Distributed Cryptographic Continuity

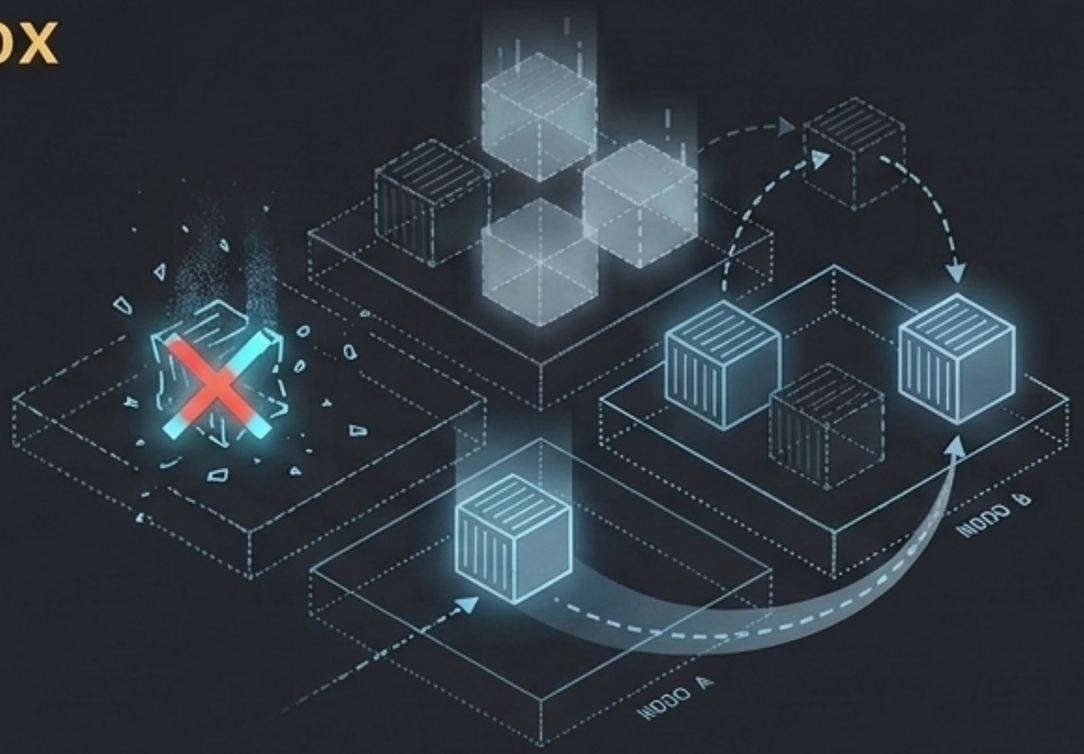
Decoupling Runtime and Identity in SkyDefended InfraApp v1.2

Author: Ismael Cruz Casasola
Date: 2026-05-22
Class: Core Architecture / Distributed Cryptographic Model

The Cloud-Native Paradox

Classic Model

The application lives for years on the same server. Keys live locally. Runtime and identity are mixed.



Ephemeral Infrastructure

The container is not persistent. The node is not stable. The workload can be restarted, migrated, or disappear without warning.

Structural question: If the container dies suddenly, where does the identity live?

