linaro Connect 2025

Existing Technologies Survey

Illustrative Examples: NVIDIA and AMD

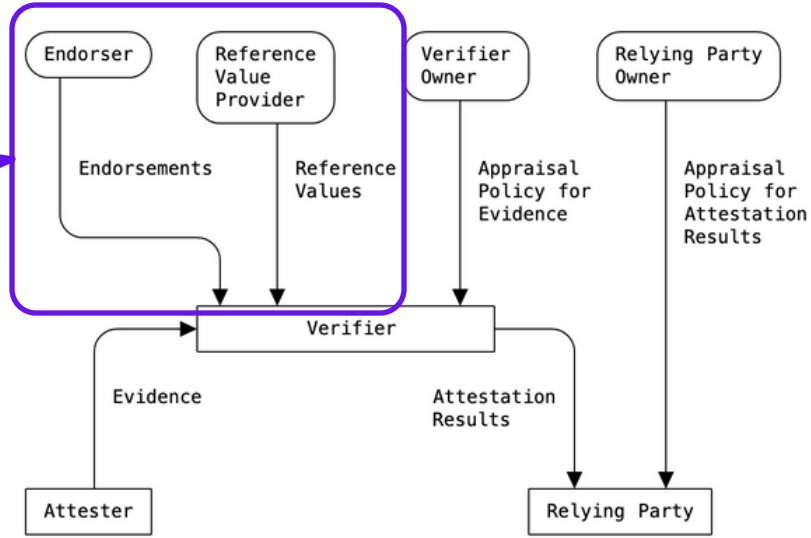
Introduction

AMD

Key Distribution Service (KDS)

