Workload Identity in Multi System Environments
Internet-Draft
Intended status: Informational
Expires: 21 April 2026

R. Krishnan
Vishanti Systems, Inc.
N. Smith
Intel
D. Lopez
Telefonica
A. Prasad
Oracle
S. Addepalli
Aryaka

18 October 2025

Zero-Trust Sovereign AI: Verifiable Geofencing & Residency Proofs for Cybersecure Workloads

draft-lkspa-wimse-verifiable-geo-fence-03

Abstract

Modern cloud and distributed environments face significant risks from stolen bearer tokens, protocol replay, and trust gaps in transit. This document presents a framework for modernizing workload security through cryptographically verifiable geofencing, proof-of-possession, and protocol-aware residency enforcement.

By binding workload identity to both geographic and host attributes, and supplementing bearer tokens with verifiable, location- and host-bound claims, the framework addresses the challenges of bearer token theft, proof-of-possession and trust-in-transit for all networking protocols. Leveraging trusted hardware, attestation protocols, and geolocation services, this approach ensures that only authorized workloads in approved locations and environments can access sensitive data or services, even in the presence of advanced threats.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at https://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on 21 April 2026.

Copyright Notice

Table of Contents

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

Acronyms used in this document:

* *TPM*: Trusted Platform Module

* *GNSS*: Global Navigation Satellite System

* *IMEI*: International Mobile Equipment Identity

* *IMSI*: International Mobile Subscriber Identity

* *PCR*: Platform Configuration Register

* *MDM*: Mobile Device Management

* *IPSEC*: Internet Protocol Security

Key Terms:

- *Data Residency* Technical and Legal Challenges Ensuring compliance with global and local data protection regulations and mandates (e.g., EU GDPR, US HIPAA, PCI DSS, and jurisdiction-specific laws; see Appendix for public references on strict data residency rules). Strict data residency rules require that specific categories of data must be stored and processed exclusively within designated geographic boundaries. Enforcing these mandates relies on a combination of trusted computing, host-affinity, geolocation-affinity, and geofencing—each defined below.
  - *Data Residency Host-Affinity Requirement*

  Data must remain bound to explicitly trusted computing environments or hosts, governing where storage and processing occur. - *Data Residency Geolocation-Affinity Requirement*

  Data must not be transferred beyond defined geographic regions. All storage and computation must remain within the specified boundaries. - *Data Residency Host Geolocation Affinity (aka Geofencing)*

  A compound enforcement mechanism requiring that data and workloads are executed only on authorized hosts located within the approved geographic regions. - *Workload Identity Agent (WIA)*

  SPIRE agent on each host, with TPM plugin to issue X.509 SVIDs and sign requests. - *Location Anchor Host (LAH)*

  Host with a trusted GNSS/5G modem attached to its TPM endorsement key. *Composite Geolocation*

  Fused location estimate from local GNSS plus mobile-API data. - *Proof-Of-Residency (POR)*

  Cryptographic proof that a workload is executing within approved geographic and host boundaries.

As organizations increasingly adopt cloud and distributed computing, the need to enforce data residency, geolocation affinity, and host affinity has become critical for regulatory compliance and risk management. Traditional approaches to geographic and host enforcement rely on trust in infrastructure providers or network-based controls, which are insufficient in adversarial or multi-tenant environments.

Modern workload security faces new challenges from stolen bearer tokens, protocol replay, and the lack of trust in transit. Attackers can exploit bearer tokens from unauthorized hosts or locations, bypassing traditional controls.

This document introduces a framework for modernizing workload security by enabling cryptographically verifiable geofencing, proof-of-possession, and protocol-aware residency enforcement. The solution cryptographically binds workload identity to both platform and geographic attributes, supplementing bearer tokens with signed, verifiable claims about workload residency and location.

This enables enforcement of data residency, geolocation affinity, and host affinity policies, even in adversarial or multi-tenant environments, and directly addresses the limitations of bearer tokens, proof-of-possession, IPSEC, and trust-in-transit.

3. 3. Use Cases

Data residency use cases can be divided into three categories: (1) server-centric location, (2) user-centric location, and (3) regulatory compliance.

3.1. 3.1. Category 1: Server-centric Location

Enterprises (e.g., healthcare, banking) need cryptographic proof of a trustworthy geographic boundary (i.e., region, zone, country, state, etc.) for cloud-facing workloads.

3.1.1. 3.1.1. Server workload <-> Server workload - General:

Enterprises handling sensitive data rely on dedicated cloud hosts (e.g., EU sovereign cloud providers) that ensure compliance with data residency laws, while also ensuring appropriate levels of service (e.g., high availability). To meet data residency legal requirements, enterprises need to verify that workload data is processed by hosts within a geographic boundary and that workload data is only transmitted between specified geographic boundaries.

Example Sovereign Cloud AI Inferencing use case depicting the key security and compliance challenges: Figure -- Sovereign Cloud AI Inferencing (https://github.com/nedmsmith/draft-klspa-wimse-verifiable-geo-fence/blob/ramki/pictures/challenges-sovereign-cloud-ai-inferencing.svg)

3.1.2. 3.1.2. Server workload <-> Server workload - Agentic AI:

Enterprises need to ensure that the AI agent is located within a specific geographic boundary when downloading sensitive data or performing other sensitive operations. A secure AI agent, running on a trusted host with TPM-backed attestation, interacts with geolocation and geofencing services to obtain verifiable proof of its geographic boundary. The agent periodically collects location data from trusted sensors, obtains attested composite location from a geolocation service, and enforces geofence policies via a geofencing service. The resulting attested geofence proof is used to bind workload identity to both the host and its geographic location,

enabling secure, policy-driven execution of AI workloads and compliance with data residency requirements.

Figure -- Cybersecure and Compliant Agentic AI Workflow (https://github.com/nedmsmith/draft-klspa-wimse-verifiable-geo-fence/blob/main/pictures/secure-agentic-workflow.svg/)

3.1.3. 3.1.3. Server workload <-> Server workload - Federated AI:

In federated learning scenarios, multiple organizations collaborate to train machine learning models without sharing raw data. Each organization needs to ensure that its training data remains within a specific geographic boundary. This requires cryptographic proof that the training process is occurring on trusted hosts within the defined boundaries.

Example Federated Learning use case depicting the key security and compliance challenges: Figure -- Federated Learning (https://github.com/nedmsmith/draft-klspa-wimse-verifiable-geo-fence/blob/ramki/pictures/challenges-federated-learning.svg)

3.1.4. 3.1.4. User workload <-> Server workload:

Enterprises ensure that they are communicating with a server (e.g., cloud services) located within a specific geographic boundary.

