Zero-Trust Sovereign AI:

Unified Identity for Workloads with HW-rooted Verifiable Geofencing & Residency Proofs

Author: Ramki Krishnan (Security expert, Advisor at Vishanti)

- References
    - IETF Draft (Telefonica, Orange, Red Hat, Oracle, Aryaka)
    - LF Edge Project (Vishanti, Red Hat et al)
    - Keylime Open-Source Project
    - Spire Open-Source Project

Sovereign cloud data residency requirements

Technical and regulatory challenges

- **Data protection regulations**: EUDR, US HIPPA, PCI DSS, Local legal mandates (Saudi Arabia, China, India …)
- **Data protection in all its lifecycle management**: creation, process, usage, storage, destruction.
- **Data protection in all its status**: in transit, in use, at rest.

Data residency technical requirements

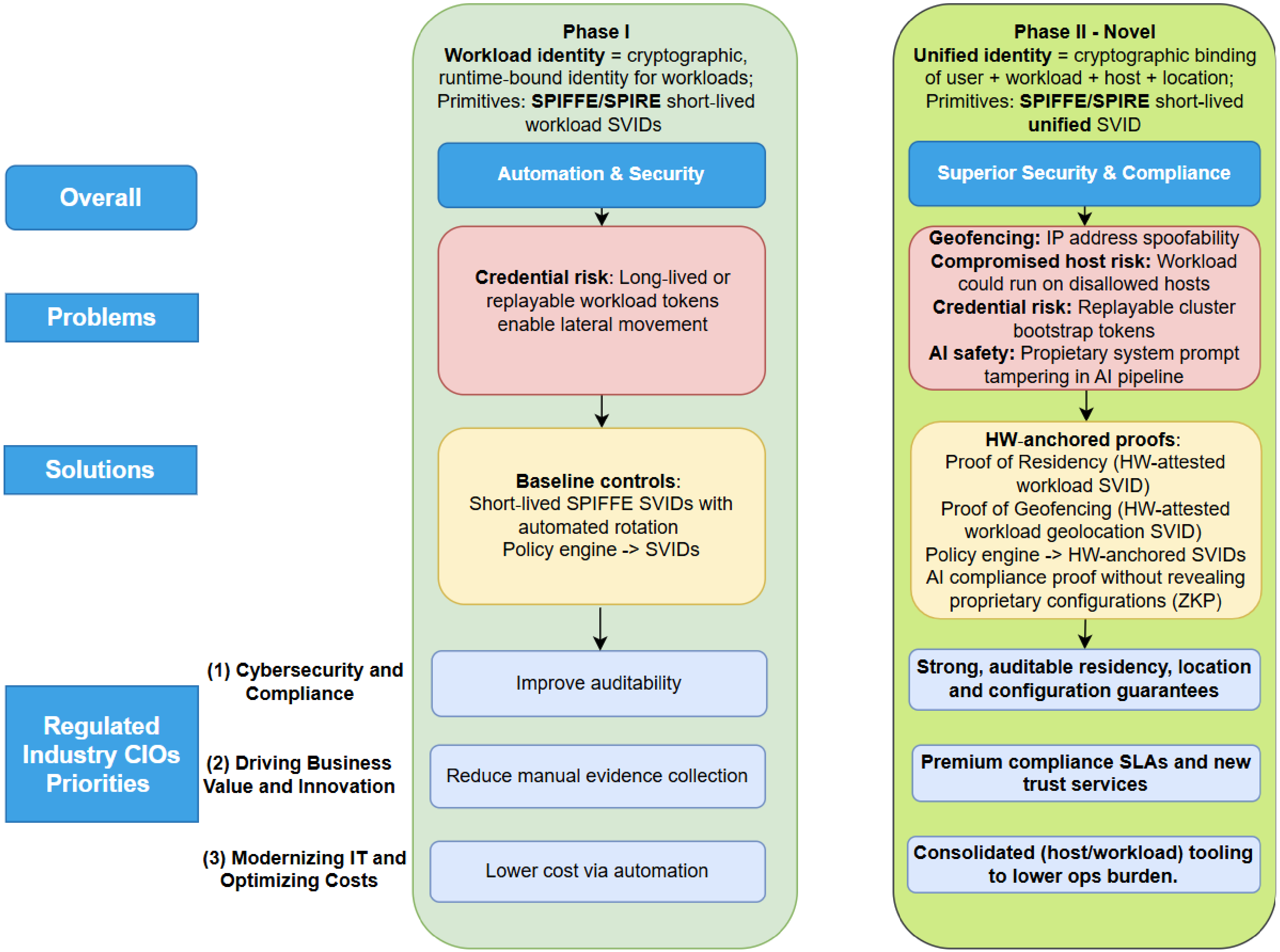
- **Host affinity** requirement
  - Data storage and processing must be tied to specific hosts.
- **Geolocation affinity** requirement
  - Data must be stored/processed only in a defined geographic region.
- **Host geolocation affinity (aka geofencing)** requirement
  - Host is bound to defined geographic region(s)

