Safeguard Your Cloud Workloads and Then Accelerate:

An In-depth Look at CPU and GPU Confidential Computing

Jingyao Zhang

Advisor: Elaheh Sadredini

UC RIVERSIDE

The content of these slides are partly adapted from online materials.

Agenda

Agenda

- Background

Agenda

- Background
- What is Confidential Computing

Agenda

- Background
- What is Confidential Computing
- Why Confidential Computing is the Future Infrastructure

Agenda

- Background
- What is Confidential Computing
- Why Confidential Computing is the Future Infrastructure
- How does Confidential Computing Work

Agenda

- Background
- What is Confidential Computing
- Why Confidential Computing is the Future Infrastructure
- How does Confidential Computing Work
- GPU Confidential Computing

Background

Confidential Computing is being widely adopted

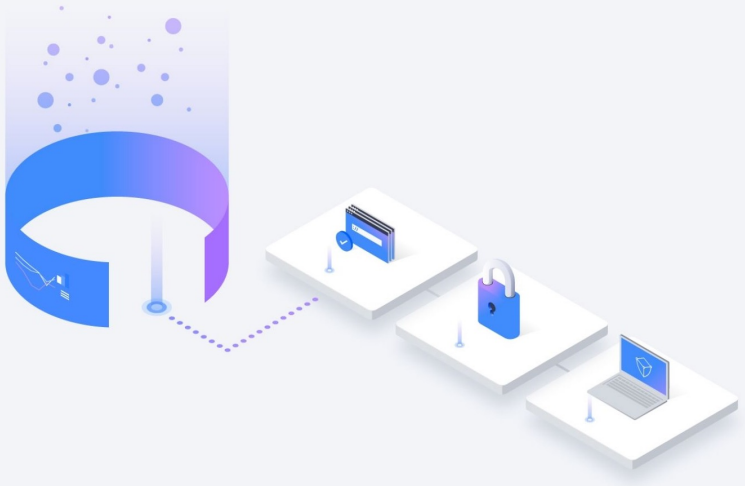
Confidential Computing is being widely adopted

Confidential computing on IBM Cloud

Protect your data at rest, in transit and in use. Get a higher level of privacy assurance.

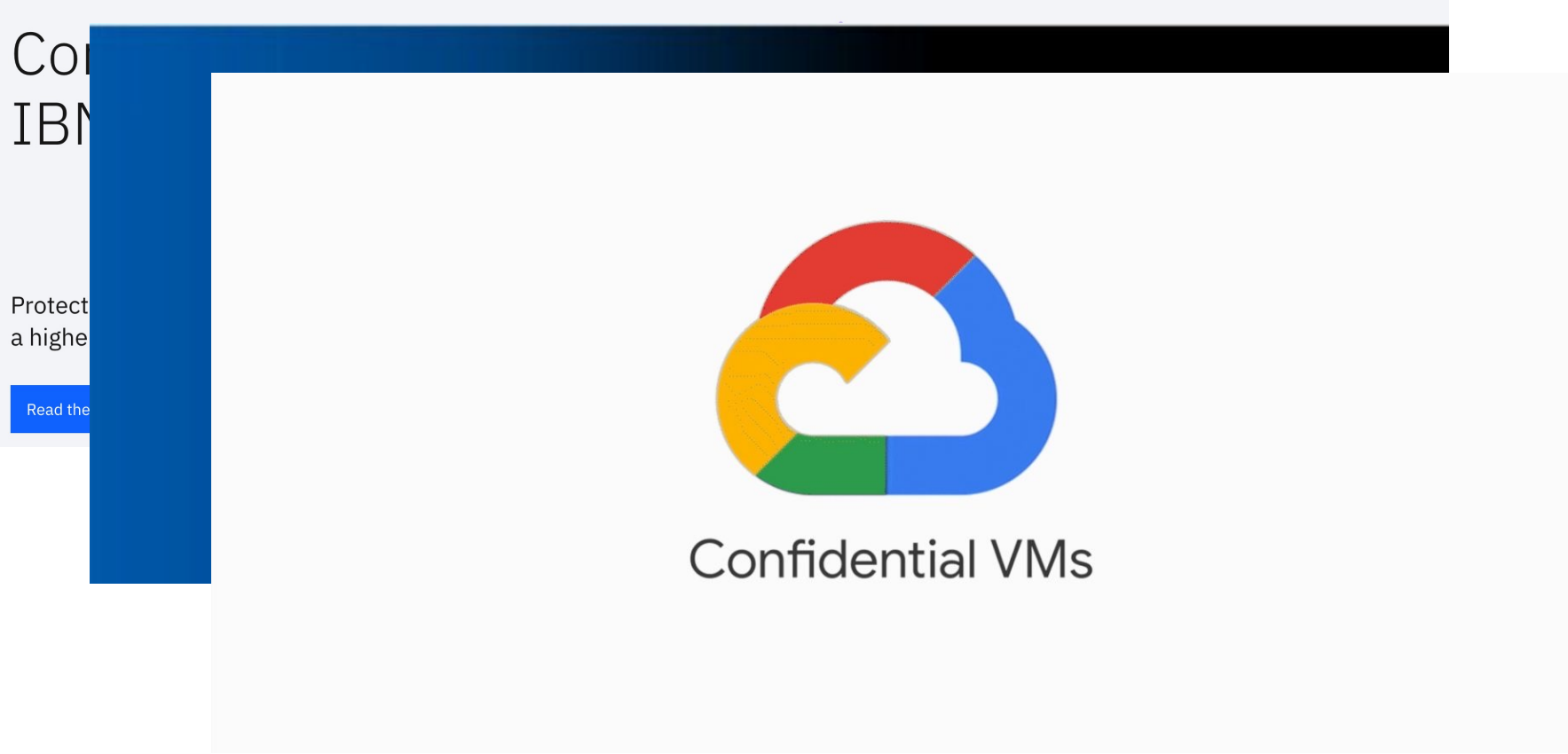
Read the smart paper

Read the report

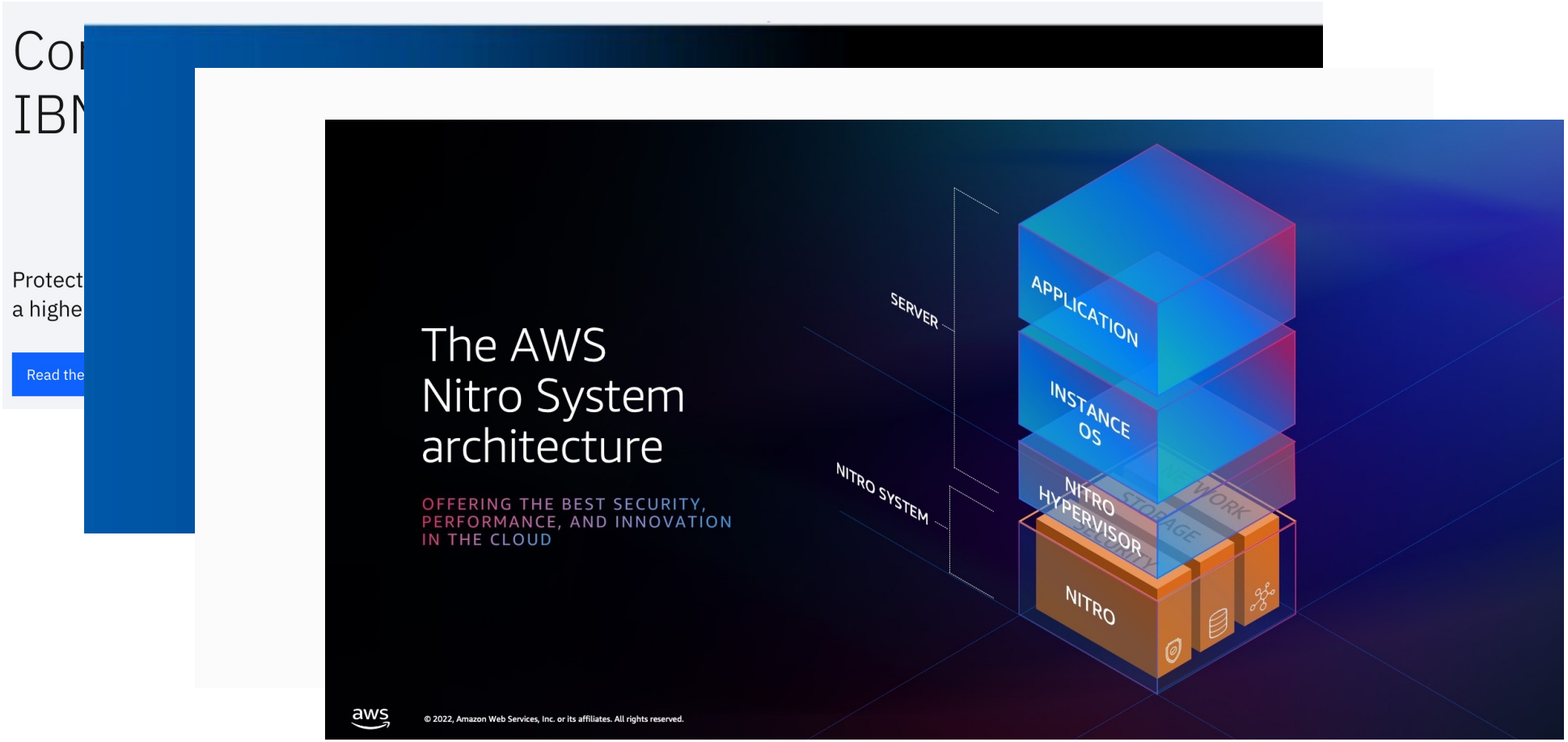


Confidential Computing is being widely adopted

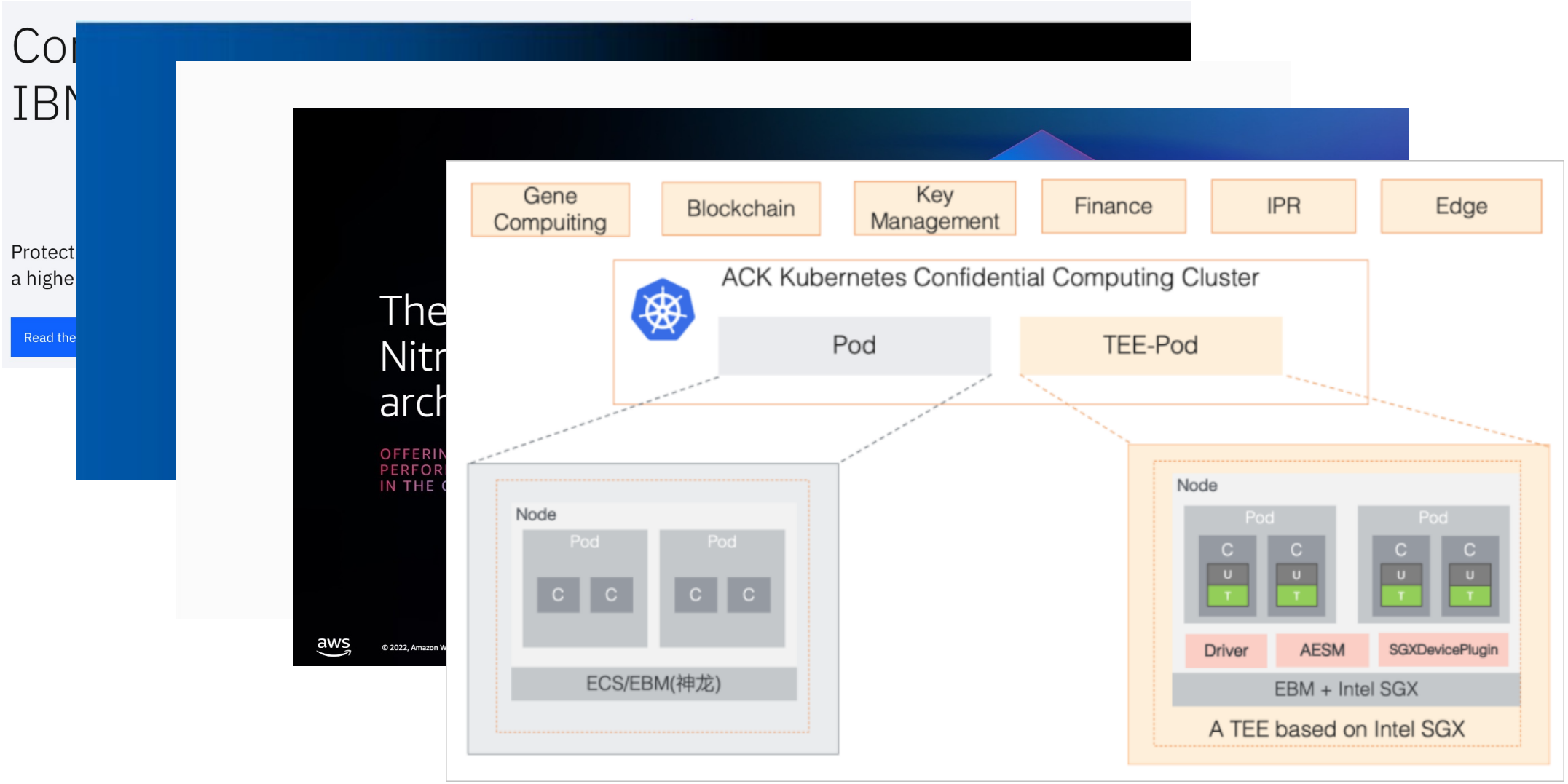
Confidential Computing is being widely adopted



Confidential Computing is being widely adopted



Confidential Computing is being widely adopted



Confidential Computing is being widely adopted

Intel is all-in on Confidential Computing

Confidential Computing Consortium

CONFIDENTIAL COMPUTING CONSORTIUM

Premier Members

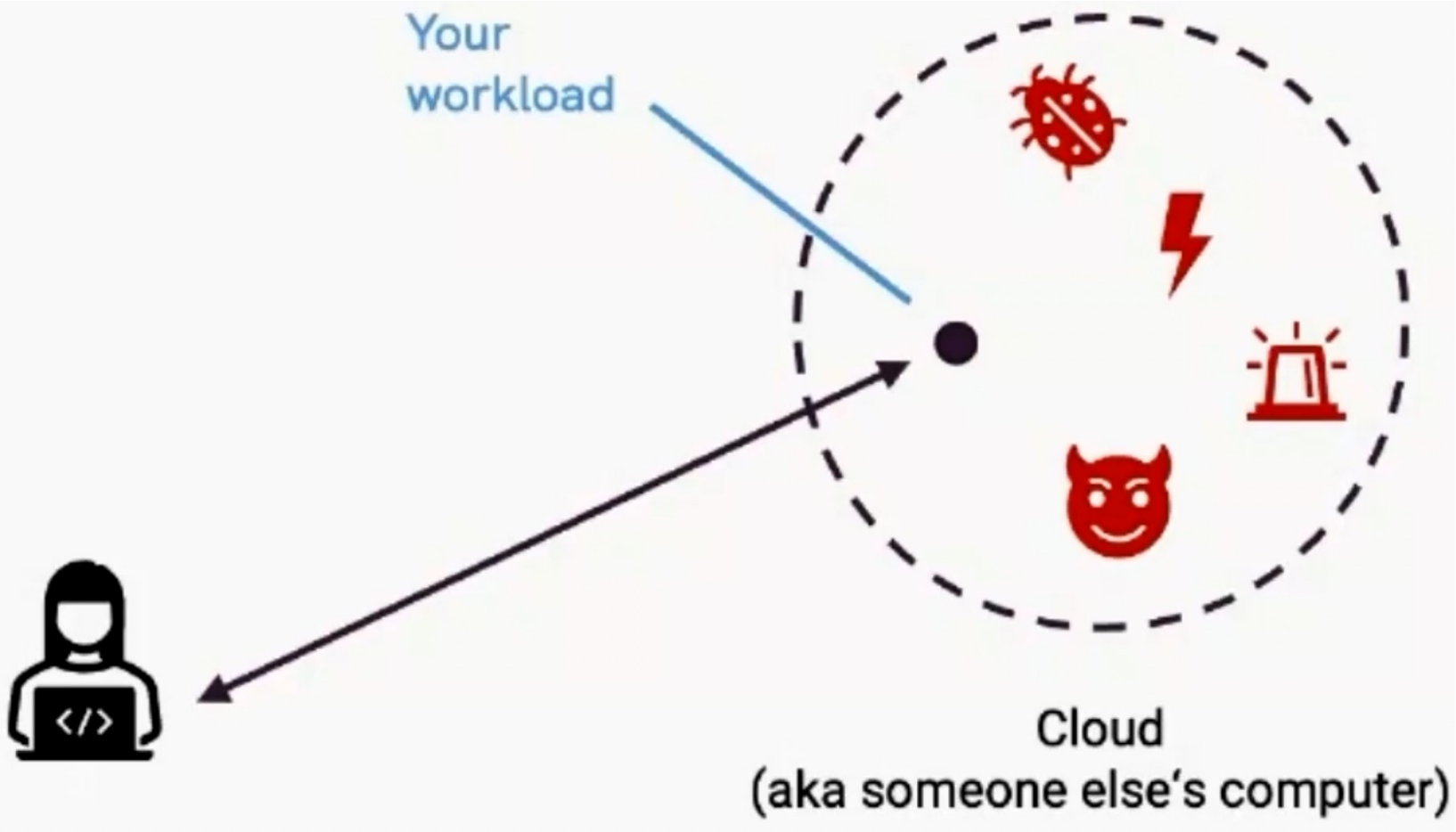
accenture
arm
Google
HUAWEI
intel
Meta
Microsoft
Red Hat

General Members

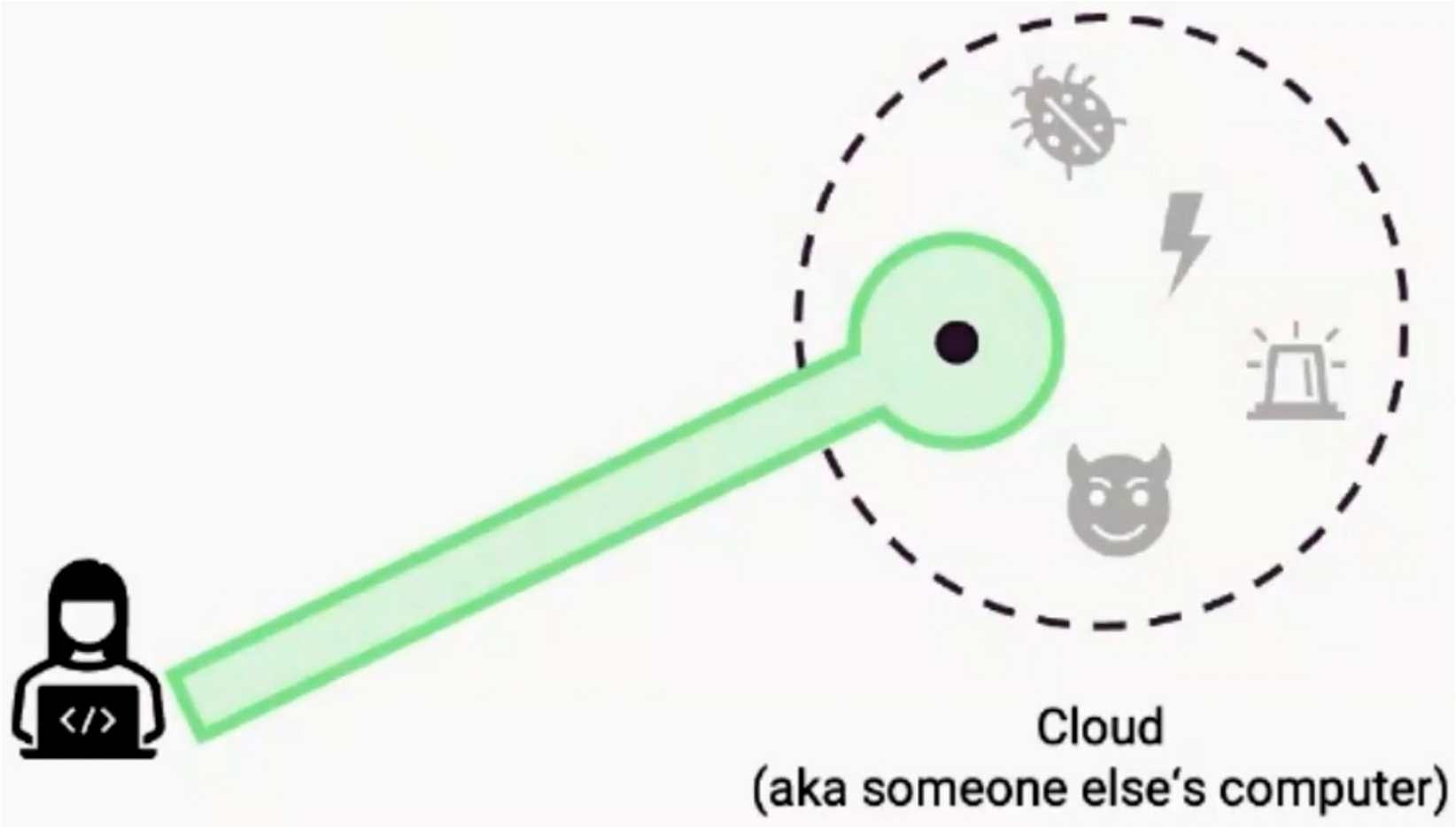
灵雀云 alauda.cn
AMD
AMPERE
anjuna
蚂蚁集团 ANT GROUP
BeeKeeperAI™
ByteDance
CanaryBit

What is Confidential Computing (CC)

