Remote ATtestation ProcedureS
Internet-Draft
Intended status: Informational
Expires: 22 November 2026

G. Kostal
Microsoft
R. Yeluri
Intel
D. Kumar
Nvidia
S. Dittakavi
Microsoft
H. Xia
J. Yu
Intel

21 May 2026

EAT Attestation Result (EAR) profile for Intelツョ Trust Domain Extensions (TDX) + Confidential GPU (C-GPU) composite attestation
draft-kykdxy-rats-tdx-cgpu-ear-profile-01

Abstract

This document defines an Entity Attestation Token (EAT) Attestation Result (EAR) profile for the composite attestation of Intelツョ Trust Domain Extensions (TDX)宴濺ased Confidential Virtual Machines (CVMs) together with confidential NVIDIA GPUs (C-GPUs) deployed in Microsoft Azure. The profile outlines claims that enable relying parties to establish trust in the integrity and confidentiality of the combined confidential computing environment. Developed collaboratively by Microsoft, Intel, and NVIDIA, this work is intended to foster interoperable composite attestation across heterogeneous Trusted Execution Environments (TEEs) and confidential accelerators, while encouraging adoption and extension by verifier providers across the confidential computing ecosystem.

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 22 November 2026.

Table of Contents

1. Introduction

This document defines an Entity Attestation Token [EAT] Attestation Result (EAR) profile for composite attestation of an Intelツョ Trust Domain Extensions [TDX]寰濈ased Confidential Virtual Machine (CVM) together with one or more Nvidia confidential GPUs (C-GPUs) running in Azure. It addresses scenarios where a relying party must verify that all components of a confidential compute workload寰任PU, guest VM, and accelerators寰蚤re cryptographically bound and jointly trusted before releasing sensitive information such as secrets or cryptographic keys. The profile assumes a composite attestation model, where multiple hardware-backed attesters contribute evidence that is verified and consolidated by a verifier. Successful

verification ensures that the components form a single, unified trust domain, preventing substitution or partial compromise. The base scenario deliberately adopts an 寰彗ll-or-nothing寰trust semantic: a relying party is expected to release secrets only when the verifier has established that all components included in the composite attestation are bound and trusted. The profile does not attempt to model partial trust graphs, or workload-specific data-flow constraints.

The objective of this profile is to provide a stable attestation result format for confidential AI deployments by defining a consistent set of claims that relying parties can process uniformly. In these environments, multiple relying parties often operate under different business and regulatory requirements, which may require the use of multiple verifiers. Without a common structure, relying parties would need to interpret diverse attestation result formats and verifier-specific claims. The Composite EAR Profile removes this complexity by defining a unified attestation result structure, allowing relying parties to evaluate results against their policies without custom parsing or translation. The profile is designed to support consistent outputs across verifiers while remaining flexible enough to incorporate future confidential computing technologies and trust signals without disrupting existing deployments.

2. Requirements Language

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.

3. Scenario overview

The canonical scenario allows a Relying Party to verify the integrity of a CVM and its associated hardware before releasing secrets. The CVM consists of confidential-computingenabled CPUs and one or more confidential GPUs provisioned at deployment and assumed to remain static throughout its lifecycle. These components collectively form the CVM’s Trusted Compute Base (TCB), and any secrets released by the RP may be accessed only within this verified TCB.

The RP must verify the trustworthiness of the CPU, the CVM boot flow, and the attached confidential GPUs. To support this, the Composite token provides:

1. an aggregate trust claim confirming that secrets remain confined to the verified TCB

2. an ephemeral provisioning key for secure secret delivery, and

3. detailed per-component appraisal data for inspection and troubleshooting.

The composite attestation relies on foundational trust assumptions. The trust model assumes a static TCB after provisioning. Any out-of-band changes— such as hot-plugging a new GPU— violate the trust contract and must be blocked, as the composite attestation result reflects the security state only at the time of evidence collection and does not support asynchronous updates without full re-attestation. Additionally, it is assumed that to prevent lateral data leakage, each GPU within the TCB confines any secrets released by the relying party to its own isolated execution environment. Confidential data sharing with other GPUs over peer-to-peer interfaces (e.g., NVLink) is assumed to be disallowed as part of this trust model.

