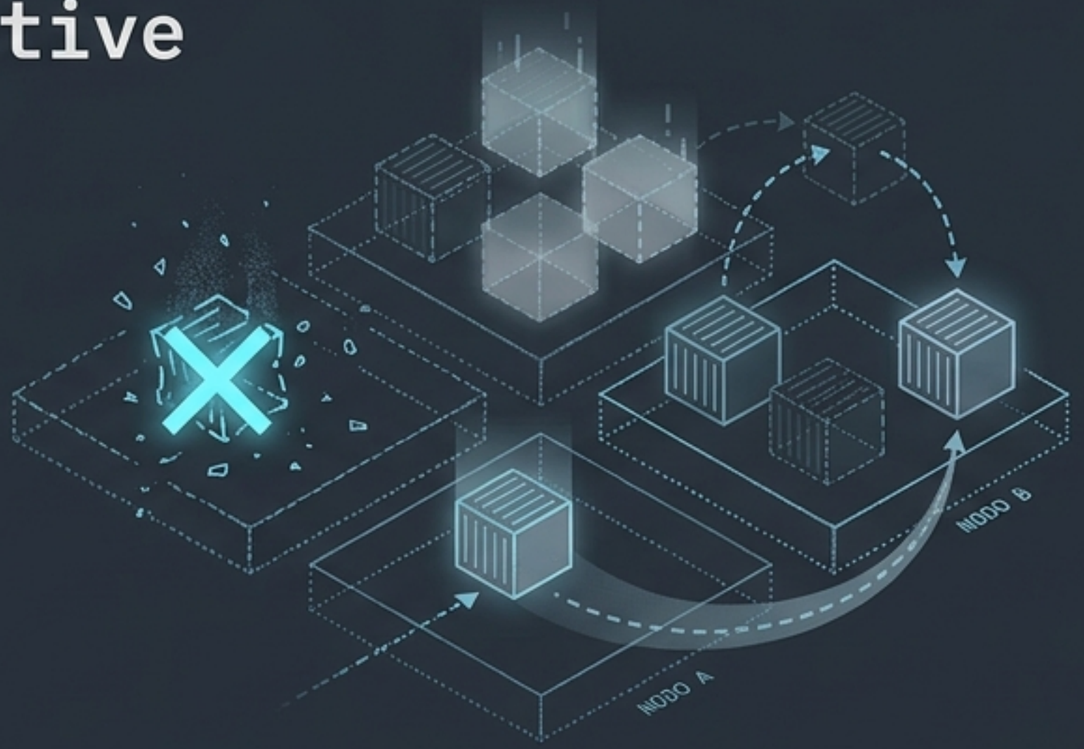
Continuidad Criptográfica Distribuida

Desacoplando Runtime e Identidad en SkyDefended InfraApp v1.2

Autor: Ismael Cruz Casasola
Fecha: 2026-05-22
Clase: Arquitectura Core / Modelo Criptográfico Distribuido

