Remote Attestation of Physical AHV Host and NVIDIA GPU

Beia Cabrera Sanchez

Mentor: Sridhar Kotturu

NUTANIX

Remote attestation of physical AHV Host using TPM

NUTANIX

What is Remote Attestation?

A process to verify a system's trustworthiness from a remote location. It confirms that the system's hardware and software are in a known, good state and have not been tampered with.

01

Device Node

The system being attested. It has a TPM and generates an attestation quote.

02

Service Node

verifier/relying party that receives the attestation evidence from the DN, checks the PCA-issued credential, verifies the quote and then returns a token/service decision.

03

Privacy-CA

the trusted issuer that anonymizes/validates device identity (EK/AK) and issues the credential used to sign the quote

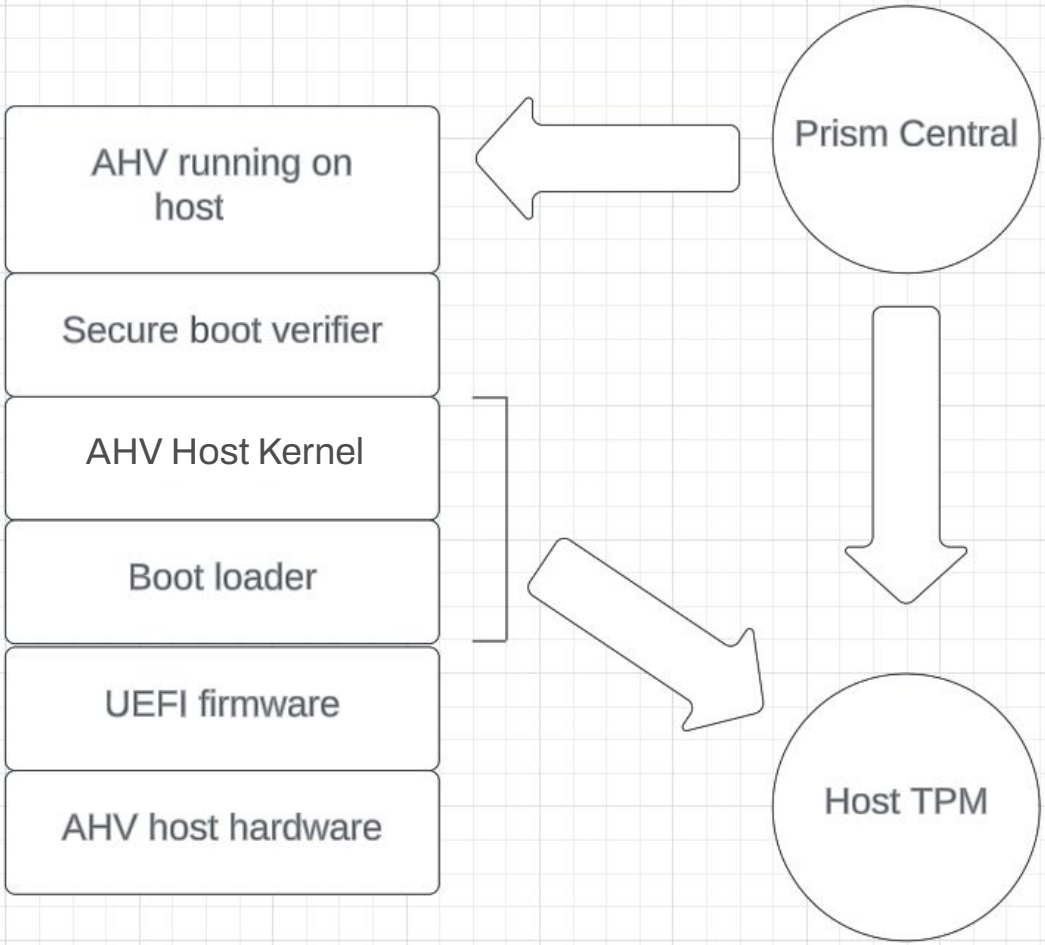
Role of Platform Configuration Registers (PCRs)