NVIDIA

RIM Service

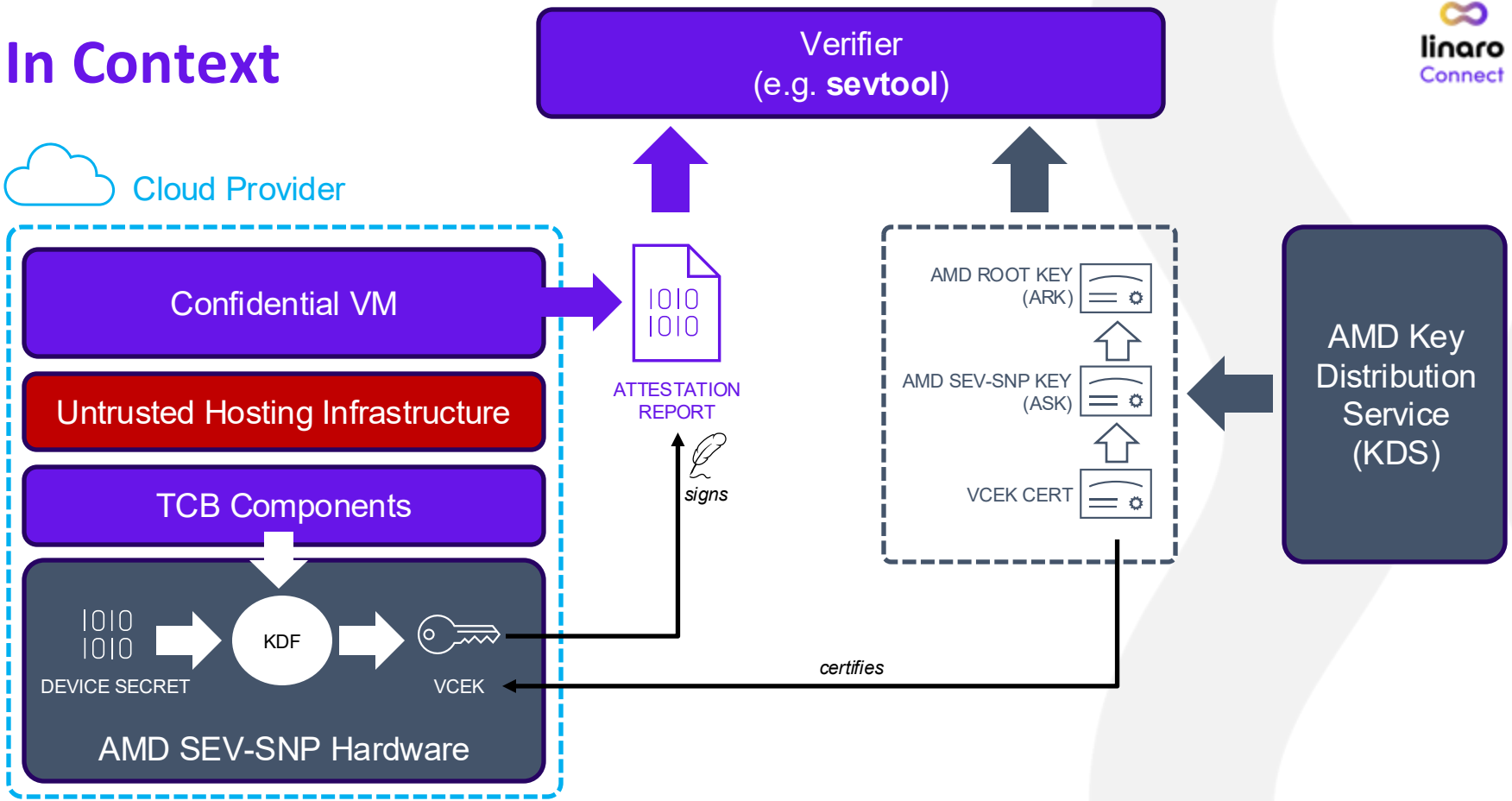


Why These Two?

- Illustrative, **not exhaustive**
- Represents a **demonstrative spread** across the problem space
  - Different artefact types: reference values vs. endorsed values
  - Different data formats: TCG RIM, CoRIM, X.509
  - Class-level vs. instance-level addressing of artefacts
  - Some non-trivial querying patterns (e.g. stateful environment)
  - Different use cases: signature verification vs. appraisal
- These disparities are useful when seeking common abstractions that are functional and flexible

AMD Key Distribution Service (KDS)

In Context


Verifier
(e.g. sevtool)
Cloud Provider
Confidential VM
Untrusted Hosting Infrastructure
TCB Components
DEVICE SECRET
KDF
VCEK
AMD SEV-SNP Hardware
ATTESTATION REPORT
signs
certifies
AMD ROOT KEY (ARK)
AMD SEV-SNP KEY (ASK)
VCEK CERT
AMD Key Distribution Service (KDS)

Overview

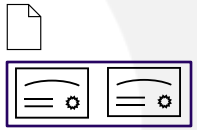
- **Purpose**
  - AMD KDS provides endorsement certificates (VCEKs) for SEV-SNP enabled AMD CPUs
- **Scope**
  - Designed for confidential computing platforms, particularly in cloud environments
- **Goal**
  - Allow Verifiers to fetch chip-specific public keys required to validate attestation reports
  - Allow Verifiers to establish trust in public keys by chaining them back to an AMD root certificate

Detail

- **Discovery**
  - Base URL https://kdsintf.amd.com
  - No enumeration of resources – caller must know what to request (based on Evidence)

- **Fetch certificate chain for product family**

```
GET vcek/v1/{product_name}/cert_chain
```



AMD root cert and product cert in PEM format

- **Fetch CRL for product family**

```
GET vcek/v1/{product_name}/crl
```

CRL in DER format (section 5 of RFC5280)

- **Fetch VCEK for CPU instance and TCB**

```
GET vcek/v1/{product_name}/{chip_id}?{tcb_parameters}
```

VCEK.CRT

Examples

- **Fetch the Cert Chain for “Milan” Processors**
  - **Query**

```
$ curl -s https://kdsintf.amd.com/vcek/v1/Milan/cert_chain
```

  - **Output (abridged)**


```
-----BEGIN CERTIFICATE-----
MIIGiTCCBDigAwIBAgIDAQABMEYGCSqGSIb3DQEBCjA5oA8wDQYJYIZIAWUDBAIC
BQChHDAaBgkqhkiG9w0BAQgwDQYJYIZIAWUDBAICBQCiAwIBMKMDAgEBMHsxFDAS
...
-----END CERTIFICATE-----
-----BEGIN CERTIFICATE-----
MIIGYzCCBBKgAwIBAgIDAQAAMEYGCSqGSIb3DQEBCjA5oA8wDQYJYIZIAWUDBAIC
BQChHDAaBgkqhkiG9w0BAQgwDQYJYIZIAWUDBAICBQCiAwIBMKMDAgEBMHsxFDAS
...
-----END CERTIFICATE-----
```

Examples

- **Fetch A VCEK**
  - **Query (**uses **Content-Disposition** header to save as **vcek.crt)**

```
$ curl -s
https://kdsintf.amd.com/vcek/v1/Milan/3ac3fe21e13fb0990eb28a802e3fb6a29483a6b0753590c951bdd3b8e53786184ca39e3596
69a2b76a1936776b564ea464cdce40c05f63c9b610c5068b006b5d -OJ
```