The workload hosts the identity, but it is not the identity

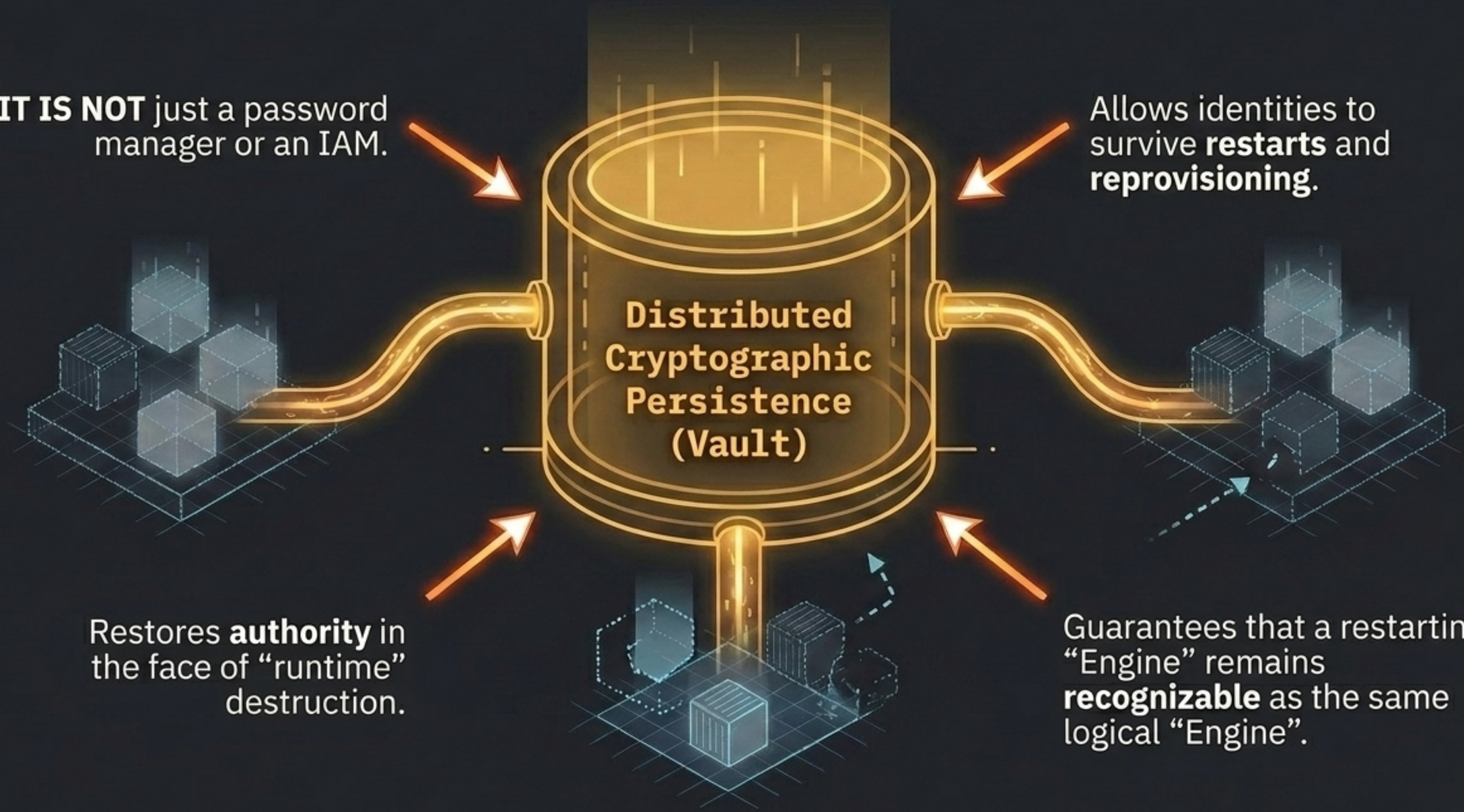
runtime ≠ identity

The cryptographic identity constitutes the real core of the system. Without cryptographic continuity, JWTs would be invalidated and federations would disappear.

The Custody System as a Cryptographic Persistence Layer

IT IS NOT just a password manager or an IAM.

Allows identities to survive **restarts** and **reprovisioning**.



Restores **authority** in the face of “runtime” destruction.

Guarantees that a restarting “Engine” remains **recognizable** as the same logical “Engine”.

Impact Classification upon Asset Compromise

Tier-0: Root Trust

Impact: Global systemic collapse. Identity forgery and loss of trust.

```
SESSION_SIGNING_KEY
Bundle signing key
JWKS operador
NCN_LICENSE_PRIVATE_KEY
```

Tier-1: Operational Critical

Impact: Compromises a specific Engine. Operationally severe, but recoverable through rotation/revocation.