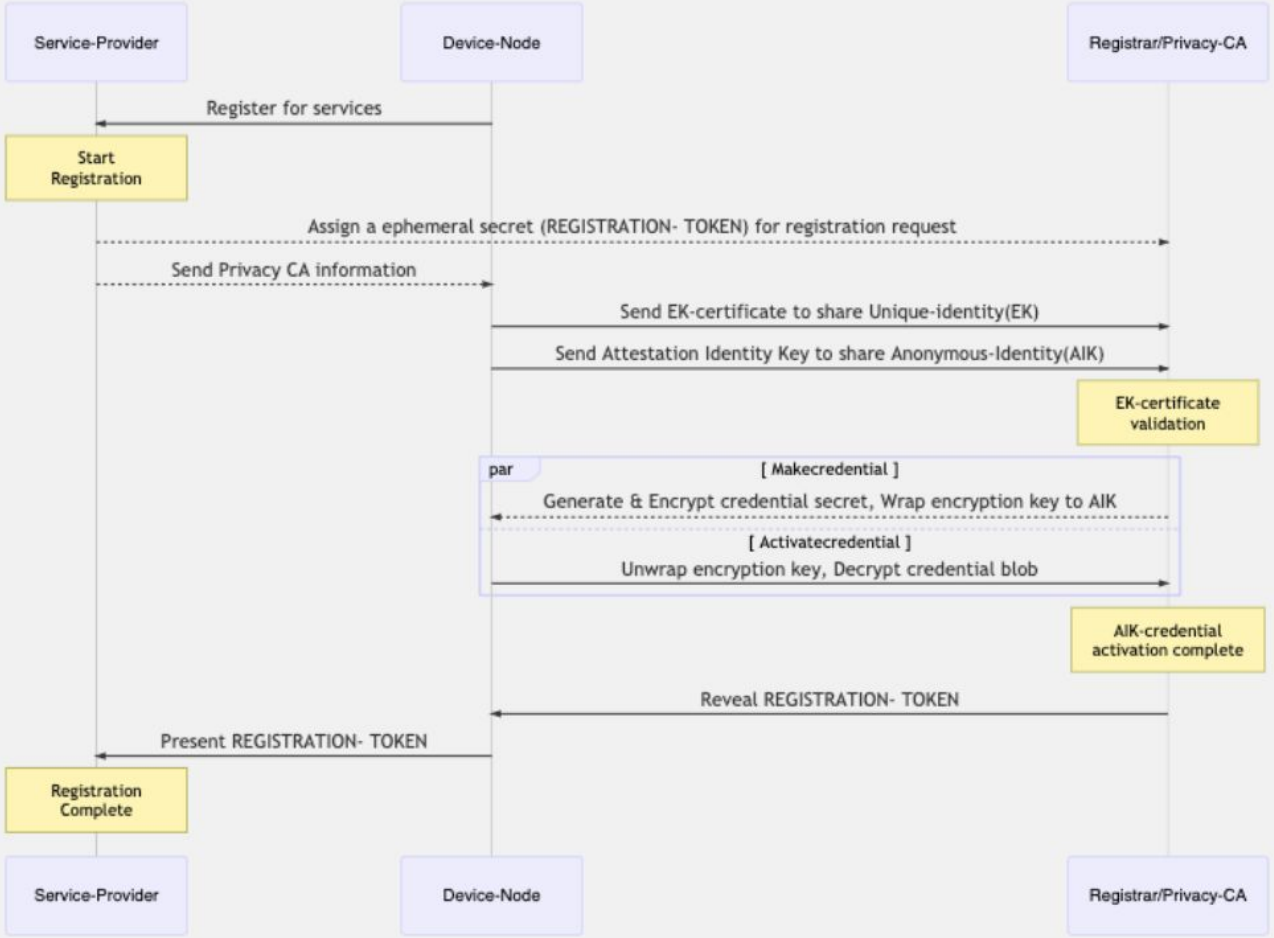
PCRs: Platform Configuration Registers
They act as a cryptographic log of the system's software state.