La Paradoja Cloud-Native



Modelo Clásico

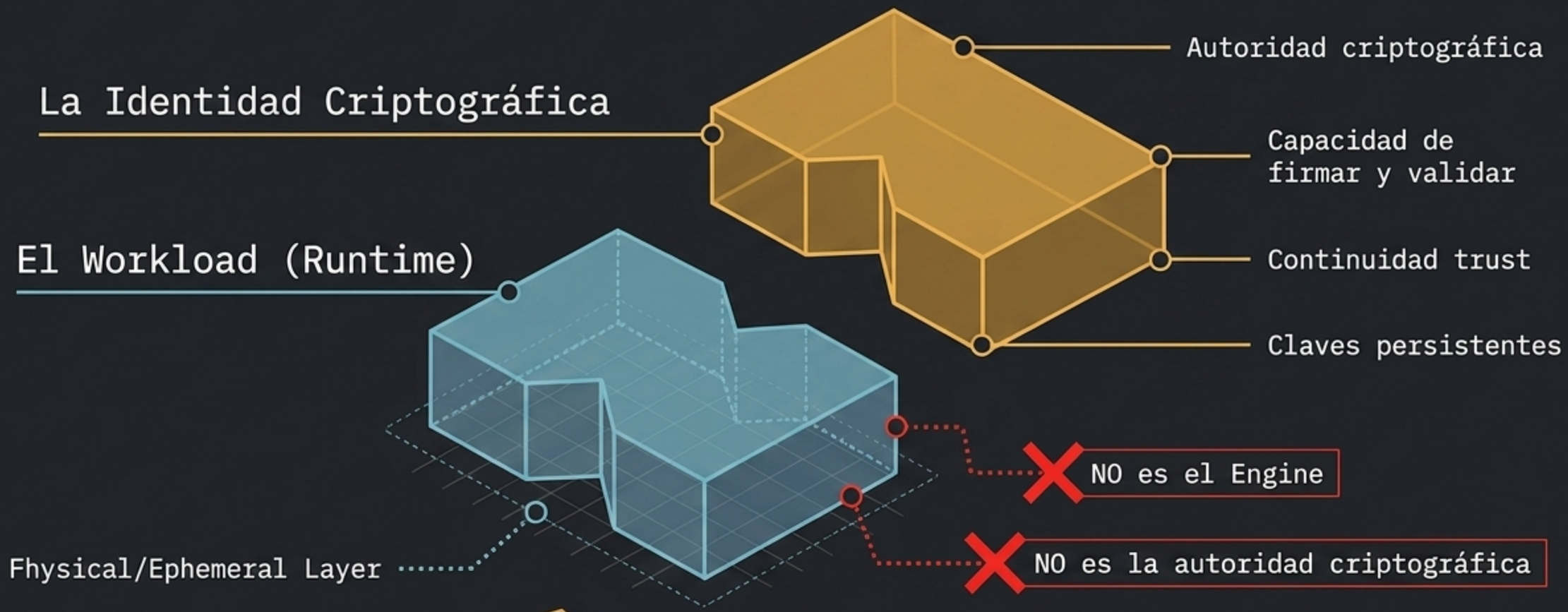
La aplicación vive años en el mismo servidor. Las claves viven localmente. `Runtime` e identidad se mezclan.

Infraestructura Efímera

El contenedor no es persistente. El `nodo` no es estable. El `workload` puede reiniciarse, migrarse o desaparecer sin previo aviso.

Pregunta estructural: Si el `contenedor` muere repentinamente, ¿dónde vive la identidad?