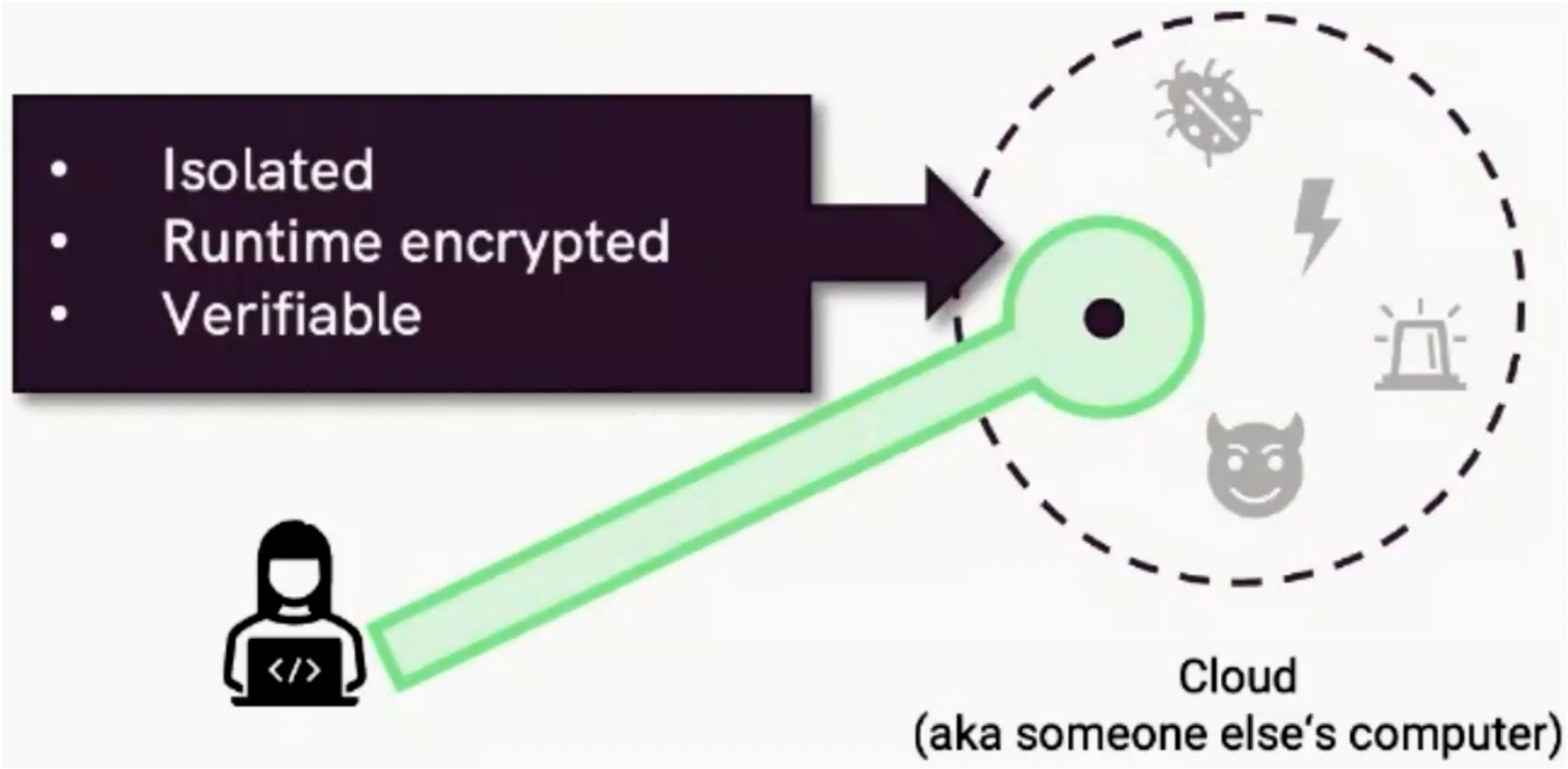
Confidential Computing takes you from here…



… to here



… to here


Isolated
Runtime encrypted
Verifiable
Cloud
(aka someone else's computer)

Why Confidential Computing is the Future

Customers want to reduce operational costs

Customers want to reduce operational costs

- Customers want to use public cloud:
  - Lower operational costs with **public cloud vs. on-premises servers**

Customers want to reduce operational costs

- Customers want to use public cloud:
  - Lower operational costs with **public cloud vs. on-premises servers**
- However, they have concerns about **data privacy and security**:

Customers want to reduce operational costs

- Customers want to use public cloud:
  - Lower operational costs with **public cloud vs. on-premises servers**
- However, they have concerns about **data privacy and security**:
  - Remote computer and software stack owned by an **untrusted party** (e.g., CSPs)
    - Manipulate everything
    - Directly see and modify application code and data

Customers want to reduce operational costs

- Customers want to use public cloud:
  - Lower operational costs with **public cloud vs. on-premises servers**
- However, they have concerns about **data privacy and security**:
  - Remote computer and software stack owned by an **untrusted party** (e.g., CSPs)
    - Manipulate everything
    - Directly see and modify application code and data
  - Software bugs
    - SMM-based rootkits
    - Xen 150K LOC, 40+ vulnerabilities per year
    - Monolithic kernel, e.g., Linux, 17M LOC, 100+ vulnerabilities per year
    - **70% of all security bugs are memory safety issues from Microsoft**

Customers want to reduce operational costs

- Customers want to use public cloud:
  - Lower operational costs with **public cloud vs. on-premises servers**
- However, they have concerns about **data privacy and security**:
  - Remote computer and software stack owned by an **untrusted party** (e.g., CSPs)
    - Manipulate everything
    - Directly see and modify application code and data
  - Software bugs
    - SMM-based rootkits
    - Xen 150K LOC, 40+ vulnerabilities per year
    - Monolithic kernel, e.g., Linux, 17M LOC, 100+ vulnerabilities per year
    - **70% of all security bugs are memory safety issues from Microsoft**
  - Compliance of General Data Protection Regulation (GDPR) or Health Insurance Portability and Accountability Act (HIPAA), etc.

Cloud Service Providers (CSPs) need higher ROI

Cloud Service Providers (CSPs) need higher ROI

- CSPs need more customers, especially security-sensitive customers:
  - Utilization of computing resources often **< 20%**
  - Security-sensitive customers are **rich**

Cloud Service Providers (CSPs) need higher ROI

- CSPs need more customers, especially security-sensitive customers:
  - Utilization of computing resources often **< 20%**
  - Security-sensitive customers are **rich**
- However, CSPs cannot **gain trust from security-sensitive customers**:

Cloud Service Providers (CSPs) need higher ROI

- CSPs need more customers, especially security-sensitive customers:
  - Utilization of computing resources often **< 20%**
  - Security-sensitive customers are **rich**
- However, CSPs cannot **gain trust from security-sensitive customers**:
  - No guarantee of data privacy and security
    - Agreements only guarantee **"won't"** instead of **"can't"**
    - Malicious infrastructure administrators, hackers

Cloud Service Providers (CSPs) need higher ROI

- CSPs need more customers, especially security-sensitive customers:
  - Utilization of computing resources often **< 20%**
  - Security-sensitive customers are **rich**
- However, CSPs cannot **gain trust from security-sensitive customers**:
  - No guarantee of data privacy and security
    - Agreements only guarantee **"won't"** instead of **"can't"**
    - Malicious infrastructure administrators, hackers
  - Compliance of GDPR or HIPAA, etc.

Confidential Computing is a Win-Win

Confidential Computing is a Win-Win

*"Confidential Computing addresses the **trust issue** between the **data/code owner** and the **platform owner** when they are not the same entity."*

Confidential Computing is a Win-Win

*"Confidential Computing addresses the **trust issue** between the **data/code owner** and the **platform owner** when they are not the same entity."*

For customers:

Confidential Computing is a Win-Win

*"Confidential Computing addresses the **trust issue** between the **data/code owner** and the **platform owner** when they are not the same entity."*

For customers:

- CC **guarantees** that remote computers **CAN'T** manipulate data/code
  - Reduce operational costs by utilizing public cloud

Confidential Computing is a Win-Win

*"Confidential Computing addresses the **trust issue** between the **data/code owner** and the **platform owner** when they are not the same entity."*

For customers:

- CC **guarantees** that remote computers **CAN'T** manipulate data/code
  - Reduce operational costs by utilizing public cloud

For cloud service providers (CSPs)

Confidential Computing is a Win-Win

*"Confidential Computing addresses the **trust issue** between the **data/code owner** and the **platform owner** when they are not the same entity."*

For customers:

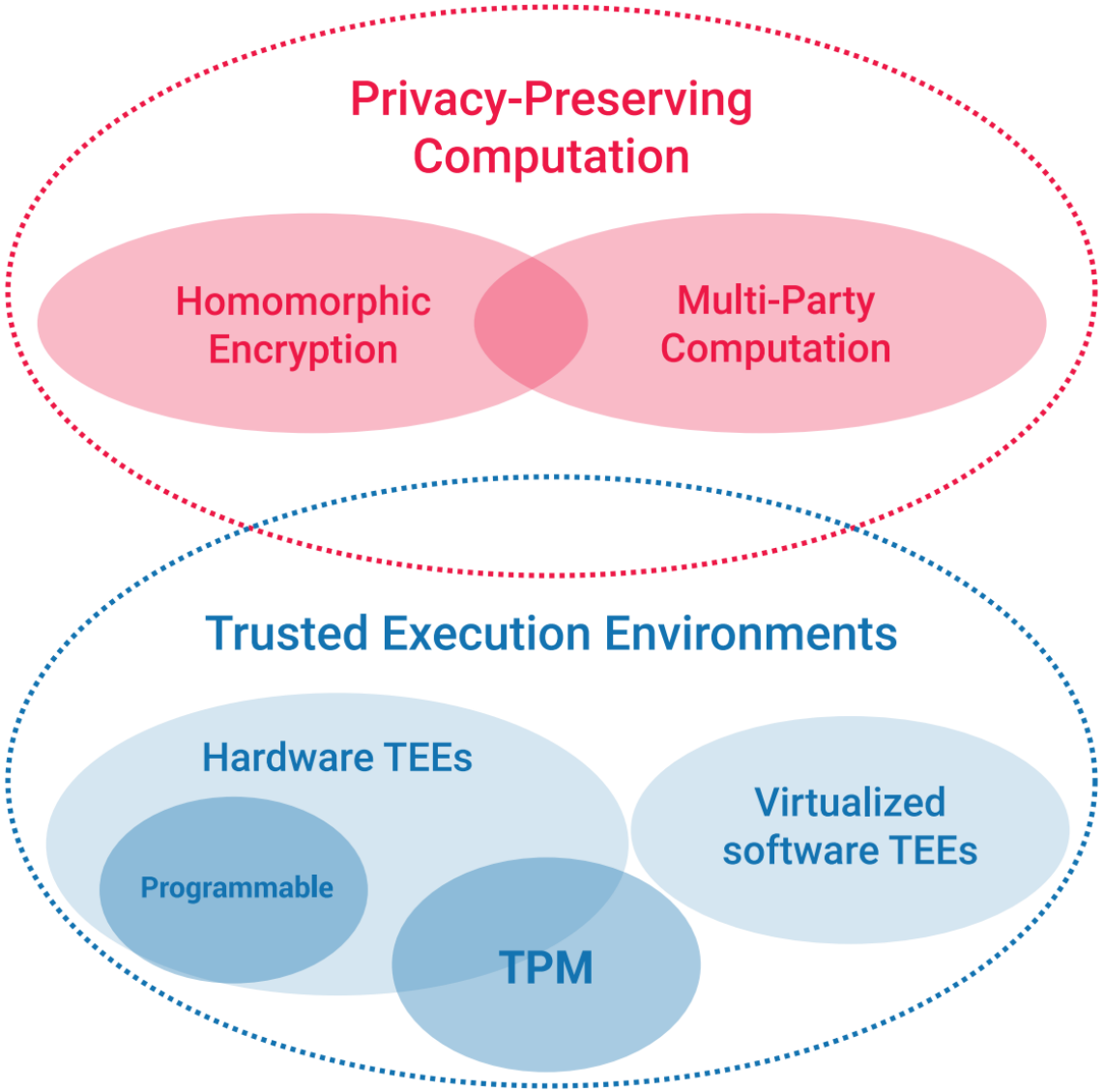
- CC **guarantees** that remote computers **CAN'T** manipulate data/code
  - Reduce operational costs by utilizing public cloud

For cloud service providers (CSPs)

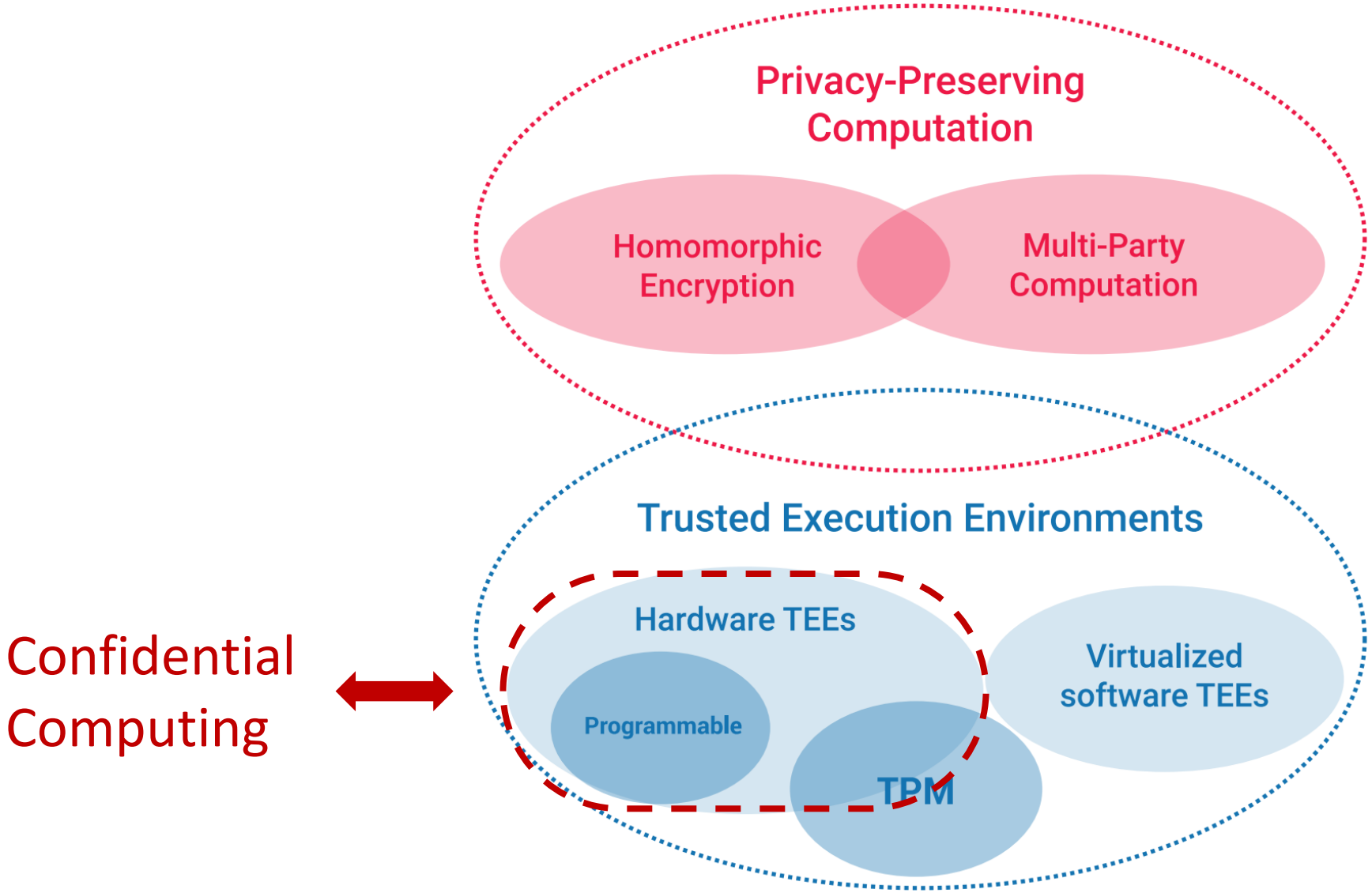
- CC helps to **gain trust** from security-sensitive customers
  - Attain higher ROI from new security-sensitive customers (e.g., health centers)

Confidential Computing is the Future

Confidential Computing can be Future Infrastructure



Confidential Computing can be Future Infrastructure



How does Confidential Computing Work

Confidential Computing Definition

*Confidential Computing is the protection of **data in use** by performing computation in a **hardware-based**, **attested** Trusted Execution Environment.*

--- defined by Confidential Computing Consortium

Confidential Computing Definition

*Confidential Computing is the protection of **data in use** by performing computation in a **hardware-based**, **attested** Trusted Execution Environment.*

--- defined by Confidential Computing Consortium

Confidential Computing

Trusted Execution Environment (TEE) Hardware

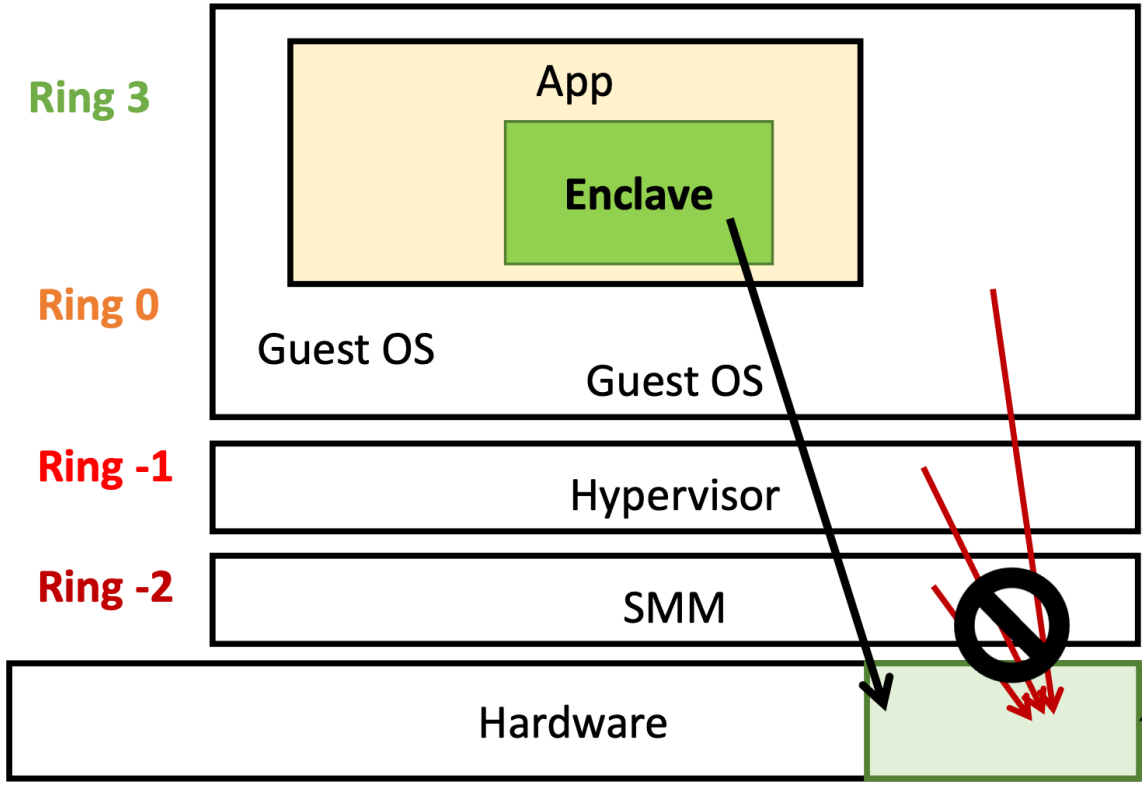
*Confidential Computing is the protection of data in use by performing computation in a **hardware-based**, attested Trusted Execution Environment.*

--- defined by Confidential Computing Consortium

Trusted Execution Environment (TEE) Hardware

*Confidential Computing is the protection of data in use by performing computation in a **hardware-based**, attested Trusted Execution Environment.*

--- defined by Confidential Computing Consortium



Remote Attestation

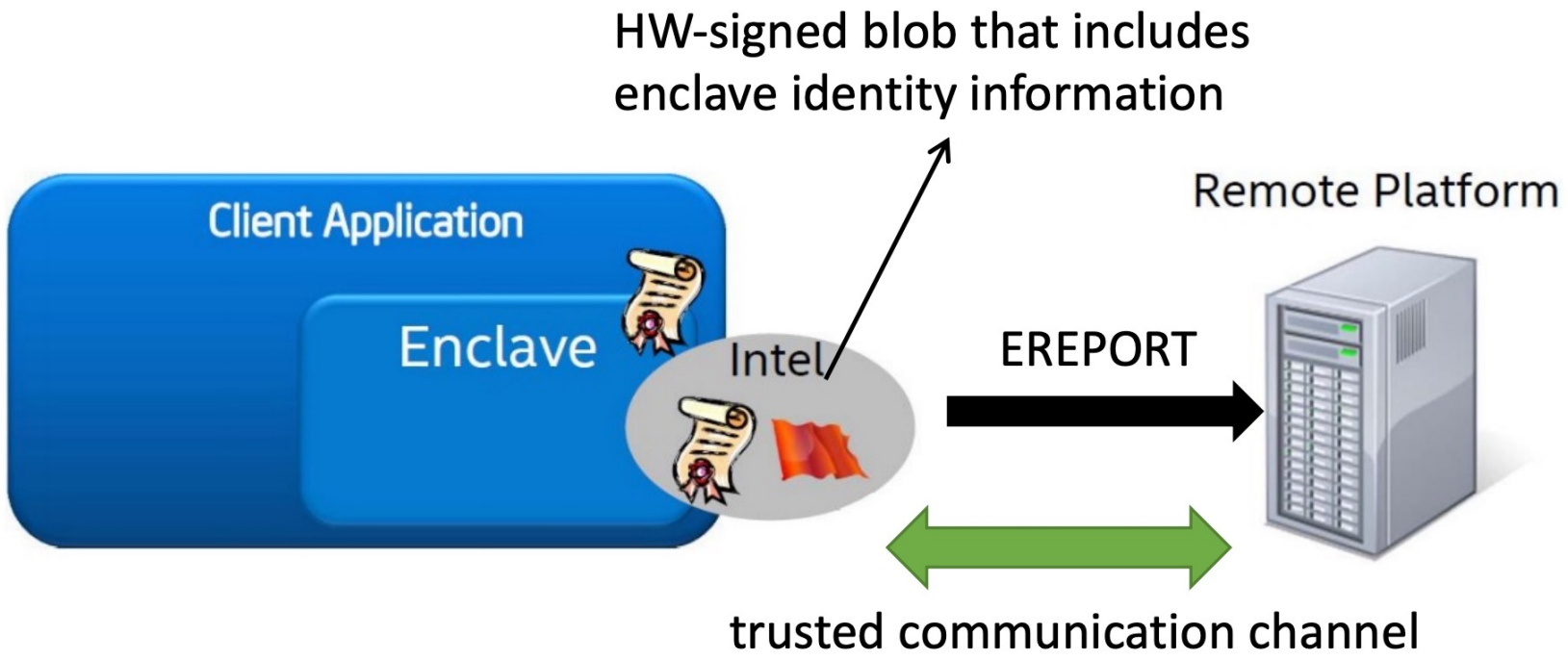
*Confidential Computing is the protection of data in use by performing computation in a hardware-based, **attested** Trusted Execution Environment.*

