DICE for Confidential VMs

Measured boot based on chaining signatures

Peter Gonda - pgonda@google.com

Linux Plumbers CC Micro Conference 2022

Agenda

- What is this talk?
- What is DICE
- DICE and Confidential VMs
- What's been done / needs updating

What is this talk?

Start a discussion around measured boot and attestation

Propose a possible solution

Meet people interested in building solutions

Goal

Users of Confidential VMs can use attestation to remotely verify their workloads code identity

Goal

Users of Confidential VMs can use **attestation** to remotely verify their workloads code identity

Goal

Users of Confidential VMs can use attestation to remotely verify their workloads **code identity**

DICE

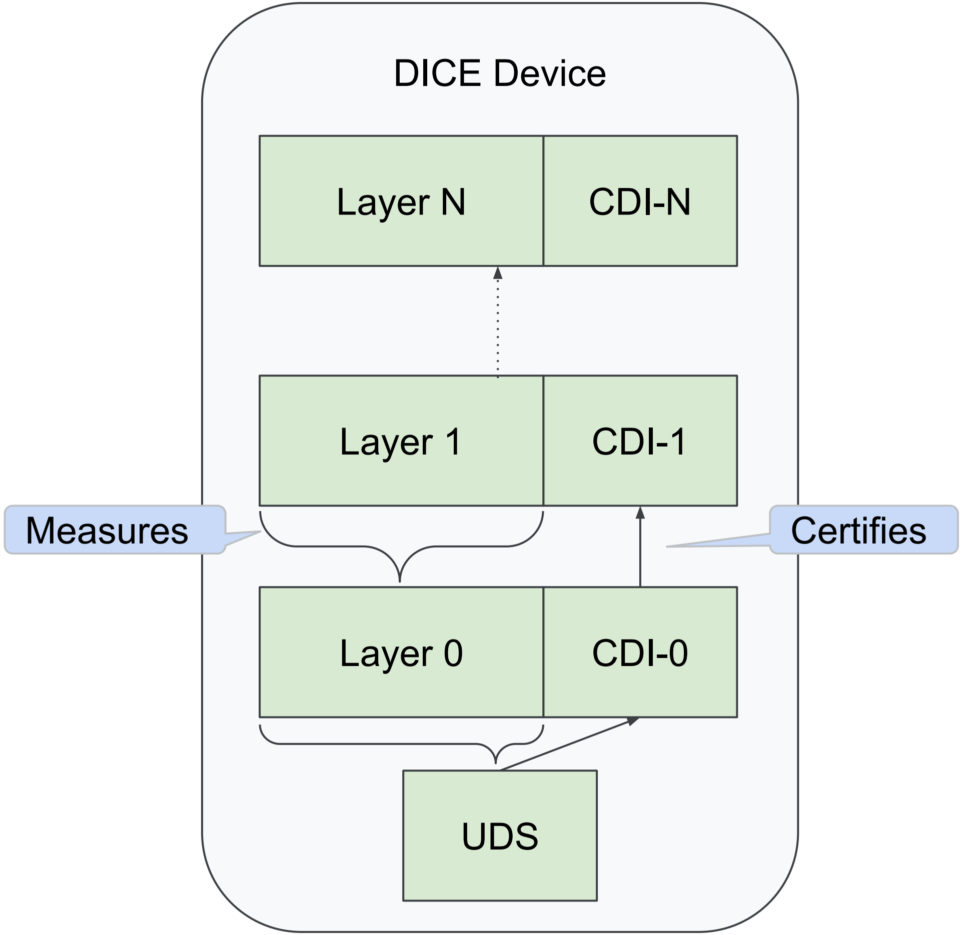
Device Identifier Composition Engine

TCG spec whose goal is “... to provide security and privacy foundations for systems without a TPM …”

Results in an identifier which represents the combination of hardware and software of a devices boot sequence