El workload hospeda la identidad, pero no es la identidad



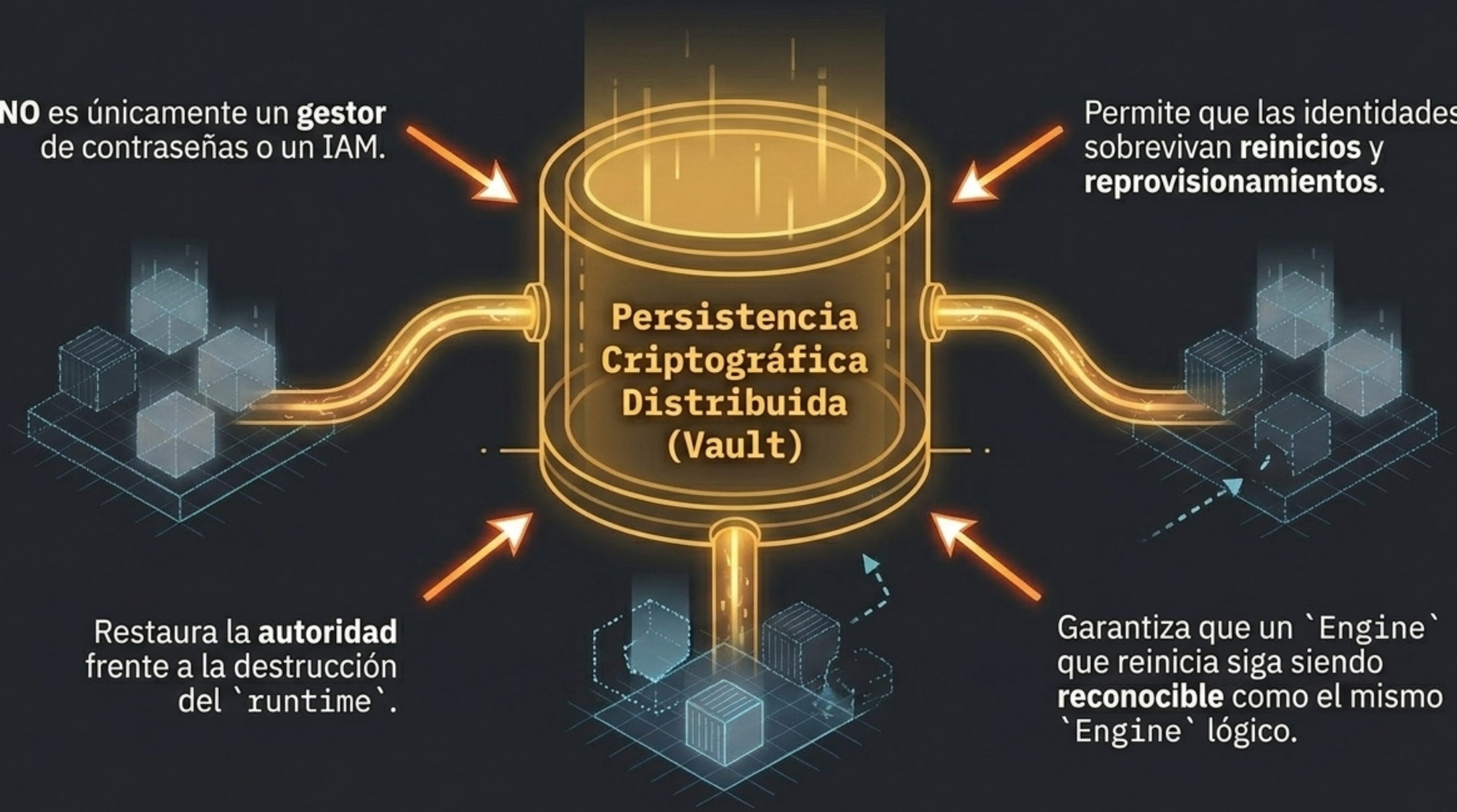
runtime ≠ identidad

La identidad criptográfica constituye el núcleo real del sistema. Sin continuidad criptográfica, los JWT quedarían invalidados y las federaciones desaparecerían.

El sistema de custodia como capa de persistencia criptográfica

NO es únicamente un **gestor** de contraseñas o un IAM.

Permite que las identidades sobrevivan **reinicios** y **reprovisionamientos.**



Restaura la **autoridad** frente a la destrucción del `runtime`.

Garantiza que un `Engine` que reinicia siga siendo **reconocible** como el mismo `Engine` lógico.

Clasificación del impacto ante el compromiso del asset

Tier-0: Trust Raíz

Impacto: Colapso sistémico global. Forja de identidad y pérdida de trust.

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

Tier-1: Crítico Operacional

Impacto: Compromete un Engine concreto. Operacionalmente grave, pero recuperable mediante rotación/revocación.

```
INTERNAL_API_KEY
STATE_SIGNING_KEY
Identidades RSA por
Engine
```

Tier-2: Impacto de Datos

Impacto: Acceso a datos. Sin colapso del trust global de la plataforma.

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

Espectro de exposición del material criptográfico

Nivel KV (Key-Value)

Vive temporalmente en RAM local del `workload`. Mayor exposición. Menor latencia.

Nivel Transit

La clave privada nunca abandona la custodia. Firma y cifrado remoto. El `workload` delega operaciones.