--- defined by Confidential Computing Consortium

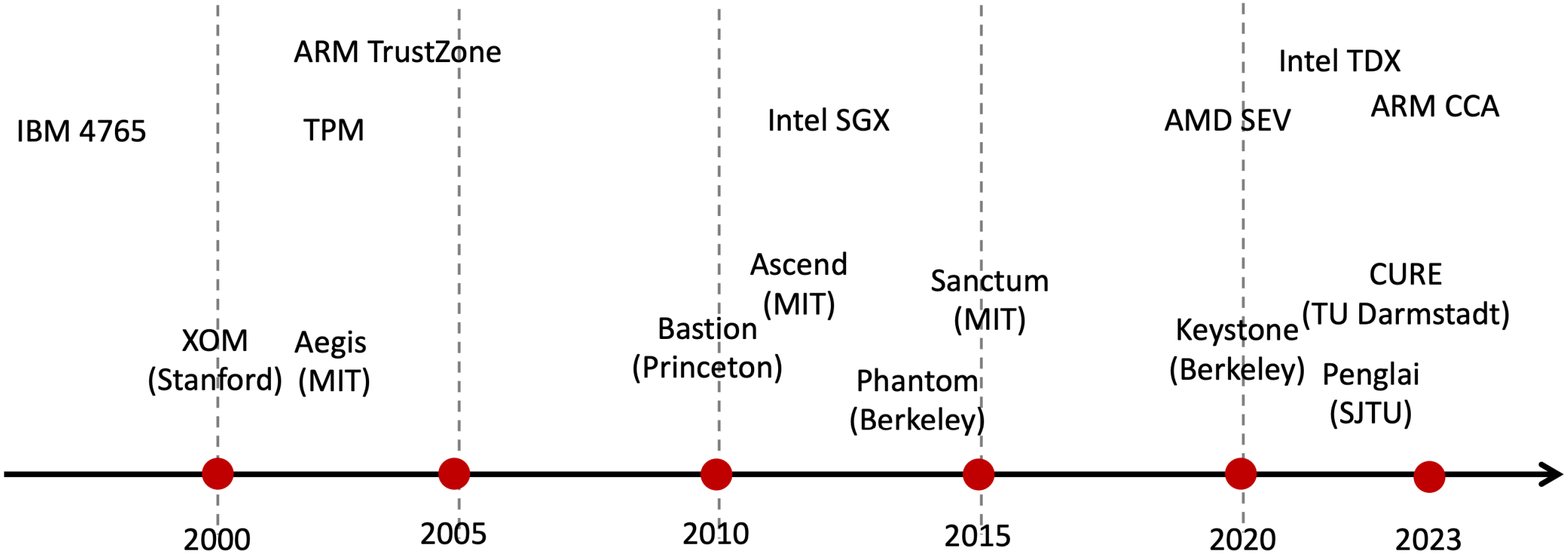
Remote Attestation

*Confidential Computing is the protection of data in use by performing computation in a hardware-based, **attested** Trusted Execution Environment.*

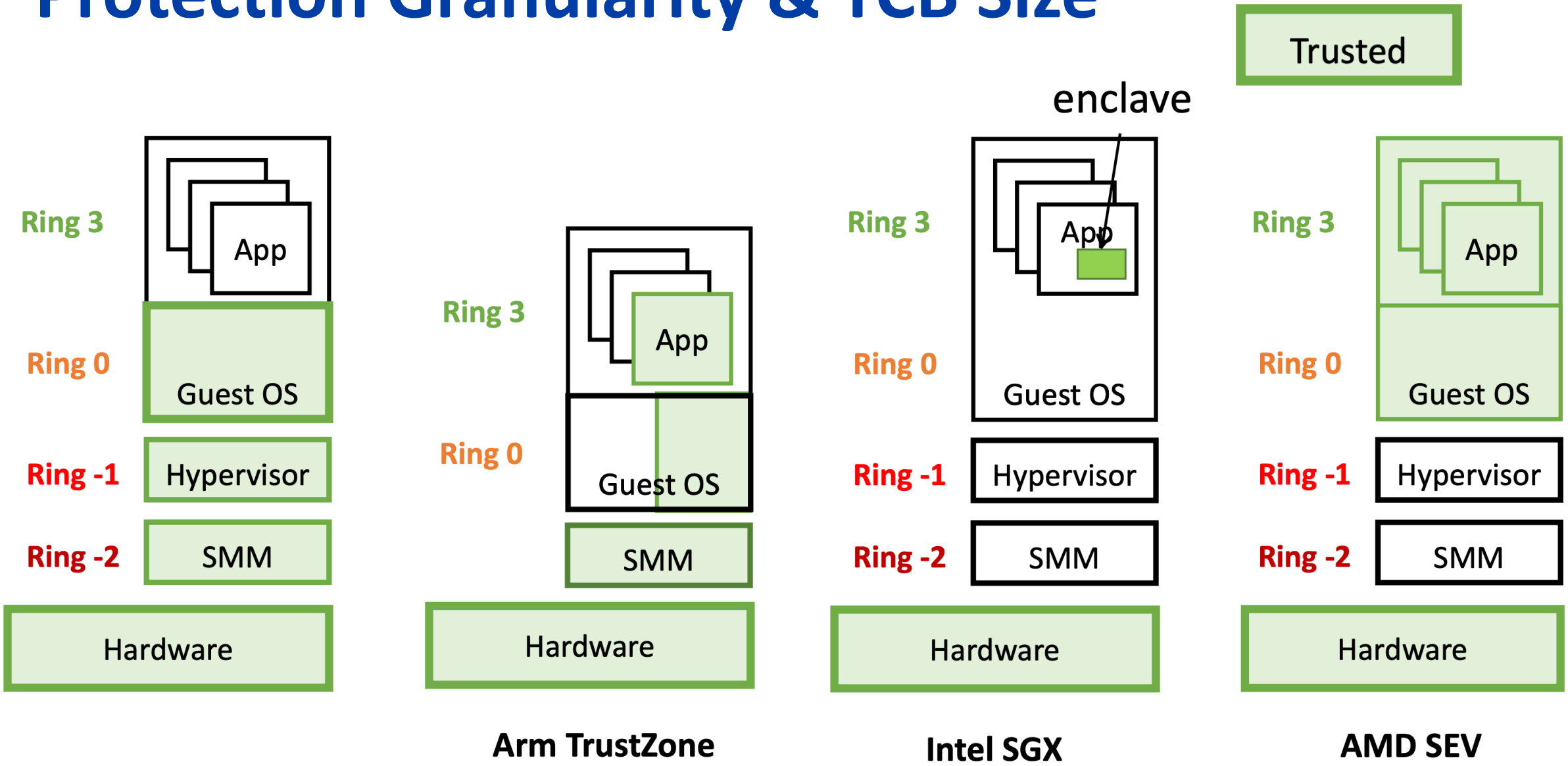
--- defined by Confidential Computing Consortium



TEE Examples



Protection Granularity & TCB Size

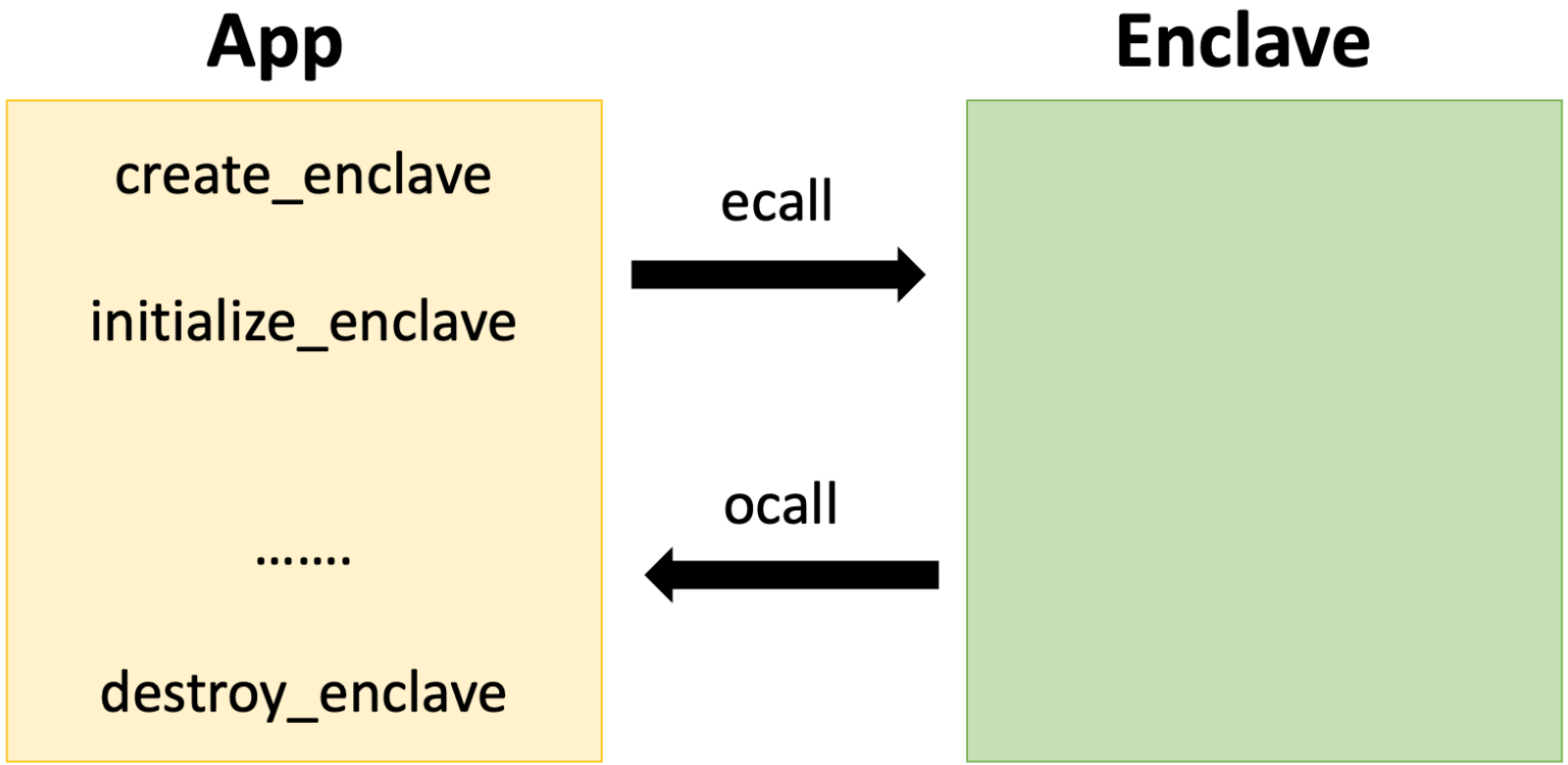


Arm TrustZone **Intel SGX** **AMD SEV**

TCB = Trusted Computing Base

SGX Enclave Programming Model

- Examples from: https://github.com/intel/linux-sgx



Security Tasks

Security Tasks

- How do we ensure the runtime execution follows our expectation (confidentiality and integrity of the execution)?

Security Tasks

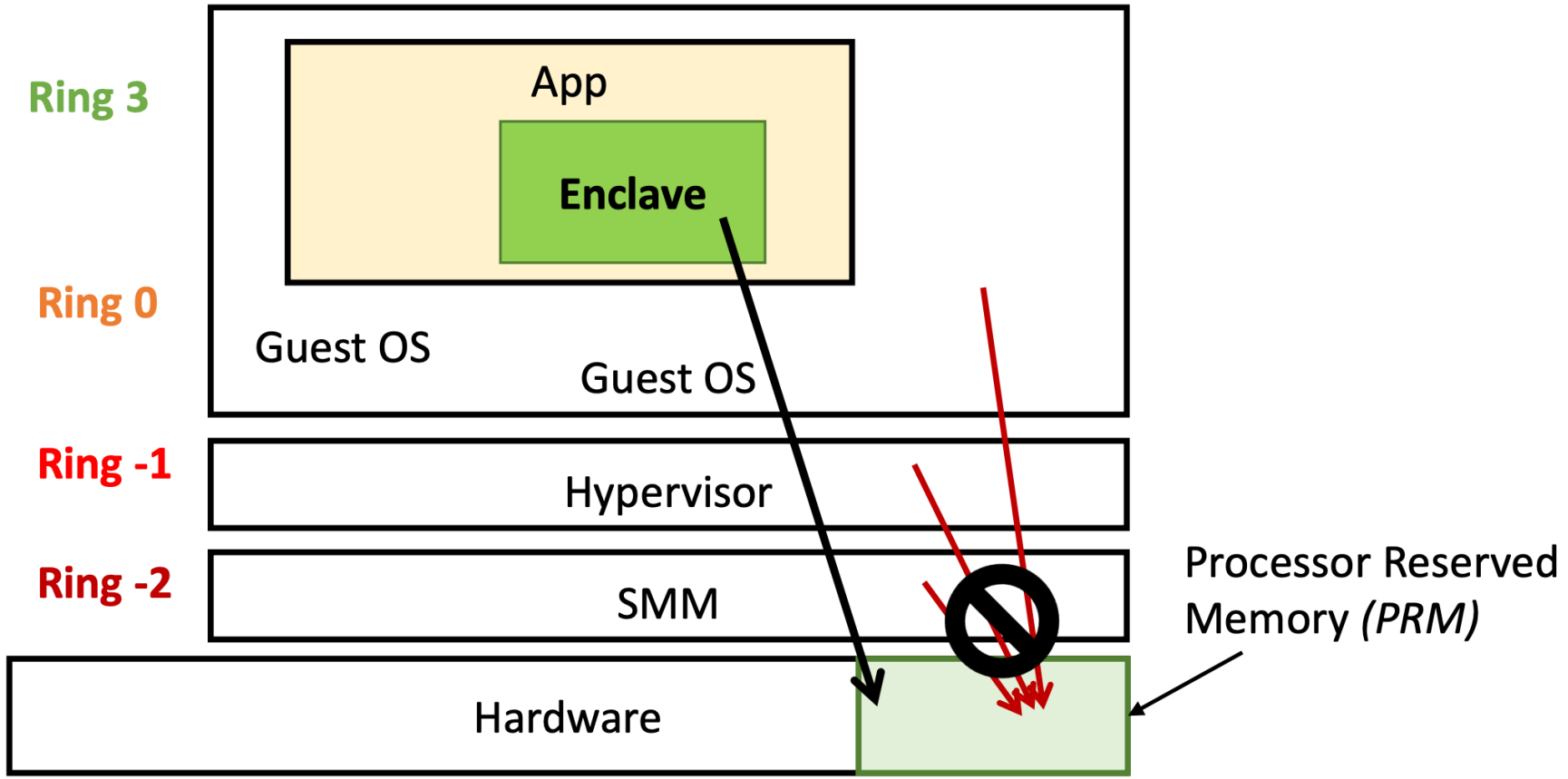
- How do we ensure the runtime execution follows our expectation (confidentiality and integrity of the execution)?

- How do we ensure the enclave code is the code that we want to execute? (code integrity during initialization)

Security Tasks

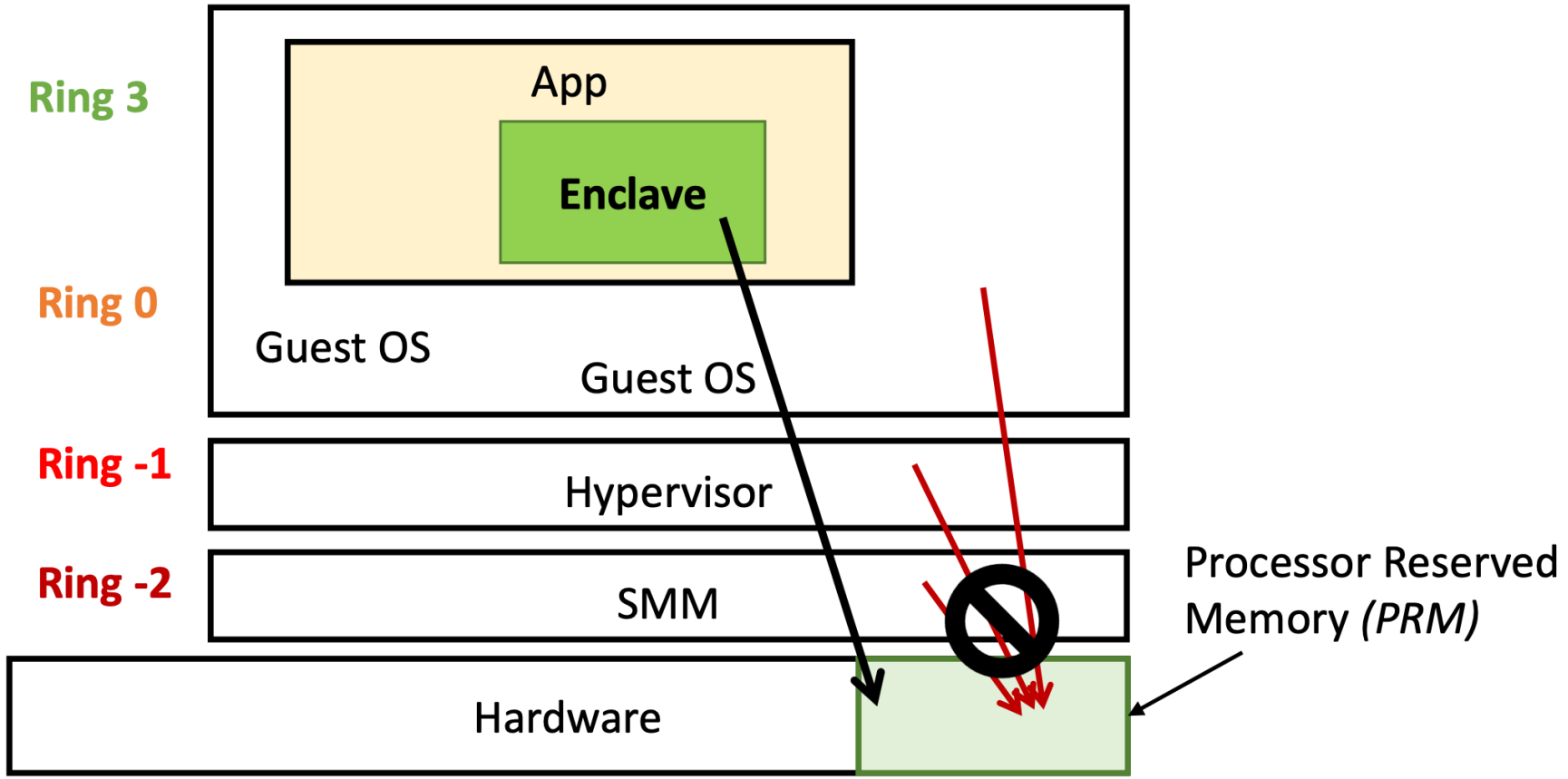
- How do we ensure the runtime execution follows our expectation (confidentiality and integrity of the execution)?
- How do we ensure the enclave code is the code that we want to execute? (code integrity during initialization)
- DRAM security? How to deal with Rowhammer and Coldboot attacks? (physical attacks)