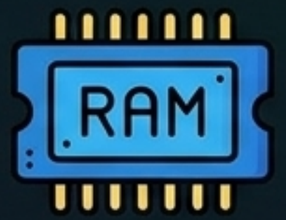
```
INTERNAL_API_KEY
STATE_SIGNING_KEY
RSA Identities per
Engine
```

Tier-2: Data Impact

Impact: Access to data. No collapse of the platform’s global trust.

```
DB credentials
OIDC client secrets
SIEM tokens
```

Spectrum of cryptographic material exposure



KV Level (Key-Value)

Lives temporarily in local RAM of the `workload`.

Highest exposure.

Lowest latency.

Transit Level

The private key never leaves custody.

Remote signing and encryption.

The `workload` delegates operations.

HSM Remote-Sign Level

The key never leaves the `hardware`.

Non-exportable.

Cryptographic boundary based on silicon.

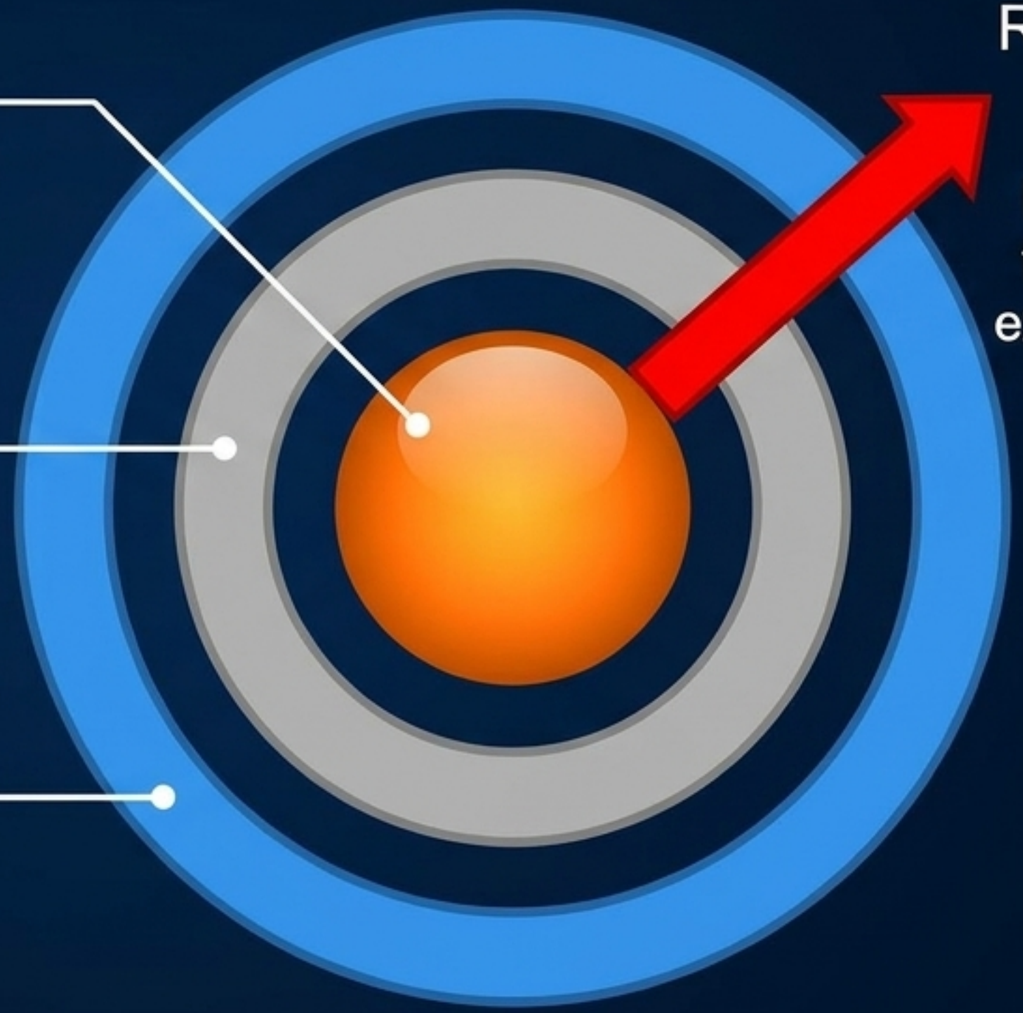
Maximum protection.