Nivel HSM Remote-Sign

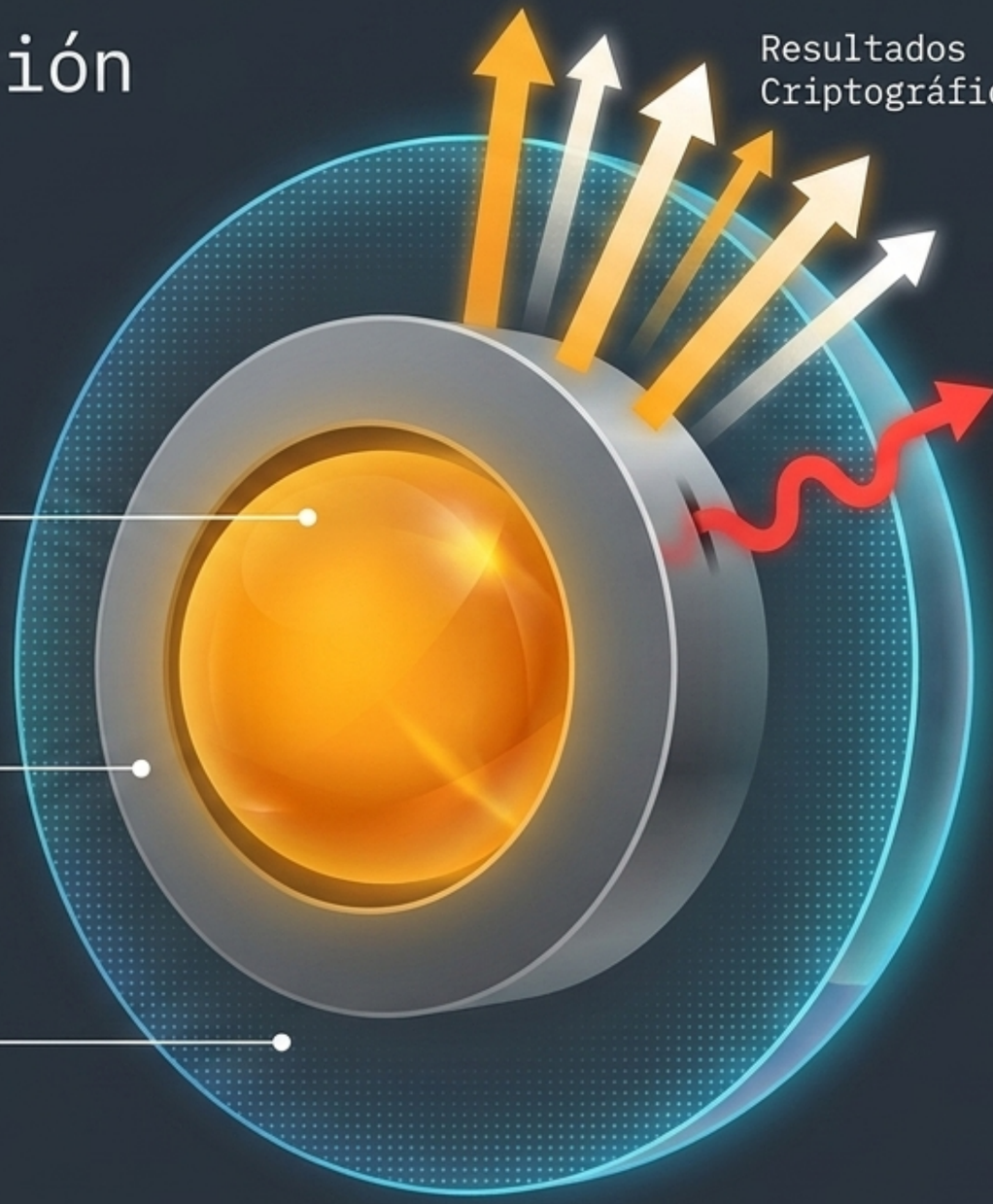
La clave nunca abandona el `hardware`. Inexportable. Frontera criptográfica basada en silicio. Máxima protección.

Alineación estratégica de resiliencia: Impacto vs. Exposición

	Nivel KV	Nivel Transit	Nivel HSM
Tier-0			Alineación Objetivo
Tier-1		Alineación Objetivo	
Tier-2	Alineación Objetivo		

La evolución del modelo NO busca 'todo a HSM'. Conviven KV, Transit y HSM según latencia, frecuencia y coste. La alineación es una decisión de resiliencia operacional asset-por-asset.