Intel SGX Overview



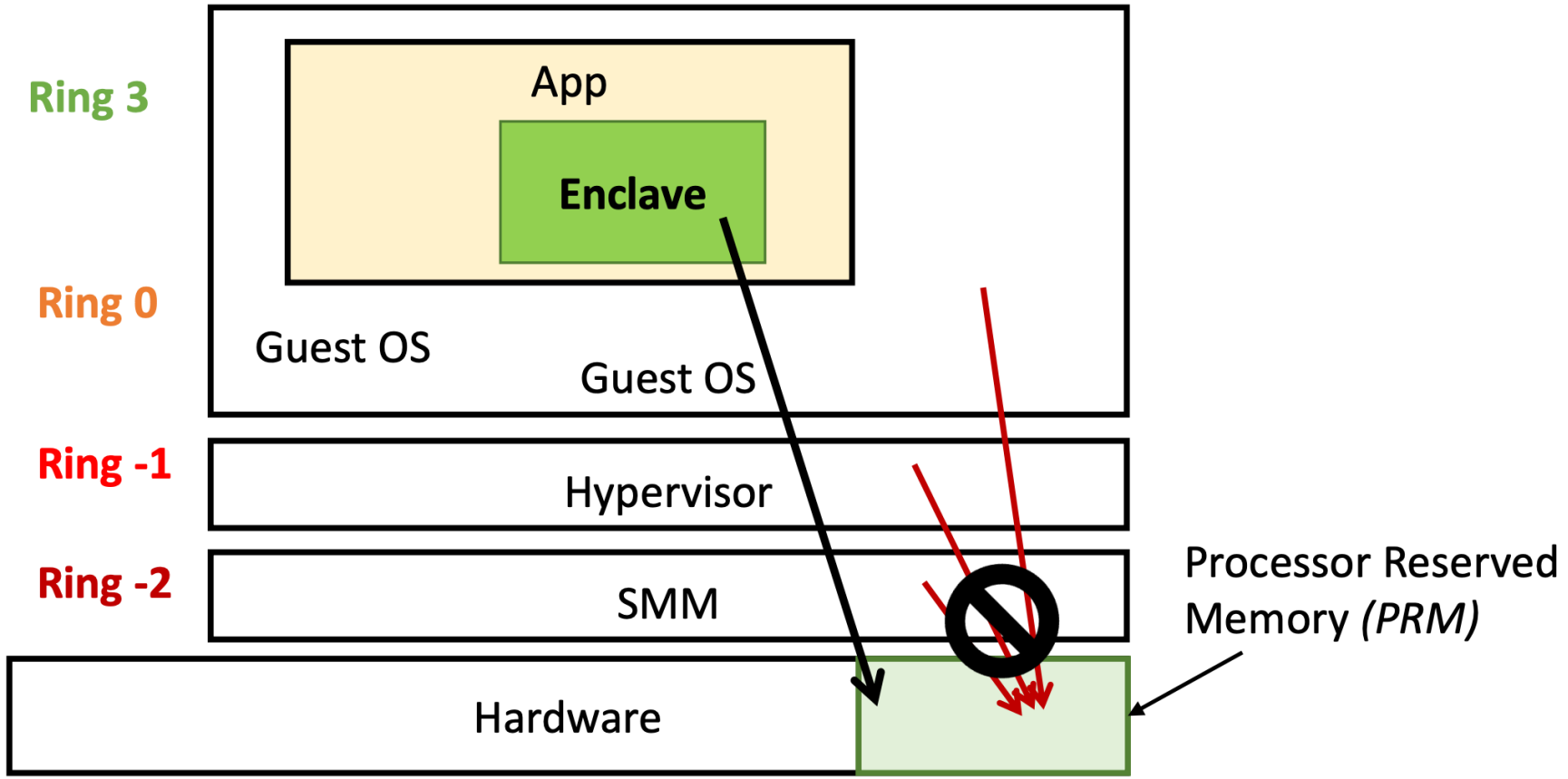
Intel SGX Overview

- Enclave code/data map to PRM

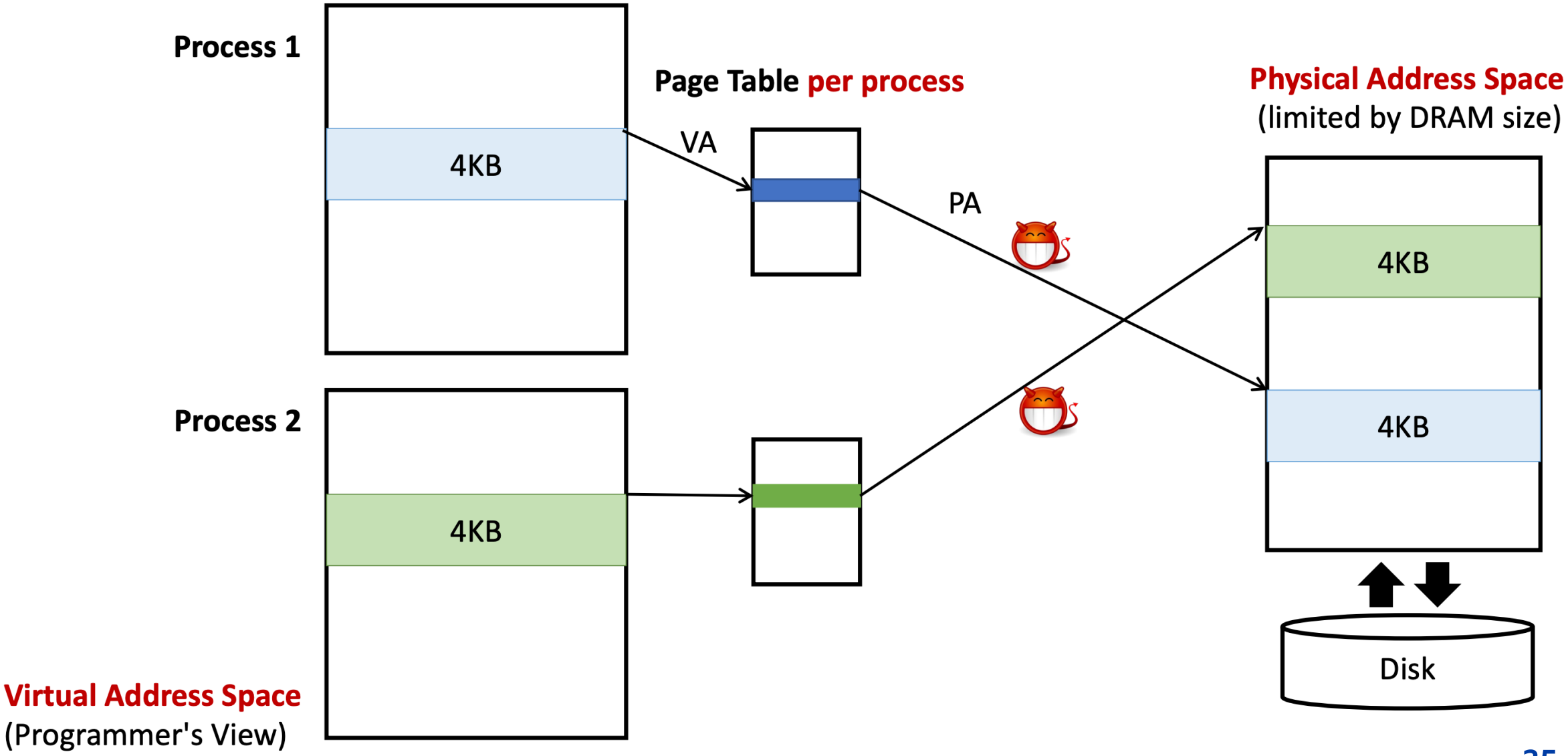


Intel SGX Overview

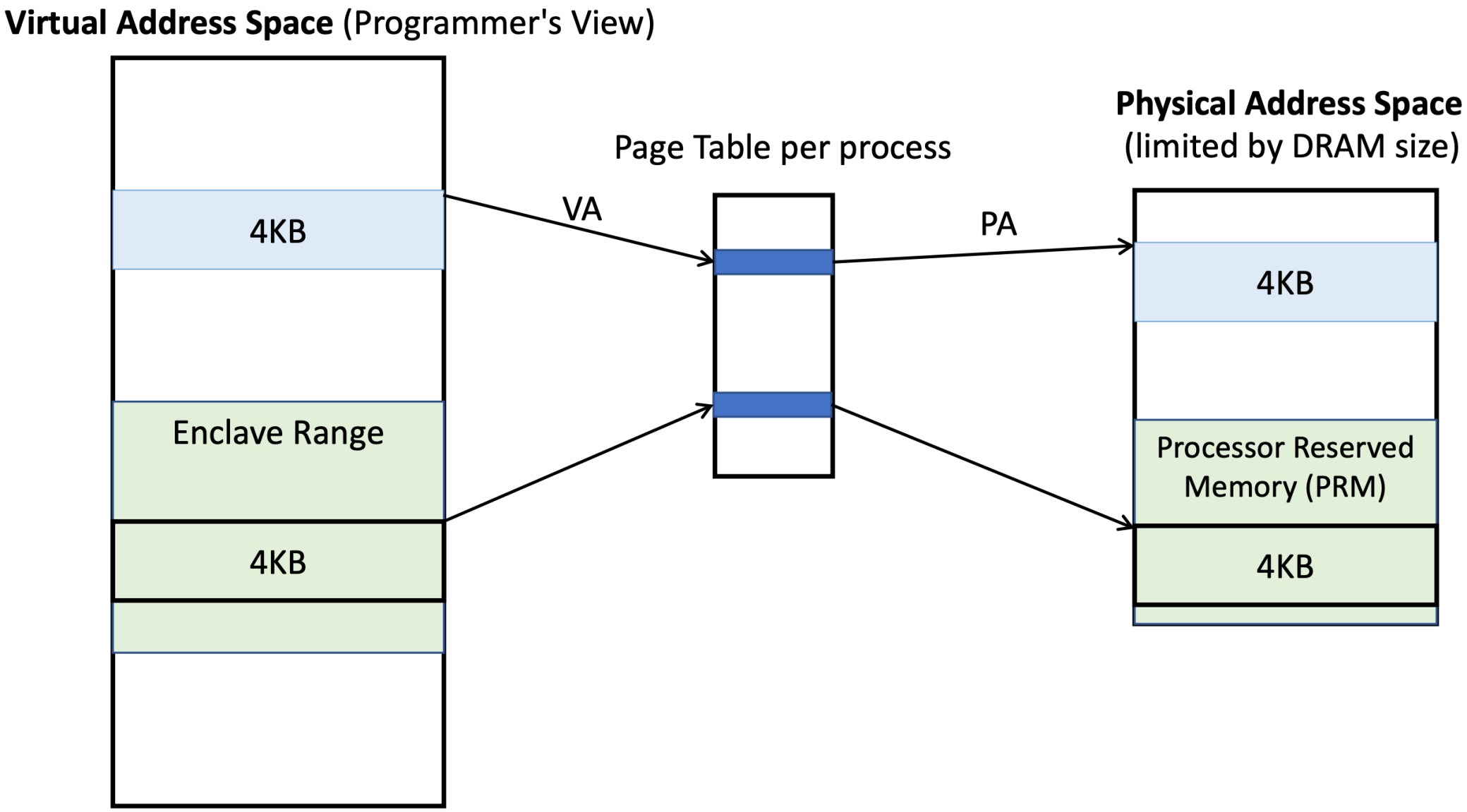
- Enclave code/data map to PRM
- Different enclaves access their own memory region



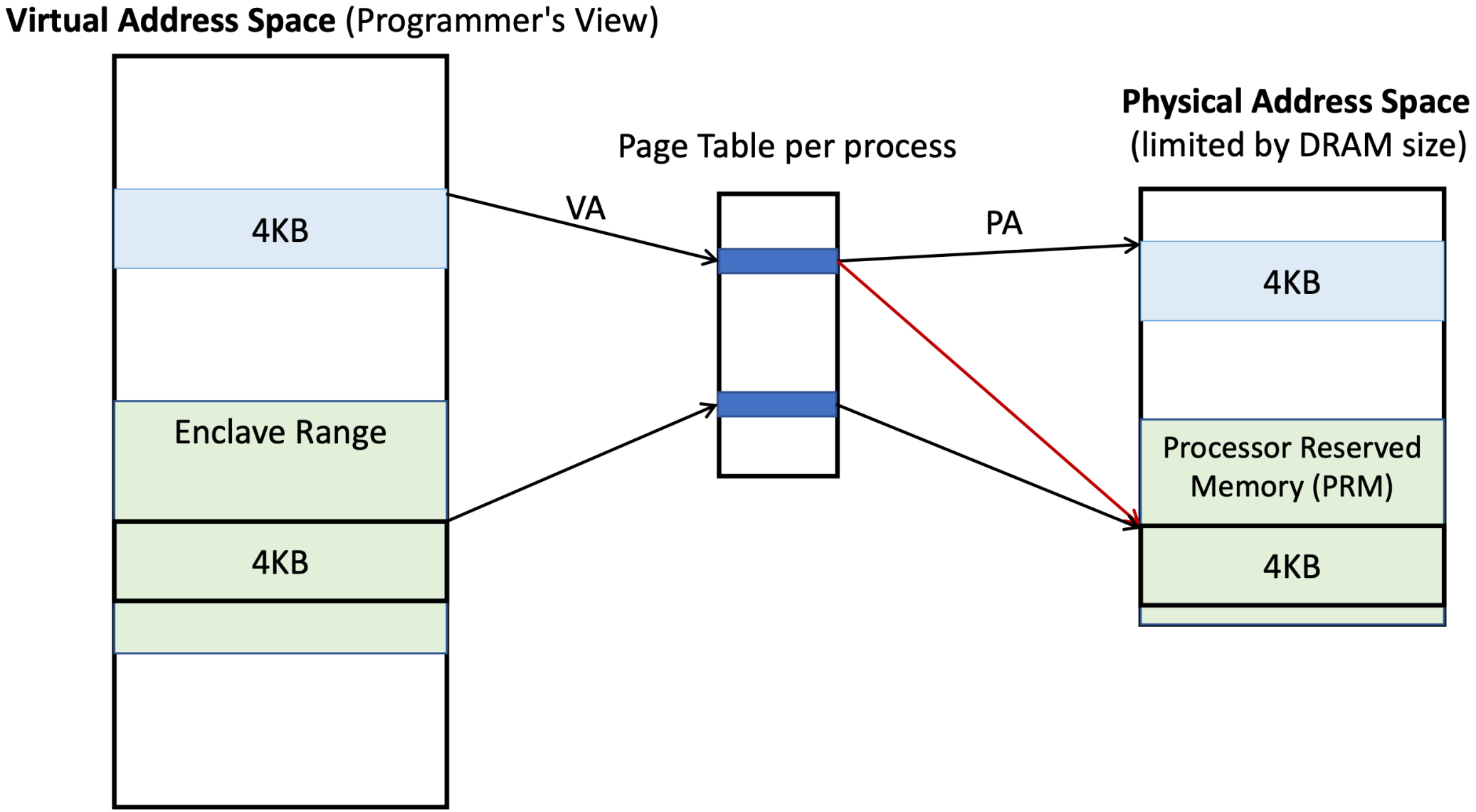
Virtual Memory Abstraction



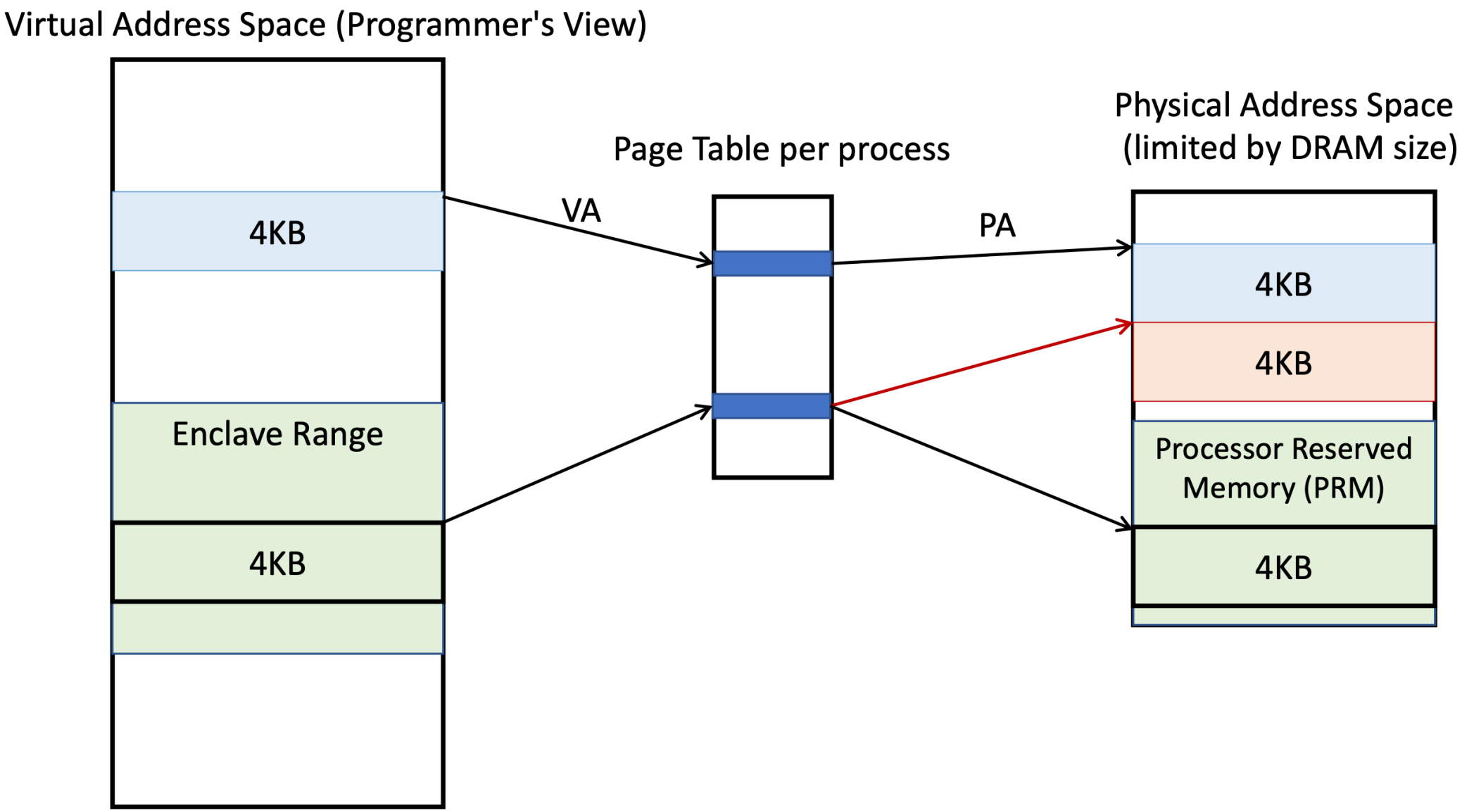
Intel SGX Address Translation Overview



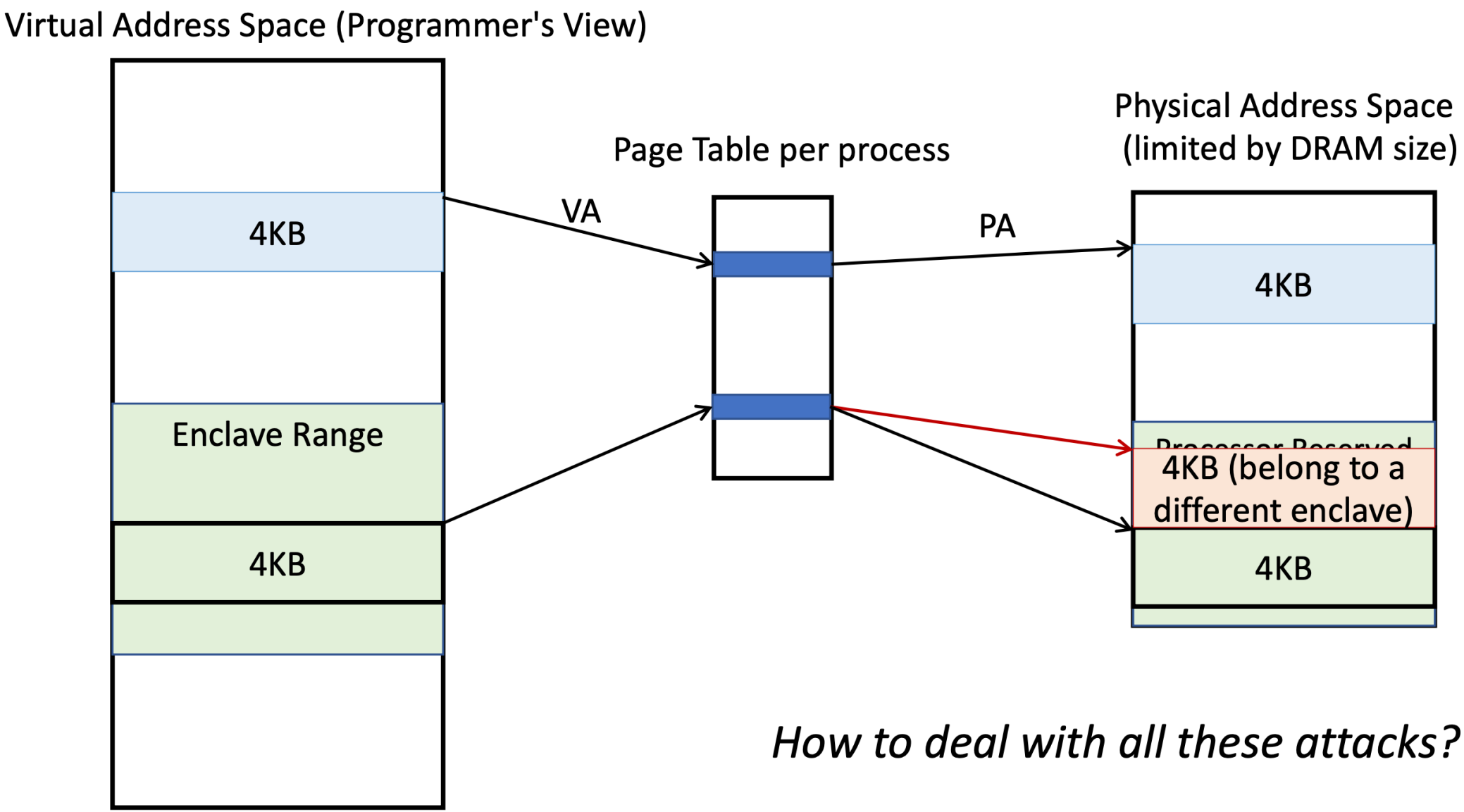
Malicious Address Translation #1



Malicious Address Translation #2

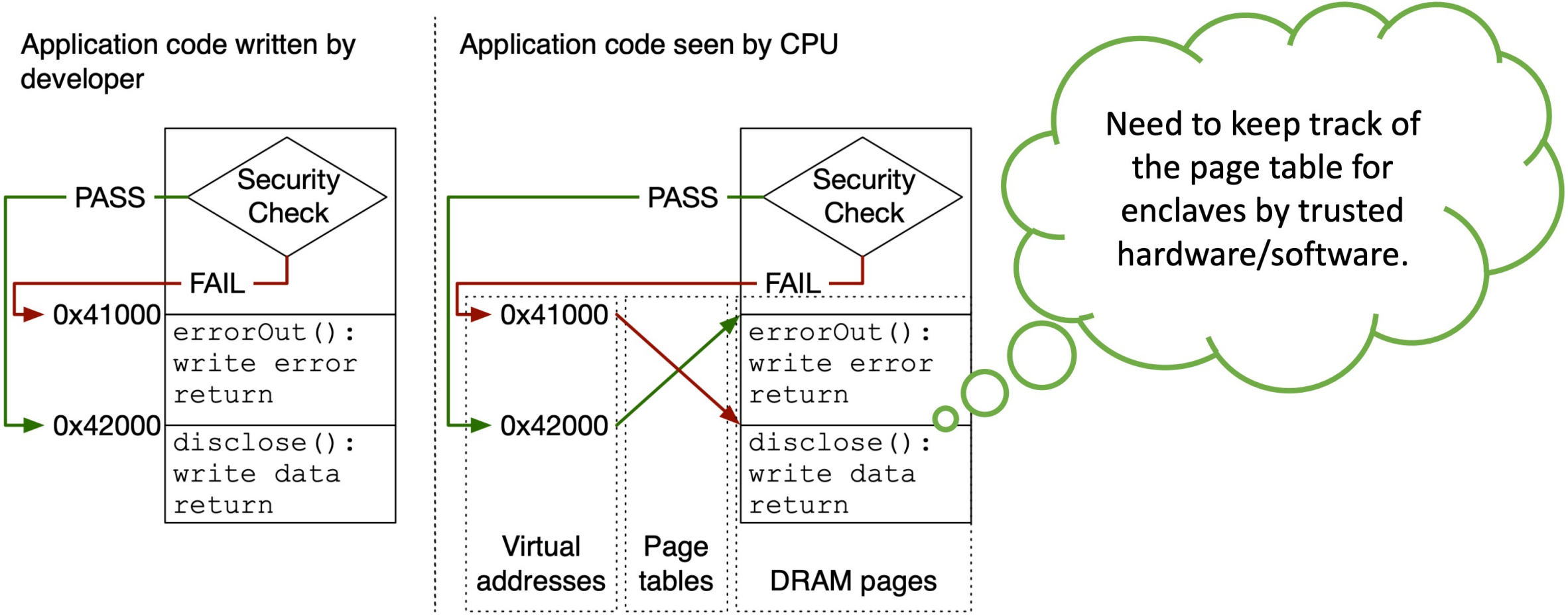


Malicious Address Translation #3



How to deal with all these attacks?

Malicious Address Translation #4



Solution: Inverted Page Table

PPN = Physical Page Number
VPN = Virtual Page Number

Solution: Inverted Page Table

- Check for security invariant:
  - Enclave VA, enclave mode -> PRM
  - Non-enclave mode is not allowed access PRM using whitherever address

PPN = Physical Page Number
VPN = Virtual Page Number

Solution: Inverted Page Table

- Check for security invariant:
  - Enclave VA, enclave mode -> PRM
  - Non-enclave mode is not allowed access PRM using whitherever address

- For each page in the PRM, remember the mapping from
  - [PPN] -> [VPN, Enclave ID]
  - Keep the reversed page table in PRM, so privilege software cannot modify

PPN = Physical Page Number
VPN = Virtual Page Number

Solution: Inverted Page Table

- Check for security invariant:
  - Enclave VA, enclave mode -> PRM
  - Non-enclave mode is not allowed access PRM using whitherever address

- For each page in the PRM, remember the mapping from
  - [PPN] -> [VPN, Enclave ID]
  - Keep the reversed page table in PRM, so privilege software cannot modify

- When to perform the check? (Review address translation process)
  - After each address translation

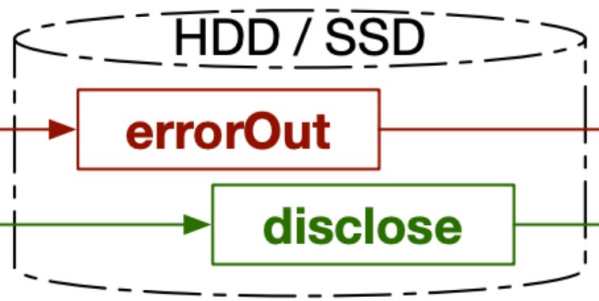
PPN = Physical Page Number
VPN = Virtual Page Number

Malicious Address Translation #5

A memory mapping attack that does not require modifying the page tables.