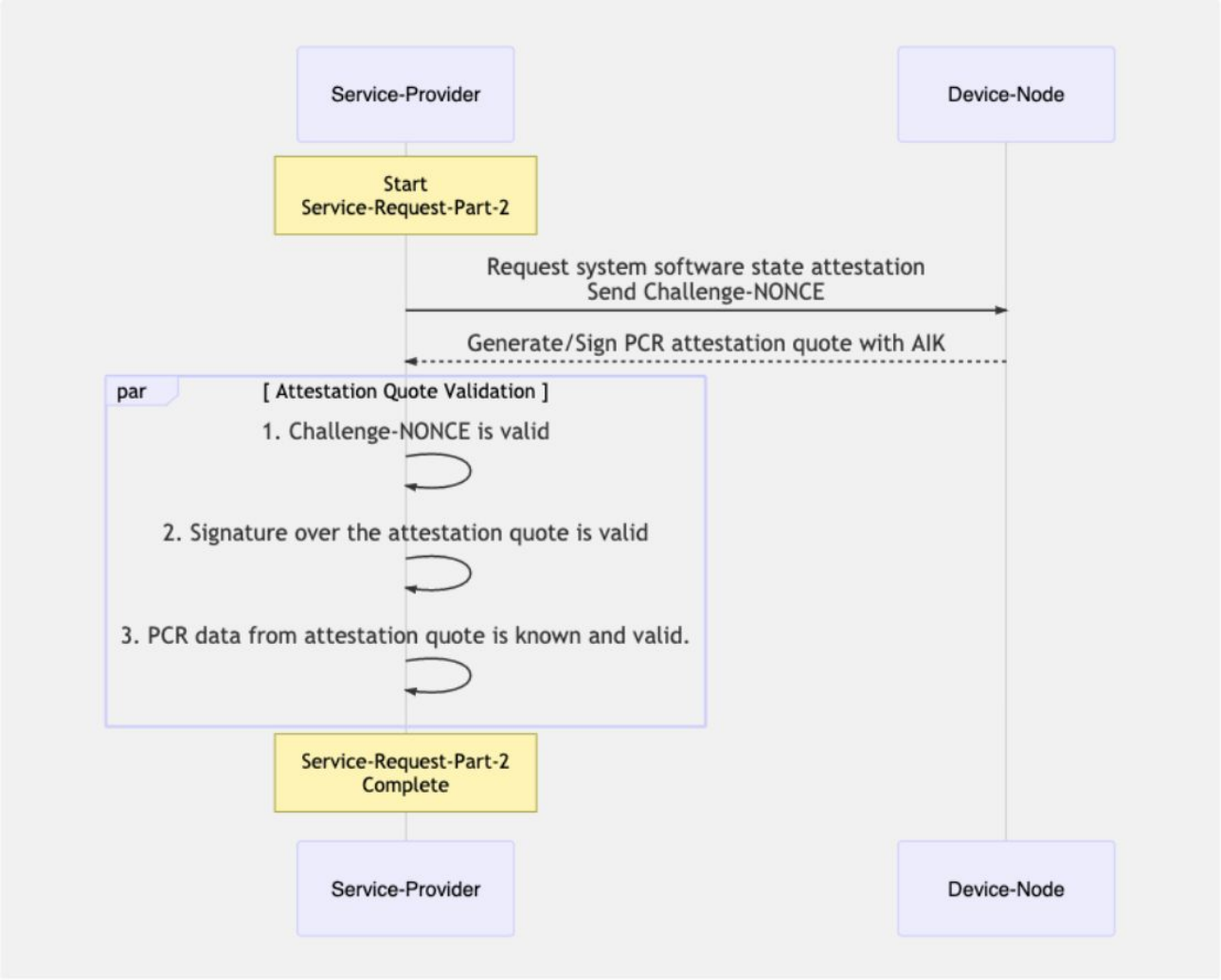
- Each value is built using hash extensions
- This forms a secure chain of trust
- Part of the core design of the TPM (Trusted Platform Module)