3.2. 3.2. Category 2: User-centric Location

Enterprises need cryptographic proof of trustworthy geographic boundary for user-facing workloads.

* A server (or proxy) authenticates to clients using different TLS certificates, each signed by a different Certificate Authority (CA), based on the geographic boundaries of user workloads.

* Enterprise Customer Premise Equipment (CPE) provides on-premises computing that is a basis for defining geolocation boundaries. A telco network provides a means for communication between premises.

* Construction & Engineering of SaaS workloads can benefit from attested geographic boundary data from end-user devices to restrict access within specific geopolitical regions (e.g., California). Enabling per-user or group-level geofencing helps prevent fraudulent access originating outside the authorized area.

* Healthcare providers need to ensure that the host is located in a specific geographic boundary when downloading patient data or performing other sensitive operations.

* U.S. Presidential Executive Order (doj-cisa) compliance directs Cloud Service Provider (CSP) support personnel be located in restricted geographies (e.g., Venezuela, Iran, China, North Korea). However, those personnel should not be allowed to support U.S. customers. Geolocation enforcement can ensure policy compliance. See [doj-cisa].

3.3. 3.3. Category 3: Regulatory Compliance

Geographic boundary attestation helps satisfy data residency and data sovereignty requirements for regulatory compliance.

4. 4. Industry Gaps and Problem Statements

Modern cloud and distributed environments face significant risks from stolen bearer tokens, protocol replay, and trust gaps in transit. Current geofencing and location verification solutions face

significant challenges across different data states, location sources, and authentication mechanisms. This section outlines the key gaps and problems that this specification aims to address.

4.1. 4.1. Data Generation/at-Rest Challenges

Textual Geotags No standard for textual geotags (EXIF covers media only).

Attesting Geotags: Existing geotag formats are unsigned and forgeable via VPN/MITM attacks.

4.2. 4.2. Data-in-Use Challenges

4.2.1. 4.2.1. Authentication and Authorization Challenges

4.2.1.1. Bearer Token Vulnerabilities Bearer tokens are typically generated via user MFA and used to establish HTTP sessions. A malicious actor can steal a bearer token (e.g., from a still-valid HAR file uploaded to a support portal, as seen in the Okta attack) and present it to a server workload. The attacker may be in a forbidden location and on an unauthorized host (e.g., their own laptop). Current solution options for addressing bearer token issue and their challenges: * PoP Token: Not easy to establish trust between the presenter (client) and the token issuer. * PoP via Mutual TLS: Client certificates are generally not supported in browsers. MITM entities such as API gateways often terminate TLS connections. * Host TPMs for API call signature: Not scalable to sign every API call with a TPM key, as typical enterprise laptops/servers TPMs support only about 5 signatures per second. * Non-HTTP protocols: No solution for IPSEC etc.

4.2.2. 4.2.2. Location and Geofencing Challenges

* *IP Address-Based Location:* This is the typical approach, but it has limitations: network providers can use geographic-region-based IANA-assigned IP addresses anywhere in the world, and enterprise VPNs can hide the user’s real IP address.

* *Wi-Fi-Based Location:* For user laptop endpoints with agents (e.g., ZTNA), traditional geographic enforcement relies on trusting the Wi-Fi access point’ s location. However, Wi-Fi access points are mobile and can be moved, undermining this trust.

* *GNSS:* Certain GNSS, e.g., civilian GPS in smartphones and navigation systems, can be spoofed. A practical example is the Israel GPS spoofing attacks.

4.2.3. 4.2.3. Implicit Trust Challenges

* *Cloud Region Trust* Implicit trust in cloud region assignment with no cryptographic proof of physical locality. There is no auditable link between stored blobs and actual geography.

* *Trust in Transit:* HTTP requests can be intercepted and modified by compromised intermediate proxies (e.g., API gateways, SASE firewalls).

5. 5. Approach Overview

This approach enables cryptographically verifiable geofencing by binding workload identity to both platform and geographic attributes using trusted hardware (e.g., TPM), attestation protocols, and geolocation services. The framework supports secure, policy-driven enforcement of data residency and location requirements for workloads in multi-system environments.

Key elements of the approach include: - *Trusted Hardware Roots:* Workload identity is anchored in hardware roots of trust such as TPMs, GNSS sensors, and mobile network modules, ensuring device integrity and authentic location data. - *Remote Attestation:* Workload Identity Agents collect measurements from the platform and location sensors, and use TPM-backed attestation to prove the integrity and residency of the workload to a remote Workload Identity Manager. - *Composite Location Claims:* The system combines multiple sources of location (e.g., GNSS, mobile network, Wi-Fi) and device composition (e.g., SIM, TPM EK) to create a composite, quality-scored location claim, which is cryptographically signed and verifiable. - *Policy Enforcement:* Workload Identity Managers and downstream policy implementers (e.g. Firewall) use these verifiable claims to enforce geofencing and data residency policies, ensuring that workloads only run or access data within approved geographic or jurisdictional boundaries. - *Continuous Monitoring:* The framework supports periodic re-attestation and monitoring of device composition and location, detecting changes such as SIM swaps or sensor removal that could affect trust. - *Interoperability:* The approach is designed to integrate with existing workload identity frameworks (e.g., SPIFFE/SPIRE), enabling adoption in cloud, edge, and enterprise environments.

For example, in this document: * The *Workload Identity Manager* is represented by the SPIFFE/SPIRE server. * The *Workload Identity Agent* is represented by the SPIFFE/SPIRE agent.

5.1. 5.1. Server Hosts - Solution highlights

Assumptions: The maximum round-trip delay within a data center typically ranges from 500-1000 microseconds.

Scalable hierarchical approach enhancements to Workload Identity (SPIFFE/SPIRE) solution * Each of the hosts runs a workload identity agent (SPIFFE/SPIRE agent) with TPM plugin which connects to a workload identity manager (SPIFFE/SPIRE server)

* Location anchor hosts are directly attached to trusted location source - Mobile modem, GNSS Galileo receiver etc.
  - Use of multiple location anchor hosts can enhance security and trust.
  - For mobile sensors, the location can be tracked outside of host using a GSMA standards based mobile service provider API.
* Host geolocation sensor composition manager periodically verifies location anchor hosts device composition (primarily location sensors).
  - Use SPIFFE/SPIRE agent host geolocation plugin (new)
* Host proximity manager periodically verifies that location anchor hosts provide proof that application hosts are within the maximum data center round-trip delay from them.
  - SW-based Attested PTP - Modified Linux PTP daemon (SPIFFE/SPIRE workload) will sign PTP messages
  - HW-based Attested PTP - Relevant for sub microsecond precision timing solutions
* Workload identity agent provides proof that workloads run only on workload/location anchor hosts.