4. EAR claims for TDX + C-GPU composite attestation

4.1. JWT claims

The following claims are reused from the IETF [JWT] specification. The complete definitions of the claims are available in the JSON Web Token (JWT) specification.

iat
: The "iat" (issued at) claim identifies the time at which the JWT was issued.

iss
: The "iss" (issuer) claim identifies the principal that issued the JWT.

jti
: The "jti" (JWT ID) claim provides a unique identifier for the JWT.

nbf
: The "nbf" (not before) claim identifies the time before which the JWT MUST NOT be accepted for processing.

exp
: The "exp" (expiration time) claim identifies the expiration time on or after which the JWT MUST NOT be accepted for processing.

4.2. EAT claims

The following claims are reused from the EAT specification. The complete definitions of the claims are available in the EAT specification.

eat_profile
: The "eat_profile" claim identifies an Entity Attestation Token (EAT) profile by either a URL or an OID.

eat_nonce (optional)
: An EAT nonce is either a byte or text string or an array of byte or text strings representing verifier response freshness. The array option supports multistage EAT verification and consumption.

4.3. EAR claims

The following claims are reused from the IETF draft EAT Attestation Results (EAR) message format. The complete definitions of the claims are available here)

ear_status
: The string represents the aggregated appraisal status across all attesters, reflecting the composite attestation result. (check the latest defn in EAR profile V4. Current preference is to reflect the min baseline of all)

ear_verifier_id
: The strings represents identifying information about the software and organizational unit that performed the attestation appraisal.

ear_raw_evidence (optional)
: The strings represents the unabridged evidence submitted for appraisal, including any signed container or envelope.

ear_all_submods_bound (optional)
: A string value indicating whether all submod components in the EAR token are provably bound to each other ("true", "false", "unknown").

ear_evidence_nonce (optional)
: if all submods share the same value for eat_nonce, the value may be replicated as a top level claim

submods
: A submodule map holding one EAR-appraisal for each separately appraised attester.

ear_status
: The strings represents the appraisal status for an attester as one of the defined trustworthiness tiers.

eat_profile (optional)
: The "eat_profile" claim identifies an Entity Attestation Token (EAT) profile by either a URL or an OID.

eat_nonce (optional)
: The claim represents evidence freshness

ear_trustworthiness_vector
: The AR4SI trustworthiness vector giving a breakdown of appraisal values for an attester.

ear_appraisal_policy_ids** (optional)
: A list of one or more unique identifiers for appraisal policies used to evaluate the attestation results.

ear_evidence_claims
: A JSON object containing the normalized, attester-reported evidence claims that the verifier accepted as input to its appraisal of this submod. The contents are organized as a flat or nested map of named claims defined by the submod’s profile (for TDX, see section 3.4; for the CVM guest, see section 3.5; for C-GPU, see section 3.6). Values in this object originate from the attester (or are derived directly from attester-supplied evidence) and are reproduced here verbatim after parsing and schema validation; the verifier does not add appraisal verdicts, reference-value comparisons, or trust judgements to claims under this object.

ear_verifier_claims
: A JSON object containing claims that are produced by the verifier itself as a result of appraising this submod. These claims are not present in the attester evidence and are added by the verifier to convey appraisal context, reference-data state, and verifier-derived dispositions.

ear_managed_keysets (optional)
: A JSON object that carries one or more named key sets extracted from the attestation evidence by the verifier on behalf of the attester, intended for use by the relying party (for example, to deliver secrets into the verified Trusted Compute Base). Each property of the object is a key-set name (e.g., ephemeral-transfer-keys) whose value is an array of JSON Web Keys (JWKs, per RFC 7517).

4.4. TDX claims

4.4.1. ear_evidence_claims

The following attester-reported claims appear as named members of the tdx submod's ear_evidence_claims container:

tdx_mrconfigid
: The hexadecimal string represents a byte array of length 48, which contains the software-defined ID for non-owner-defined configuration of the TDX, e.g., runtime or OS configuration.

tdx_mrowner
: The hexadecimal string represents a byte array of length 48, which contains the software-defined ID for the TDX' s owner.

tdx_mrownerconfig
: The hexadecimal string represents a byte array of length 48, which contains the software-defined ID for owner-defined configuration of the TDX, e.g., specific to the workload rather than the runtime or OS.

tdx_mrseam
: The hexadecimal string represents a byte array of length 48, which contains the measurement of the Intel TDX module.

tdx_mrsignerseam
: The hexadecimal string represents a byte array of length 48, which contains the measurement of the TDX module signer.