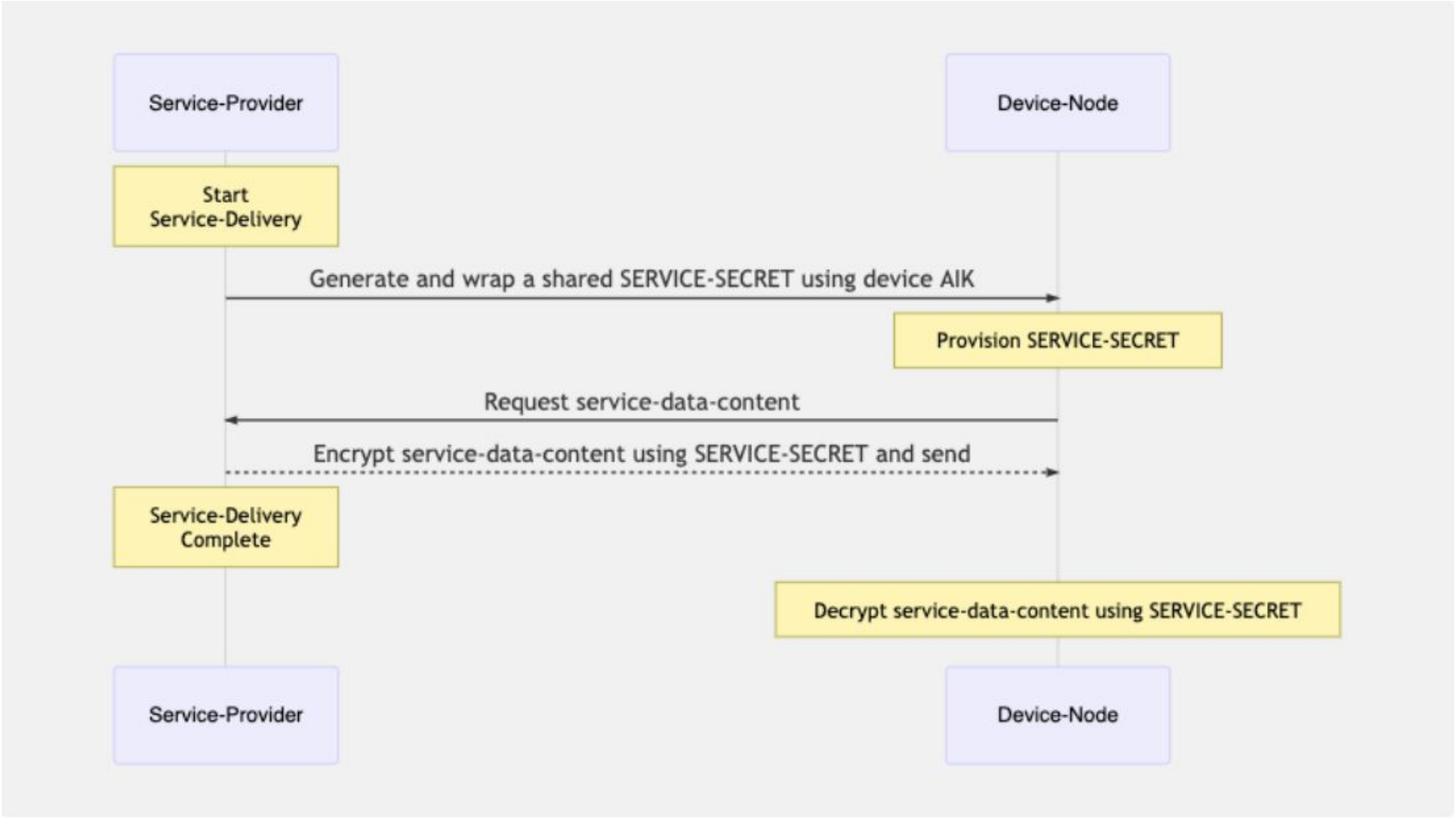
TPM Components

1. Key Components of a TPM:
   a. PCR cryptographically records software state & use measured boots as they store hashes that cannot change once written until power off or reboot
   b. Endorsement Key: serves as an identity for machines starting from the manufacturing level and acting as the root of trust
2. Measured Boot: creates a tamper-evident log of the boot process by taking cryptographic measurements (hashes) of each loaded component.
3. Secure Boot: ensures that only verified and trusted software runs during the computer's startup. It prevents malicious software from loading before the operating system.


AHV running on host
Secure boot verifier
AHV Host Kernel
Boot loader
UEFI firmware
AHV host hardware
Prism Central
Host TPM


Service-Provider
Device-Node
Registrar/Privacy-CA
Register for services
Start Registration
Assign a ephemeral secret (REGISTRATION- TOKEN) for registration request
Send Privacy CA information
Send EK-certificate to share Unique-identity(EK)
Send Attestation Identity Key to share Anonymous-Identity(AIK)
EK-certificate validation
par
[Makecredential]
Generate & Encrypt credential secret, Wrap encryption key to AIK
[Activatecredential]
Unwrap encryption key, Decrypt credential blob
AIK-credential activation complete
Reveal REGISTRATION- TOKEN
Present REGISTRATION- TOKEN
Registration Complete
Service-Provider
Device-Node
Registrar/Privacy-CA


Service-Provider
Device-Node
Start
Service-Request-Part-2
Request system software state attestation
Send Challenge-NONCE
Generate/Sign PCR attestation quote with AIK
par
[Attestation Quote Validation]
1. Challenge-NONCE is valid
2. Signature over the attestation quote is valid
3. PCR data from attestation quote is known and valid.
Service-Request-Part-2
Complete
Service-Provider
Device-Node


Service-Provider
Device-Node
Start
Service-Delivery
Generate and wrap a shared SERVICE-SECRET using device AIK
Provision SERVICE-SECRET
Request service-data-content
Encrypt service-data-content using SERVICE-SECRET and send
Service-Delivery
Complete
Decrypt service-data-content using SERVICE-SECRET
Service-Provider
Device-Node