Zero-Trust Sovereign AI:

Unified Identity for Workloads with HW-rooted Verifiable Geofencing & Residency Proofs

Summary and Business Value

Industry Evolution to Novel Zero-Trust Sovereign AI Solution (Phase II)



Phase II public references:

- IETF Draft (Telefonica, Orange, Red Hat, Oracle, Aryaka);
- LF Edge Project (Vishanti, Red Hat et al)

Phase II implementation progress:

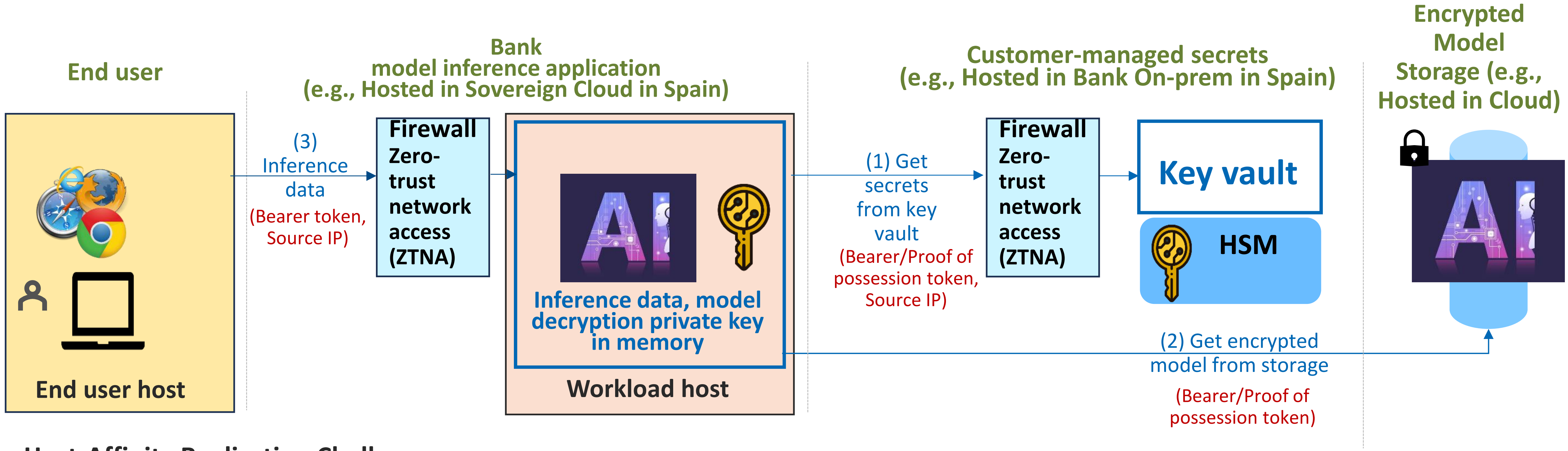
- **Joint Hybrid Cloud PoC** - Telefonica, Red Hat and Vishanti

Zero-Trust Sovereign AI:

Unified Identity for Workloads with HW-rooted Verifiable Geofencing & Residency Proofs