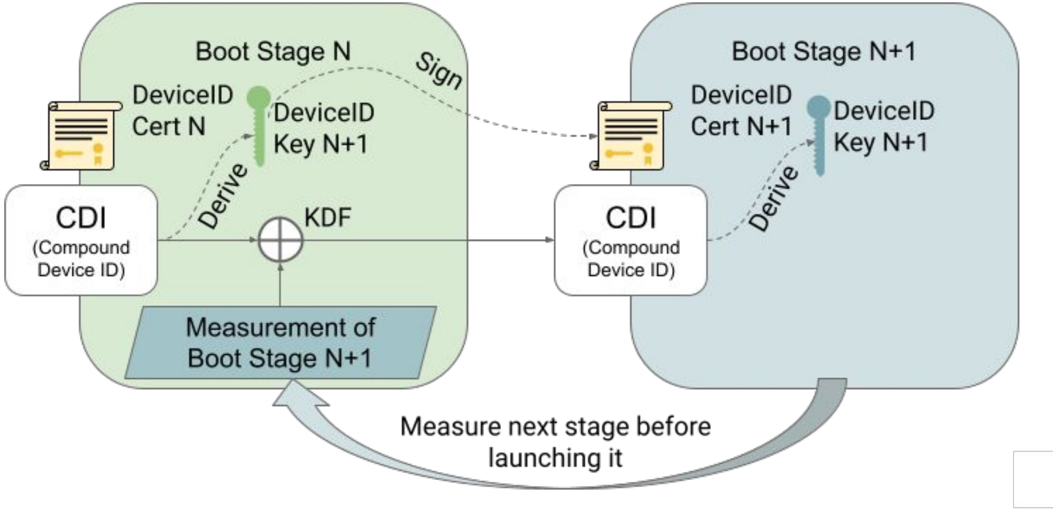
DICE: Layering

- Layered approach - DICE Chain
- Boot divided into layers
  - OVMF
  - Grub2
  - Linux
- Each boot layer N:
  - Measures N+1
  - Certifies N+1
  - Clears N's private keys
- UDS: Unique Device Secret
- CDI: Compound Device Identifier



