Zero-Trust Sovereign AI – WIMSE impact:

Verifiable Geofencing & Residency Proofs for Cybersecure Workloads

- **Reference**: WIMSE https://github.com/nedmsmith/draft-klspa-wimse-verifiable-geo-fence/blob/main/draft-lkspa-wimse-verifiable-geo-fence.md
- **Authors**: Ramki (Presenter), Ned, Diego, Prasad, Srini

Agenda

- Sovereign cloud data residency requirements
- Key industry problems and solution summary with “Sovereign Cloud AI inferencing” as an example
- Summary of key protocol changes and suggested next steps

Sovereign cloud data residency requirements

Technical and regulatory challenges

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

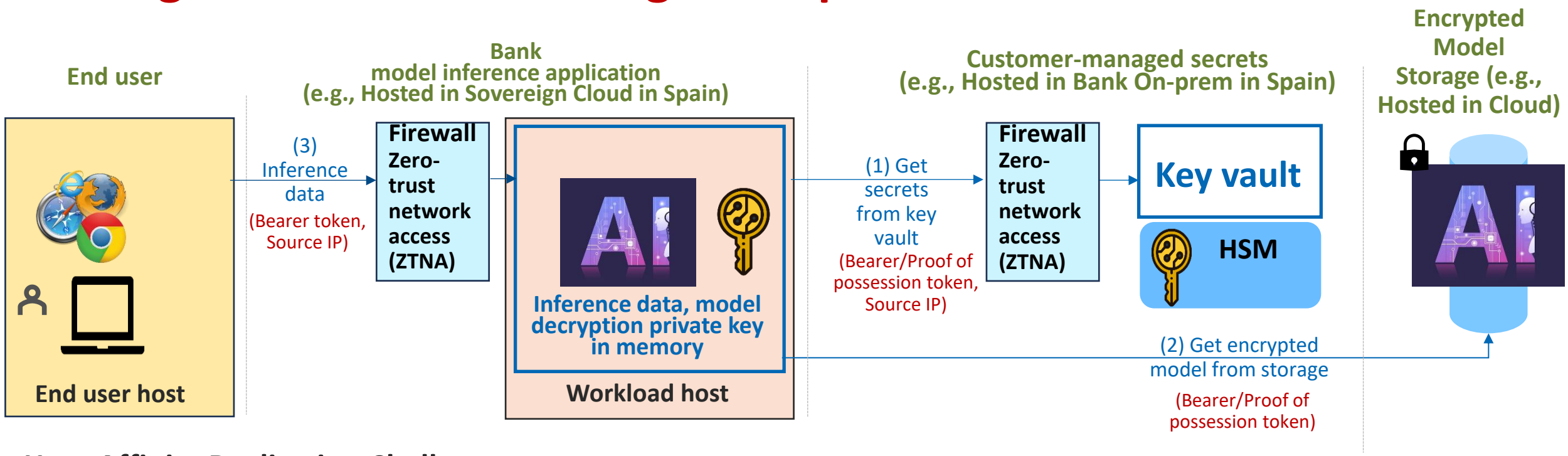
Data residency technical requirements

- **Host affinity** requirement
  - Data must storage and processing must be tied to specific hosts.

- **Geolocation affinity** requirement
  - Data must be stored/processed only in a defined geographic region.