Fronteras de contención criptográfica

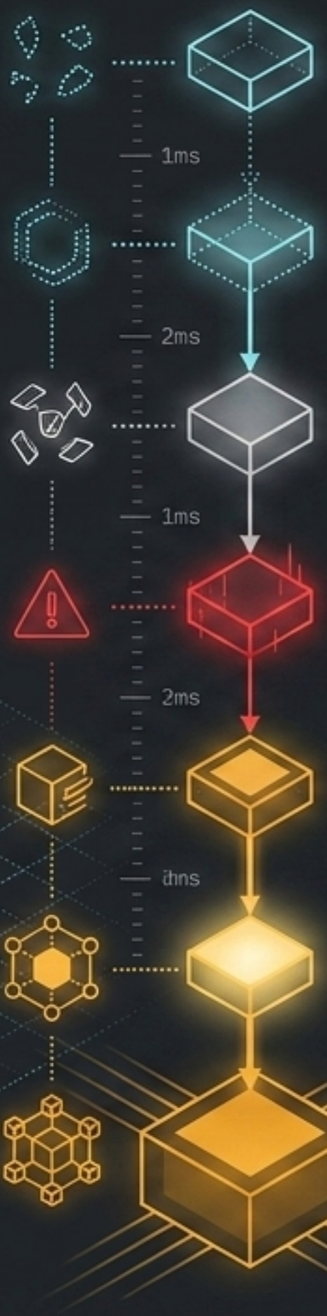


Hardware HSM
La clave privada nace y muere aquí. (Nivel HSM)

Vault Transit Engine
Peticiones de firma/cifrado remoto. La clave no sale (Nivel Transit)

Workload RAM / KV
La clave se materializa temporalmente en memoria local (Nivel KV)

Flujo de Bootstrap: Adquisición de identidad temporal



1. Arranque
Pod arranca como cascarón vacío.

2. K8s JWT
Kube-apiserver inyecta credencial efímera del ServiceAccount.

3. Auth (Vault)
Validación delegada vía TokenReview API.