We think long-term

GPU Remote Attestation: High Level

NUTANIX

3 Main Stages

1

Add a Verifier

1. tell SDK what to attest and where to verify

2

Collect evidence & attest

SPDM evidence → verifier/NRAS)

3

Validate Results

check token against policy/JWKS

Hard Prerequisites

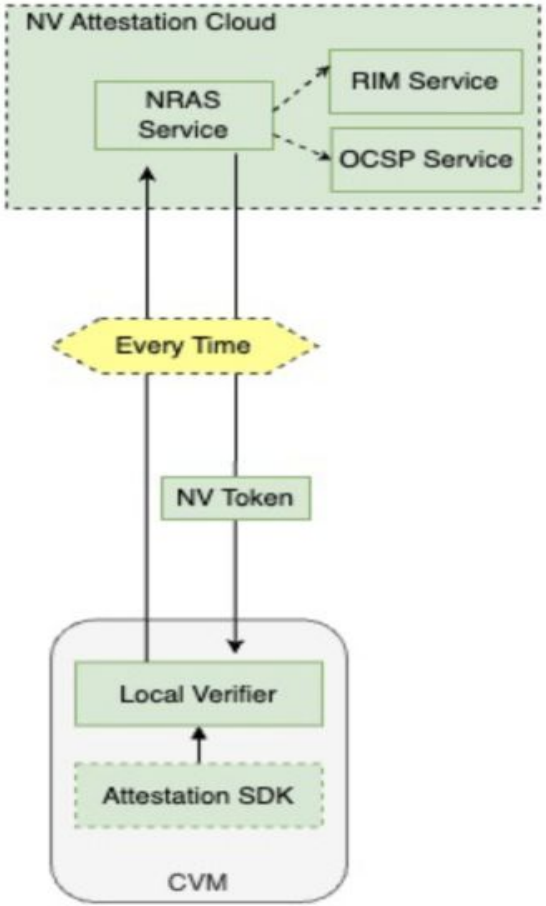
Hardware / firmware

- GPU: NVIDIA H100 (CC supported)
- CPU/Platform (one of):
  - AMD SEV-SNP enabled in BIOS.
  - Or Intel with TDX enabled in BIOS.
- BIOS: UEFI Secure Boot enabled; IOMMU enabled
- BMC up to date (used to out-of-band toggle CC on many servers).
  - Toggle GPU CC mode (out-of-band)
    - In the BMC UI, look for something like GPU Settings → Confidential Compute (or Security):
    - Set Confidential Compute = Enabled (CC-On), or
    - Set Protected PCIe = Enabled (lab/test mode).

Option 1 - NVIDIA-Managed Attestation (Most Secure)

What it is: The Confidential VM (CVM) (or PPCIE host) calls NRAS for every attestation. NRAS talks to NVIDIA RIM/OCSP, verifies everything, and returns a signed token.

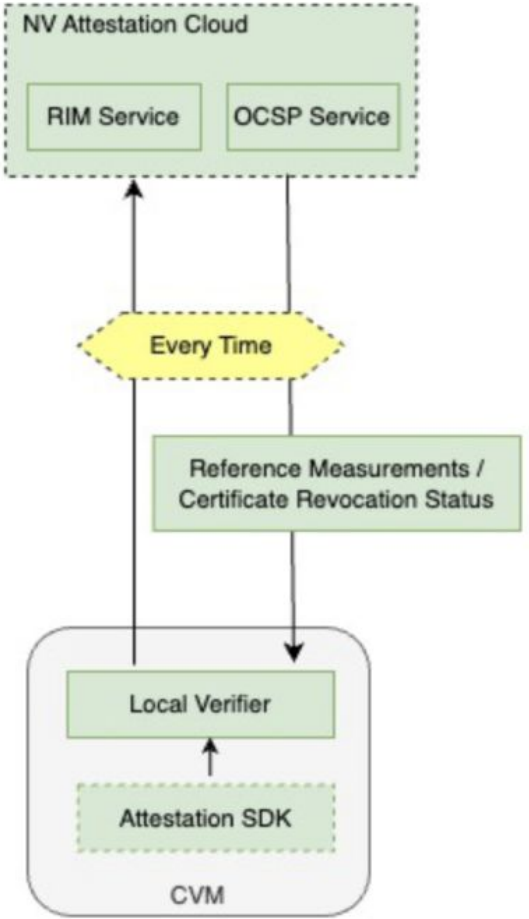
When to choose: You want strongest security + simplest operations.