tdx_mrtd
: The hexadecimal string represents a byte array of length 48, which contains the measurement of the initial contents of the TDX.

tdx_report_data
: The hexadecimal string represents a byte array of length 64. In this context, the TDX has the flexibility to include 64 bytes of custom data in a TDX Report. For instance, this space can be used to hold a nonce, a public key, or a hash of a larger block of data.

tdx_rtmr0 tdx_rtmr3
: Each hexadecimal string represents a byte array of length 48, which contains the runtime extendable measurement register.

tdx_seam_attributes
: The hexadecimal string represents a byte array of length 8, which contains additional configuration of the TDX module.

tdx_seamsvn
: The number represents the Intel TDX module security version number (SVN).

tdx_td_attributes
: The hexadecimal string represents a byte array of length 8. These are the attributes associated with the Trusted Domain (TD).

tdx_td_attributes_debug
: The boolean value represents whether the TD runs in TD debug mode (set to 1) or not (set to 0). In TD debug mode, the CPU state and private memory are accessible by the host VMM.

tdx_td_attributes_key_locker
: The boolean value represents whether the TD is allowed to use Key Locker.

tdx_td_attributes_perfmon
: The boolean value represents whether the TD is allowed to use Perfmon and PERF_METRICS capabilities.

tdx_td_attributes_protection_keys
: The boolean value represents whether the TD is allowed to use Supervisor Protection Keys.

tdx_td_attributes_septve_disable
: The boolean value represents whether to disable EPT violation conversion to #VE on TD access of PENDING pages.

tdx_tee_tcb_svn
: The hexadecimal string represents a byte array of length 16, which describes the TCB SVNs of TDX.

tdx_xfam
: The hexadecimal string represents a byte array of length 8, which contains a mask of CPU extended features that the TDX is allowed to use.

sgx_tcb_comp_svn
: The hexadecimal string represents the array of security version numbers (SVNs) for Intel SGX TCB components.

pce_svn
: The integer value represents the security version number (SVN) of the Intel SGX Provisioning Certification Enclave (PCE), which is part of the TDX TCB.

platform_instance_id
: The hexadecimal string represents a byte array of length 16, generated during Intel TDX Initial Platform Establishment (IPE), that uniquely identifies a specific physical platform instance.

4.4.2. ear_verifier_claims

The following verifier-derived claims appear as named members of the tdx submod's ear_verifier_claims container:

attester_tcb_date
: The date-time string is in UTC and encoded using ISO 8601, and it represents the date of the evaluated TCB level.

attester_tcb_status
: The string describes the evaluated status of the attesting platform TCB level.

attester_advisory_ids
: The array of advisory IDs refers to Intel security advisories that explain the reason(s) for the attester_tcb_status value of the evaluated platform TCB level.

tdx_collateral
: The metadata of Intel Provisioning Certification Service (PCS) TDX collateral that the verifier used to appraise the attesting platform' s quote. Specifically: tcbevaluationdatanumber (TCB Evaluation Data Number) represents the version of the TDX verification collateral, and fmspc indicates the FMSPC associated with that collateral.

4.5. CVM claims

The following claim appears as a peer of the other submod-level EAR claims (e.g., ear_appraisal_policy_id) within the cvm_guest submod:

ear_azurevm_policy_hash
: The base64url-encoded string represents the hash (SHA-256) of the Azure VM guest attestation appraisal policy that the verifier evaluated to produce the cvm_guest submod result.

4.5.1. ear_evidence_claims

The following attester-reported claims appear as named members of the cvm_guest submod’s ear_evidence_claims container (see section 3.3):

secureboot
: The boolean value represents whether secure boot is enabled.

azurevm_attestation_protocol_ver
: The string value represents the version of the Azure VM attestation protocol used to generate the attestation token.

azurevm_attested_pcrs
: The array represents PCR indices included in the TPM quote and successfully validated by the service.

azurevm_bootdebug_enabled
: The boolean value represents whether boot debugging was enabled for the Azure VM at boot time.

azurevm_dbvalidated
: The boolean value represents whether the UEFI Secure Boot signature database (DB) was successfully validated during boot.

azurevm_dbxvalidated
: The boolean value represents whether the UEFI Secure Boot revocation database (DBX) was successfully validated.

azurevm_debuggersdisabled
: The boolean value represents whether kernel and user-mode debuggers were disabled in the guest operating system at boot.