Strategic Resilience Alignment: Impact vs. Exposure

	KV Level	Transit Level	HSM Level
Tier-0			Target Alignment
Tier-1		Target Alignment	
Tier-2	Target Alignment		

The evolution of the model does NOT seek ‘everything to HSM’. KV, Transit, and HSM coexist according to latency, frequency, and cost. The alignment is an asset-by-asset operational resilience decision.

Cryptographic Containment Boundaries

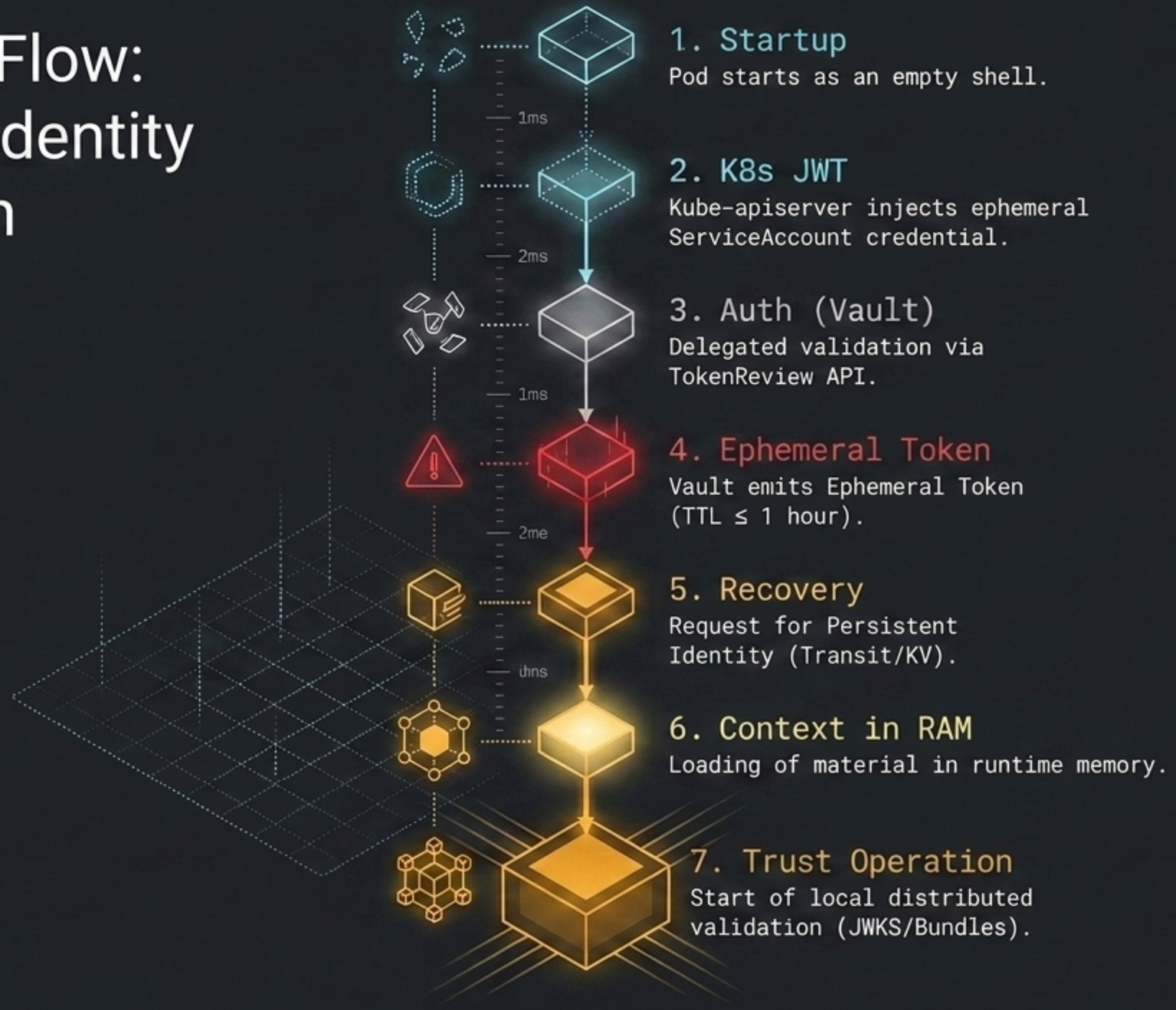


Hardware HSM
Private key is born and dies here.
(HSM Level)

Vault Transit Engine
Remote signing/encryption requests.
Key does not leave
(Transit Level)

Workload RAM / KV
Key is temporarily materialized
in local memory
(KV Level)

Bootstrap Flow: Temporal Identity Acquisition