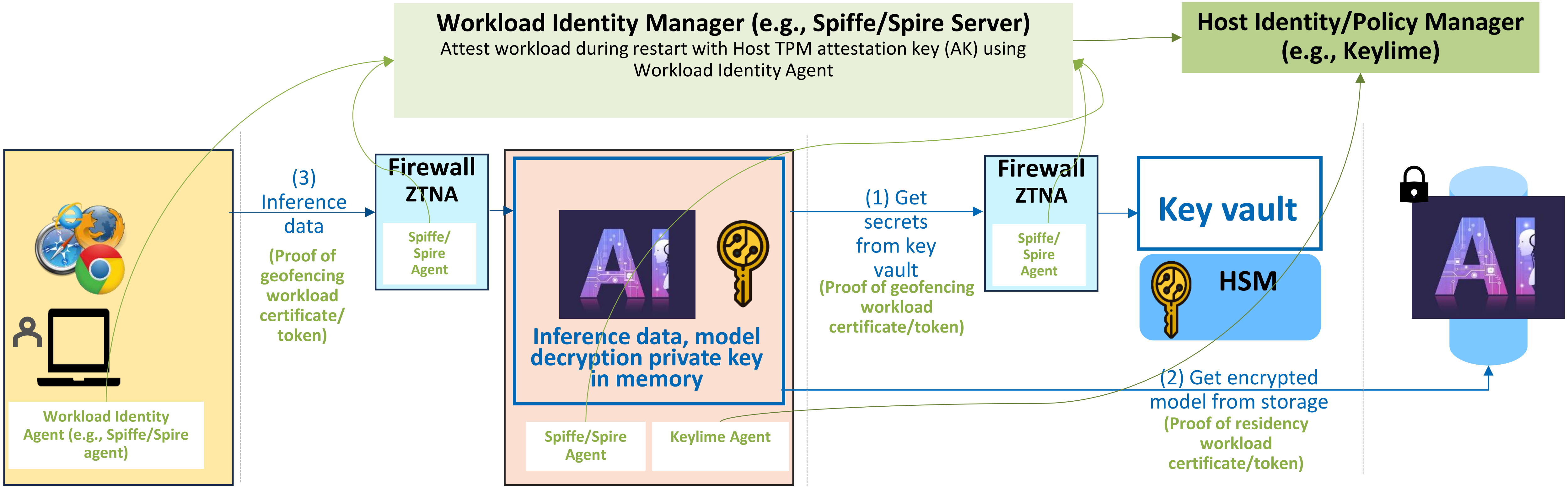
Technical Details

The Problem: A Fragile and Non-Verifiable Security Model



- **Host-Affinity Realization Challenges**
  - **Bearer tokens** (RFC 6750) protect inference apps, secret stores, and model repositories—but if stolen (via breach of an identity provider like Okta, via breach of Metadata server - Kubernetes bootstrap token, spire bootstrap token), they can be replayed.
  - **Proof-of-Possession tokens** (RFC 7800) bind a private key to the token. However, Container orchestration abuse (Mitre T1611), RBAC and policy abuse (T1059 & T1203), Supply chain compromise (T1584) can still undermine them by:
    - Allowing valid workloads to execute in disallowed hosts/regions
- **Geolocation-Affinity Realization Challenges**
  - **IP-based geofencing** (firewall rules that check source IP) offers only weak location guarantees. Attackers easily bypass it using VPNs, proxies, or IP spoofing.
- **Static and Isolated Security Challenges**
  - GPU health, utilization and host integrity are typically checked by isolated, non-verifiable monitoring systems -- creates a critical gap where a compromised host can easily feed false data to the monitoring system.

The Solution: A Zero-Trust, HW-Rooted, Unified, Extensible & Verifiable Identity



Address Bearer/Proof of possession token issue by Proof of Residency (PoR)

- Cryptographically bind (vs convention & configuration) Workload identity (executable code hash etc.) + Approved host platform hardware identity (TPM PKI key etc.)/platform policy (Linux kernel version etc.) to generate a PoR workload certificate/token.

Address Bearer/Proof of possession token and Source IP issue by Proof of Geofencing (PoG)

- Cryptographically bind PoR + Approved host platform location hardware identity (GNSS or mobile sensor hardware/firmware version) to generate a PoG workload certificate/token.

Dynamic Hardware Integrity

- Real-time data about the host's health, including GPU status (e.g., temperature and utilization), is collected by specialized plugins and included in the attestation process.