Option 2 - Hybrid Attestation

What it is: You collect evidence locally and verify locally, but still fetch RIM (golden measurements) and OCSP (cert revocation) from NVIDIA cloud.

When to choose: You want local verification & policy control, but don't want to host RIM/OCSP.

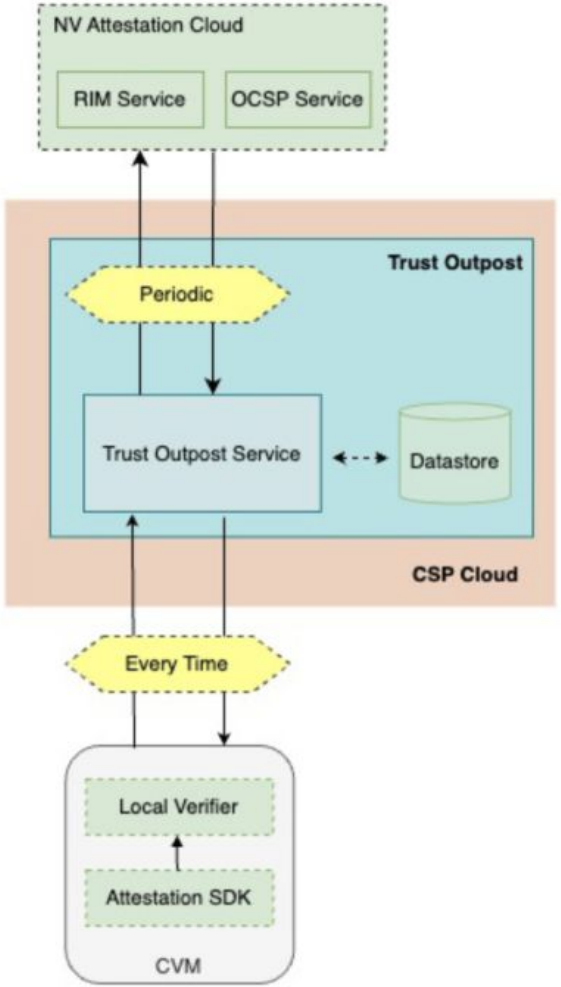


Option 3 - Customer-Managed (“Trust Outpost”)

What it is: NVIDIA provides Trust Outpost so you can keep RIM/OCSP caches on-prem and run the verification pipeline locally. Ideal for strict data-sovereignty or air-gapped environments.

When to choose: You need the highest local control / lowest cloud dependency.

Requires NVIDIA enablement and deployment artifacts.
attestation-support@nvidia.com.



Which option to start with?

If you have outbound access and want the cleanest path: *Option 1 (NVIDIA-Managed)*.

If you want local control but can still reach RIM/OCSP: *Option 2 (Hybrid)*.

If you must be air-gapped / sovereign: *Option 3 (Trust Outpost)* via NVIDIA.

Verify / Attest with nvtrust - local verifier (cc_admin)

PPCIE path:

```shell
source /opt/venvs/nvtrust/bin/activate
python3 -m verifier.cc_admin --user_mode -v
```

Full CC attestation (requires a CVM: SEV-SNP/TDX), run inside the CVM:

```shell
source /opt/venvs/nvtrust/bin/activate
python3 -m verifier.cc_admin -v
```

Testing with (no hardware):

```shell
python3 -m verifier.cc_admin --test_no_gpu -v
```

Quick Checklist for workflow

1. Update BIOS: enable SEV-SNP (AMD) or TDX (Intel); ensure UEFI Secure Boot, IOMMU
2. Update BMC & system firmware.
3. Install NVIDIA driver; reboot; confirm `nvidia-smi` has H100.
4. Toggle GPU CC mode (preferred: CC-On).
    - Re-check with `nvidia-smi -q` (Confidential Compute / Protected PCIe sections).
5. Create a CVM (SEV-SNP or TDX); pass through the H100.
6. Inside the CVM: install NVIDIA driver; start `nvidia-persistenced`; confirm CC-On in `nvidia-smi -q`.
7. Run `python3 -m verifier.cc_admin --user_mode -v`.

References

- NVIDIA nvtrust github
- NVIDIA: Secure communication with guests using AMD SEV-SNP
- Confidential Computing on NVIDIA H100 GPUs for Secure and Trustworthy AI

Questions