Page tables and DRAM before swapping

Virtual	Physical	Contents
0x41000	0x19000	errorOut
0x42000	0x1A000	disclose

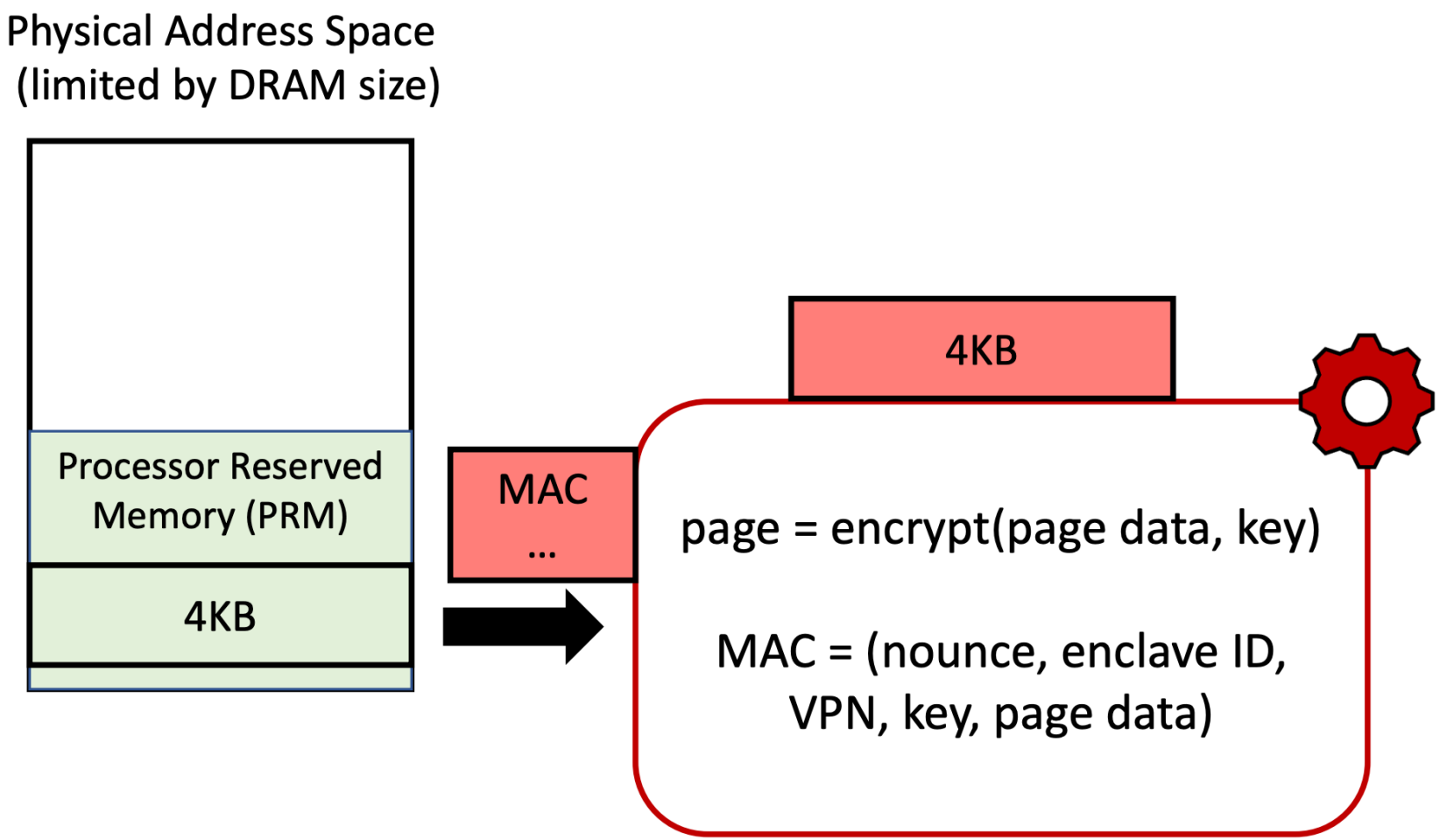


Need to bind the virtual address mapping with the page content.

Page tables and DRAM after swapping

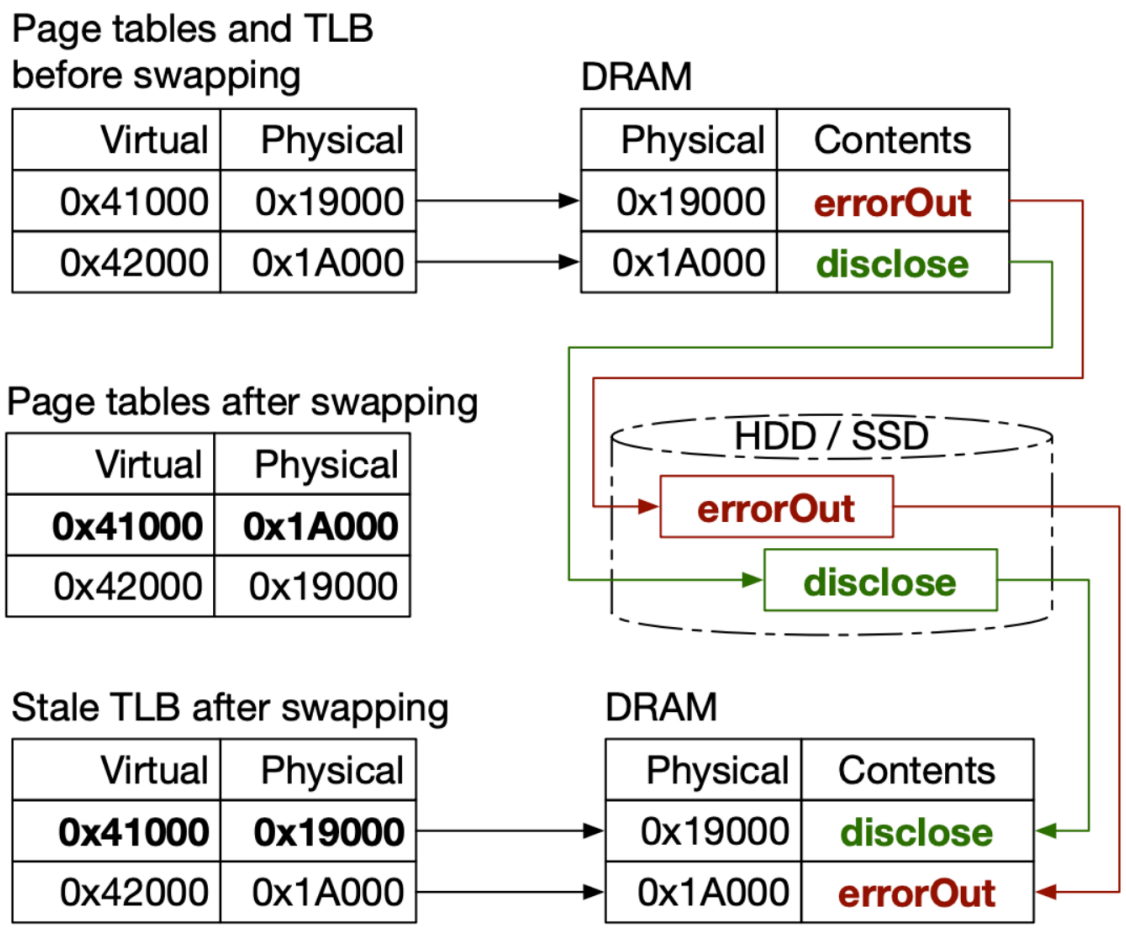
Virtual	Physical	Contents
0x41000	0x19000	disclose
0x42000	0x1A000	errorOut

Solution: Page Encryption and Authentication



Malicious Address Translation #6

A memory mapping attack that exploits stable TLB entries.



Need to make sure TLB is never obsolete.

TLB = Translation lookaside buffer

Solution: Keep TLB up-to-date

PPN = Physical Page Number
VPN = Virtual Page Number

Solution: Keep TLB up-to-date

- Keep an extra state in the inverted page table
  - [PPN] -> [VPN, Enclave ID]
  - [PPN, state] -> [VPN, Enclave ID]
  - Mark “blocked”
  - Unset only until all the VPNs (can be mapped by multiple enclaves) exist and flush TLBs

PPN = Physical Page Number
VPN = Virtual Page Number

Solution: Keep TLB up-to-date

- Keep an extra state in the inverted page table
  - [PPN] -> [VPN, Enclave ID]
  - [PPN, state] -> [VPN, Enclave ID]
  - Mark “blocked”
  - Unset only until all the VPNs (can be mapped by multiple enclaves) exist and flush TLBs

- If the TLB has stale data, post address translation check will see the physical address is “blocked”

PPN = Physical Page Number
VPN = Virtual Page Number

Summary: SGX Memory Management

Summary: SGX Memory Management

- #1: Maintain an inverted page table and check after every address translation
  - Physical page in PRM -> (enclave ID, virtual page number)

Summary: SGX Memory Management

- #1: Maintain an inverted page table and check after every address translation
  - Physical page in PRM -> (enclave ID, virtual page number)

- #2: Encrypt/decrypt upon page swap to non-PRM region
  - (nonce, enclave ID, virtual page number, key, page content) -> MAC

Summary: SGX Memory Management

- #1: Maintain an inverted page table and check after every address translation
  - Physical page in PRM -> (enclave ID, virtual page number)

- #2: Encrypt/decrypt upon page swap to non-PRM region
  - (nonce, enclave ID, virtual page number, key, page content) -> MAC

- #3: Keep TLB state up-to-date
  - Upon page swap, block the page in the inverted page table and unblock only until all the corresponding TLB entries are flushed

Security Tasks

- How do we ensure the runtime execution follows our expectation (confidentiality and integrity of the execution)?

- How do we ensure the enclave code is the code that we want to execute? (code integrity during initialization)

- DRAM security? How to deal with Rowhammer and Coldboot attacks? (physical attacks)

Security Tasks

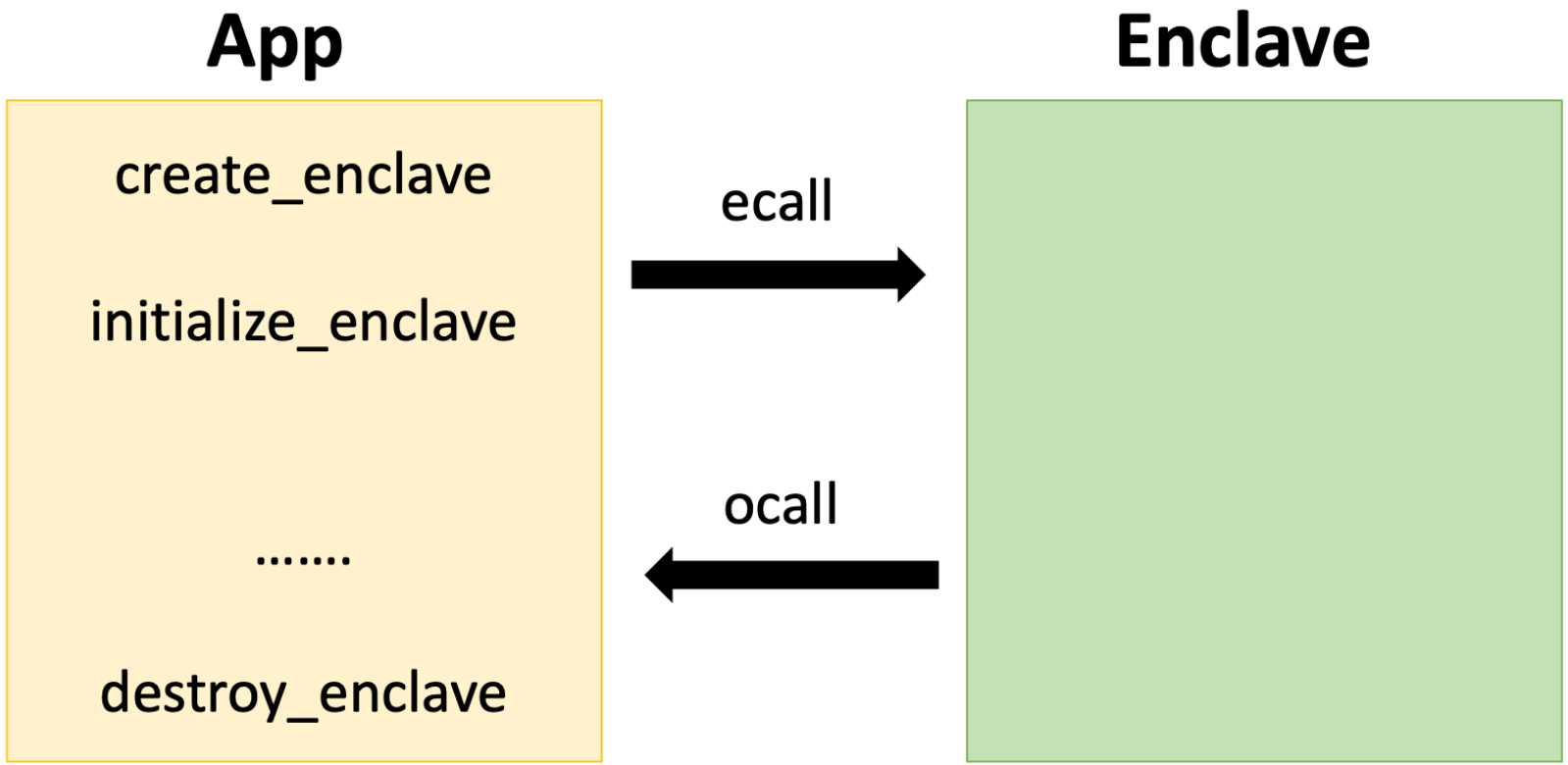
- How do we ensure the runtime execution follows our expectation (confidentiality and integrity of the execution)?

- How do we ensure the enclave code is the code that we want to execute? (code integrity during initialization)

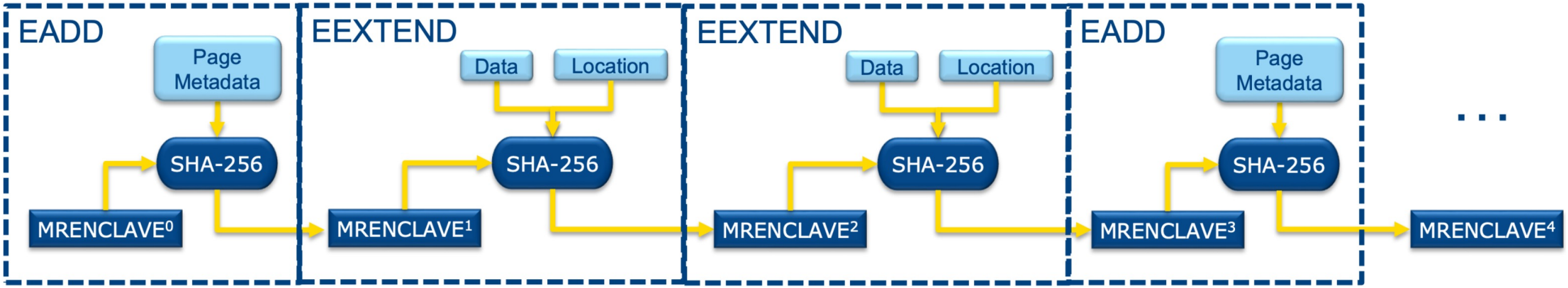
- DRAM security? How to deal with Rowhammer and Coldboot attacks? (physical attacks)

Review: SGX Enclave Programming Model

- How to ensure the enclave is initialized correctly?

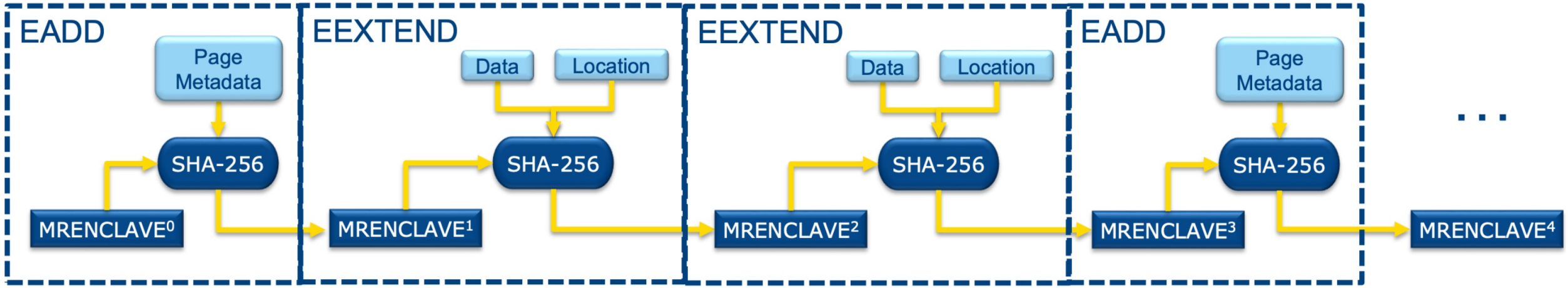


Enclave Measurement



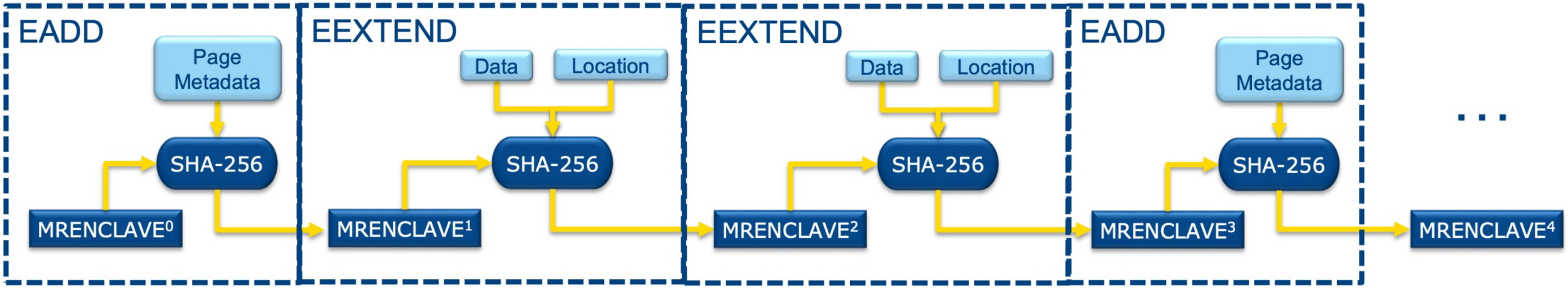
Enclave Measurement

- Hardware generates a cryptographic log of the build process
  - Code, data, stack, and heap contents
  - Location of each page within the enclave
  - Security attributes (e.g., page permissions) and enclave capabilities



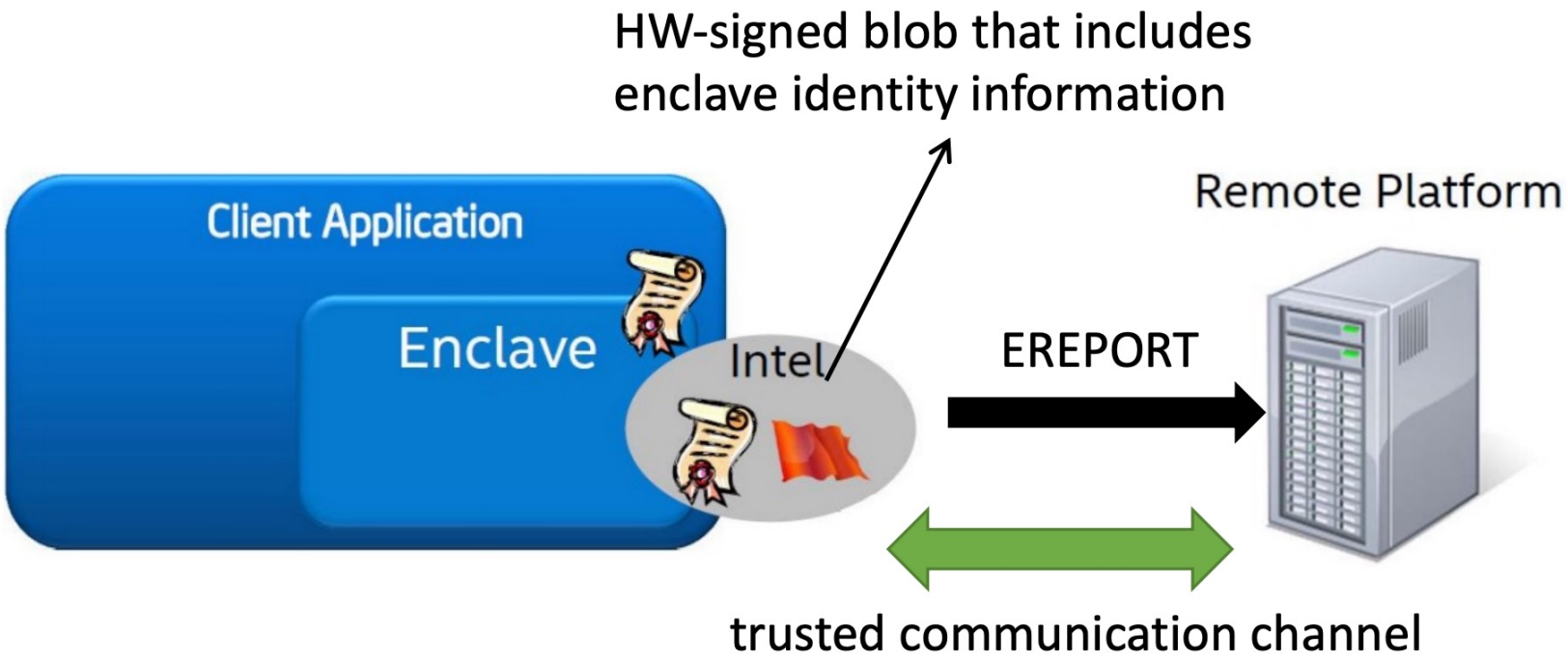
Enclave Measurement

- Hardware generates a cryptographic log of the build process
  - Code, data, stack, and heap contents
  - Location of each page within the enclave
  - Security attributes (e.g., page permissions) and enclave capabilities
- Enclave identity (MRENCLAVE) is a 256-bit digest of the log that represents the enclave