Three market entry points for Zero-Trust Sovereign AI adoption

Case	Description	Primary Targets	Regulatory / Market Drivers	Key Differentiators
1. Cross-Layer Trust Gap - Bearer Token Removal	Remove static bearer tokens (e.g., SPIRE bootstrap token) due to replay and compromise risks -- create hardware-anchored trust primitive to prove workload provenance at runtime.	Multi-cloud enterprises, hybrid edge operators, regulated industries with high supply chain risk	SLSA, NIST 800-204D, Zero-Trust mandates, runtime integrity requirements	Replaces fragile bearer tokens with hardware-rooted Proof of Residency (PoR) and Proof of Geofencing (PoG) -- cryptographically binding workload code hash, host integrity, GPU health, and geolocation into a unified SVID verifiable across all layers
2. Sovereign Cloud – Primary Generic Use Case	Enterprise hybrid cloud differentiation via *verifiable everything* (geolocation, GPU status, telemetry) as a premium trust service	EU CSPs, edge CSPs, edge enablers	EU Data Act, GDPR, AI Act, customer trust in regulated sectors	Hardware-rooted proofs for location, hardware state, and metrics; enables premium SLA tiers and compliance-as-a-service
3. Sovereign Cloud – Geographic / Sector-Specific Data Residency	Region-bound workload execution and data storage for compliance and trust	National CSPs, regulated enterprises (healthcare, finance, retail)	China PIPL, U.S. DOJ foreign access rules, HIPAA, PCI DSS, sectoral mandates	Cryptographic proof of residency, jurisdiction-aware workload orchestration, compliant cross-border analytics

The Solution Value: Technical and Operational Benefits

Technical Value

- **From Isolated Data to a Verifiable Credential:** Instead of relying on a human or a centralized system to check GPU health and location, the architecture automates the process and provides a **verifiable, cryptographically-signed claim**. This is the key difference between a simple monitoring solution and a robust security control.

- **Real-time, Holistic Policy Enforcement:** The unified SVID allows for highly granular, context-aware policies at the service mesh layer. A scheduler can now make decisions based on the combined information from the SVID, ensuring a workload is placed on a healthy GPU that is also on a trusted host in the correct location.

Operational Value

- **Improved Compliance and Resilience:** The cryptographically verifiable **Proof of Geofencing** provides irrefutable evidence that data is processed in a compliant location, which is crucial for meeting regulatory requirements – reduce compliance audit prep time by ~30%. This also provides a solution for physical attacks, such as unauthorized move of host.

- **Complete Automation:** The entire process, from host attestation to policy enforcement, is fully automated. This drastically reduces the manual overhead and human error associated with managing security at scale.

- **Simplified Auditing:** The attestation reports and SVID claims provide a comprehensive, verifiable record of a workload's identity and operational context, simplifying security audits and providing a clear, auditable trail.

Spiffe/Spire agent unified svid (workload/host identity) architectural flow

Workload Identity Manager - Spire Server

Verify workload attestation (local) and host attestation from keylime verifier – allocate unified svid (workload version, host cpu/gpu integrity, geolocation)

Host Hardware Identity - Keylime registrar DB

Host TPM AK -> Host TPM Endorsement Key (EK) (Note: security is in TPM AK/EK attestation)

Host TPM EK -> GNSS sensor HW id; Mobile sensor HW id, IMEI/IMSI/ph. number

Composite Geolocation Manager

Background job : Call Mobile SP API periodically w/ Host mobile sensor phone number for geolocation

Benefit: Higher trust, Mobile location outside of mobile device

Host (bare metal/VM) verifier – Keylime verifier

TPM-based measured boot checks, runtime IMA policy and other policy validation – ensure host meets policy constraints

Host Proximity Manager

Host Geolocation sensor Composition Manager

(8) Verify workload attestation from spire agent

(9) Verify host attestation from keylime agent (proxied through spire agent)

Control Plane

(7) Share host attestation w/ nonce from keylime agent

(1) Get nonce from spire server

(10) Fetch unified spire agent svid from spire server (includes workload and host attestation)

Spire Agent

(2) Certify Spire TPM App key w/ nonce by TPM AK (allocate key during agent start)

Spire agent tpm plugin

(3) Share nonce from spire server w/ spire tpm app key cert and get TPM signed quote w/ geolocation and gpu status

(4) Get current geolocation

Keylime Agent geolocation plugin

(6) Get host attestation w/ nonce from keylime agent **(includes host/gpu integrity, geolocation)**