1. Startup
Pod starts as an empty shell.

2. K8s JWT
Kube-apiserver injects ephemeral ServiceAccount credential.

3. Auth (Vault)
Delegated validation via TokenReview API.

4. Ephemeral Token
Vault emits Ephemeral Token (TTL ≤ 1 hour).

5. Recovery
Request for Persistent Identity (Transit/KV).

6. Context in RAM
Loading of material in runtime memory.

7. Trust Operation
Start of local distributed validation (JWKS/Bundles).

Logical Separation: Ephemeral Authentication vs. Continuous Authority

Authentication (K8s/Vault Ephemeral Token)

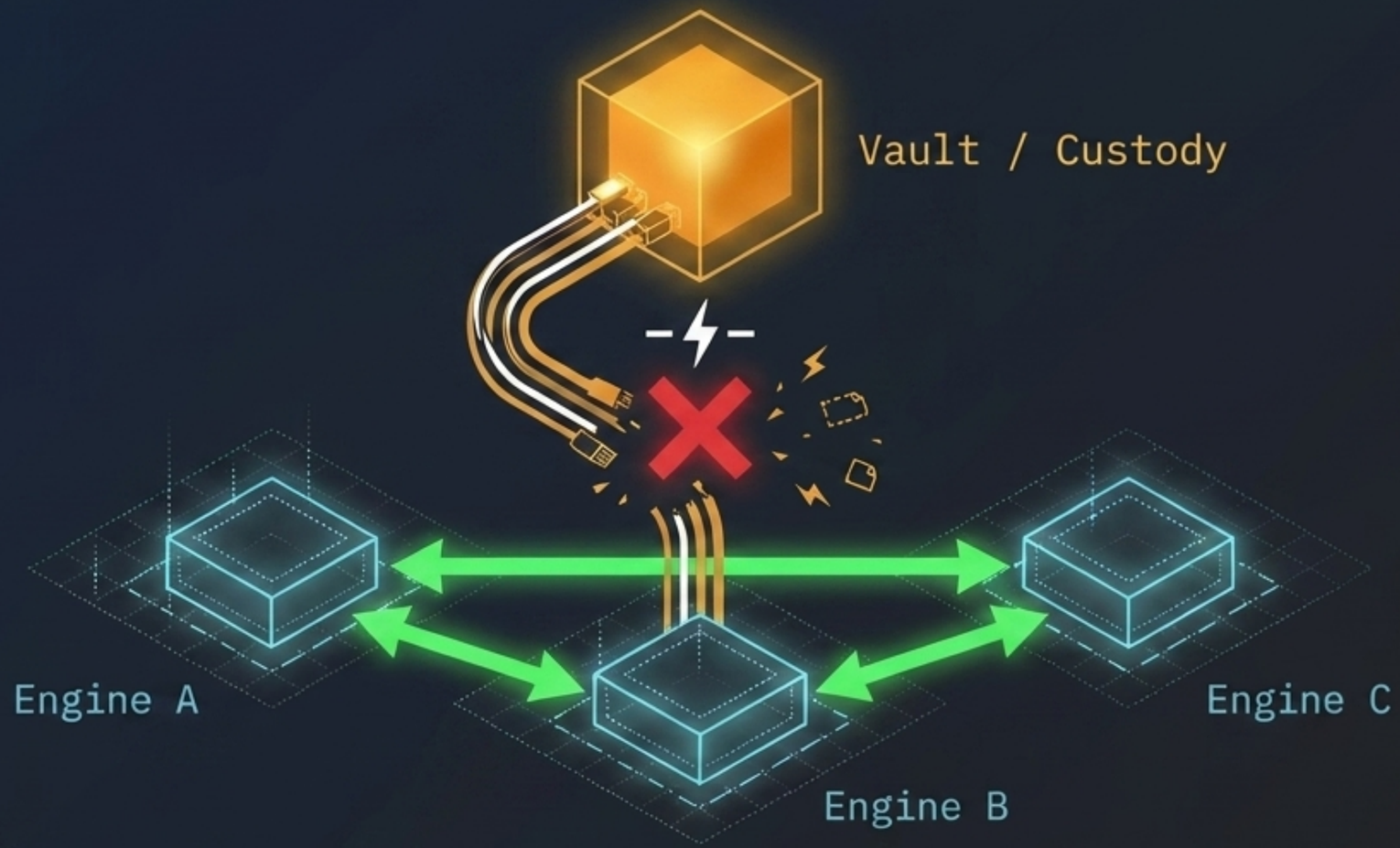
The hotel key. Enables the Bootstrap process. Rotates, expires (TTL ≤ 1h), and DOES NOT represent the real identity of the system.