  - **Display X.509**

```
$ openssl x509 -in vcek.crt -text -noout
```

  - **Output (next slide)**

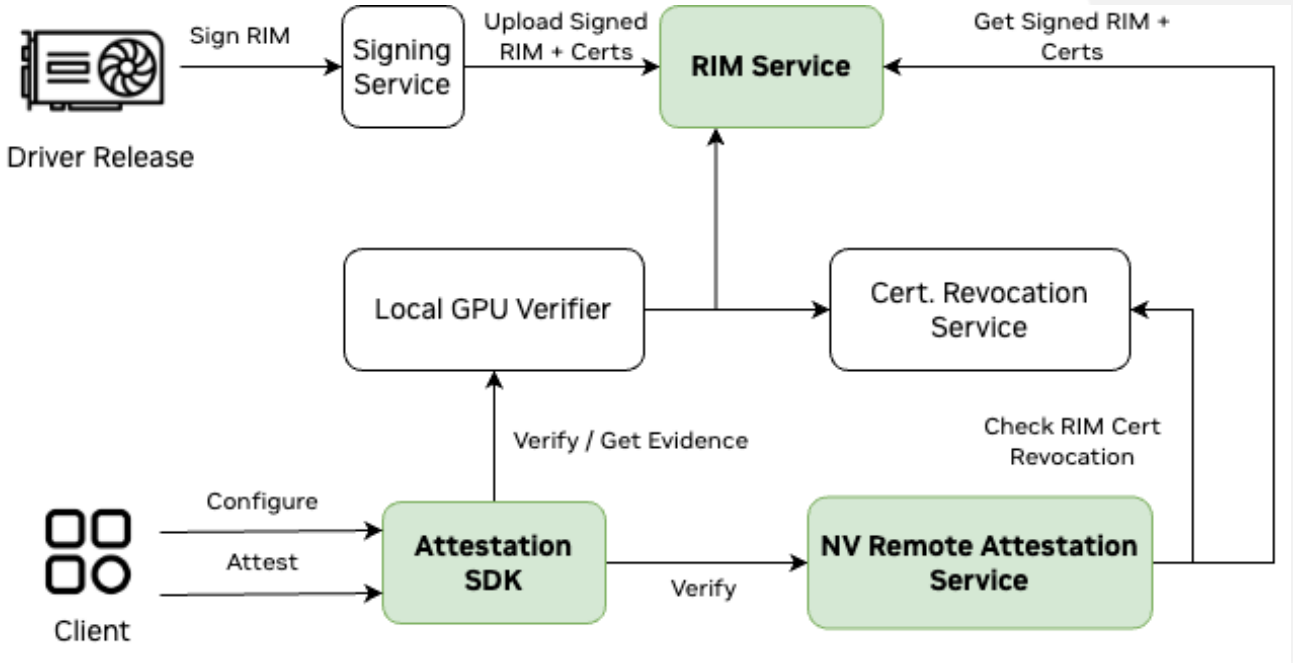
Examples

- **Output (abridged)**

```
Certificate:
  Data:
    Version: 3 (0x2)
    Serial Number: 0 (0x0)
    [...]
    Issuer: OU=Engineering, C=US, L=Santa Clara, ST=CA, O=Advanced Micro Devices, CN=SEV-Milan
    Validity
      Not Before: May  8 15:06:22 2025 GMT
      Not After : May  8 15:06:22 2032 GMT
    Subject: OU=Engineering, C=US, L=Santa Clara, ST=CA, O=Advanced Micro Devices, CN=SEV-VCEK
    Subject Public Key Info:
      [...]
    X509v3 extensions:
      1.3.6.1.4.1.3704.1.1:
        ...
      1.3.6.1.4.1.3704.1.2:
        ..Milan-B0
      1.3.6.1.4.1.3704.1.3.1:
        ...
      1.3.6.1.4.1.3704.1.3.2:
        ...
```

NVIDIA RIM Service

In Context



https://docs.nvidia.com/attestation/overview-attestation-suite/latest/introduction.html

Overview

- **Purpose**
  - The NVIDIA RIM Service provides signed Reference Integrity Manifests (RIMs) to support platform attestation
- **Scope**
  - Designed for NVIDIA DPUs, GPUs and related secure devices
- **Goal**
  - Allow Verifiers to validate attestation Evidence using known-good measurements and metadata signed by NVIDIA