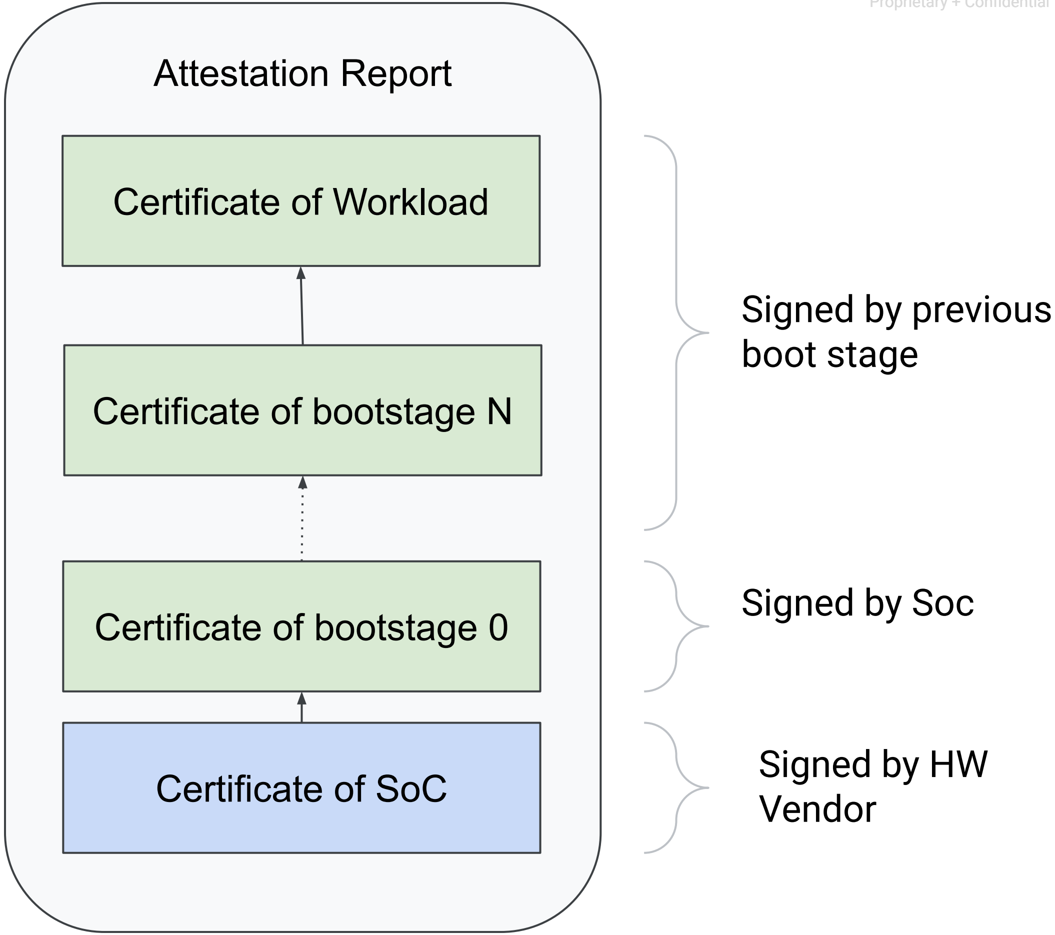
DICE: A Layers Job

- Inputs for each layer
  - CDI-N
  - DeviceID key pair w/ cert
  - Code + config of next layer
- Outputs:
  - CDI-N+1
  - N+1 Alias Key Certificate
- **Layer N must zero out CDI-N and DeviceID Key priv**