azurevm_default_securebootkeysvalidated
: The boolean value represents whether the default Microsoft Secure Boot keys were present and validated during Secure Boot initialization.

azurevm_elam_enabled
: The boolean value represents whether Early Launch Anti-Malware (ELAM) was enabled, ensuring that trusted anti-malware drivers are loaded before other boot drivers.

azurevm_flightsigning_enabled
: The boolean value represents whether flight signing was enabled, allowing test or preview-signed binaries to load in the guest OS.

azurevm_hvci_policy
: The integer value represents the Hypervisor-Enforced Code Integrity (HVCI) policy configured and enforced by the guest operating system.

azurevm_hypervisordebug_enabled
: The boolean value represents whether hypervisor debugging was enabled for the Azure VM.

azurevm_is_windows
: The boolean value represents whether the guest operating system running inside the Azure VM is Microsoft Windows.

azurevm_kerneldebug_enabled
: The boolean value represents whether kernel debugging was enabled in the guest operating system at boot time.

azurevm_osbuild
: The string value represents the operating system build number of the guest OS running in the Azure VM.

azurevm_osdistro
: The string value represents the guest operating system distribution name (for example: specific Linux distribution or Windows edition).

azurevm_ostype
: The string value represents the guest operating system family or type (for example: Windows, Linux).

azurevm_osversion_major
: The integer value represents the major version number of the guest operating system.

azurevm_osversion_minor
: The integer value represents the minor version number of the guest operating system.

azurevm_signingdisabled
: The boolean value represents whether code signing enforcement was disabled, allowing unsigned binaries to be loaded.

azurevm_testsigning_enabled
: The boolean value represents whether test signing mode was enabled, allowing test-signed binaries to execute in the guest OS.

azurevm_vmid
: The string value represents the unique identifier (VM ID) assigned to the Azure Virtual Machine instance.

runtime
: A JSON object containing claims that are defined and generated within the attested environment. This includes information such as keys and client payload, which are formatted as UTF-8encoded, well-formed JSON.

4.5.2. ear_verifier_claims

The following verifier-derived claims appear as named members of the cvm_guest submod’s ear_verifier_claims container (see section 3.3):

x_ms_compliance_status
: The string value summarizes the Microsoft-defined compliance disposition of the attested CVM guest (for example, azure-compliant-cvm-guestvm indicates the guest satisfies the Azure confidential VM guest compliance baseline).

4.6. C-GPU claims

eat_profile
: The eat_profile from EAR token generated by NVIDIA verifier. This profile represents the EAR profile not evidence profile.

ear_status
: The ear_status from EAR token generated by NVIDIA verifier.

ear_nvidia_purpose
: The context associated with the appraisal. A GPU can respond out of band for infrastructure attestation and inband for various modes such as CC-TDISP. This claim allows a RP to ensure that an EAR meant for a different purpose does not get used by such RP.

ear_trustworthiness_vector (optional)
: The ear_trustworthiness_vector from EAR token generated by NVIDIA verifier.

eat_nonce (optional)
The eat_nonce from EAR token generated by NVIDIA verifier. This nonce represents evidence freshness not freshness of response from NVIDIA verifier.

ear_verifier_claims
A collection of claims generated by the verifier during the process of evidence appraisal other than any claim from evidence that verifier copies into ear_evidence_claims (explained below). ear_verifier_claims includes claims that were not part of the evidence (e.g., certificate chain related claims).

ear_verifier_claims.ear_nvidia_evidence
A collection of claims generated by the verifier based on evidence validation step prior to comparison to reference values.

ear_verifier_claims.ear_nvidia_evidence.signature_verified
This boolean value indicates whether the signature on SPDM response has been verified successfully.

ear_verifier_claims.ear_nvidia_evidence.parsed (optional)
This boolean value indicates whether the evidence has been successfully parsed. If signature verification of SPDM response fails, this claim will not be emitted.

ear_verifier_claims.ear_nvidia_evidence.cert_chain (optional)
An array of claims related to each of the certificates in the device certificate chain. Every array entry corresponds to one certificate in the chain. The certs are listed in the order from the root to the end entity cert.

ear_verifier_claims.ear_nvidia_evidence.cert_chain[].status
The string value represents the validation result of the certificate.

ear_verifier_claims.ear_nvidia_evidence.cert_chain[].ocsp_crl_status (optional)
The string value represents the certificate status from Online Certificate Status Protocol (OCSP) or CRL.