- This is done using enhancements to existing SPIFFE/SPIRE agent and TPM plugin and addresses bearer token issue

Addressing the key security and compliance challenges in the Sovereign Cloud AI Inferencing use case: Figure -- Verifiable Geofencing with Proof of Residency for Sovereign Cloud AI Inferencing (https://github.com/nedmsmith/draft-klspa-wimse-verifiable-geo-fence/blob/ramki/pictures/addressing-challenges-sovereign-cloud-ai-inferencing.svg)

5.2. 5.2. End user/IoT hosts - Solution highlights

Browser solution new browser extension for proof residency and geofencing * Application proxy which intercepts every HTTP request; connects to workload identity agent to add geolocation; signs request using workload identity agent key which is attested by TPM attestation key.

Similar to the server hosts solution * Each of the hosts runs a workload identity agent (SPIFFE/SPIRE agent) which connects to a workload identity manager (SPIFFE/SPIRE server).

Addressing the key security and compliance challenges in the Federated Learning use case: Figure -- Verifiable Geofencing with Proof of Residency for Federated Learning (https://github.com/nedmsmith/draft-klspa-wimse-verifiable-geo-fence/blob/ramki/pictures/addressing-challenges-federated-learning.svg)

6. 6. Control Plane End-to-End Workflow

The end-to-end workflow for the proposed framework consists of several key steps, including attestation for system bootstrap and Workload Identity Agent initialization, Workload Identity Agent geolocation and geofencing processing, workload attestation, and remote verification.

Figure -- Control Plane End-to-end Workflow (https://github.com/nedmsmith/draft-klspa-wimse-verifiable-geo-fence/blob/main/pictures/control-plane-end-to-end-flow.svg)

6.1. 6.1. SPIFFE/SPIRE Architecture Modifications

In the context of the SPIFFE/SPIRE architecture, the SPIFFE/SPIRE Agent includes a new geolocation plugin -- this is depicted in the figure below. The Agent is a daemon running on bare-metal Linux OS host (H) as a process with direct access to TPM (root permissions for TPM 2.0 access may be needed for certain Linux distributions for certain host hardware configurations). The Agent, using the geolocation plugin, can gather the location from host-local location sensors (e.g., GNSS). The Agent has a TPM plugin which interacts with the TPM. The Workload Identity Manager (SPIFFE/SPIRE server) is running in a cluster which is isolated from the cluster in which the Agent is running.

Figure -- Modified SPIFFE-SPIRE architecture with new geolocation plugin (https://github.com/nedmsmith/draft-klspa-wimse-verifiable-geo-fence/blob/main/pictures/spiffe-spire.svg)

6.2. 6.2. Attestation of OS Integrity and Proof of Residency on Host

As part of system boot/reboot process, boot loader-based measured system boot with remote Workload Identity Manager verification is used to ensure only approved OS is running on an approved hardware platform.

Measurement Collection: During the boot process, the boot loader collects measurements (hashes) of the boot components and configurations. The boot components are Firmware/BIOS/UEFI, bootloader, OS, drivers, location devices, and initial programs. All the location devices (e.g., GNSS sensor, mobile sensor) version/ firmware in a platform are measured during each boot -- this is a boot loader enhancement. Any new location device which is hot-swapped in will be evaluated for inclusion only during next reboot.

Log Creation: These measurements are recorded in a log, often referred to as the TCGLog, and stored in the TPM's Platform Configuration Registers (PCRs).

Attestation Report: The TPM generates an attestation report, which includes the signed measurements and the boot configuration log. The signature of the attestation report (aka quote) is by a TPM Attestation Key (AK). This attestation includes data about the TPM's state and can be used to verify that the AK is indeed cryptographically backed by the TPM Endorsement Key (EK) certificate.

Transmission: The attestation report is then sent to an external verifier (Workload Identity Manager), through a secure TLS connection.

Remote Verification: The remote Workload Identity Manager checks the integrity of the attestation report and validates the measurements against known good values from the set of trusted hosts in the Host hardware identity datastore. The Workload Identity Manager also validates that the TPM EK certificate has not been revoked and is part of the approved list of TPM EK identifiers associated with the hardware platform. At this point, we can be sure that the hardware platform is approved for running workloads and is running an approved OS.

6.3. 6.3. Start/Restart time attestation/remote verification of Workload Identity Agent for integrity and proof of residency on Host

The Workload Identity Agent TPM plugin is a process with elevated privileges with access to TPM and location sensor hardware. Linux IMA and Workload Identity Agent public/private key attestation are the changes compared to the original SPIFFE/SPIRE architecture with the TPM plugin.

Measurement Collection: For the Workload Identity Agent start case, the Agent executable is measured by Linux IMA, for example through cloud init and stored in TPM PCR through tools e.g., Linux ima-evm-utils before it is loaded. For the Workload Identity Agent restart case, it is not clear how the storage in TPM PCR will be accomplished - ideally this should be natively handled in the IMA measurement process with an ability to retrigger on restart or refresh cycles (OPEN ISSUES 1).

Local Verification: Enforce local validation of a measurement against an approved value stored in an extended attribute of the file.

TPM attestation and remote Workload Identity Manager verification:

Step 1 (Workload Identity Agent TPM APP ID issuance): 1. The Workload Identity Agent TPM plugin generates a TPM APP private key for proof of residency on the host for each start/restart. 2. The Workload Identity Agent TPM plugin sends the TPM APP public key, TPM AK public key and TPM EK certificate attestation parameters to the Workload Identity Manager. 3. The Workload Identity Manager verifies the attestation parameters. It then validates that the TPM

EK certificate is in the trusted TPM EK certificate list with the Host Identity Manager (e.g. Keylime Verifier). 4. If validation passes, the Workload Identity Manager generates a credential activation challenge. The challenge’s secret is encrypted using the Workload Identity Agent TPM APP public key. 5. The Workload Identity Manager sends the challenge to the Workload Identity Agent. 6. The Workload Identity Agent decrypts the challenge’s secret using its TPM APP private key. 8. The Workload Identity Agent sends back the decrypted secret. 9. The Workload Identity Manager verifies that the decrypted secret matches the original secret used to build the challenge. 10. The Workload Identity Manager issues a Workload Identity Agent TPM APP ID using the TPM APP public key, TPM AK public key and TPM EK certificate.