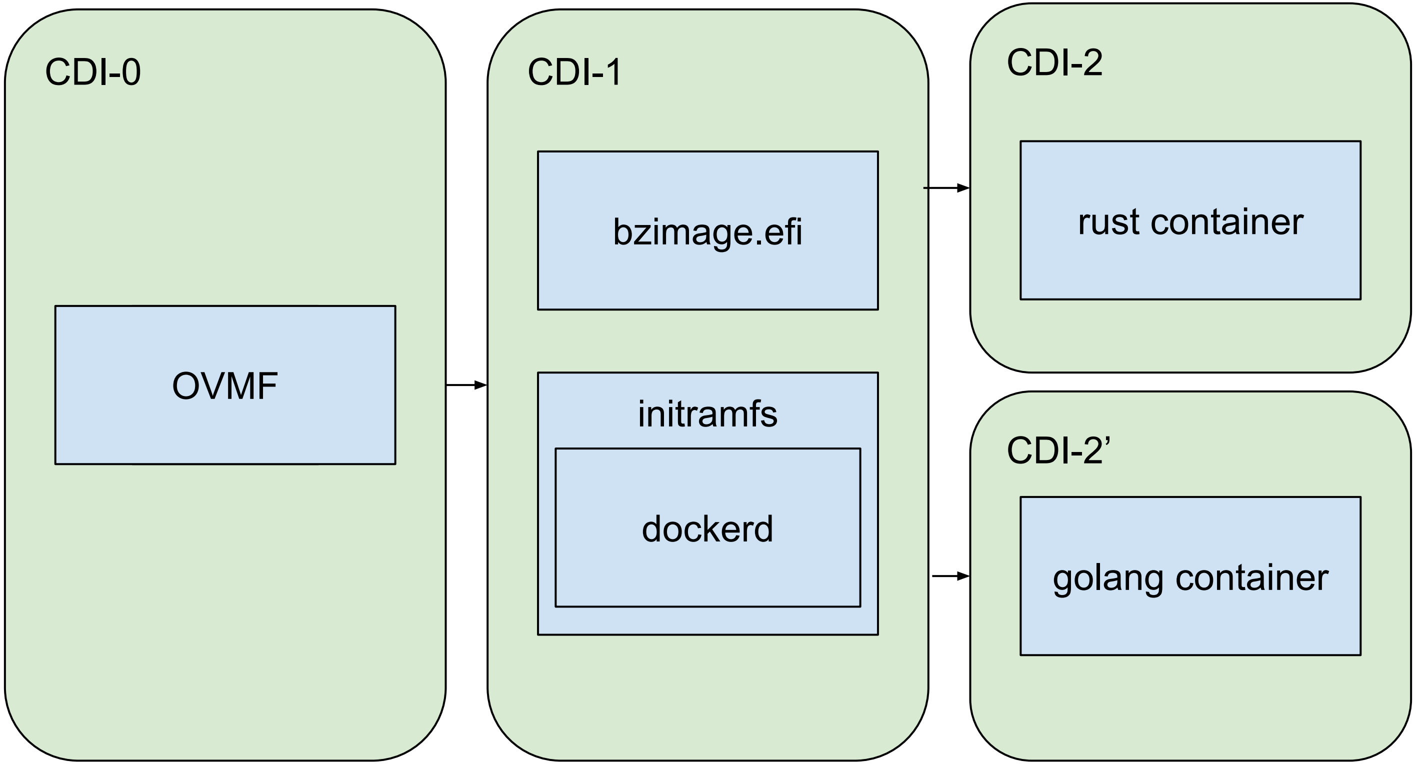


DICE: End State

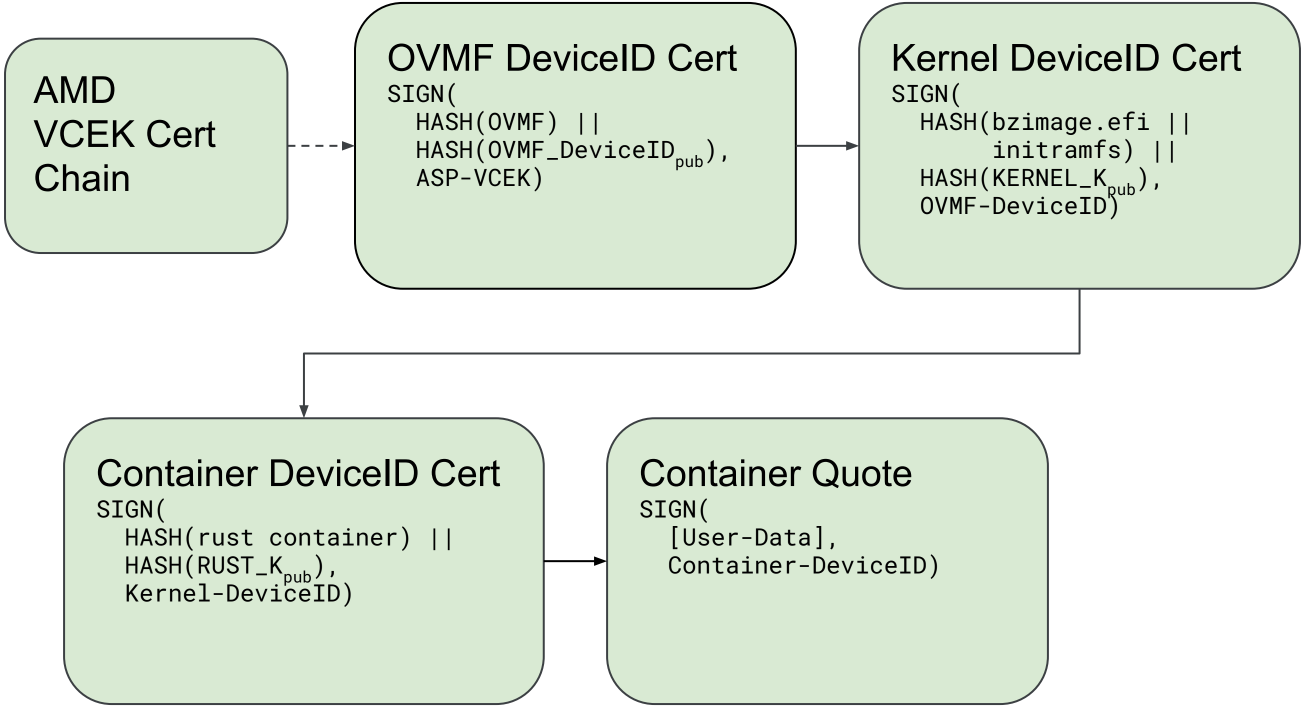
- Workload at end of DICE chain has:
  - DeviceID asymmetric Key Pair N
  - Certificate Chain [0, N-1]
- Workload can:
  - Use key pair to attest identity to remote parties
  - Use CDI-N derived key for sealing



A linux DICE flow



A linux DICE Cert Chain



AMD
VCEK Cert
Chain

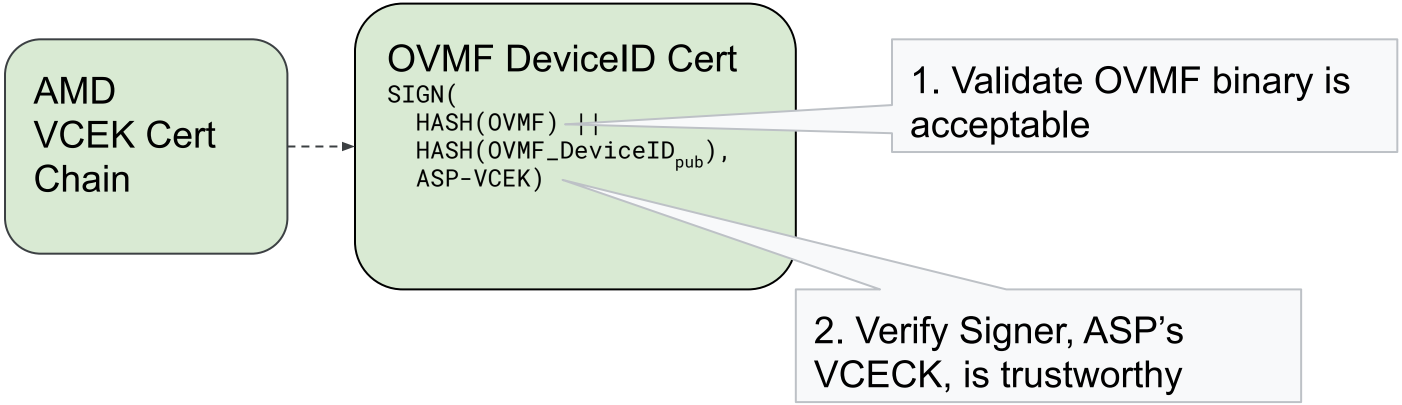
OVMF DeviceID Cert
SIGN(
  HASH(OVMF) ||
  HASH($\text{OVMF\_DeviceID}_{\text{pub}}$),
  ASP-VCEK)