ear_verifier_claims.ear_nvidia_evidence.cert_chain[].ocsp_nonce_matches (optional)
The boolean value represents whether the nonce in the OCSP response matches the nonce sent in the OCSP request.

ear_verifier_claims.ear_nvidia_evidence.cert_chain[].expiration_date
The string value represents the expiration timestamp of the certificate.

ear_verifier_claims.ear_nvidia_evidence.cert_chain[].revocation_reason (optional)
: The string value represents the revocation reason returned by certificate status validation if the certificate has been revoked.

ear_verifier_claims.ear_nvidia_evidence.akpub (optional)
: This claim represents the public key from the end entity certificate used by the verifier to verify the signature on the SPDM response from the attester.

ear_verifier_claims.ear_nvidia_evidence.nonce_match

ear_verifier_claims.ear_nvidia_rims (optional)
: A collection of claims generated by the verifier during its attempts to acquire and validate RIMs. This claim must be emitted if the verifier decides to attempt to acquire RIMs.

ear_verifier_claims.ear_nvidia_rims[].fetched
: The boolean value indicates whether the verifier successfully retrieved the corresponding NVIDIA RIM required for evidence validation.

ear_verifier_claims.ear_nvidia_rims[].signature-verified (optional)
: The boolean value indicates that the digital signature of the RIM was successfully verified using NVIDIA's signing certificates. This claim must be emitted if the fetched claim above is true.

ear_verifier_claims.ear_nvidia_rims[].id (optional)
: The string value represents the identifier of the NVIDIA Reference Integrity Manifest (RIM).

ear_verifier_claims.ear_nvidia_rims[].cert_chain (optional)
: An array of claims related to each of the certificates in the RIM certificate chain. Every array entry corresponds to one certificate in the chain. The certs are listed in the order from the root to the end entity cert.

ear_verifier_claims.ear_nvidia_rims[].cert_chain[].status
: The string value represents the validation result of the certificate.

ear_verifier_claims.ear_nvidia_rims[].cert_chain[].ocsp_crl_status (optional)
: The string value represents the certificate status from Online Certificate Status Protocol (OCSP) or CRL.

ear_verifier_claims.ear_nvidia_rims[].cert_chain[].ocsp_nonce_matches (optional)
: The boolean value represents whether the nonce in the OCSP response matches the nonce sent in the OCSP request.

ear_verifier_claims.ear_nvidia_rims[].cert_chain[].expiration_date
: The string value represents the expiration timestamp of the certificate.

ear_verifier_claims.ear_nvidia_rims[].cert_chain[].revocation_reason (optional)
: The string value represents the revocation reason returned by certificate status validation if the certificate has been revoked.

ear_verifier_claims.ear_nvidia_evidence_rim_cmp (optional)
: A collection of claims generated by the verifier during the process of reference value corroboration. This claim must be emitted if the verifier reaches the corroboration phase.

ear_verifier_claims.nvidia_evidence_rim_cmp.matched-env
: The array of environment-maps from evidence that can be satisfied by CoRIM(s).

ear_verifier_claims.nvidia_evidence_rim_cmp.unmatched-env
: The array of environment-maps from evidence that were not found in CoRIM(s).

ear_verifier_claims.nvidia_evidence_rim_cmp.mismatched-env
: The array of environment-maps from evidence that were found in CoRIM(s) but can not be satisifed by the reference values in such CoRIM(s).

ear_verifier_claims.nvidia_evidence_rim_cmp.cert_chain_dti_match
: The boolean value indicates status of comparison of all DiceTcbInfo structures found in the cert chain of the attester to suitable environment-maps from CoRIM(s).

ear_evidence_claims (optional)
: A collection of claims copied from evidence without any comparison to ref values.

ear_evidence_claims.oemid
: This claim identifies the Original Equipment Manufacturer (OEM) of the hardware.

ear_evidence_claims.hwmodel
: This claim identifies the model of the GPU.