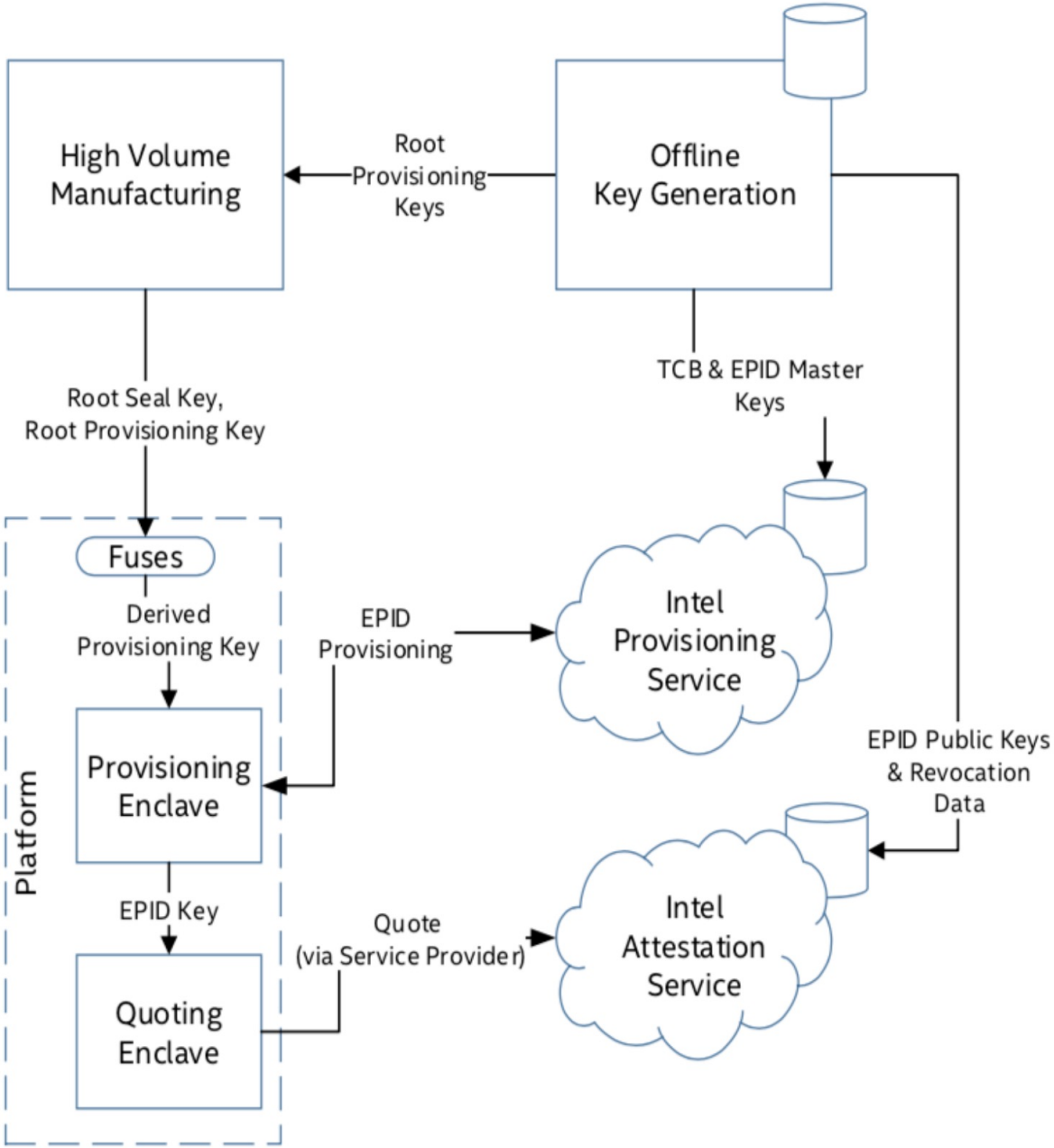


Enclave Measurement

- HW based attestation provides evidence that “this is the right application executing on an authentic platform” (approach similar to secure boot attestation)

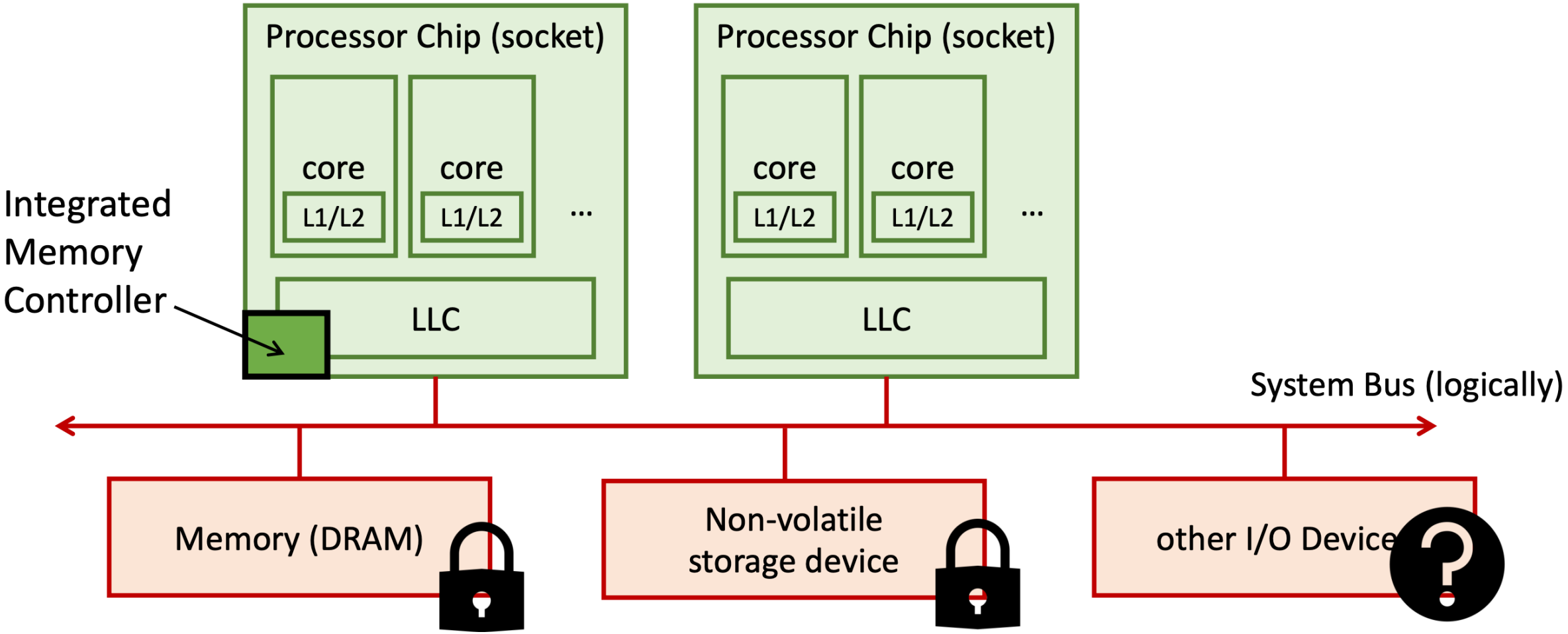


SGX Infrastructure Services – Chain of Trust



Additional Security Threats

- DRAM attacks: Rowhammer, Coldboot attacks



Additional Security Threats

Additional Security Threats

- Confidentiality:
  - DATA written to the DRAM cannot be distinguished from random data.

Additional Security Threats

- Confidentiality:
  - DATA written to the DRAM cannot be distinguished from random data.

- Integrity + freshness:
  - DATA read back from DRAM to LLC is the same DATA that was most recently written from LLC to DRAM.

Additional Security Threats

- Confidentiality:
  - DATA written to the DRAM cannot be distinguished from random data.

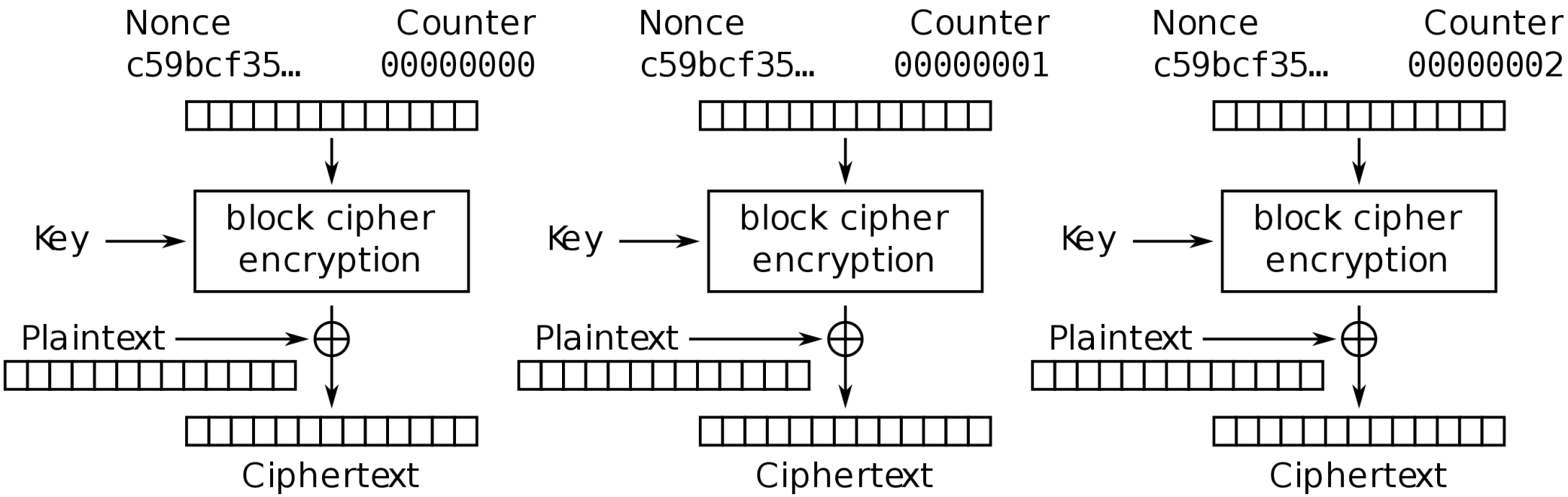
- Integrity + freshness:
  - DATA read back from DRAM to LLC is the same DATA that was most recently written from LLC to DRAM.

What attacks can be mitigated?

Rowhammer? Bus tapping?

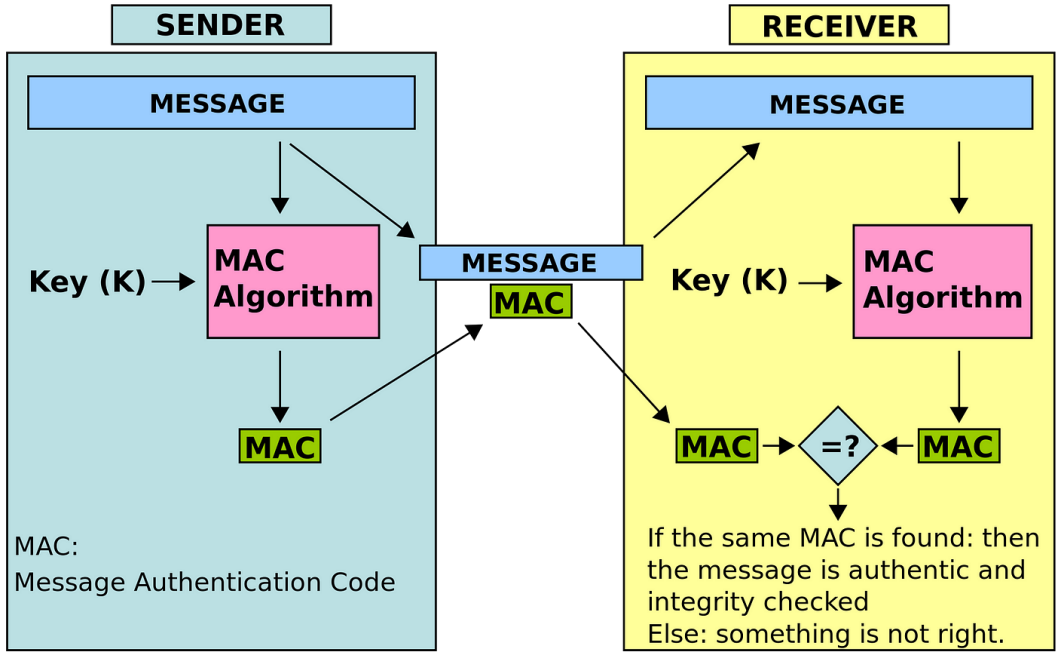
Confidentiality

- AES 128-CTR mode



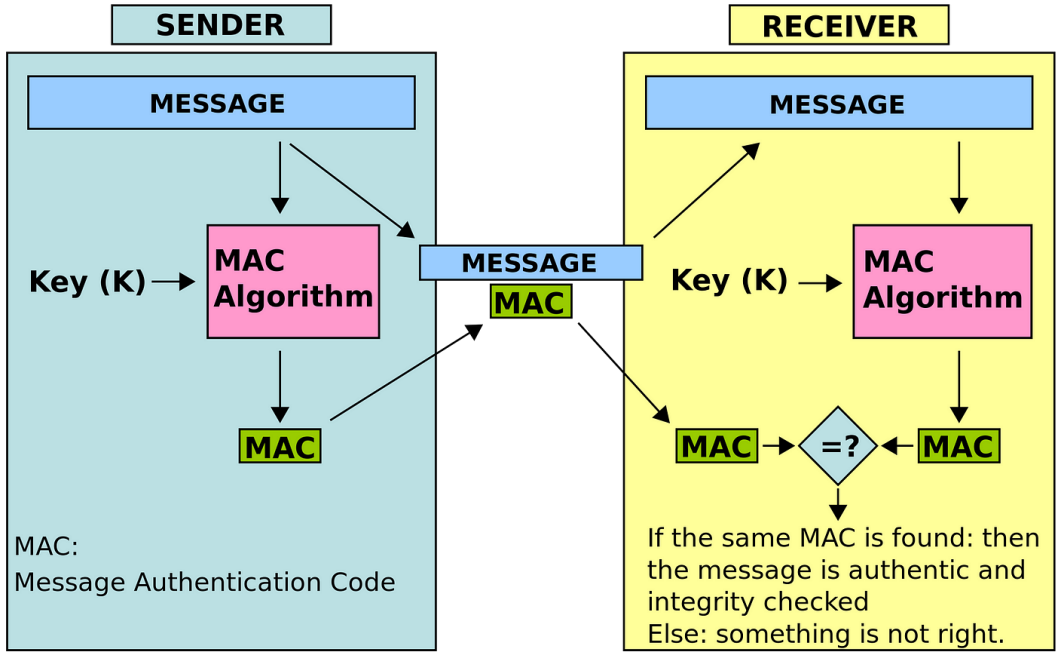
Counter (CTR) mode encryption

Message Authentication Code (MAC)



Message Authentication Code (MAC)

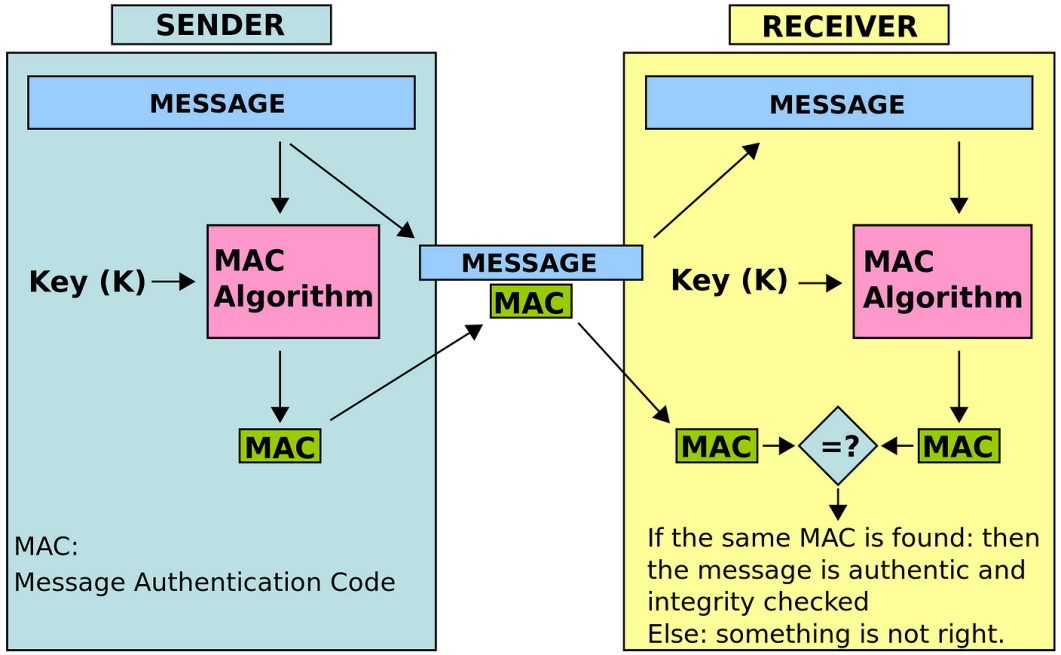
- `Hash(plaintext)`



Message Authentication Code (MAC)

- `Hash(plaintext)`

- Keyed Hash
  - `hash = SHA(message)`
  - `HMAC = enc(hash, key)`

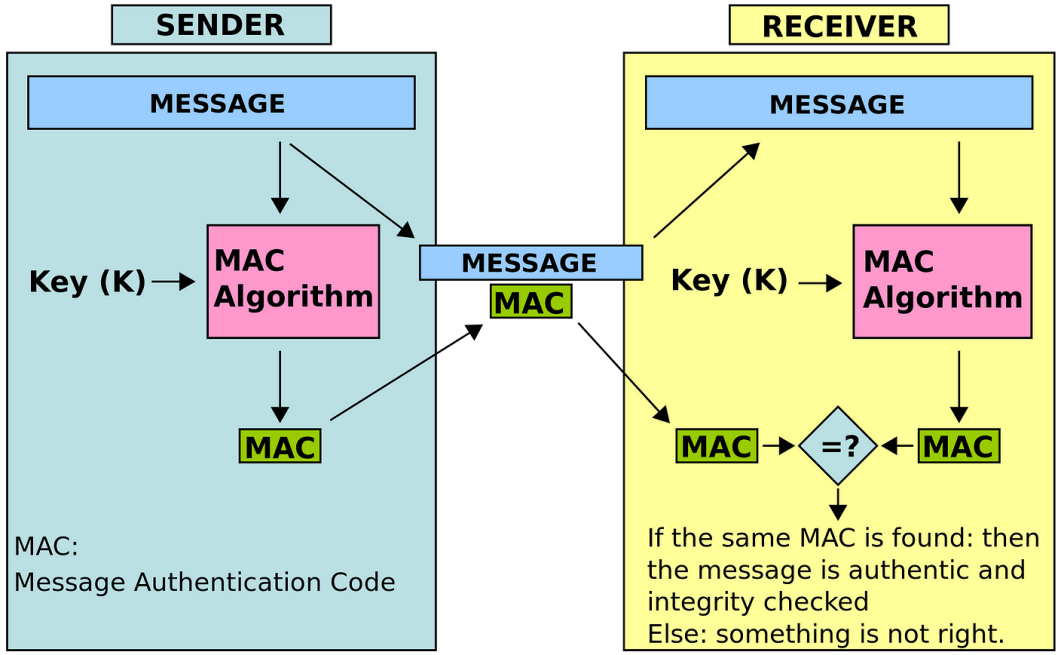


Message Authentication Code (MAC)

- `Hash(plaintext)`

- Keyed Hash
  - `hash = SHA(message)`
  - `HMAC = enc(hash, key)`

- Freshness
  - `hash = SHA(message||nonce)`
  - `HMAC = enc(hash, key)`



Integrity Storage Problem

Integrity Storage Problem

- For each cache line: {ciphertext + CTR + MAC}
  - MAC 56 bits
  - CTR 56 bits

Integrity Storage Problem

- For each cache line: {ciphertext + CTR + MAC}
  - MAC 56 bits
  - CTR 56 bits
- Can we store all the three components off-chip?

Integrity Storage Problem

- For each cache line: {ciphertext + CTR + MAC}
  - MAC 56 bits
  - CTR 56 bits
- Can we store all the three components off-chip?
- Problem: if store CTR on-chip -> high on-chip storage requirement