Step 2 (Workload Identity Agent ID issuance): 1. The Workload Identity Agent generates a private/public key pair. 2. The Workload Identity Agent uses the TPM APP private key, stored in the TPM, to sign the public key. 3. The Workload Identity Agent sends the public key, signed by the TPM APP private key, to the Workload Identity Manager. 4. The Workload Identity Manager ensures the public key is associated with a Workload Identity Agent TPM APP ID. 5. If validation passes, the Workload Identity Manager generates a credential activation challenge. The challenge’s secret is encrypted using the Workload Identity Agent public key. 6. The Workload Identity Manager sends the challenge to the Workload Identity Agent. 7. The Workload Identity Agent decrypts the challenge’s secret using its private key. 8. The Workload Identity Agent sends back the decrypted secret. 9. The Workload Identity Manager verifies that the decrypted secret matches the original secret used to build the challenge. 10. The Workload Identity Manager issues the Workload Identity Agent ID using the Workload Identity Agent public key, the TPM APP signature of the Workload Identity Agent public key, and the Workload Identity Agent TPM APP ID.

Design Options for TPM-Based Workload Identity with Privacy * From a privacy standpoint, sharing TPM details—especially the EK certificate—across organizational boundaries can be problematic. Below are two approaches that let you attest workload identities without exposing raw EK data. * Option 1 Pseudonymity - Only the Workload Identity Agent’s TPM APP public key and Host identity agent’s (e.g. Keylime agent) TPM AK public key are shared outside the host-owner organization. - The consuming organization’s Workload Identity Manager verifies that AK against its own list of trusted AK public keys. - A unique TPM AK is generated and attested per tenant, so no single AK maps across tenants. - Supported by all TPM 2.0+ devices, this gives you per-tenant pseudonymity without ever revealing the EK. * Option 2 Full Anonymity - Leverage TPM 2.0+ Direct Anonymous Attestation (DAA) or EPID. Note that this requires TPM 2.0+ devices with DAA support. - Each host’s TPM runs the DAA Join protocol with a Privacy CA to obtain a group credential. - The Workload Identity Agent signs its public key with that DAA credential (using a session-specific basename). - The Workload Identity Manager verifies the signature against the DAA group public key—proving membership without exposing or linking any device identity. * Both options remove direct TPM EK exposure. Pseudonymity uses the standard TPM AK model, while TPM DAA offers unlinkable, anonymous proofs of TPM possession.

6.4. 6.4. Host geolocation sensor composition manager and Host Composition Change Tracking

The Host geolocation sensor composition manager runs outside of the host. In addition to obtaining location from device location sources (e.g., GNSS), it connects to mobile location service providers (e.g., Telefonica) using the GSMA Location API ([gsma-loc]). The process described below is run periodically (e.g., every 5 minutes) to check

if the host hardware composition has changed. Host hardware composition comprises TPM EK, GNSS sensor hardware ID, mobile sensor hardware ID (IMEI), and mobile-SIM IMSI. Note that this workflow is feasible only in enterprise environments where the host hardware is owned and managed by the enterprise.

1. The Workload Identity Agent periodically gathers host composition details (e.g., mobile sensor hardware ID (IMEI), mobile-SIM IMSI) and sends them to the Host geolocation sensor composition manager.

2. The Host geolocation sensor composition manager cross-verifies that the components of the host are still intact or detects if anything has been removed. (Plugging out components can decrease the quality of location. Host hardware composition comprises TPM EK, GNSS sensor hardware ID, mobile sensor hardware ID (IMEI), and mobile-SIM IMSI. Note that e-SIM does not have the plugging out problem like standard SIM but could be subject to e-SIM swap attack.)

6.5. 6.5. Workload Identity Agent Geolocation Gathering Workflow

The process described below is run periodically (e.g., every 30 seconds for frequently mobile hosts such as smartphones; every 5 minutes for less frequently mobile hosts such as laptops; every 50 minutes for stationary hosts) to check if the host’s location has changed and to obtain an attested location.

1. The Workload Identity Agent gathers the location using the geolocation plugin (a) directly from host-local location sensors (e.g., GNSS), which provide a hardware-attested location, and/or (b) using existing Operating System (OS) APIs, which gather a composite location from location providers (e.g., Google, Apple). Location has a quality associated with it. For example, IP address-based or Wi-Fi-based location is of lower quality compared to other sources.

2. For each of the registered workload IDs (or website URL), based on the configured location policy (precise, approximated within a fixed radius, geographic region-based indicating city/state/ country - see OPEN ISSUES 2), the location is converted appropriately to a workload ID-specific location. For thin clients (browser clients), the workload ID is the website URL. This ensures that the privacy of the workload is preserved while still allowing for geolocation enforcement.

3. All the above details are captured in the Geolocation Information Cache which contains the following fields:

    1. Time of collection (timestamp)

    2. Workload ID specific location details for each client workload where each entry contains:

        1. client workload ID - relevant for thick clients (e.g. Microsoft Teams client)

        2. server workload ID (or website URL) - relevant for thin clients (e.g. Microsoft Teams browser version)

        3. client location type (e.g. precise, approximated, geographic region based)

        4. client location (e.g. latitude/longitude, city/state/ country)

5. client location quality (e.g. GNSS, mobile network, Wi-Fi, IP address)

It is important to note that the Geolocation Information Cache is kept in the Workload Identity Agent memory and is not stored on disk. The information is refreshed periodically to ensure that the location is up-to-date. This information is used only by workloads in the host and never leaves the host.

If the location is gathered only using existing OS APIs, it may be done in the workload (thick client) or browser extension (thin client). The Geolocation Information Cache is stored in thick client memory (relevant only to specific client) or browser extension memory (relevant to all thin clients and indexed using user in OAuth bearer token/server website URL).

6.6. 6.6. Workload Public Key Attestation and Remote Verification

Workload Identity Agent public/private key attestation, rather than TPM Attestation Key (AK) attestation, is the key change compared to the original SPIFFE/SPIRE architecture with the TPM plugin. Optionally, the workload geolocation can be attested to the Workload Identity Manager BY supplying the geolocation information in the Geolocation Information Cache.

1. The Workload Identity Agent ensures that the workload connects to it on a host-local socket (e.g., Unix-domain socket).

2. The Workload Identity Agent generates a private/public key pair for the workload.

3. The Workload Identity Agent signs the workload public key with its own private key.

4. The Workload Identity Agent sends the signed workload public key, along with its cryptographically attested Workload Identity Agent ID, to the Workload Identity Manager. (Note: The Workload Identity Agent ID has already been verified by the Workload Identity Manager during the agent attestation process, establishing proof of residency of the Workload Identity Agent on the host.)