Kernel DeviceID Cert
SIGN(
  HASH(bzimage.efi ||
       initramfs) ||
  HASH($\text{KERNEL\_K}_{\text{pub}}$),
  OVMF-DeviceID)

Container DeviceID Cert
SIGN(
  HASH(rust container) ||
  HASH($\text{RUST\_K}_{\text{pub}}$),
  Kernel-DeviceID)

Container Quote
SIGN(
  [User-Data],
  Container-DeviceID)

A linux DICE Cert Chain



AMD
VCEK Cert
Chain

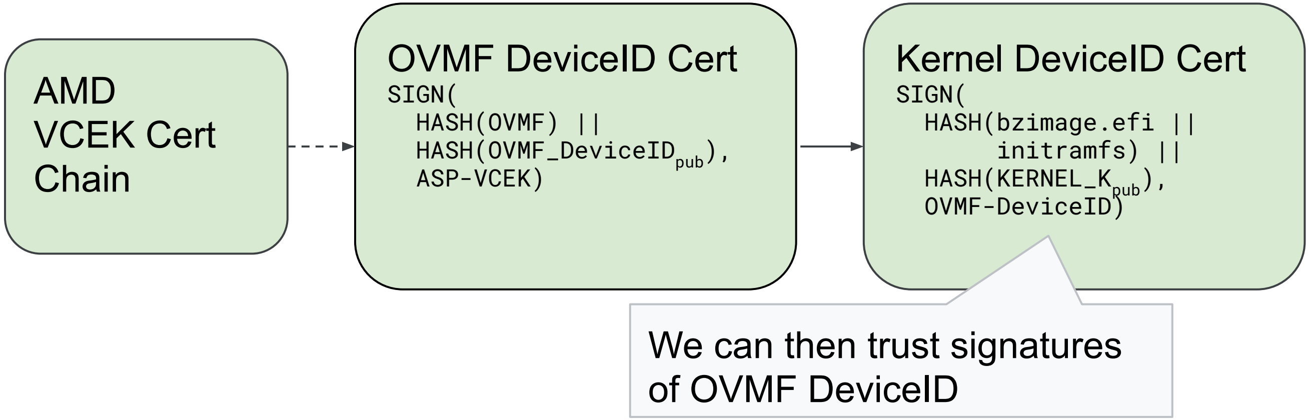
OVMF DeviceID Cert

SIGN(
  HASH(OVMF) ||
  HASH($\text{OVMF\_DeviceID}_{\text{pub}}$),
  ASP-VCEK)

1. Validate OVMF binary is acceptable

2. Verify Signer, ASP’s VCECK, is trustworthy

A linux DICE Cert Chain



AMD VCEK Cert Chain

OVMF DeviceID Cert

SIGN(
  HASH(OVMF) ||
  HASH(OVMF_DeviceID$_{pub}$),
  ASP-VCEK)