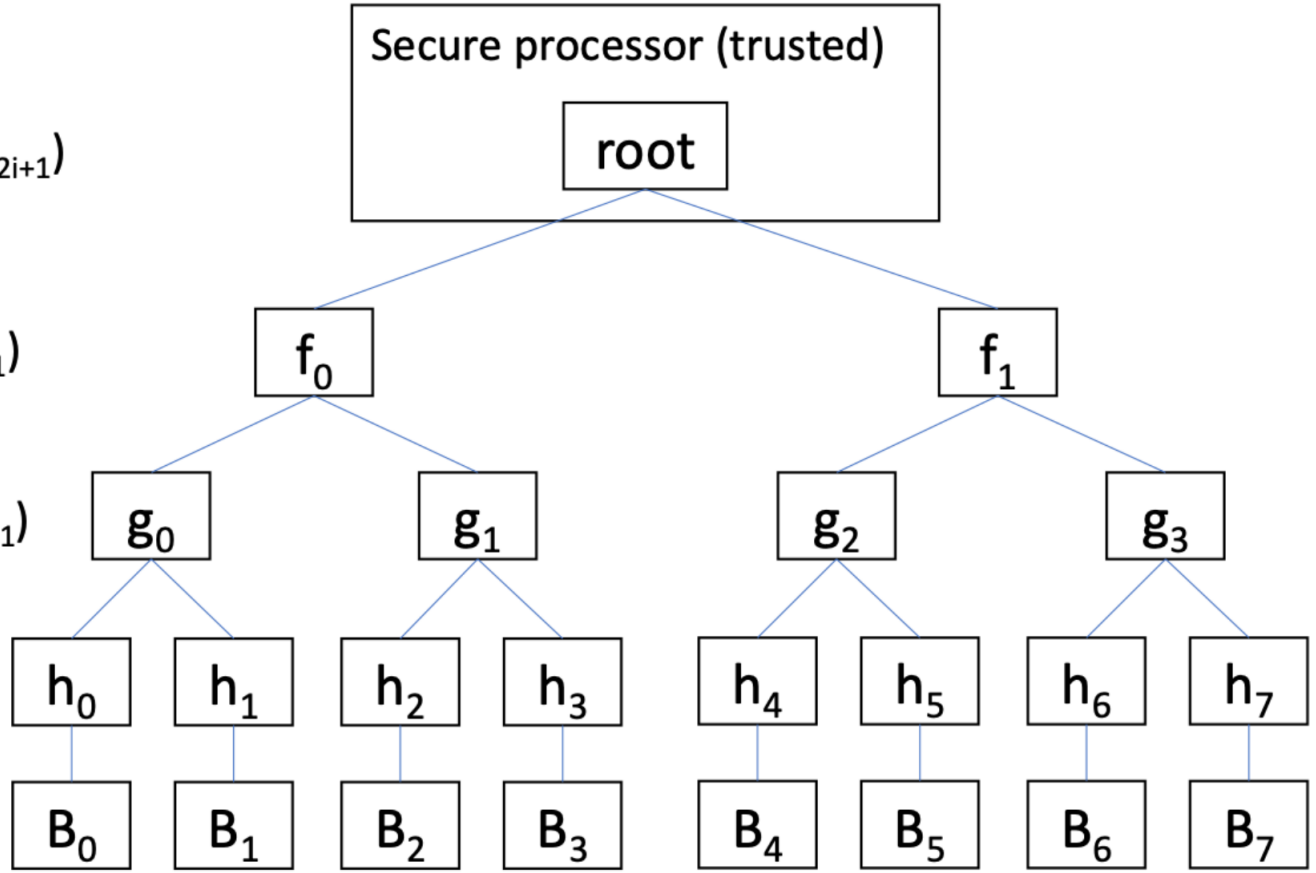
Operations on Merkle Tree

$root = Hash(f_{2i} || f_{2i+1})$

$f_i = Hash(g_{2i} || g_{2i+1})$

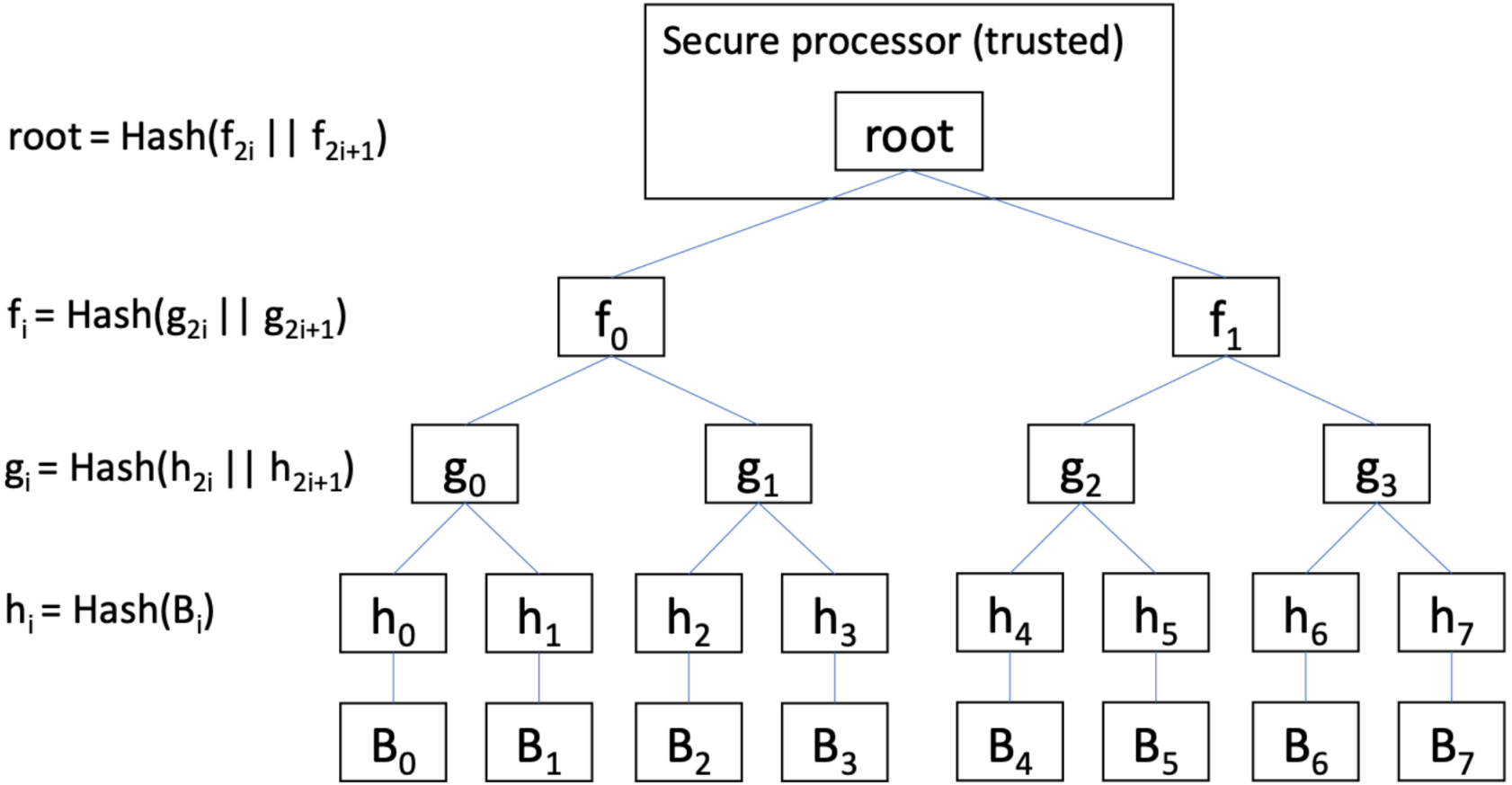
$g_i = Hash(h_{2i} || h_{2i+1})$

$h_i = Hash(B_i)$



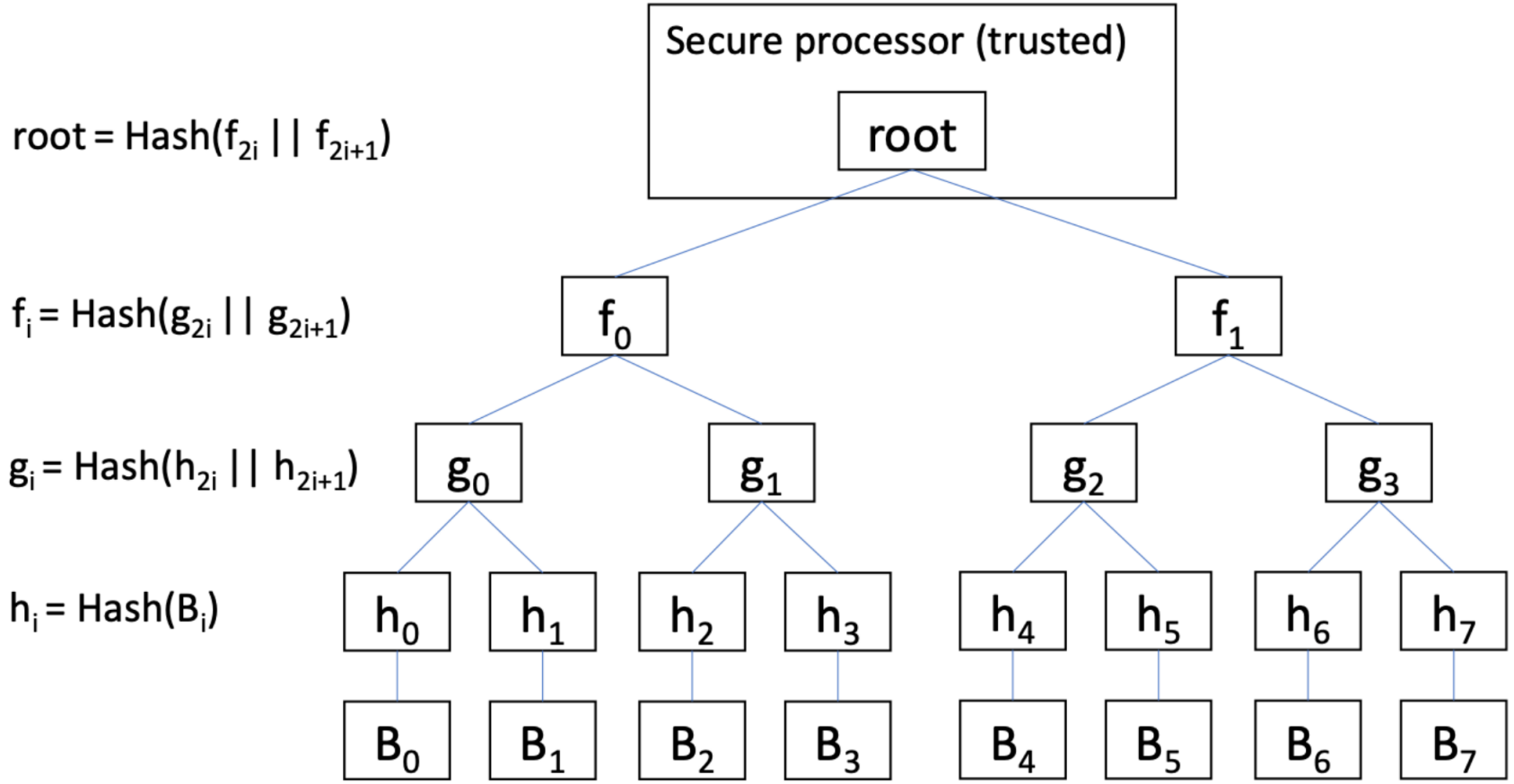
Operations on Merkle Tree

- Only need to store the root node on chip



Operations on Merkle Tree

- Only need to store the root node on chip
- How to verify block B1?



Operations on Merkle Tree

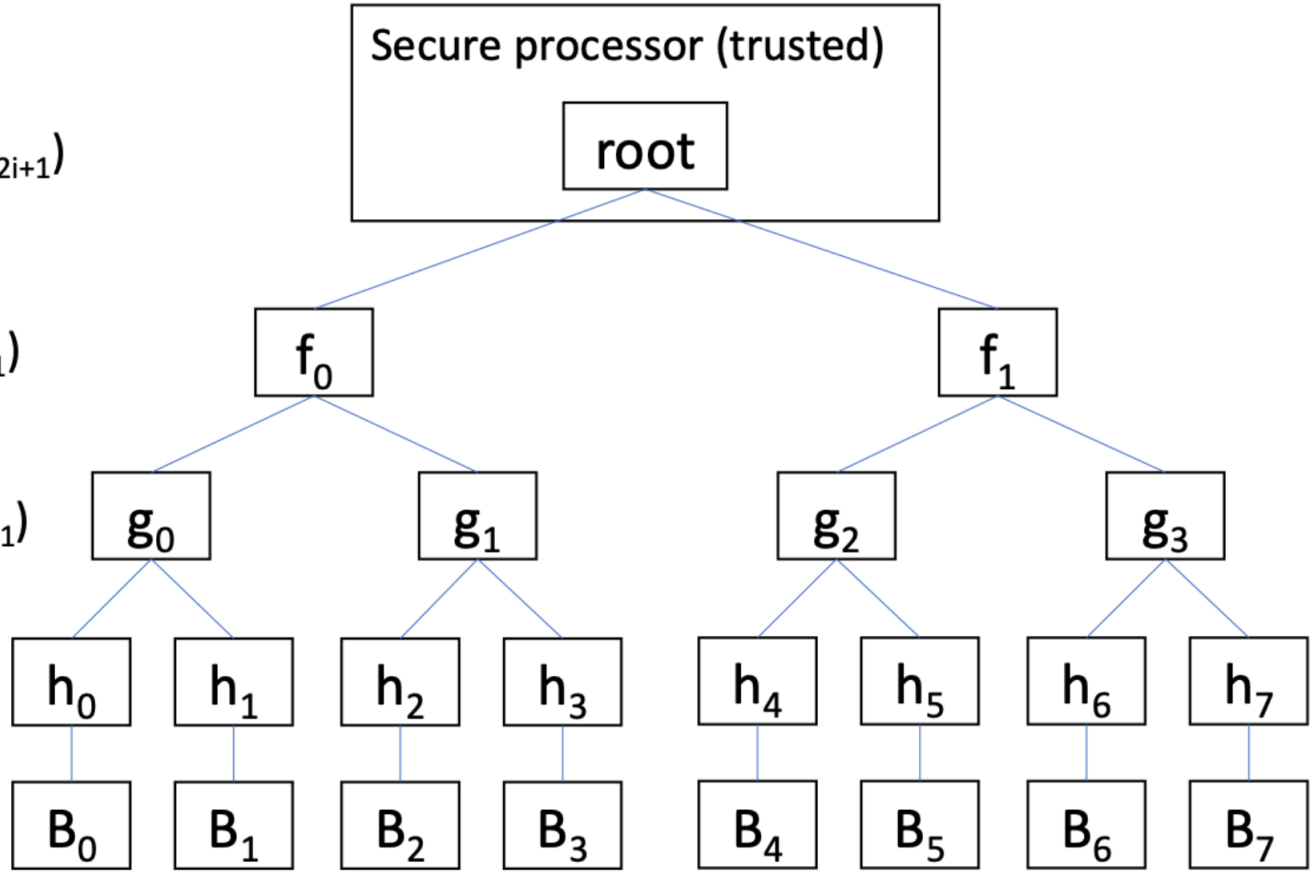
- Only need to store the root node on chip
- How to verify block B1?
- Write to block B3?

$root = Hash(f_{2i} || f_{2i+1})$

$f_i = Hash(g_{2i} || g_{2i+1})$

$g_i = Hash(h_{2i} || h_{2i+1})$

$h_i = Hash(B_i)$



Summary

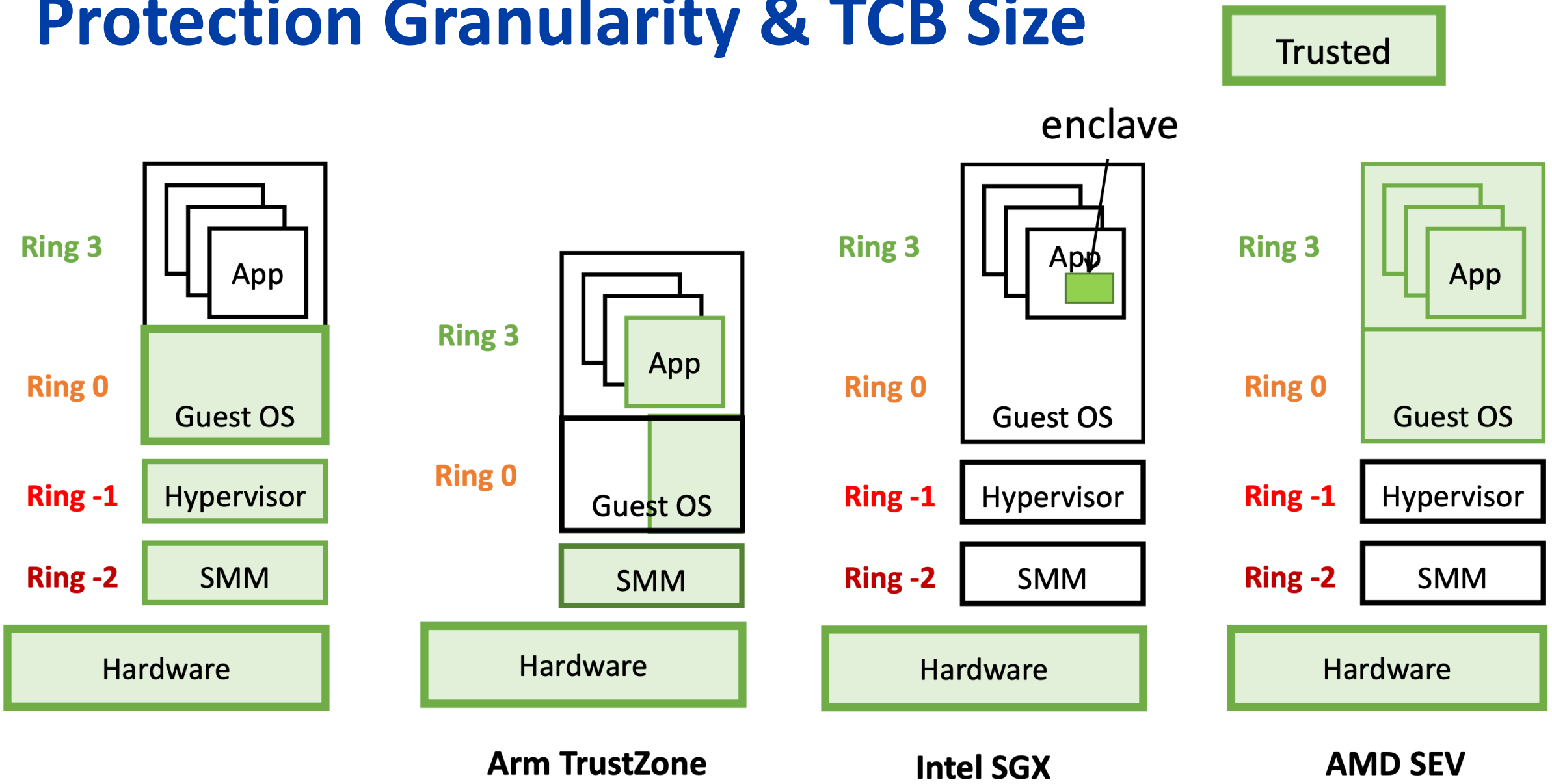
Summary

- How does typical Confidential Computing (Intel SGX) works

Summary

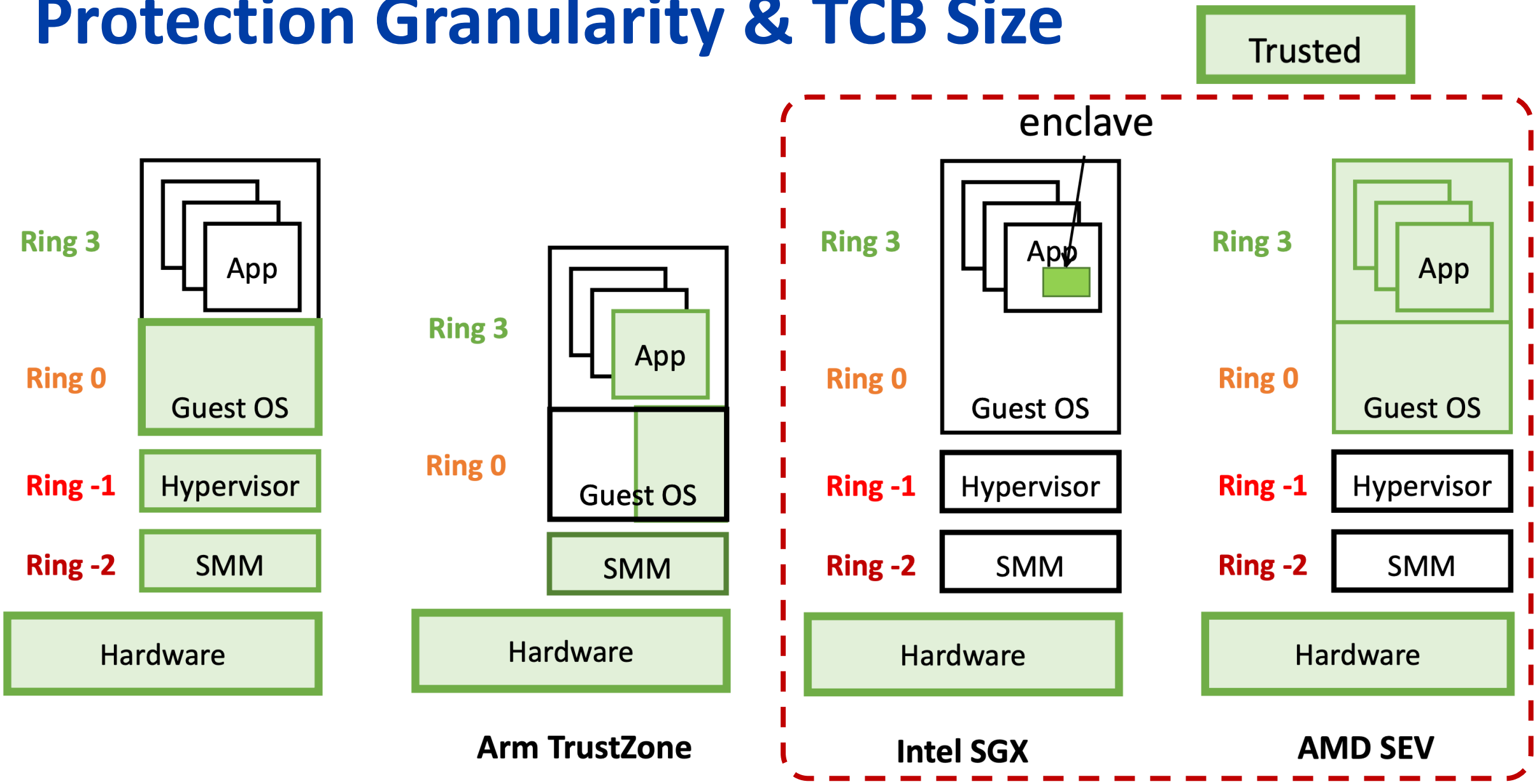
- How does typical Confidential Computing (Intel SGX) works
- Design tradeoffs between TCB size, flexibility, perf overhead, cost, etc.
  - Intel SGX, AMD SEV, ARM CCA
  - Keystone, Sanctum, Penglai, etc.

Protection Granularity & TCB Size



TCB = Trusted Computing Base

Protection Granularity & TCB Size



TCB = Trusted Computing Base

Function and Use Cases Comparison

Intel SGX	AMD Memory Encryption Technology (SEV)
Initial design targeted microservices and small workload. (small amount of secure memory and was featured mainly in mobile and desktop family processors)	Initial design targeted cloud and Infrastructure as a Service. (Large amount of secure memory featured in server family processors)
Requires major software changes and code refactoring. (Not suitable for securing legacy applications)	Does not require software changes and code refactoring. (Suitable for securing legacy applications)
SGX works with ring 3 and is not suitable for workloads with many system calls.	SEV works with ring 0 and is suitable for broader range of workloads especially those with many system calls.
SGX is suitable for small but security-sensitive workload. (SGX has small TCB)	SEV is suitable for securing legacy, large and enterprise level application. (SEV has large TCB)

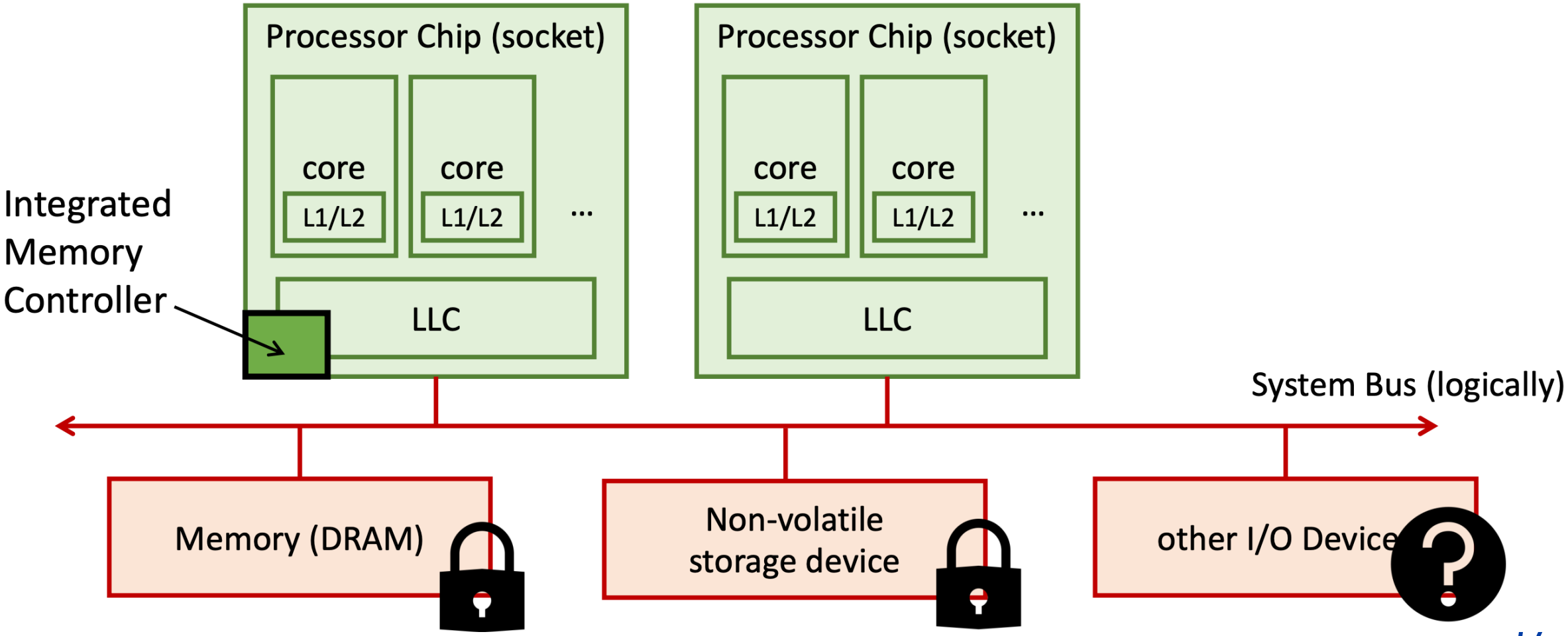
Security and Vulnerability Comparison

Intel SGX	AMD SEV, SEV-ES, SEV-SNP
Provides Memory Integrity Protection.	Provides Memory Integrity Protection.
Vulnerable to Memory Side Channels.	Vulnerable to Memory Side Channels.
Vulnerable to Denial of Service Attacks. (OS Handles System Calls)	Vulnerable to Denial of Service Attacks. (Hypervisor Handles VM Requests)
Small TCB. (TCB is CPU package)	Large TCB. (VM's OS is located inside TCB)
Vulnerable to Synchronization Attacks. (TOCTTOU, Use-After-Free)	AMD Secure Processor Firmware Bug Discovered. (MASTERKEY and FALLOUT)