5. The Workload Identity Manager verifies that the TPM Endorsement Key (EK) associated with the Workload Identity Agent ID is present in the trusted host hardware database.

6. The Workload Identity Manager verifies the workload public key signature using the Workload Identity Agent’s public key (which was previously attested).

7. The Workload Identity Manager sends an encrypted challenge to the Workload Identity Agent. The challenge’s secret is encrypted using the workload’s public key. (This step proves that the workload controls the corresponding private key.)

8. The Workload Identity Agent decrypts the challenge using the workload’s private key and sends the response back to the Workload Identity Manager.

9. The Workload Identity Manager verifies that the decrypted secret matches the original secret used to build the challenge.

10. The Workload Identity Manager issues a workload ID (e.g., SPIFFE ID) for the workload’s public key. The workload ID is signed by the Workload Identity Manager and contains the workload’s public key and the Workload Identity Agent ID. * Optionally, the

workload geolocation can be attested to the Workload Identity Manager by supplying the geolocation information in the Geolocation Information Cache. The SVID of the workload is signed by the Workload Identity Manager and contains the workload's public key, the Workload Identity Agent ID, and the workload geolocation information.

11. Design Options: * *Option 1:* The workload can use the Workload Identity Agent to manage its keys and perform cryptographic operations on its behalf. In this case, the workload receives its workload ID from the Workload Identity Agent. * *Option 2:* The workload can manage its own keys and perform cryptographic operations independently. In this case, the workload receives its private key and workload ID from the Workload Identity Agent. * *Comparison:* Option 1 (software HSM delegated model) is more secure, as the workload's private key is never exposed outside the Workload Identity Agent. Option 2 provides more flexibility for workloads that require independent key management and may offer slightly better performance, since the workload can optimize cryptographic operations without going through the Workload Identity Agent.

7. 7. Scaling the Solution

Having a geolocation sensor on every host is not scalable from a deployment and management perspective and can be cost prohibitive. In the case of end user hosts, the geolocation sensor can be on a mobile host (e.g., smartphone with Mobile network capabilities and optionally GNSS capabilities) which can be leveraged by a laptop/desktop host which is proximal to the mobile host. The mobile host serves as the location anchor host. In the case of data center hosts, the geolocation sensor can be on a host with Mobile network and/or GNSS capabilities which can be leveraged by other data center hosts. This host serves as the location anchor host.

7.1. 7.1. End user location anchor host

Goal is to provide an easy to use wireless solution that can be used by end users without requiring them to install a geolocation sensor on their laptop/desktop host.

The smartphone can be used as a location anchor host for the laptop/desktop host. The smartphone connects to the laptop/desktop host using Bluetooth Low Energy (BLE) or Ultra-Wideband (UWB) technology and continuously measures the following: * signal strength of the laptop/desktop host * round-trip time (RTT) between the smartphone and laptop/desktop host

Host proximity manager periodically verifies that the smartphone provides proof that the laptop/desktop host is in proximity using the measured signal strength and RTT.

7.2. 7.2. Data center location anchor host

Goal is to provide an easy to use solution that can be used by data center operators without requiring them to install a geolocation sensor on every data center host.

PTP is a network protocol that enables precise synchronization of clocks across a computer network and can be used to measure the round-trip time (RTT) between the location anchor host and other data center hosts with sub-microsecond accuracy. To provide cryptographically verifiable proof of residency on the host - referred to as "attested PTP" - the PTP software/hardware can be enhanced so that all PTP messages are signed with a private key.

This signing can be done in two ways: * Software-based: PTP software (e.g. Linux PTP daemon), after adding timestamp to PTP message, signs the PTP message with its private key -- Linux PTP daemon is a workload managed by workload identity manager. This approach may not provide sub-microsecond accuracy due to inherent software jitter, but it can still provide a reasonable approximation of the proximity of the other data center hosts to the location anchor host. * Hardware-based: PTP hardware (e.g., SmartNIC), after adding timestamp to PTP message, signs the PTP message with its private key (e.g., SmartNIC DPU). The corresponding public key, used to verify the signatures, can be attested by the Host TPM Attestation Key (AK). This approach provides sub-microsecond accuracy and the perfect proximity measure of the other data center hosts to the location anchor host, and is suitable for data center environments where precise timing is critical.

Host proximity manager periodically verifies that the PTP daemon in the location anchor hosts provide proof that the PTP daemon in the application hosts are within the maximum DC round-trip delay from them. The PTP daemon is another workload managed by the workload identity manager.

Note that this is a proposed enhancement to the existing PTP hardware and software, and there is currently no standard for attested PTP (see OPEN ISSUES 3). Further work is needed to define and standardize this enhancement to ensure interoperability and security.

8. 8. Data Plane End-to-End Workflow

The following sections describe the end-to-end workflow for HTTP and IPSEC networking protocols.

8.1. 8.1. HTTP Networking Protocol - request signing along with geolocation information

This workflow enhances the Demonstrating proof of posession (DPoP) https://datatracker.ietf.org/doc/html/rfc9449) mechanism as follows (1) By providing a more accurate and cryptographically verifiable location of the client workload and (2) Using a workload signing key that is attested by the Workload Identity Manager for proof of residency on approved hosts. Note that this public key is not part of the OAuth bearer token.

A new HTTP header field ’Workload-Geo-ID’ is proposed for conveying the workload geolocation information in the Geolocation Information Cache. The HTTP request is signed--the signature is generated using the Workload private key (thick client) or the Workload Identity Agent private key. The following steps describe the end-to-end workflow for HTTP requests between client workloads (e.g. Microsoft Teams thick client app, Microsoft Teams thin client browser app) and server workloads (e.g. Microsoft Teams server), including intermediate proxies (e.g., API gateways, SASE firewalls). The server workload (e.g. Microsoft Teams server) acts as a thick client when it connects to other server workloads (e.g. Microsoft OneDrive for Business).

1. Client workload gets OAuth bearer token for the server workload from the Authentication/Authorization server.

2. Client workload (browser extension for thin client) contacts the Workload Identity Agent to get the latest Geolocation Information Cache relevant to it. If the location is gathered only using existing OS APIs, it may be done in the workload (thick client) or browser extension (thin client). The client workload (browser extension for thin client) constructs a Workload-Geo-ID extension header containing the following

fields:

* The latest Geolocation Information Cache relevant to the client workload ID (thick clients) or user in OAuth bearer token/server website URL (thin clients) which has the following details:

  - client workload location,

  - client workload location type (e.g. precise, approximated, geographic region based),

  - client workload location quality (e.g. GNSS, mobile network, Wi-Fi, IP address).