(5) Get current gpu status

Keylime Agent host proximity plugin

Keylime Agent gpu integrity plugin

Keylime Agent

5G/LTE Modem with antenna & SIM slot or GNSS

Server Location Anchor Host

Data Plane

Technical highlights:

- **Spire agent unified Spiffe ID (SVID) - workload version, host cpu/gpu integrity, geolocation**
  - ***mTLS private key is in TPM HW root of trust***
- **Address Bearer/Proof of possession token issue by Proof of Residency (PoR)**
- **Address Bearer/Proof of possession token and Source IP issue by Proof of Geofencing (PoG)**

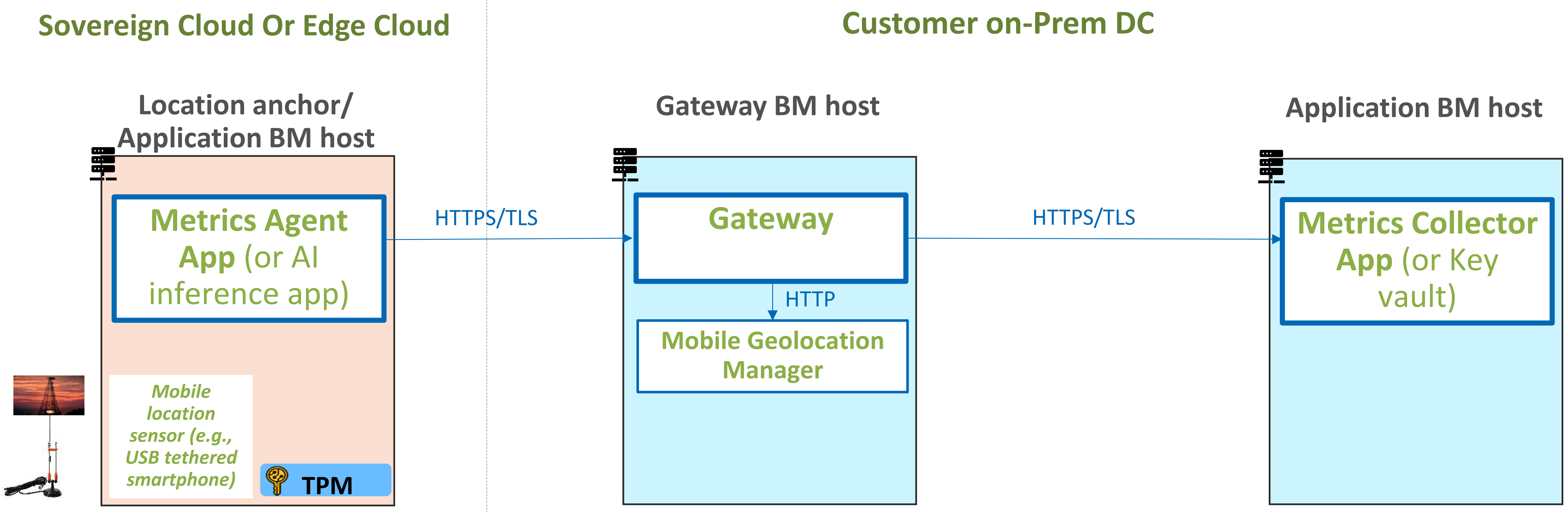
Note: Keylime agent host proximity plugin is needed for hosts without location sensor (can use attested PTP)

Zero-Trust Sovereign AI:

Unified Identity for Workloads with HW-rooted Verifiable Geofencing & Residency Proofs

Implementation Details – First Iteration

Implementation Details – First Iteration



Agent flow summary

- Get nonce from collector
- Sign payload with nonce using key in TPM
- Sign Workload-Geo-ID HTTP header using key in TPM

Gateway Enforcement (First Layer – IETF Workload-Geo-ID header based):

- **Geolocation Policy**: Rejects if location doesn't match allowlist
- **Public Key Hash**: Rejects unregistered agents
- **Cryptographic Signature Verification**: Verifies signature using TPM2-based verification
- **Timestamp Proximity**: Rejects if agent time is too far from gateway time

Collector Enforcement (Second Layer – Payload based):

- **Nonce Validity**: Existence, expiration, reuse prevention
- **Payload Signature**: Full cryptographic verification

Reference: https://github.com/lfedgeai/AegisEdgeAI/blob/main/zero-trust/README.md