5. Examples

5.1. Sample TDX + C-GPU attestation token

Below is a sample TDX + C-GPU attestation token which includes claims from this EAR profile.


```
    {
 "eat_profile": "points to TDX+GPU composite EAR profile in IETF website",
 "eat_nonce": "a1b2c3d4e5f67890123456789abcdef0",
 "iat": 1666529300,
 "iss": "https://mytenant.rats.verifier.com",
 "jti": "950a20caadb27206dda48f8d9f15d550d935ba5d6d074321ea34398ca5bc5975",
 "nbf": 1764709981,
 "exp": 1764738781,
 "ear_verifier_id": {
     "developer": "https://rats.verifier.com",
     "build": "v1.23.0"
 },
 "ear_raw_evidence": "NzQ3MjY5NzM2NTYzNzQKNzQ3MjY5NzM2NTYzNzQK...",
 "ear_status": "affirming",
 "ear_all_submods_bound": "true",
 "submods": {
     "tdx": {
         "ear_status": "affirming",
         "ear_profile": "https://portal.trustauthority.intel.com/ear_profile.html",
         "ear_trustworthiness_vector": {
             "instance-identity": 2,
             "executables": 2,
             "hardware": 2
         },
         "ear_appraisal_policy_ids": [
             "tdx-default-v1"
         ],
         "ear_evidence_claims": {
             "tdx_mrconfigid": "018779f38c1cc5d1e643fbfc7238bae2c227f7ffa4c72c04980294
2658acfc5bee0000000000000000000000000000000000",
             "tdx_mrowner": "0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000",
             "tdx_mrownerconfig": "0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000",
             "tdx_mrseam": "7bf063280e94fb051f5dd7b1fc59ce9aac42bb961df8d44b709c9b0ff8
7a7b4df648657ba6d1189589feab1d5a3c9a9d",
             "tdx_mrsignerseam": "0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000",
             "tdx_mrtd": "75f3acc2e1dfc3acf404d7eaa69a2eefcd0475a0dd6516ef5ba3cb83399c
61b4aa1c638e3622bb650a514bfc6e858886",
             "tdx_report_data": "b4ee5ceb6bee96cac5605b2af1d2a483f0a6f790c6de738e7d243
0ba33a350a86d1ace43b90ea9421624738151b677a23a577e2fc0c6180ae786ca1cf91e0eac",
             "tdx_rtmr0": "59dc57e1e3029f15034b127f0264a8e6f00db6c178bb087bc03205243d5
d9daeff8bbeccced48d881c2a63d8974dbe26",
             "tdx_rtmr1": "70c3be9ff6c2fab010196a5fa1dd04c1bad37a4a1d2531cb56a9762d9f3
b59c4f20b630a572396da77f8238d8e056bc2",
             "tdx_rtmr2": "3142d2479ce77c402ce8d67a04a67d57de798fa98748ca5f0486d498591
9cad7de9a31adaa51d1073c62eafd0dfc8891",
             "tdx_rtmr3": "b56c050e7ae3846678603ab7ef552bf6452710f5601337b69b00f6e8894
f5b0a0f87b9ed39faebba3e2d740dba701d5b",
             "tdx_seam_attributes": "0000000000000000",
             "tdx_seamsvn": 258,
```

```
                "tdx_td_attributes": "0000000000000000",
                "tdx_td_attributes_debug": false,
                "tdx_td_attributes_key_locker": false,
                "tdx_td_attributes_perfmon": false,
                "tdx_td_attributes_protection_keys": false,
                "tdx_td_attributes_septve_disable": false,
                "tdx_tee_tcb_svn": "02010600000000000000000000000000",
                "tdx_xfam": "e718060000000000",
                "sgx_tcb_comp_svn": "06060202030100030000000000000000",
                "pce_svn": 11,
                "platform_instance_id": "2ba7336ce9acf49fe7d3e3625337e510"
            },
            "ear_verifier_claims": {
                "attester_tcb_date": "2025-05-14T00:00:00Z",
                "attester_advisory_ids": [ "INTEL-SA-01192","INTEL-SA-01245"],
                "attester_tcb_status": "OutOfDate",
                "tdx_collateral": {
                    "fmspc": "B0C06F000000",
                    "tcbevaluationdatanumber": 20
                }
            }
        },
        "cvm_guest": {
            "eat_profile": "https://aka.ms/eat-profile-cvm-guest/1.0.0",
            "ear_status": "affirming",
            "ear_trustworthiness_vector": {
                "instance-identity": 2,
                "executables": 2
            },
            "ear_appraisal_policy_ids": [
                "policy:cvm-guest/7e8f1b2a-9c4d-4327-b59a-8d6e1a3f0c2b"
            ],
            "ear_azurevm_policy_hash": "ndXtG3MNtueeIPCj2Y-3fDFl16CREC5FF_sUyU4fLQ8",
            "ear_managed_keysets": {
                "ephemeral-transfer-keys": [
                    {
                        "e": "AQAB",
                        "key_ops": [
                            "encrypt"
                        ],
                        "kid": "TpmEphemeralEncryptionKey",
                        "kty": "RSA",
                        "n": "zcjFQAABYsqZUkS4w"
                    }
                ]
            },
            "ear_evidence_claims": {
                "secureboot": true,
```

```
                "azurevm_attestation_protocol_ver": "2.0",
                "azurevm_attested_pcrs": [
                    0,
                    1,
                    2,
                    3,
                    4,
                    5,
                    6,
                    7
                ],