* Thick or Thin client flag: Indicates whether the client workload is a thick client (e.g., Microsoft Teams thick client app) or a thin client (e.g., Microsoft Teams thin client browser app).

* Type flag: Request signing with geolocation information (this section) or SVID with geolocation information (next section)

* Using a browser extension (thin client) is attractive especially for web applications which already use a browser extension for functions such as URL filtering, ad blocking, privacy protection, etc. The tradeoffs are:

  - Pros: Leverages Workload Identity agent (SPIFFE/SPIRE agent) hardened key store (TPM etc.); Keeps keys out of browser javascript entirely; Centralized management of keys and policies; Provides a consistent interface for all web applications.

  - Cons: Requires browser extension installation; May not be supported by all browsers

3. For thick clients, the Client workload signs the hash of the HTTP request using the workload private key.

4. For thin clients, the Client workload passes the hash of the HTTP request to the Workload Identity Agent for signature. The Workload Identity Agent signs the HTTP Request using the Workload Identity Agent Private Key and returns the signature of the HTTP request to the workload.

5. Details of the HTTP request signature:

   * The resulting signature is included in a separate header, Signature (RFC 9421)

   * The signature input is included in a separate header, Signature-Input (RFC 9421), which contains the following fields:

     - keyid: Thin clients--The Workload Identity Agent ID public key hash is used as the keyid; Thick clients--The Workload public key hash is used as the keyid.

     - created: The timestamp of the signature creation.

     - expires: The timestamp of the signature expiration (e.g., 5 minutes after creation).

     - alg: The algorithm used for signing (e.g., Ed25519).

- nonce: The unique nonce used for replay protection and troubleshooting.

- context: The context of the signature, which includes the HTTP method, and URL

* The public key used to verify the signature can be derived using the Workload Identity Agent ID public key hash. This enables recipients (intermediate proxies or server workloads) to validate the authenticity of the signature and the binding to the specific Workload Identity Agent.

6. Client workload appends the Signature header and the Signature-Input header to the HTTP request.

7. Intermediate proxies (e.g., API gateways, SASE firewalls) inspect the Workload-Geo-ID, Signature and Signature-Input header fields and perform the following checks:

   * Verify that the Workload Identity Agent ID hash in the Signature-Input header matches a configured Workload Identity Agent ID. They can retrieve the host TPM EK certificate from the Workload Identity Agent ID and compare it with the host TPM EK certificate in the Host hardware identity datastore.

   * Verify that the HTTP request signature in the Signature header is valid by verifying it against the Workload Identity Agent Public Key in the Signature-Input header.

   * Verify that the timestamp in the Signature-Input header is within an acceptable range (e.g., 5 minutes).

   * Verify that the nonce in the Signature-Input header is unique and monotonically increasing to prevent replay attacks.

8. Note that these HTTP extension header checks can be performed by the server as well, but it is more efficient to do them at the intermediate proxy level and aligns well with how Zero Trust Network Access (ZTNA) solutions operate. If the verification passes, the request is forwarded to the destination server. If the verification fails, the request is dropped, and an error response is generated.

9. Intermediate proxies (e.g., API gateways, SASE firewalls) or server workloads connect to Composite Geolocation Manager, to supply GNSS geolocation/workload identity agent ID and get attested composite location. Composite Geolocation Manager can use the host TPM EK certificate in the Workload Identity Agent ID to retrieve the mobile geolocation sensor IMEI/IMSI from the Host hardware identity datastore. Using the IMEI/IMSI, they can retrieve the location of the host from the mobile network operator’s location service. This is useful for mobile devices that may not have GNSS sensors or when GNSS is not available (e.g., indoors) or when GPS/GNSS location is subject to spoofing. As compared to IP address, Wi-Fi and GPS/GNSS geolocation methods, mobile network location services provide a more reliable and cryptographically verifiable location. Based on the mobile geolocation and existing geolocation in the Workload-Geo-ID header, a more accurate composite location can be constructed. - Note that for performance considerations, the Composite Geolocation Manager can be a library function which is integrated into intermediate proxies or server workloads.

10. Intermediate proxies (e.g., API gateway, Firewall) or server workloads can enforce policies based on: - Workload Identity Agent ID (running on the same host as the client workload), -

user in OAuth bearer token, - server website URL, - client workload ID (relevant only for thick clients), - client workload location, - client workload location type (e.g. precise, approximated, geographic region based), - client workload location quality (e.g. GNSS, mobile network, Wi-Fi, IP address).

Besides native HTTP protocols, this solution will also address the following common browser-based protocols: * browser-based Secure Shell (ssh) terminal (common for cloud access by customers) which tunnels ssh traffic over HTTP/TLS. * browser-based Remote Desktop Protocol (RDP) terminal (common for cloud access by customers) which tunnels RDP traffic over HTTP/TLS.

8.2. 8.2. HTTP Networking Protocol - geolocation information in SVID

The workload SVID can be used to convey the geolocation information to the workload. The SVID is signed by the Workload Identity Manager and contains the workload's public key, the Workload Identity Agent ID, and the workload geolocation information.

The workload SVID can be conveyed in the 'Workload-Geo-ID' header field.

8.3. 8.3. IPSEC Tunnel Networking Protocol

In the IPSEC key exchange protocol (IKE), the following changes are proposed: * Proof of residency * In the IPSEC client, in the Elliptic Curve Diffie-Hellman Ephemeral key exchange (ECDHE) phase, the Workload Identity Agent Public Key is used as the ephemeral public key. * Geolocation information * The IPSEC client includes the Geolocation Information in the Workload Identity Agent Geolocation Information Cache in the IPSEC IKEv2 notification payload.

IPSEC server policy enforcement can be done in the following way: * Proof of residency * In the IPSEC server, from the IPSEC IKEv2 notification payload, the Workload Identity Agent Public Key is extracted. The Workload Identity Agent Public Key is checked against the configured list of allowed Workload Identity Agent IDs (IPSEC client certificates). The signature of the IPSEC client is then verified using the Workload Identity Agent Public Key. This provides a cryptographically verifiable proof of residency of the IPSEC client on the required host. * Geolocation policy enforcement * In the IPSEC server, from the IPSEC IKEv2 notification payload, the Geolocation Information is extracted. * IPSEC server connects to Composite Geolocation Manager, supply Geolocation Information/workload identity agent ID, and get attested composite location which includes mobile geolocation also. In case the mobile network location service is not use, the composite Geolocation Information is the same as the original Geolocation Information. * The IPSEC server can use the composite Geolocation Information to verify that the host is within the allowed geographic boundary.