≠

Identity (Persistent Cryptographic Authority)

The Engine’s DNA. Enables trust continuity. Survives runtime destruction. It is the identity that the rest of the platform recognizes.

Operational Resilience Outside the Continuous Hot Path

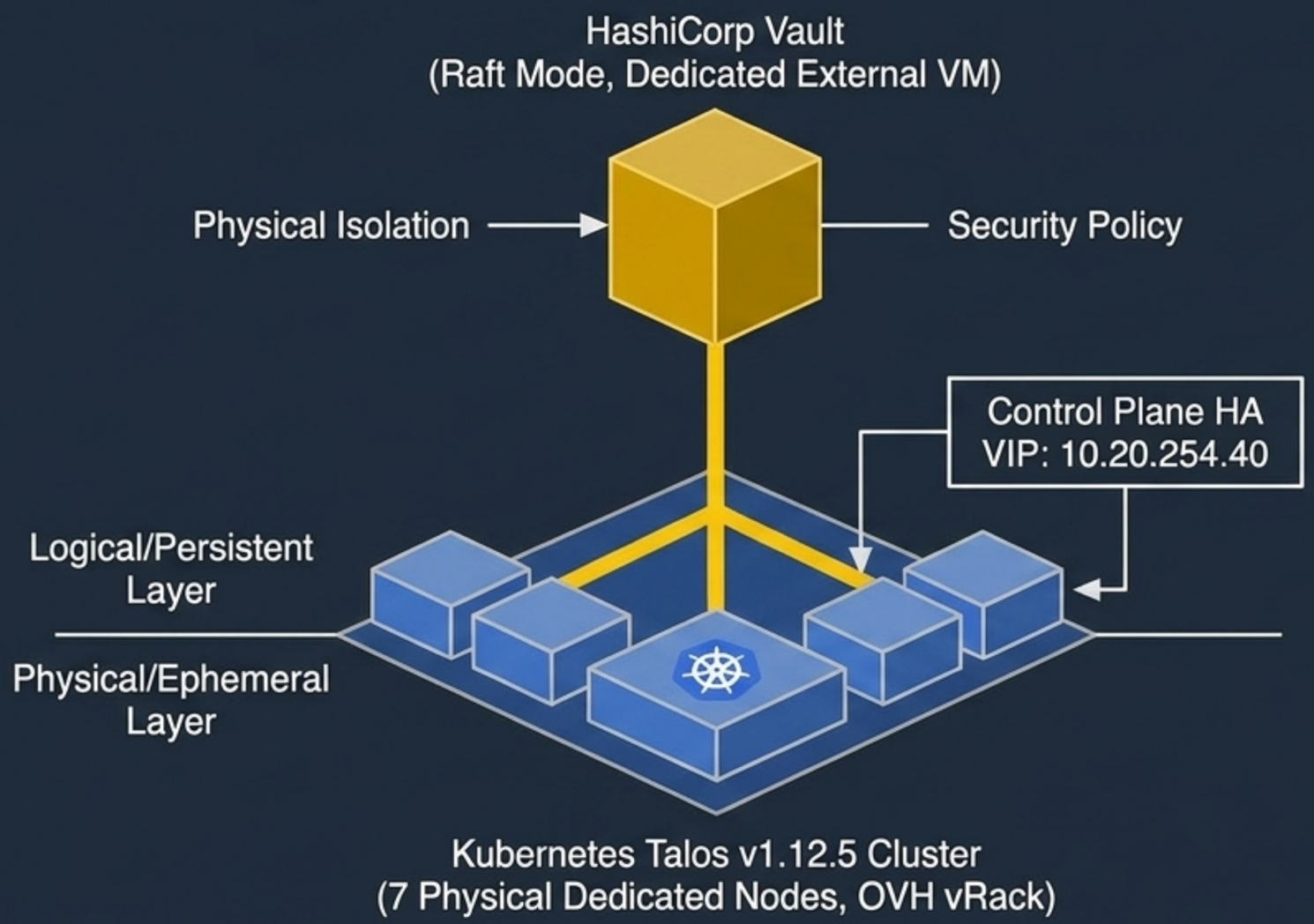


If custody temporarily fails, the platform maintains operation. JWT validation occurs locally via JWKS, bundles in RAM, and distributed trust. Vault is only critical for bootstrap, recovery, and rotation.

Structural Survival: Logical Engine vs. Physical Workload

Multiple ephemeral physical workloads represent a single stable logical Engine. The network continues to recognize the same cryptographic actor.

Physical Implementation of the SkyDefended Stack InfraApp 2026



Orchestration

- Kubernetes Talos v1.12.5 on 7 physical dedicated OVH nodes (internal vRack).
- Control Plane HA VIP: 10.20.254.40.

External Custody

- HashiCorp Vault in Raft mode (dedicated external VM).
- Physical isolation: compromising the cluster does not compromise the vault.

Integrity and Signing

- Cosign keyless OIDC (Fulcio + Rekor) integrated in CI.
- Pinning by digest (@sha256).
- Admission via Kyverno ClusterPolicy (verify-skydefended-image-signatures in Audit mode).

Operational Status of Cryptographic Custody and Roadmap

SECTION 1: DEPLOYED AND OPERATIONAL

On

KV Level

On

Vault KV v2. Syncronization to workloads via External Secrets Operator (ESO) for Tier-2 (DB, OIDC, SIEM).

Transit Level

On

Vault Transit (transit/keys/...). Encryption of mfaSecret using AES-256-GCM (First widespread use of the non-exportable model).

SECTION 2: IN PROGRESS / ARCHITECTURAL GOAL

- Azure Managed HSM (FIPS 140-3 Level 3 single-tenant) lune 2 Level 3 single-tenant) intended for Tier-0.
- Deployment of Vault HA (local follower).
- SPIFFE/SPIRE for native workload identity.

The infrastructure will disappear.
Runtimes will be continuously destroyed and rebuilt.
But the cryptographic authority and distributed trust will remain intact.