                "azurevm_bootdebug_enabled": false,
                "azurevm_dbvalidated": true,
                "azurevm_dbxvalidated": true,
                "azurevm_debuggersdisabled": true,
                "azurevm_default_securebootkeysvalidated": true,
                "azurevm_elam_enabled": false,
                "azurevm_flightsigning_enabled": false,
                "azurevm_hvci_policy": 0,
                "azurevm_hypervisordebug_enabled": false,
                "azurevm_is_windows": false,
                "azurevm_kerneldebug_enabled": false,
                "azurevm_osbuild": "NotApplication",
                "azurevm_osdistro": "Debian GNU/Linux",
                "azurevm_ostype": "Linux",
                "azurevm_osversion_major": 13,
                "azurevm_osversion_minor": 0,
                "azurevm_signingdisabled": true,
                "azurevm_testsigning_enabled": false,
                "azurevm_vmid": "59ECD20B-CD92-4A84-82CB-9F3F06E9CDEC",
                "runtime": {
                    "client_payload": {
                        "Nonce": "MaaSandbox Nonce : 12/2/2025 9:13:01 PM",
                        "RelyingPartyId": "bcd368ce93bdad7c2f67bfd7af0d6b052c127aec28802c
376f54a6ca8712ae32"
                    },
                    "keys": [
                        {
                            "e": "AQAB",
                            "key_ops": [
                                "encrypt"
                            ],
                            "kid": "TpmEphemeralEncryptionKey",
                            "kty": "RSA",
                            "n": "zcjFQAABYsqZUke3aw"
                        }
                    ]
                }
            },
```

```
            "ear_verifier_claims": {
                "x_ms_compliance_status": "azure-compliant-cvm-guestvm"
            }
        },
        "gpu_0": {
            "eat_profile": "tag:nvidia.com,2026-05:ear/profiles/gpu/1.0",
            "ear_status": "affirming",
            "ear_nvidia_purpose": "CC-Bounce-Buffer",
            "ear_trustworthiness_vector": {
                "configuration": 2,
                "executables": 2,
                "hardware": 2
            },
            "ear_appraisal_policy_ids": [
                "tag:nvidia.com,2026-05:ear/profiles/composite/generic/1.0.0",
                "https://nras.attestation.nvidia.com/ear/policies/gpu/1.1"
            ],
            "eat_nonce": "80FH7byULVei4u1YP4EirV8B7oHxIq0/1C3wE6vJ8ouq9j+F6m1X/dWO6B2qoov
v",
            "ear_verifier_claims": {
                "ear_nvidia_evidence": {
                    "signature_verified": true,
                    "parsed": true,
                    "cert_chain": [
                        {
                            "status": "valid",
                            "ocsp_status": "good",
                            "expiration_date": "2036-07-15T23:02:10Z",
                            "revocation_reason": null
                        },
                        {
                            "status": "valid",
                            "ocsp_status": "good",
                            "expiration_date": "2032-07-15T23:02:10Z",
                            "revocation_reason": null
                        },
                        {
                            "status": "valid",
                            "ocsp_status": "good",
                            "expiration_date": "2028-07-15T23:02:10Z",
                            "revocation_reason": null
                        },
                        {
                            "status": "valid",
                            "ocsp_status": "good",
                            "expiration_date": "2026-07-15T23:02:10Z",
                            "revocation_reason": null
                        }
                    ],
```

```
                    "akpub": "-----BEGIN PUBLIC KEY-----
       \nMHYwEAYHKoZIzj0CAQYFK4EEACIDYgAEX0dHnbKG8XlTZk1LrNBFYxca/xomeYVQ\nHHnCksh1BXEBsJ
t4wIUjkPuTXqy1NLThQXL6m3zgP7unKAeThOKSiGr4/D9n6XMg\noFJGZMFgQYQsc3ZY+SogfgDTf5cEGaeQ\n---
--END PUBLIC KEY-----\n",
                    "nonce_match": true
                },
                "ear_nvidia_rims": [
                    {
                        "fetched": true,
                        "signature_verified": true,
                        "id": "ID-Driver",
                        "cert_chain": [
                            {
                                "status": "valid",
                                "ocsp_status": "good",
                                "expiration_date": "2036-07-15T23:02:10Z",
                                "revocation_reason": null
                            },
                            {
                                "status": "valid",
                                "ocsp_status": "good",
                                "expiration_date": "2032-07-15T23:02:10Z",