Benefit: * Since IPSEC tunnel can encapsulate any IP traffic, it provides proof of residency and geolocation on the IPSEC client host for all the traffic that is tunneled through it (e.g., RDP, SCTP, NFS, SSH).

Challenge: * Location information granularity is at the IPSEC client host level and not at the individual workload level, which may be a challenge for some use cases.

9. 9. Confidential Computing Considerations

In confidential computing, the host operating system cannot be trusted. Instead, the platform owner must maintain and verify the relationships between three hardware identifiers: * Hardware-rooted

certification key * Hardware-rooted certification key — Vendor-issued, CPU-bound asymmetric key pair used to either sign attestation evidence directly (AMD VCEK) or certify an attestation signing key (Intel PCK). Always anchored in the vendor's root CA. * AMD SEV-SNP "Versioned Chip Endorsement Key VCEK" is defined in https://www.amd.com/content/dam/amd/en/documents/epyc-technical-docs/specifications/57230.pdf (Location: Chapter 1 (Glossary), specifically in Table 2 Terms and Definitions) * Intel TDX, "Provisioning Certification Key PCK" is defined in https://download.01.org/intel-sgx/latest/dcap-latest/linux/docs/Intel_TDX_DCAP_Quoting_Library_API.pdf (Location: Page 6, under Table 1-1: Terminology) * Role: Certifies the Attestation Key inside the TDX Quoting Enclave, which signs the quote. * TPM Endorsement Key (EK) * Geolocation Sensor ID + public key

Proof of Residency * The confidential workload generates a hardware attestation that includes its hardware-rooted certification key. * The platform owner receives this attestation and binds the reported hardware-rooted certification key to the TPM EK. * This binding produces a cryptographic proof that the workload is running on the expected physical CPU.

Proof of Geolocation * The geolocation sensor creates a signed location report using its private key. This is supported in popular GNSS sensors such as https://www.u-blox.com/en. * An agent on the bare-metal host periodically polls the sensor and collects these signed reports. * The platform owner maps each sensor's ID and its signed geolocation to the corresponding CPU ID and TPM EK. * This mapping yields a verifiable proof that the workload is executing at the claimed physical location.

Note: The Intel Software Guard Extensions (Intel SGX) Attestation Service utilizing the Enhanced PrivacyID (EPID) group-signature mechanism is a legacy, privacy-preserving attestation path. Intel has announced that this service will reach end-of-life on April22025, after which EPID-based attestation will no longer be supported. See Intel's IAS End-of-Life announcement (https://community.intel.com/t5/Intel-Software-Guard-Extensions/IAS-End-of-Life-Announcement/m-p/1545831) for details. This discussion focuses on current attestation models (e.g., ECDSA-based DCAP for SGX and PCK-based attestation for TDX) and excludes EPID/DAA from scope. ECDSA-based DCAP for SGX and PCK-based attestation for TDX are closely related in structure and trust model — both are part of Intel's Data Center Attestation Primitives (DCAP).

10. 10. Solution Mapping to Industry Gaps and Problem Statements

* *Host TPMs for Signature* challenges are addressed
  - Workload Identity Agent private key, which is used for signing. Workload identity agent public key is signed by the Host TPM APP private key providing a cryptographically verifiable proof of residency of Workload Identity Agent on the host. The Workload Identity Agent generates a public/private key pair for each workload which connects through a host local socket and signs the workload public key with its private key. The Workload Identity Manager verifies the signature using the Workload Identity Agent Public Key, providing a cryptographically verifiable proof of residency of workload on the host.
* *Bearer Tokens*, *PoP Token*, *PoP via Mutual TLS* challenges are addressed
  - HTTP request signature with the Workload Identity Agent Private Key, which provides a scalable and cryptographically verifiable

proof of residency on host and workload identity. The signature is verified by the intermediate proxies (e.g., API gateways, SASE firewalls) or server workloads using the Workload Identity Agent Public Key.

* *IP Address-Based Location* and *Wi-Fi-Based Location* challenges are addressed

  - Combination of host-local location sensors (e.g., GNSS) with direct hardware-based attestation and mobile network location services provides a more reliable and cryptographically verifiable location than IP address, Wi-Fi-based methods or existing Host OS location services.

* *Trust in Transit* challenges are addressed

  - The HTTP request signature with the Workload Identity Agent Private Key provides a cryptographically verifiable proof of residency on host and workload identity, which is verified by the intermediate proxies (e.g., API gateways, SASE firewalls) using the Workload Identity Agent Public Key. This ensures that the request is not tampered with in transit.

* *IPSEC Tunnel Networking Protocol* challenges are addressed

  - The IPSEC client uses the Workload Identity Agent Public Key as the ephemeral public key in the ECDHE phase of the IPSEC IKEv2 key exchange protocol, providing a cryptographically verifiable proof of residency on host. The Geolocation Information is included in the IPSEC IKEv2 notification payload, which is verified by the IPSEC server.

11. 11. Authorization Policy Implementers

Policy implementers use attested geographic boundary from Workload to make decisions. Example implementers: * Intermediate proxies (e.g., API gateway, Firewall) * SaaS application. * K8s node agent. * OS process scheduler.

If the policy implementer is at the SaaS application level, things are simpler. However, if it is pushed down to, say, K8s or OS process scheduler or JVM class loader/deserializer, then malware can be prevented (similar to a code-signed application).

12. 12. Security Considerations

The proposed framework introduces several security considerations that must be addressed to ensure the integrity and trustworthiness of geofencing:

* *TPM and Hardware Trust*: The security of the solution depends on the integrity of the TPM and other hardware roots of trust. Physical attacks, firmware vulnerabilities, or supply chain compromises could undermine attestation. Regular updates, secure provisioning, and monitoring are required.

* *Geolocation Spoofing*: Location sensors (e.g., GPS) are susceptible to spoofing or replay attacks. Use of cryptographically authenticated signals (e.g., Galileo GNSS, mobile network) and cross-verification with multiple sources can mitigate this risk.

* *SIM and e-SIM Attacks*: Physical SIM removal or e-SIM swap attacks can break the binding between device and location. Continuous monitoring of device composition and periodic re-attestation are recommended.

* *Software Integrity*: The geolocation agent and supporting software must be protected against tampering. Use of Linux IMA, secure boot, and measured launch environments helps ensure only approved software is executed.

* *Communication Security*: All attestation and geolocation data must be transmitted over secure, authenticated channels (e.g., TLS) to prevent interception or manipulation.