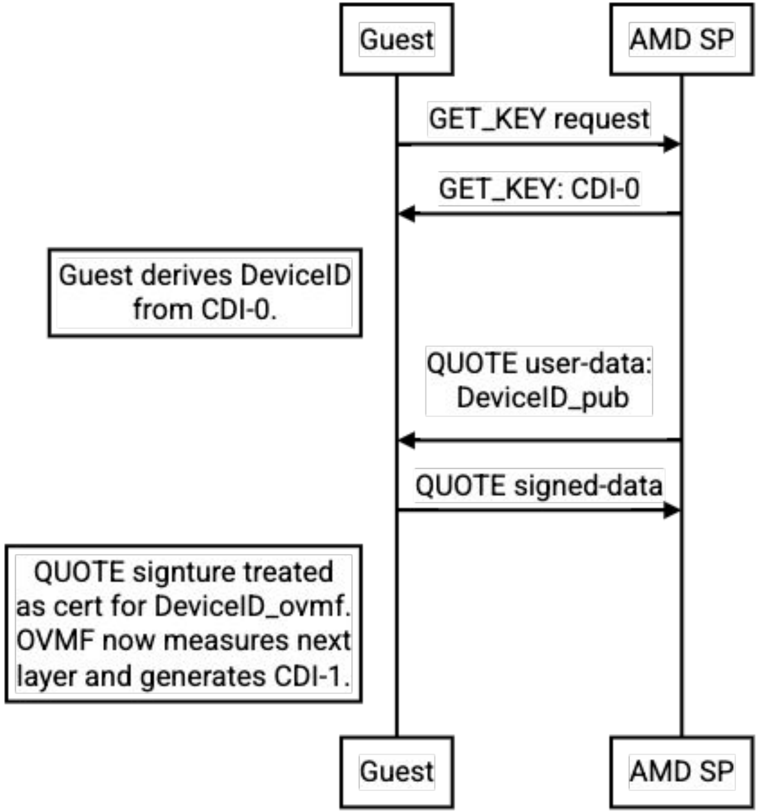
Kernel DeviceID Cert

SIGN(
  HASH(bzimage.efi ||
       initramfs) ||
  HASH(KERNEL_K$_{pub}$),
  OVMF-DeviceID)

We can then trust signatures of OVMF DeviceID

Layer 0 for AMD SNP

- GET_KEY command can be used to get CDI-like data.
  - Must enforce next layer cannot use same GET_KEY



Guest

AMD SP

GET_KEY request

GET_KEY: CDI-0

Guest derives DeviceID from CDI-0.

QUOTE user-data: DeviceID_pub

QUOTE signed-data

QUOTE signture treated as cert for DeviceID_ovmf. OVMF now measures next layer and generates CDI-1.

Guest

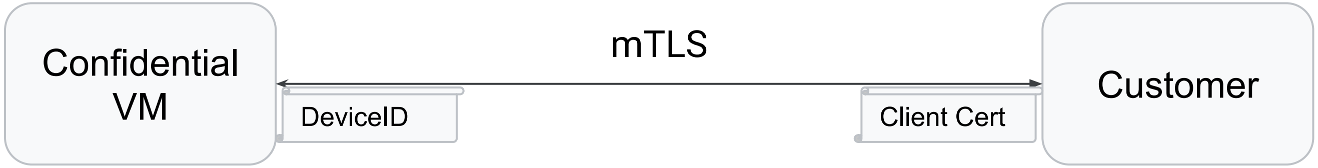
AMD SP

What needs updating?

- Already have **`/dev/open-dice0`**
- OVMF
- grub
- distro specific boot processes
    - systemd

Recap

- DICE gives workloads DeviceID key pairs
- DeviceID cryptographic combination of software and hardware state, ie **Code Identity**
- DeviceID can perform **Remote Attestation**
- Can be used in-place of TPM or to compliment one



Questions / Comments?

Let's discuss on linux-coco@

Get in touch directly pgonda@google.com

Links:

