Kilimanjaro

TECHNICAL WHITEPAPER

SECURITY & TRUST

Sovereign Key Custody

Where the trust actually sits in Kilimanjaro's key hierarchy — the platform-managed tier, the attestation it rests on, and the BYOK/HYOK answer for buyers who want to hold the key themselves.

June 2026 · v1.0 · 14 min read

01 The question behind "is it encrypted?"
02 The key hierarchy: a wrapped chain, hardware-rooted at the confidential tier
03 Where the trust sits in the platform-managed tier
04 Two attestation directions, never conflated
05 The sovereign answer: BYOK and HYOK
06 What our FIPS and cryptographic claims cover

Executive summary

Encryption is easy to claim and hard to verify. The question a security architect actually asks is not “is it encrypted,” but “who holds the key, where does that key live, and what am I being asked to trust?” This paper answers that question for Kilimanjaro — in the open.

Kilimanjaro encrypts data with a layered, wrapped key chain. In the platform-managed tier the root is held and operated by the platform: at the hardened baseline the root is a 256-bit AES key held as a regional software-protected Azure Key Vault secret, loaded into service memory only; the confidential tier adds a hardware root above it — a 4096-bit RSA region master key (MKEK) in a regional FIPS-140-3 Level 3 HSM, released to your attested workload via Microsoft Azure Attestation. That attestation dependency on Microsoft is real and named — not a sovereign claim — and we disclose it rather than bury it.

For buyers who require that the platform never hold the key, the BYOK and HYOK tiers root custody in your own KMS or certified HSM. This paper draws both pictures precisely, separates the two attestation directions that are easy to conflate, and scopes exactly what our FIPS and cryptographic claims do and do not cover.

01 The question behind “is it encrypted?”

Every platform says it encrypts your data. The claim is nearly meaningless on its own, because it leaves the load-bearing questions unanswered: who can decrypt it, where the key that decrypts it physically lives, and whose attestation you are relying on when you believe the answer. A serious key-custody story answers those questions concretely, and is direct about the parts where trust rests on a third party.

Kerckhoffs’s principle applies: the design is publishable, and security must not depend on hiding it. The key hierarchy, the custody model, and the attestation chain are laid out here, and the trust boundary is named explicitly instead of gestured past.

Load-bearing question	Answered by
Who can decrypt it?	The key hierarchy and custody tier (§ The key hierarchy)
Where does the key physically live?	Software-protected secret at baseline, regional HSM at the confidential tier (§ Where the trust sits)
Whose attestation are you relying on?	Named explicitly, not gestured past (§ Two attestation directions)

The three questions “is it encrypted?” leaves unanswered — and where this paper answers each one.

02 The key hierarchy: a wrapped chain, hardware-rooted at the confidential tier

Kilimanjaro wraps data in a layered key chain: a per-field data key (DEK) encrypts each column; a per-project key (PKEK) wraps the DEKs; and a region key-encryption key (KEK) — a 256-bit AES key — wraps the PKEKs via AES-256-GCM-SIV. What differs by tier is how that region KEK is protected. At the hardened baseline the region KEK is held directly as a regional software-protected Azure Key Vault secret (in service memory only, never on disk).

The confidential tier adds a region master key-encryption key (MKEK) — a 4096-bit RSA key in a regional FIPS-140-3 Level 3 HSM, never exported — that wraps the region KEK and is released to your attested workload only via Secure Key Release.† So the confidential tier is a regional four-tier chain (MKEK → KEK → PKEK → DEK); the hardened baseline is the same chain without the HSM-held MKEK, the AES-256 KEK being the regional software-protected root.



A per-project key (PKEK) wraps the per-field DEKs; the region KEK wraps the PKEKs. At the hardened baseline the AES-256 region KEK is a regional software-protected Azure Key Vault secret (no HSM); the confidential tier adds an RSA-4096 MKEK in a regional FIPS-140-3 Level 3 HSM that wraps it. Destroying a project's PKEK is Zero-Residue erasure for that project.

The per-project key is the unit of isolation and the unit of erasure. One project's PKEK cannot decrypt another's data, and destroying a PKEK renders every field encrypted under it permanently undecryptable — the Zero-Residue erasure model, with a signed KEY_PURGED receipt as the record. The envelope cipher that protects the data itself is AES-256-GCM-SIV.