                                "revocation_reason": null
                            },
                            {
                                "status": "valid",
                                "ocsp_status": "good",
                                "expiration_date": "2028-07-15T23:02:10Z",
                                "revocation_reason": null
                            },
                            {
                                "status": "valid",
                                "ocsp_status": "good",
                                "expiration_date": "2026-07-15T23:02:10Z",
                                "revocation_reason": null
                            }
                        ]
                    },
                    {
                        "fetched": true,
                        "signature_verified": true,
                        "id": "ID-Vbios",
                        "schema_validated": true,
                        "measurements_available": true,
                        "cert_chain": [
                            {
                                "status": "valid",
                                "ocsp_status": "good",
                                "expiration_date": "2036-07-15T23:02:10Z",
                                "revocation_reason": null
```

```
                            },
                            {
                                "status": "valid",
                                "ocsp_status": "good",
                                "expiration_date": "2032-07-15T23:02:10Z",
                                "revocation_reason": null
                            },
                            {
                                "status": "valid",
                                "ocsp_status": "good",
                                "expiration_date": "2028-07-15T23:02:10Z",
                                "revocation_reason": null
                            },
                            {
                                "status": "valid",
                                "ocsp_status": "good",
                                "expiration_date": "2026-07-15T23:02:10Z",
                                "revocation_reason": null
                            }
                        ]
                    }
                ],
                "ear_nvidia_evidence_rim_cmp": {
                    "matched_env": [
                        {
                            "class": {
                                "vendor": "NVIDIA",
                                "model": "GB100 A01 FSP",
                                "layer": 0
                            },
                            "instance": { "type": "ueid", "value": "AQIDBAUGBwgJCgsMDQ4P"
 }
                        },
                        {
                            "class": {
                                "class_id": { "type": "oid", "value": "2.23.133.5.4.1" },
                                "vendor": "NVIDIA",
                                "model": "GB100 HW config"
                            }
                        }
                    ],
                    "unmatched_env": [
                        {
                            "class": {
                                "vendor": "NVIDIA",
                                "model": "GB100 Fuses"
                            }
                        },
```

```
                        {
                            "class": {
                                "vendor": "NVIDIA",
                                "model": "GB100 Firmware microcodes (BootComplex reset do
main)"
                            }
                        }
                    ],
                    "mismatched_env": [],
                    "cert_chain_dti_match": true
                }
            },
            "ear_evidence_claims": {
                "oemid": "5703",
                "hwmodel": "R0gxMDA="
            }
        }
    }
}
```


6. References

6.1. 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/info/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/info/rfc8174>.

[EAT] Lundblade, L., Mandyam, G., O'Donoghue, J., and C. Wallace, "The Entity Attestation Token (EAT)", 30 June 2023, <https://datatracker.ietf.org/doc/html/draft-ietf-rats-eat>.

[JWT] Jones, M., Bradley, J., and N. Sakimura, "JSON Web Token (JWT)", May 2015, <https://datatracker.ietf.org/doc/html/rfc7519>.

[TDX] Intel, "Intel Trust Domain Extensions", February 2023, <https://www.intel.com/content/www/us/en/developer/tools/trust-domain-extensions/overview.html>.

Authors' Addresses

Greg Kostal
Microsoft
Email: gkostal@microsoft.com

Raghuram Yeluri
Intel
Email: raghuram.yeluri@intel.com

Dhawal Kumar
Nvidia
Email: dkumar@nvidia.com

Sindhuri Dittakavi
Microsoft
Email: sindhuri.dittakavi@microsoft.com

Haidong Xia
Intel
Email: haidong.xia@intel.com

Jerry Yu
Intel
Email: jerry.yu@intel.com