Detail

- **Discovery**
  - Base URL https://rim.attestation.nvidia.com
  - RIM naming convention:
    - `NV_GPU_DRIVER_{PRODUCT}_{DRIVER_VERSION}`
    - `NV_GPU_VBIOS_{BOARD_NUMBER}_{SKU}_{BOARD_ID}`
    - `NV_SWITCH_{BIOS}_{VERSION}_{PROJECT}_{SKU}_{CHIP}_{SKU}`

- **Enumeration**

```
GET v1/rim/ids?{rim_format}
```

JSON response listing IDs of known RIMs in the requested format (TCG or CORIM)

- **Fetch Specific RIM**

```
GET v1/rim/{id}
```

JSON-wrapped, base64-encoded TCG RIM (XML) file or CoRIM file

Examples

- **Enumerate RIMs**
  - **Query**

```
$ curl -s 'https://rim.attestation.nvidia.com/v1/rim/ids | jq .
```

  - **Output (abridged JSON)**


```
{
  "ids": [
    "NV_GPU_DRIVER_GH100_535.104.05",
    "...",
    "NV_GPU_VBIOS_1010_0200_882_96005E0001",
    "...",
    "NV_NIC_FIRMWARE_CX7_28.39.4082-LTS_MCX713104AC-ADA",
    "...",
    "NV_SWITCH_BIOS_5612_0002_890_96105E0001",
    "...”
  ],
  "request_id": "54922519-b87e-4057-b323-33b9a2bb9743",
  "last_updated": "2025-04-08T09:05:11.595000”
}
```

Examples

- **Fetch A GPU Driver RIM (TCG Format)**
  - **Query**

```
$ curl -s 'https://rim.attestation.nvidia.com/v1/rim/NV_GPU_DRIVER_GH100_535.104.05’ \
    | jq -r '.rim’ \
    | base64 -D -o rim.xml
```

  - **Output (abridged rim.xml)**

```
<SoftwareIdentity>
   <Entity name=”NVIDIA Corporation” role=“softwareCreator tagCreator”/>
   <Meta colloquialVersion=“535.104.05” edition=“GPU” product=“GH100” .../>
   <Payload>
      <Resource type=“Measurement” Hash0=“1234abcd” name=“Measurement1” .../>
      ...
   </Payload>
</SoftwareIdentity>
```

Examples

- **Fetch A CONNECTX-7 NIC Firmware RIM (CoRIM Format)**
  - **Query**

```
$ curl -s 'https://rim.attestation.nvidia.com/v1/rim/NV_NIC_FIRMWARE_CX7_28.39.4082-LTS_MCX713104AC-ADA' \
  | jq -r '.rim' \
  | base64 -d \
  | cbor2pretty.rb \
  | egrep -v 'tag\(500\)|tag\(502\)' \
  | pretty2cbor.rb \
  > corim.cbor
```

  - **Display using cocli - https://github.com/veraison/cocli**

```
$ cocli corim display –f corim.cbor –-show-tags
```

Examples

- **CoRIM Structure**
  - **Top-Level**

```
Meta:
{
  "signer": {
    "name": "NVIDIA”
  }
}
Corim:
{
  "corim-id": "ConnectX-7_28.39.4082",
```

  - **CoMID**

```
"tags": [
  "2QH6ogGhAHgdMTViMzEwMjExNWIzMDAzMzAwLTI4LjM5…"
],
```

Comparison and Discussion

Comparative

	NVIDIA RIM Service	AMD KDS
Distributes	Reference values + endorsements	Endorsements only
Output Format	TCG RIM or CoRIM	X.509
Includes Hashes	☑ yes	☒ no
Query Input	Device model, firmware version	Chip ID + TCB versions
Target Devices	NVIDIA GPUs, DPUs	AMD SEV-SNP CPUs
Attestation Use Case	Appraisal (firmware validation)	Signature verification
API Style	HTTP GET URL path	HTTP GET URL path + parameters

https://wiki.ietf.org/group/rats/referencevalues/nvidia-endorsement-distribution-api

https://wiki.ietf.org/group/rats/referencevalues/amd-key-distribution-service

Thoughts/Discussion

- Endorsements/RVs are not always addressable by a taxonomic path – dynamic query parameters are sometimes needed (as in the case of AMD, where TCB patch levels help to address the certificate).
- Fragmentation is horizontal, convenience is vertical
  - Vendor-specific evidence is naturally geared towards fetching the same vendor-specific endorsements – if you have a SEV-SNP attestation report, it's easy to call the AMD KDS with the correct parameters
  - Commonality may result in more conversions and less convenience, unless it is embraced by the whole ecosystem
- Other comments/questions?

linaro Connect 2025

Thank You!