† Per-project key anchoring (confidential tier, optional). On the confidential tier, an organization may elect to anchor each project's keys in its own dedicated hardware key: a per-project Project Master Key (PMKEK) — a 4096-bit RSA key held in a regional FIPS-140-3 Level 3 HSM, with a release policy bound to that project's own attested workload. Anchoring means the project's key-encryption key (PKEK) is wrapped by, and released only through, that PMKEK — so instead of a shared regional root unlocking every project's keys, each anchored project's key chain roots in its own HSM-resident master key, released only to that customer's verified, attested code. The effect is per-project hardware custody and per-project attestation-bound release: one project's keys cannot be unlocked by another project's workload, or by the shared regional root. This remains platform-managed custody — a stronger isolation tier, not customer-held key ownership; for custody the platform never holds, that is BYOK/HYOK.

03 Where the trust sits in the platform-managed tier

In the platform-managed tier, the root of the hierarchy is held and operated by the platform. We are precise about what that means and what backs it: at the confidential posture the platform-held key custody is attested by Microsoft Azure Attestation (MAA) and the region master key is protected by a regional FIPS-140-3 Level 3 HSM, while the hardened baseline holds its root key as a software-protected secret. That is the precise description — the platform operates the key, and at the confidential posture the assurance that it is held correctly rests, in part, on Microsoft's attestation service.

This is not a sovereignty claim, and we do not dress it as one. In the platform-managed tier we do not say "your keys," "we can never see your data," or "the key never leaves your control." Those statements are true only of the customer-rooted tiers below. The baseline platform-managed posture is a hardened, software-protected custody model; the HSM-backed, attestation-gated confidential posture is a premium tier, not the default — and we label which one a given deployment is on.

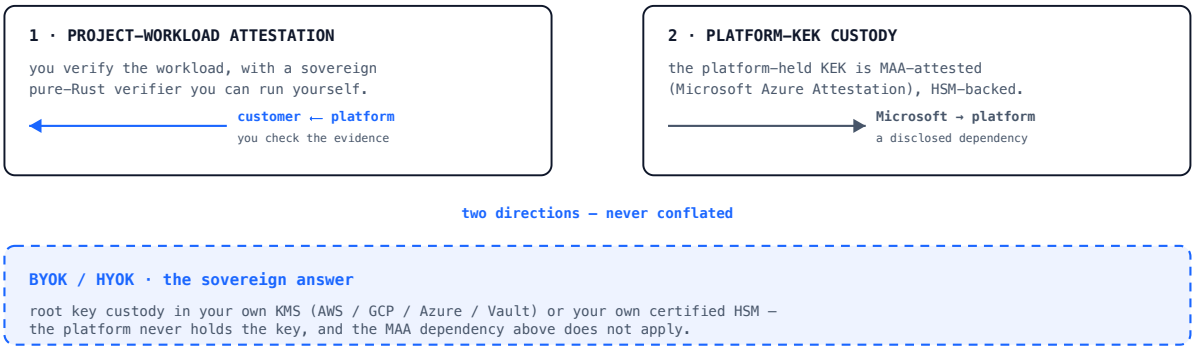
Posture	Root held as	Assurance rests on	"Your keys"?
Hardened baseline	Software-protected secret	Platform operation alone	No
Confidential	Regional FIPS-140-3 Level 3 HSM	Microsoft Azure Attestation (MAA)	No
Customer-rooted (BYOK/HYOK)	Your own KMS or certified HSM	Your own control	Yes

What backs the platform-managed root at each posture, and whether "your keys" is a true statement there.

04 Two attestation directions, never conflated

Attestation is easy to wave at and hard to reason about, because two different things both get called “attestation.” The first is project-workload attestation: you verify that the workload running on the platform is the one you expect, using a sovereign, pure-Rust verifier you can run yourself. There, the trust flows toward you — you check the evidence.

The second is platform-KEK custody attestation: the platform’s key custody is attested by Microsoft Azure Attestation. There, the trust flows from Microsoft to the platform — it is a disclosed third-party dependency, not something you independently verify. These are different directions, and conflating them is how vendors imply more sovereignty than they deliver. We keep them separate, and we always pair the platform-managed picture with the customer-rooted answer below.