* *Policy Enforcement*: The enforcement of geofence policies must be robust against attempts by malicious workloads or agents to bypass controls. Policy decisions should be based on verifiable, signed attestation evidence.

* *Time Source Integrity*: Trusted time sources are necessary to prevent replay attacks and ensure the freshness of attestation data.

* *Datastore Security*: The Host hardware identity datastore containing trusted host compositions and location sensor details must be protected against unauthorized access and tampering, using encryption and access controls.

By addressing these considerations, the framework aims to provide a secure and reliable foundation for verifiable geofencing in diverse deployment environments.

13. 13. IANA Considerations

This document has no IANA actions.

14. 14. Appendix - Items to follow up

14.1. OPEN ISSUES 1: Restart time attestation/remote verification of workload identity agent for integrity and proof of residency on Host

For the workload identity agent restart case, it is not clear how the storage in TPM PCR will be accomplished - ideally this should be natively handled in the IMA measurement process with an ability to retrigger on restart or refresh cycles.

14.2. OPEN ISSUES 2: Location privacy options

The current approach includes some location privacy options for the geolocation in the Geolocation Information Cache. This may need to be expanded further in the future.

14.3. OPEN ISSUES 3: Attested PTP

Attested PTP is a software/hardware-based solution using Precision Time Protocol (PTP) for measuring proximity between hosts in a data center. However, this is a proposed enhancement to the existing PTP hardware and software, and there is currently no standard for attested PTP. There is a proposed authentication framework for PTP using symmetric key distribution (https://datatracker.ietf.org/doc/draft-ietf-ntp-nts-for-ptp/).

14.4. OPEN ISSUES 4: Geotagging textual data

Popular standard for geotagging photos/videos is EXIF. There is no standard for geotagging textual data. If there is no geolocation tag, data can be stored/processed in non-compliant locations.

14.5. OPEN ISSUES 5: Attesting Geotags

There is no standard for attesting (signing) geolocation tag. If geolocation tag is not signed, it can be manipulated through techniques such as VPNs.

15. 15. Appendix - Public References for Strict Data Residency Rules

India — Reserve Bank of India (RBI): Payment System Data Localization (2018) * From RBI Circular RBI/2017-18/153 (April 6, 2018): * “All system providers shall ensure that the entire data relating to payment systems operated by them are stored in a system only in India. This data should include the full end-to-end transaction details / information collected / carried / processed as part of the message / payment instruction.” (https://www.rbi.org.in/SCRIPTs/NotificationUser.aspx?Id=11244)

South Korea’s Data Localization Regulations * Geospatial Information Management Act (Spatial Data Act) * Article 16, Paragraph 1: Prohibits the export of state-led survey data (https://elaw.klri.re.kr/eng_service/lawView.do?hseq=45348&lang=ENG).

16. References

16.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/rfc/rfc2119>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/rfc/rfc8174>.

16.2. Informative References

[doj-cisa] DOJ and CISA, "DOJ and CISA Issue New National Security Program to Regulate Foreign Access to Sensitive Data", n.d., <https://www.justice.gov/opa/pr/justice-department-implements-critical-national-security-program-protect-americans-sensitive>.

[galileo] European Commission, EU Space, "Galileo Satellite Navigation", n.d., <https://defence-industry-space.ec.europa.eu/eu-space/galileo-satellite-navigation_en>.

[gsma-loc] GSMA open gateway documentation, "GSMA location API", n.d., <https://www.gsma.com/solutions-and-impact/gsma-open-gateway/gsma-open-gateway-api-descriptions/>.

[I-D.ietf-wimse-arch] Salowey, J. A., Rosomakho, Y., and H. Tschofenig, "Workload Identity in a Multi System Environment (WIMSE) Architecture", Work in Progress, Internet-Draft, draft-ietf-wimse-arch-06, 30 September 2025, <https://datatracker.ietf.org/doc/html/draft-ietf-wimse-arch-06>.

[keylime] Keylime open source project, "Keylime", n.d., <https://keylime.org/>.

[linux-ima] Sourceforge Linux IMA documentation, "Linux Integrity Measurement Architecture", n.d., <https://linux-ima.sourceforge.net/>.

[RFC-7800] IETF, "Proof-of-Possession Key Semantics for JSON Web Tokens (JWT)", n.d., <https://datatracker.ietf.org/doc/html/rfc7800>.

[RFC-8705] IETF, "OAuth 2.0 Mutual-TLS Client Authentication and Certificate-Bound Access Tokens", n.d., <https://datatracker.ietf.org/doc/html/rfc8705>.

[spiffe-jwt-svid] SPIFFE Project, "SPIFFE JWT-SVID Standard", n.d., <https://github.com/spiffe/spiffe/blob/main/standards/JWT-SVID.md>.

[spiffe-x509-svid] SPIFFE Project, "SPIFFE X.509-SVID Standard", n.d., <https://github.com/spiffe/spiffe/blob/main/standards/X509-SVID.md>.

[spire] Spire open source project, "SPIFFE/SPIRE workload identity", n.d., <https://spiffe.io/>.

[spire-tpm] Spire open source project plugin, "SPIFFE/SPIRE TPM plugin", n.d., <https://github.com/bloomberg/spire-tpm-plugin>.

[tcg-geo-loc] TCG, "TCG keynote and whitepaper-Trusted Computing Future-Emerging Use Cases and Solutions", n.d., <https://trustedcomputinggroup.org/resource/trusted-computing-future-emerging-use-cases-and-solutions/>.

[tcg-tpm] TCG, "Trusted Platform Module 2.0-A Brief Introduction", n.d., <https://trustedcomputinggroup.org/resource/trusted-platform-module-2-0-a-brief-introduction/>.

[tpm-performance] Stian Kristoffersen (Substack), "TPM Performance - How Fast is Your TPM?", n.d., <https://stiankri.substack.com/p/tpm-performance>.

Contributors

Ghada Arfaoui
Orange
Email: ghada.arfaoui@orange.com

Michael Epley
Red Hat
Email: mepley@redhat.com

Vijay Masilamani
Independent
Email: saanvijay20@gmail.com

Authors’ Addresses

Ramki Krishnan
Vishanti Systems, Inc.
Email: ramkri123@gmail.com

Ned Smith
Intel
Email: ned.smith@intel.com

Diego R. Lopez
Telefonica
Email: diego.r.lopez@telefonica.com

A Prasad
Oracle
Email: a.prasad@oracle.com

Srinivasa Addepalli
Aryaka
Email: srinivasa.addepalli@aryaka.com