4. Token Efímero
Vault emite Token Efímero (TTL ≤ 1 hora).

5. Recuperación
Petición de la Identidad Persistente (Transit/KV).

6. Contexto en RAM
Carga del material en memoria runtime.

7. Operación Trust
Inicio de la validación local distribuida (JWKS/Bundles).

Separación lógica: Autenticación efímera vs. Autoridad continua

Autenticación (Token Efímero de K8s/Vault)

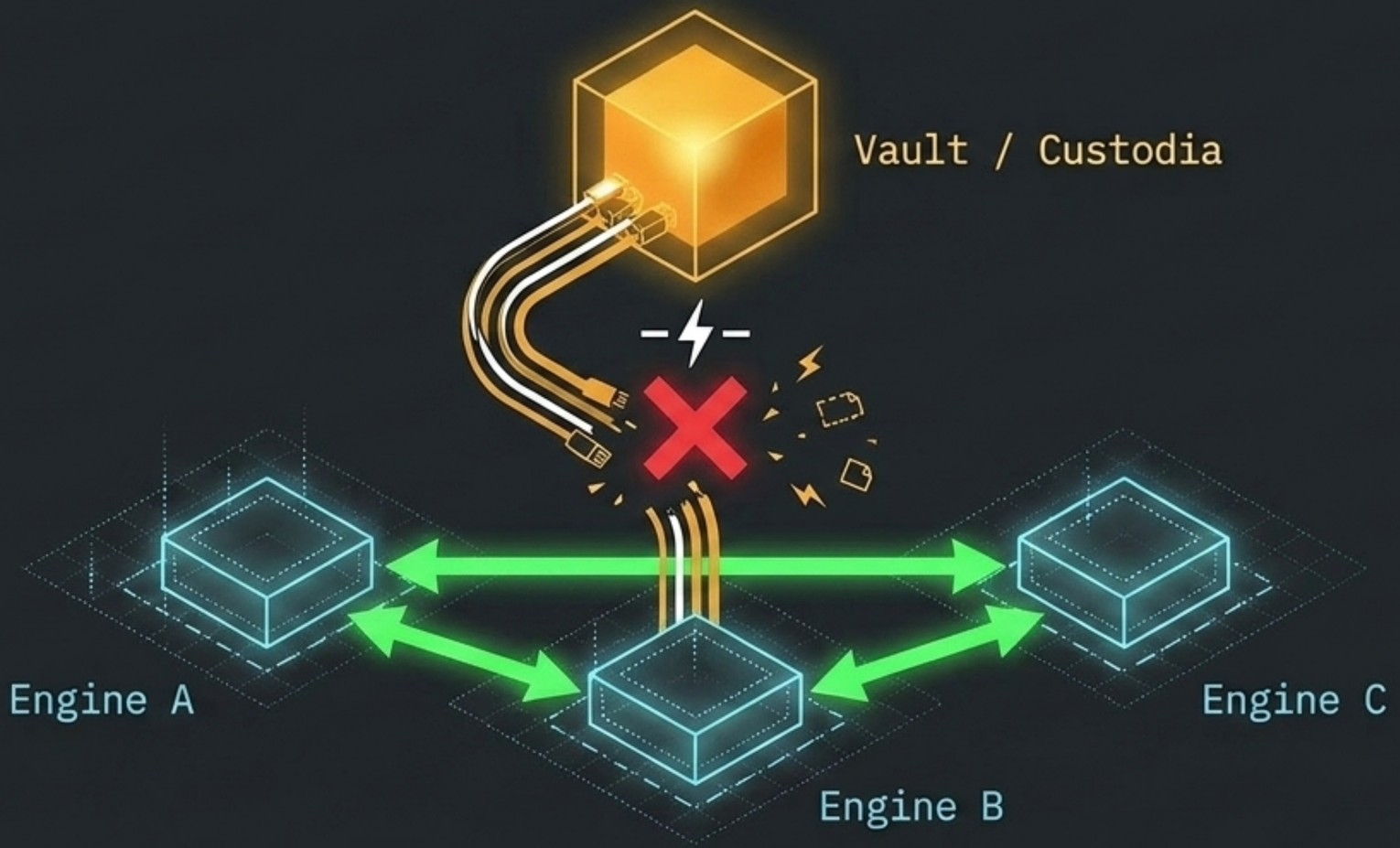
La `llave del hotel`. Permite el proceso de `Bootstrap`. Rota, expira (TTL ≤ 1h) y NO representa la identidad real del sistema.

