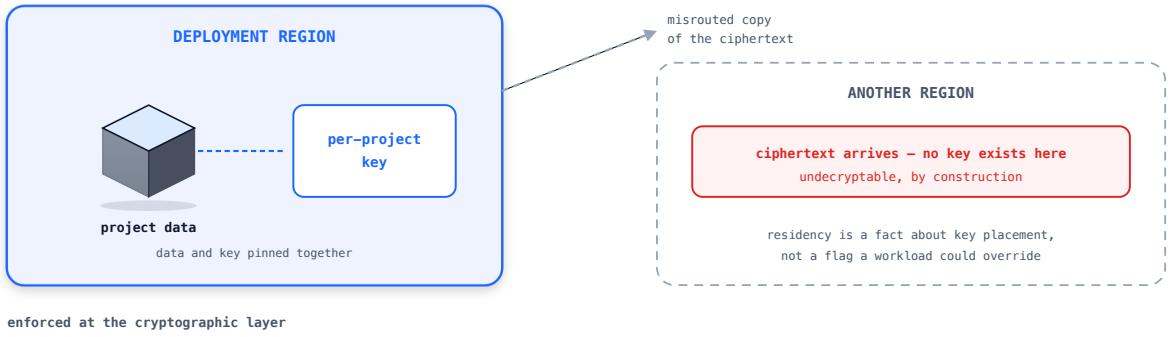


Gate verifies the accessing principal’s identity, then evaluates the access against role, attribute, and purpose policy. A read whose stated purpose has lapsed is denied; an authorized read is attributed on the signed ledger — attribution happens at the kernel, so no anonymous write touches a regulated record.

07 Data sovereignty: keys and data, in-region

A growing number of sovereign regulators are no longer satisfied by encryption alone — they require regulated data to remain within a jurisdiction. Saudi Arabia's PDPL and Nigeria's data-localisation mandates are explicit; others express a strong residency preference. Kilimanjaro pins a project's data and its per-project key to a deployment region, so the data and the key that protects it stay where the law requires.

Residency is enforced at the cryptographic layer, not as a configuration flag a workload can override: the key for an in-region project does not leave its region, so even a misrouted copy of the ciphertext is undecryptable elsewhere. For data-protection statutes without a hard localisation mandate, the same mechanism supports a residency posture — it is offered as a capability, not presented as a statutory requirement the law does not impose.



A project's data and its per-project key are pinned together in one region. Residency is enforced at the cryptographic layer, not a configuration flag: a misrouted copy of the ciphertext has no key to decrypt it with outside that region.

08 Coverage: 24 frameworks, one posture

The same five controls map to 24 data-protection laws and security frameworks across the Americas, Europe, Africa, Asia-Pacific, the Middle East, and the cross-jurisdiction security standards. Each framework's specific citations, the control each one names, and the proof Kilimanjaro emits are documented per-framework in the Compliance Frameworks index.

The point is not that Kilimanjaro "makes you compliant" — no platform can, and the next section is explicit about that. The point is that the technical controls a framework requires are implemented once and evidenced the same way for every framework you answer to.

Geography	Framework cited in this paper
Americas	CPRA § 1798.105 · HIPAA § 164.312 · SOX § 404
Europe	GDPR Art. 17 / Art. 32
Africa	NDPA · CBN (Nigeria) · POPIA
Asia-Pacific	APP 11.2 (Australia) · DPDPA (India) · APPI (Japan)
Middle East	Saudi PDPL
Cross-jurisdiction security standards	PCI DSS § 3.5.1 · SOC 2 CC6.1 · ISO 27001 A.8.15 / A.5.15 · NIST SP 800-53 SC-28

A representative framework from each geography this paper cites — the full 24-framework mapping is published at kilimanjaro.dev/solutions/compliance.

What we disclose

Compliance is a shared responsibility, and we are precise about the line. Kilimanjaro implements and evidences the technical controls a framework specifies — encryption at rest, access attribution, provable erasure, audit accountability, and data residency. It does not provide the organizational controls a framework also requires: your policies, your staff training, your risk assessments, your business-associate and processor agreements. A framework is answered by both; we answer the technical half and produce the evidence for it.

On custody, we draw the tier line in the open. In the platform-managed posture, your per-project key is held and operated by the platform; "implements the control of" a framework means the encryption, erasure, and audit controls are in force — not that the platform is blind to your data. For buyers who require that the platform never hold the key, the BYOK and HYOK tiers root key custody in your own KMS or HSM; that is the sovereign answer, and it is a deliberate, separate tier.

Field-level encryption protects data at rest. Where a workload must compute over that data — search, indexing, agent memory — it is decrypted in use, under the project key, inside the perimeter. That is a precise "encrypted at rest, processed in-perimeter" posture, not a zero-knowledge claim. Our erasure proof is cryptographic key destruction with a platform-signed, customer-verifiable receipt; the envelope cipher is AES-256-GCM-SIV, and FIPS scoping applies to the custodied keys, not to a blanket "FIPS-validated system" claim.

Map your frameworks to the controls

Talk to the team and we will map your jurisdiction, data classes, and deployment region to the specific controls Kilimanjaro implements — and the evidence each one produces.

Talk to Compliance →

Browse the 24 frameworks