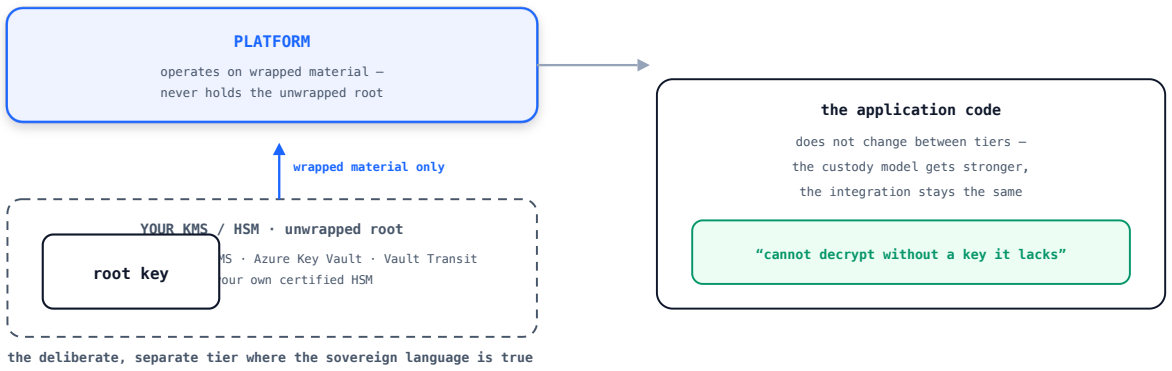


Project-workload attestation (you verify the platform’s workload with a sovereign verifier) is a different direction of trust from platform-KEK custody attestation (Microsoft attests the platform’s HSM custody). BYOK/HYOK removes the second dependency entirely.

05 The sovereign answer: BYOK and HYOK

If the Microsoft-attested custody dependency is unacceptable for your threat model, the answer is to hold the key yourself. With BYOK, the root of the hierarchy is rooted in your own KMS — AWS KMS, GCP KMS, Azure Key Vault, or HashiCorp Vault Transit. With HYOK, it is rooted in your own certified HSM. In both, the platform operates on wrapped material and never holds the unwrapped root key; the MAA dependency described above simply does not apply.

The application code does not change between tiers — the custody model gets stronger while the integration stays the same. This is the deliberate, separate tier where the sovereign language is true: the platform cannot decrypt your data without a key it does not hold.



The platform holds and operates on wrapped key material only; the unwrapped root key lives outside its boundary, in your own KMS (BYOK) or certified HSM (HYOK). The application code is unchanged between tiers — only the custody model gets stronger.

06 What our FIPS and cryptographic claims cover

Precision matters most where it is easiest to overstate. Our FIPS-140-3 Level 3 claim scopes to the custodied keys — the HSM that protects the root key material — not to the system as a whole. We do not claim “the system is FIPS-validated.” The envelope cipher that encrypts your data, AES-256-GCM-SIV, is a RustCrypto implementation and is not itself FIPS-validated; it is a modern, misuse-resistant AEAD, and we describe it as exactly that.

The same discipline applies to every adjective in this paper: the audit ledger is platform-signed (RSA-SHA256) and customer-verifiable against a published key, hash-chained and tamper-evident — not “tamper-proof,” and not “tenant-signed.” Stating the boundary precisely is the point; a claim a buyer’s own security team can puncture is worse than no claim at all.

Claim	What it covers
FIPS-140-3 Level 3	The custodied keys — the HSM protecting root key material. Not “the system is FIPS-validated.”
AES-256-GCM-SIV envelope cipher	A RustCrypto implementation; a modern, misuse-resistant AEAD. Not itself FIPS-validated.
Audit ledger signature	Platform-signed (RSA-SHA256), customer-verifiable, hash-chained, tamper-evident. Not “tamper-proof,” not “tenant-signed.”

Each cryptographic claim in this paper, scoped exactly to what it covers.

What we disclose

The load-bearing disclosure is this: in the platform-managed tier, the platform holds and operates the root of your key hierarchy, and at the confidential posture the assurance that it does so correctly rests in part on Microsoft Azure Attestation and an Azure FIPS-140-3 Level 3 HSM. That is a real third-party dependency in the trust path. We name it on purpose. If it is unacceptable for your threat model, BYOK/HYOK roots custody in your own KMS or HSM and removes it.

We do not conflate the two attestation directions: workload attestation (which you verify) is not the same as platform-KEK custody attestation (which Microsoft provides). The HSM-backed, attestation-gated confidential posture is a premium tier, not the baseline platform-managed default. And our FIPS claim scopes to the custodied keys, not the whole system — the AES-256-GCM-SIV envelope cipher is a non-validated (if modern and misuse-resistant) RustCrypto implementation.

Everything in this paper is publishable by design (Kerckhoffs): the security of the system does not depend on keeping any of it secret.

Take the key custody conversation deeper

Talk to a Sovereign Engineer about which custody tier fits your threat model — platform-managed with disclosed attestation, or BYOK/HYOK rooted in your own KMS or HSM.

Talk to a Sovereign Engineer →

Read about Kibo