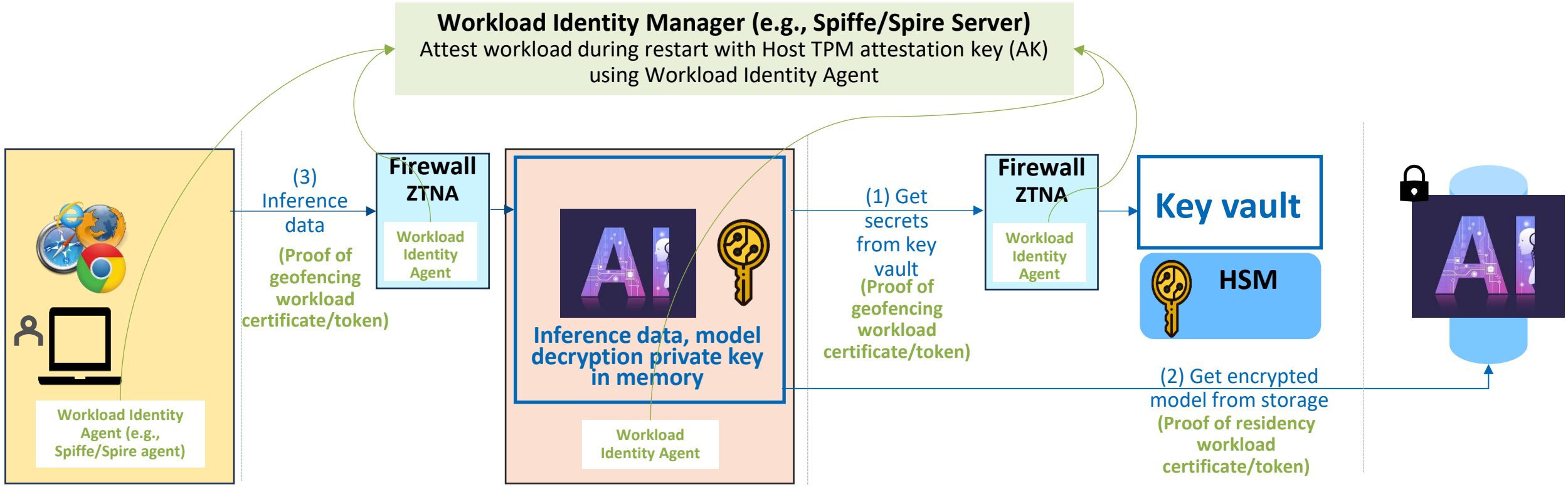
- **Host geolocation affinity (aka geofencing)** requirement
  - Host is bound to defined geographic region(s)

Sovereign Cloud AI inferencing: Example use case & Problem



- **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), they can be replayed from any workload, host, or region.
  - **Proof-of-Possession tokens** (RFC 7800) bind a private key to the token. However, workload orchestrator security misconfigurations (OWASP security misconfiguration) can still undermine them by:
    - Exposing that private key to unauthorized workloads
    - Allowing valid workloads to execute in disallowed 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.

Sovereign Cloud AI inferencing: Example use case & Solution



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, hardware/firmware version) 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.

Summary of key protocol changes and suggested next steps

Summary of key protocol changes (non exhaustive)

1. **Proof of Residency (PoR)**
    1. Access token
        1. **Change - Naming - PoR instead of PoP**
        2. No change – other fields
    2. **Demonstrating Proof of Residency (DPoR) JWT**
        1. Same as DPoP JWT
    3. Resource server verification
        1. **Change – Verify that DPoR public key is attested by an approved host TPM; Verify that workload identity (e.g., spiffe/spire) certificate chain has the DPoR public key, the TPM Attestation public key (AK) and the TPM Endorsement public key (EK).**

2. **Proof of Geofencing (PoG) = PoR + the following described below**
    1. **Demonstrating Proof of Geofencing (DPoG) JWT - workload obtains this through workload identity (e.g., spiffe/spire) agent**
        1. **Location fields – the workload identity agent uses the right location type based on workload policy**
            1. Server workload ID (or website URL) - relevant for thin clients
            2. Location type (e.g., precise, approximated, geographic region based)
            3. Location coordinates (e.g., latitude/longitude, city/state/country)
            4. Location quality (e.g., GNSS, mobile network, Wi-Fi, IP address)
        2. **Signing public key - TPM Attestation key (AK) managed by workload identity agent**
        3. **Signature**
    2. Resource server verification
        1. **Verify that DPoG public key is attested by an approved host TPM; Verify that workload identity agent certificate chain has the DPoG public key and the TPM Endorsement public key (EK).**

Suggested next steps

- Working Group request
  - Please review draft

- Architecture draft - https://github.com/nedmsmith/draft-klspa-wimse-verifiable-geo-fence/blob/main/draft-lkspa-wimse-verifiable-geo-fence.md
  - Progress as informational draft

- Other drafts - DPoR, DPoG etc.
  - Progress as standards track