TCG DICE

Open DICE code and spec

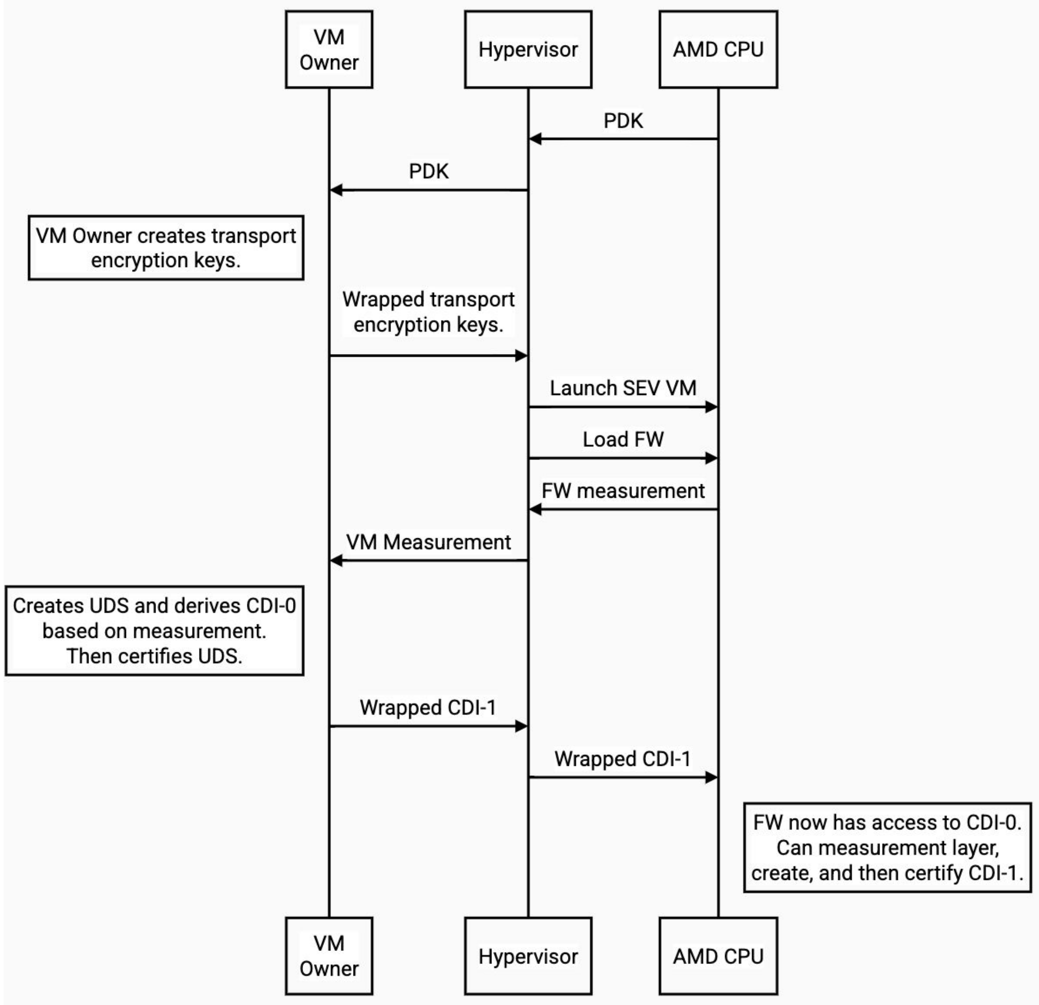
Thank You

Google

What about the UDS?

AMD SEV

- Use guest owner as HRoT
- Guest owner can provision and certify instances



VM Owner

Hypervisor

AMD CPU

PDK

PDK

VM Owner creates transport encryption keys.

Wrapped transport encryption keys.

Launch SEV VM

Load FW

FW measurement

VM Measurement

Creates UDS and derives CDI-0 based on measurement. Then certifies UDS.

Wrapped CDI-1

Wrapped CDI-1

FW now has access to CDI-0. Can measurement layer, create, and then certify CDI-1.