≠

Identidad (Autoridad Criptográfica Persistente)

El ADN del `Engine`. Permite la continuidad `trust`. Sobrevive a la destrucción del `runtime`. Es la identidad que el resto de la plataforma reconoce.

Resiliencia operacional fuera del hot path continuo

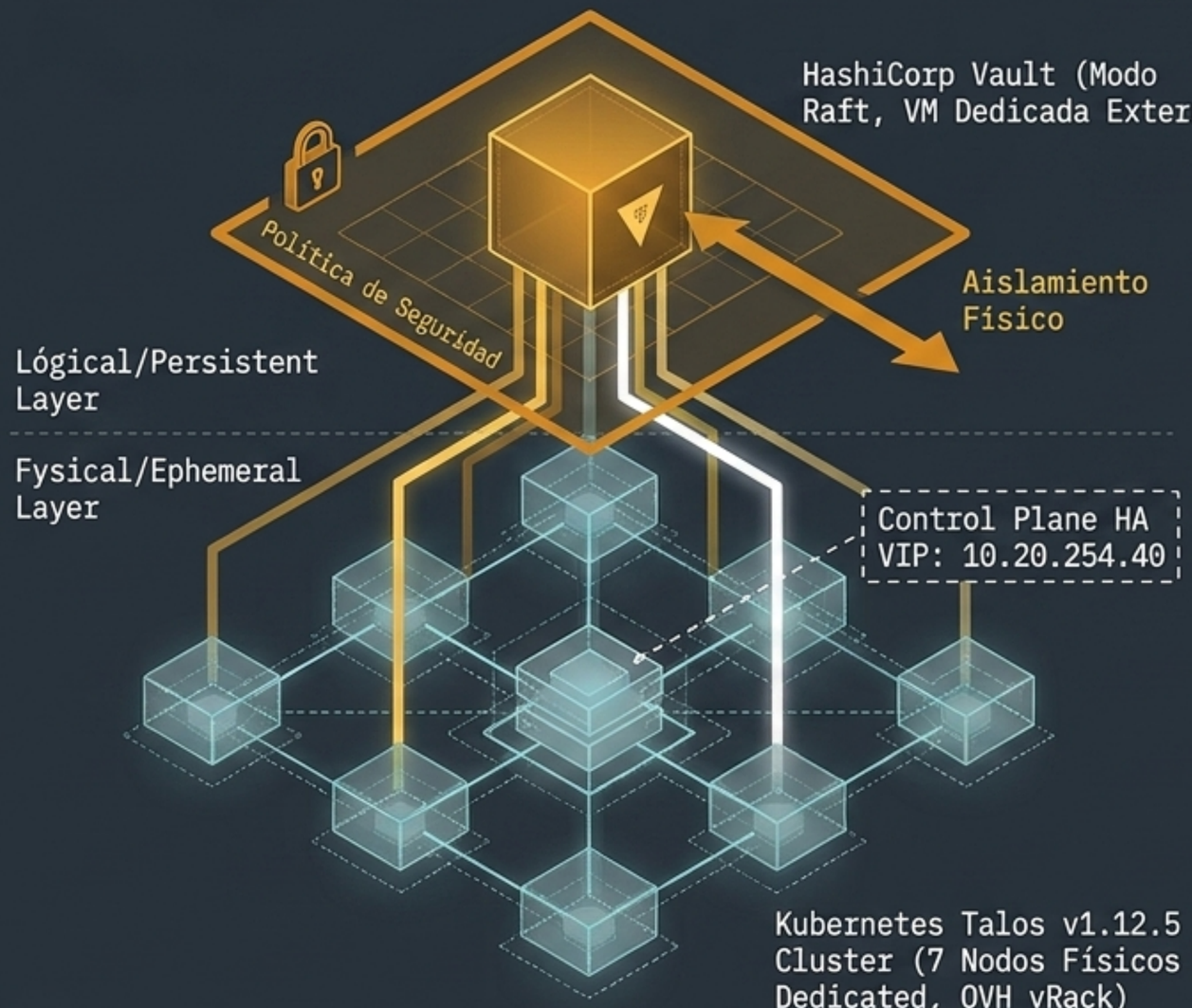


Si la custodia cae temporalmente, la plataforma mantiene la operación. La validación de los JWTs ocurre localmente mediante JWKS, bundles en memoria RAM y trust distribuido. Vault solo es crítico para bootstrap, recovery, y rotación.

Supervivencia estructural: Engine Lógico vs. Workload Físico

Múltiples workloads físicos efímeros representan un único Engine lógico estable. La red sigue reconociendo al mismo actor criptográfico.

Implementación física del stack SkyDefended InfraApp 2026



Orquestación

- Kubernetes Talos v1.12.5 sobre 7 nodos físicos dedicados OVH (vRack interno).
- Control Plane HA VIP: 10.20.254.40.

Custodia Externa

- HashiCorp Vault en modo Raft (VM dedicada externa).
- Aislamiento físico: el compromiso del clúster no compromete la bóveda.

Integridad y Firma

- Cosign keyless OIDC (Fulcio + Rekor) integrado en CI.
- Pinning por digest (@sha256).
- Admisión vía Kyverno ClusterPolicy (verify-skydefended-image-signatures en modo Audit).

Estado operativo de la custodia criptográfica y roadmap

SECCIÓN 1: DESPLEGADO Y OPERATIVO

On

Nivel KV

On

Vault KV v2. Sincronización a cargas de trabajo vía **External Secrets Operator** (ESO) para **Tier-2** (DB, OIDC, SIEM).

Nivel Transit

On

Vault Transit (**transit/keys/...**). Encriptación de **mfaSecret** usando **AES-256-GCM** (Primer uso generalizado del modelo inexportable).

SECCIÓN 2: EN PROGRESO / OBJETIVO ARQUITECTÓNICO

- **Azure Managed HSM** (FIPS 140-3 Nivel 3 single-tenant) Iune 2 Nivel 3 single-tenant) destinado al **Tier-0**.
- Despliegue de **Vault HA** (follower local).
- **SPIFFE/SPIRE** para **workload identity** nativa.

La infraestructura desaparecerá.
Los runtimes se destruirán y reconstruirán continuamente.
Pero la autoridad criptográfica y el trust distribuido permanecerán intactos.