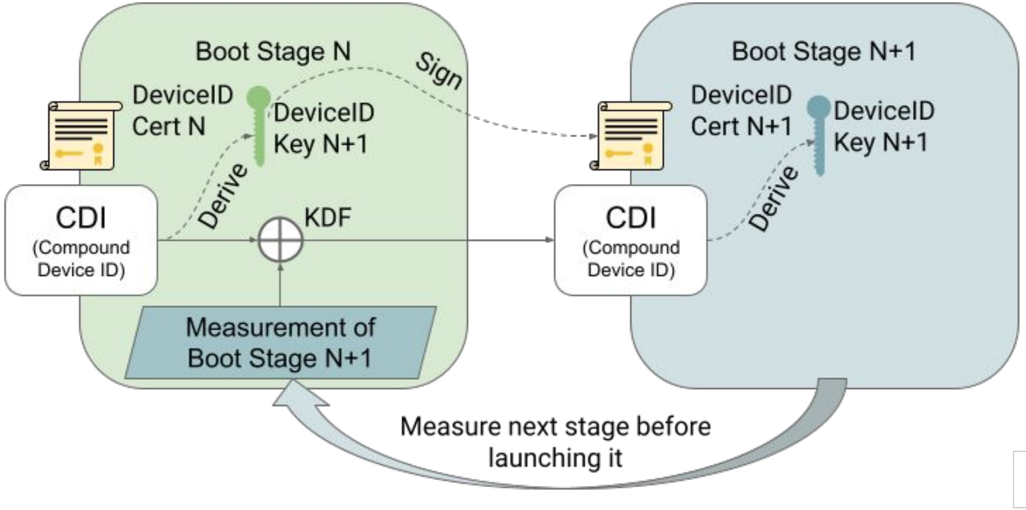
VM Owner

Hypervisor

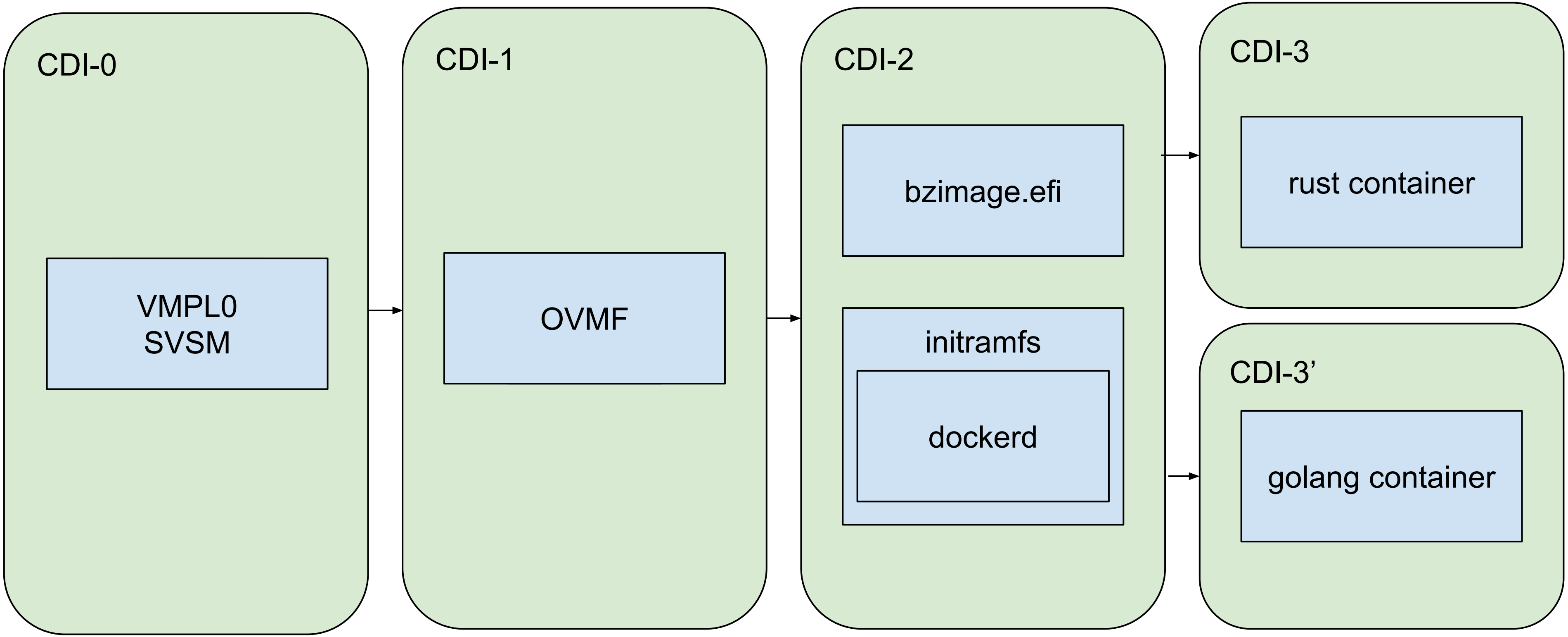
AMD CPU

DICE: A Layers Job

```
void dice_layer(CDI: cdi,
                ecdsa-pair: device-id,
                ecdsa-cert: device-id-cert,
                boot-layer: next) {
  TCI next-layer-hash = HASH(next.code ||
                             next.config)
  CDI cdi-next = HMAC(cdi, next-layer-hash)
  ecdsa-pair device-id-next = HMAC(
                                cdi-next,
                                `device-id`)
  ecdsa-cert next-cert = certify(
                           device-id-next,
                           device-id)
  …
  clear_mem(cid)
  clear_mem(device-id)
}
```



A linux DICE flow now with an SVSM


CDI-0
VMPL0
SVSM
CDI-1
OVMF
CDI-2
bzimage.efi
initramfs
dockerd
CDI-3
rust container
CDI-3’
golang container