GPU Confidential Computing

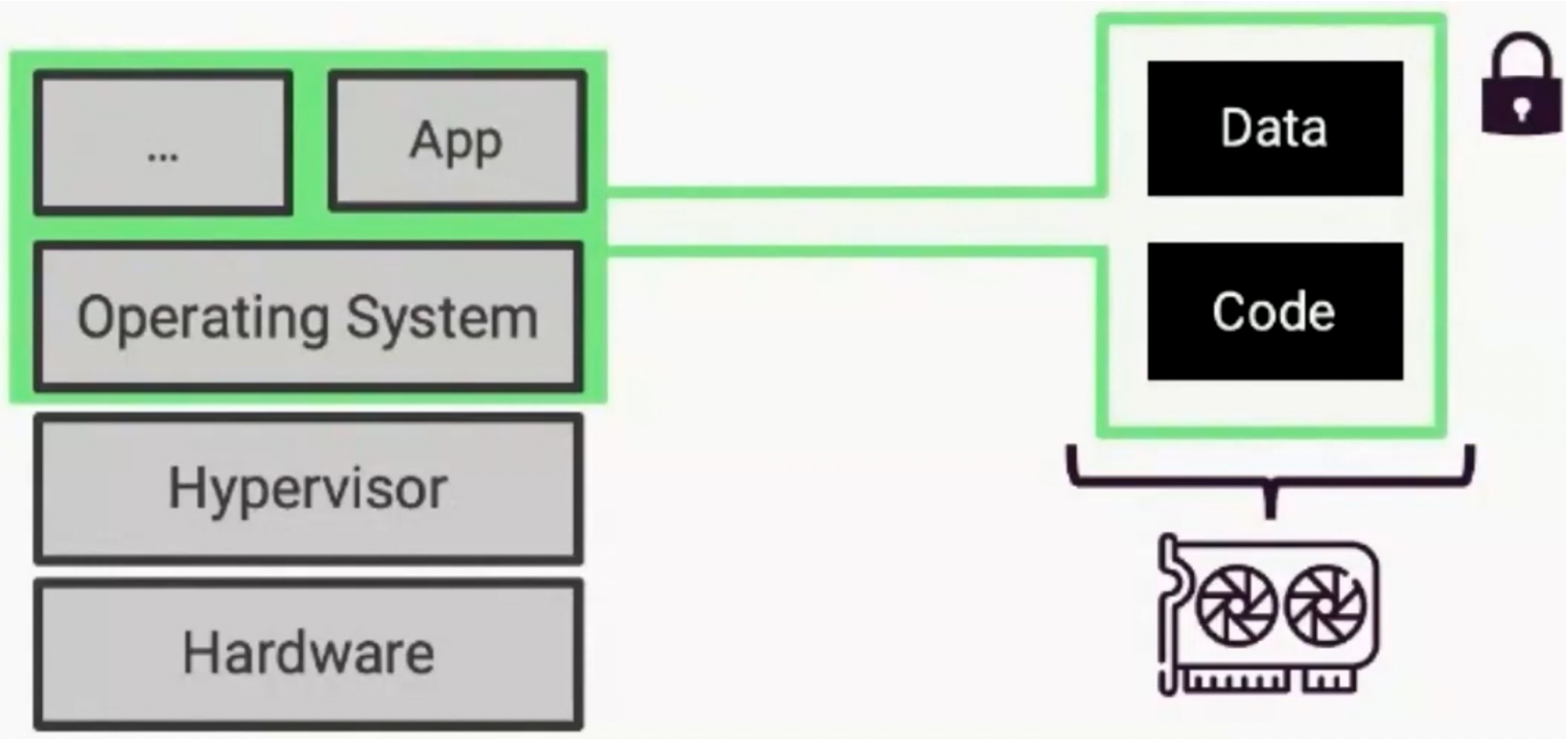
I/O Device TEE

- 66% overhead when running Deep Learning Recommendation Model (DLRM) on AMD SEV-SNP compared to non-secure environment



GPU TEE

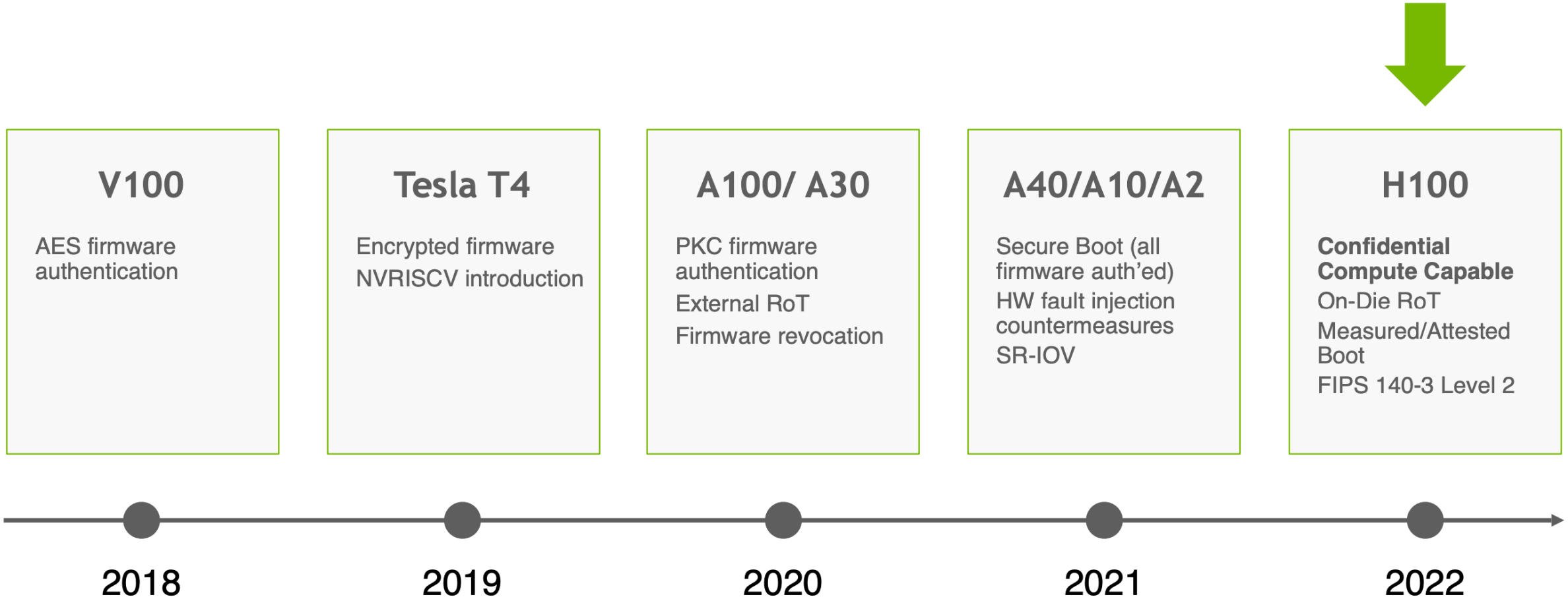
- All data and code in GPU TCB



GPU TEE Examples

Graviton	2018 OSDI	○ Build GPU TEE
Slalom	2019 ICLR	○ Offload linear layers to untrusted GPU via differential privacy
HIX	2019 ASPLOS	○ Extend enclave memory to GPU
DeepAttest	2019 ISCA	○ Design a device-specific fingerprint which is encoded in the weights of the DNN deployed on the target platform
Telekine	2020 NSDI	○ Address one of the side channel attacks
HETEE	2020 S&P	○ Separate FPGA for access control. ○ PCIe fabric is within TCB which is not the case in common situation (e.g, SGX)
Goten	2021 AAAI	○ Over Slalom, support training
Gramine + SGX	Report	○ Just interface implementation without protection
StrongBox	2022 CCS	○ Support ARM GPU for general computation ○ Extant Arm-based GPU defenses are intended for secure machine learning, and lack generality
Honeycomb	2023 OSDI	○ Provide a software-based GPU TEE by validating the offloaded GPU program, so there is no run-time overhead when executing GPU program ○ The method highly depends on the quality of validation software (SFI) ○ Why validating before execution works is that the offloaded GPU workload does not include many long divisions, nested branches and indirect memory references
SAGE	2023 ATC	○ Support software-based attestation ○ Protect code integrity and secrecy, computation integrity, as well as data integrity and secrecy

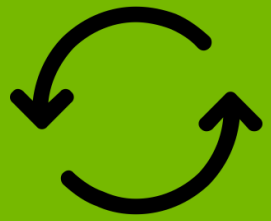
NVIDIA Roadmap to Confidential Computing



RoT = Root of Trust

NVIDIA Confidential Computing Goals

Protect data in use for accelerated computing



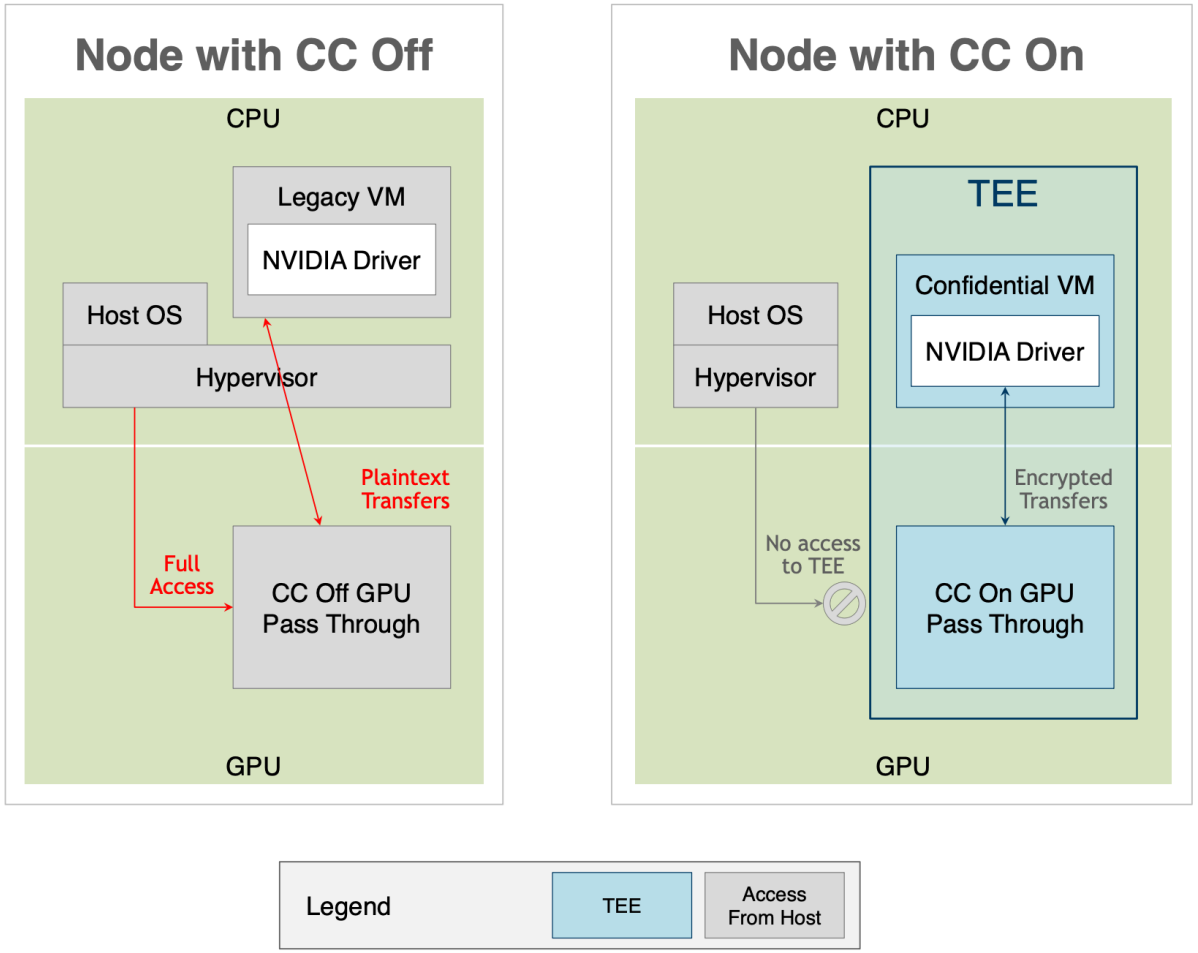
Run CUDA applications unchanged

Offer scale from multi-instance GPUs to multi-node

Threats and Mitigations of H100's CC Modes

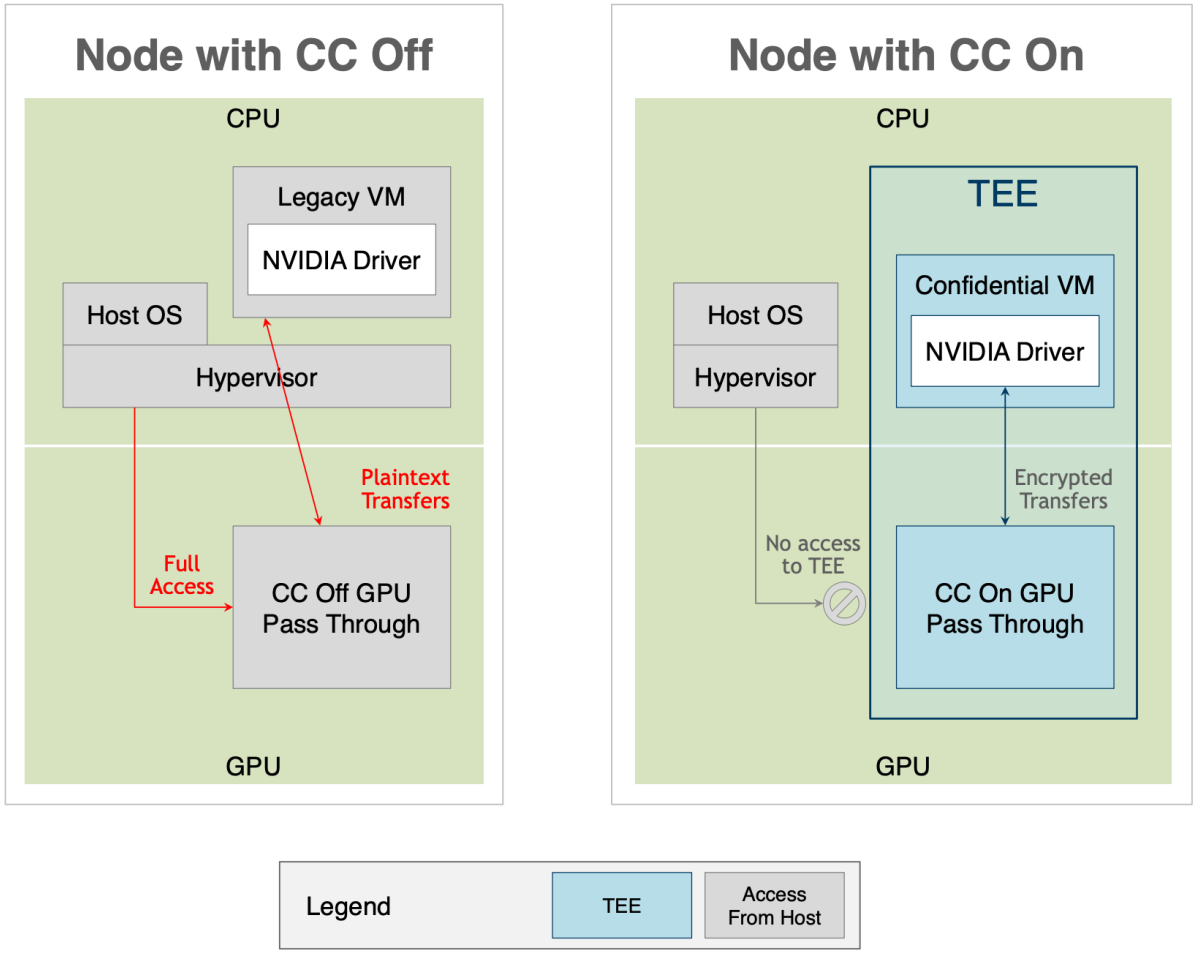
Category	Threat	Mitigation
Confidentiality	Use PCIE/NVLINK to read tenant data (e.g. Hypervisor, another VM, PCIE interposer)	✓
	Use Out-of-band management/debug channels to read tenant data (e.g. SMBus, JTAG)	✓
	Use memory remapping to read tenant data	✓
	Use GPU Cache/Memory based side channels to read tenant data	✓
	Use GPU TLB based side channels to read tenant data	✓
	Use GPU Performance Counters to read tenant data or fingerprint tenant	✓
	Read tenant data via hypothetical physical attacks (physical side channels / DPA / EM, HBM interposer)	✗
Integrity	Use PCIE/NVLINK to modify tenant data (e.g. Hypervisor, another VM, PCIE interposer)	✓
	Use Out-of-band management/debug channels to modify tenant data (e.g. SMBus, JTAG)	✓
	Corrupt tenant data by replaying previous data or MMIO transactions (replay attacks)	✓
	Corrupt tenant data via hypothetical physical attacks (fault injection, HBM interposer)	✗
Availability	Denial of Service to hypervisor by tenant	✓
	Denial of Service to tenant by another tenant	✓
	Permanent denial of service of GPU by tenant	✓
	Denial of Service to tenant by hypervisor	✗
General	Use a spoofed, non-genuine, or known vulnerable TCB component	✓
	Use hardware side channels (e.g. DPA) to extract persistent device keys	✓
	Use hardware side channels (e.g. DPA) to extract tenant ephemeral session key	✗

NVIDIA CC Introduction



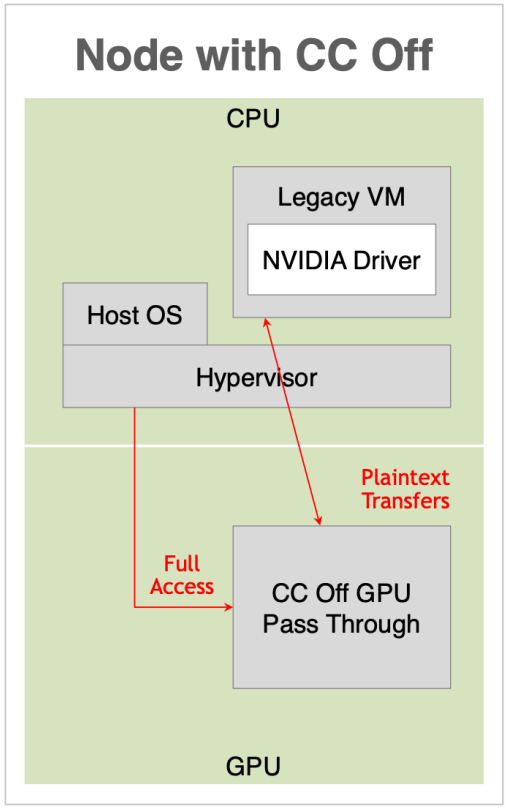
NVIDIA CC Introduction

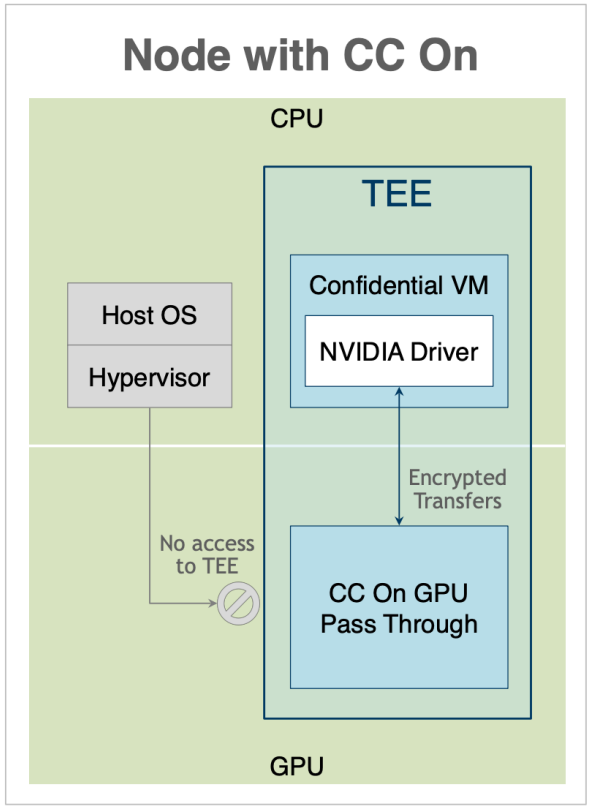
- Prerequisites:
  - CPU with support for a Virtualized-based TEE ("Confidential VM")
  - Supported variants are AMD Milan or later, or Intel SPR and later



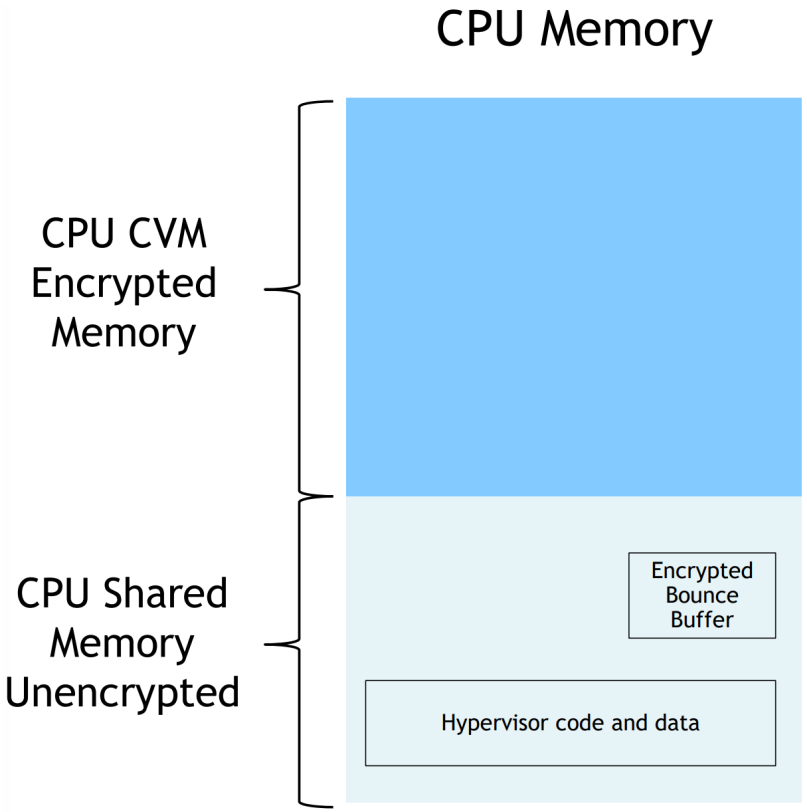
NVIDIA CC Introduction

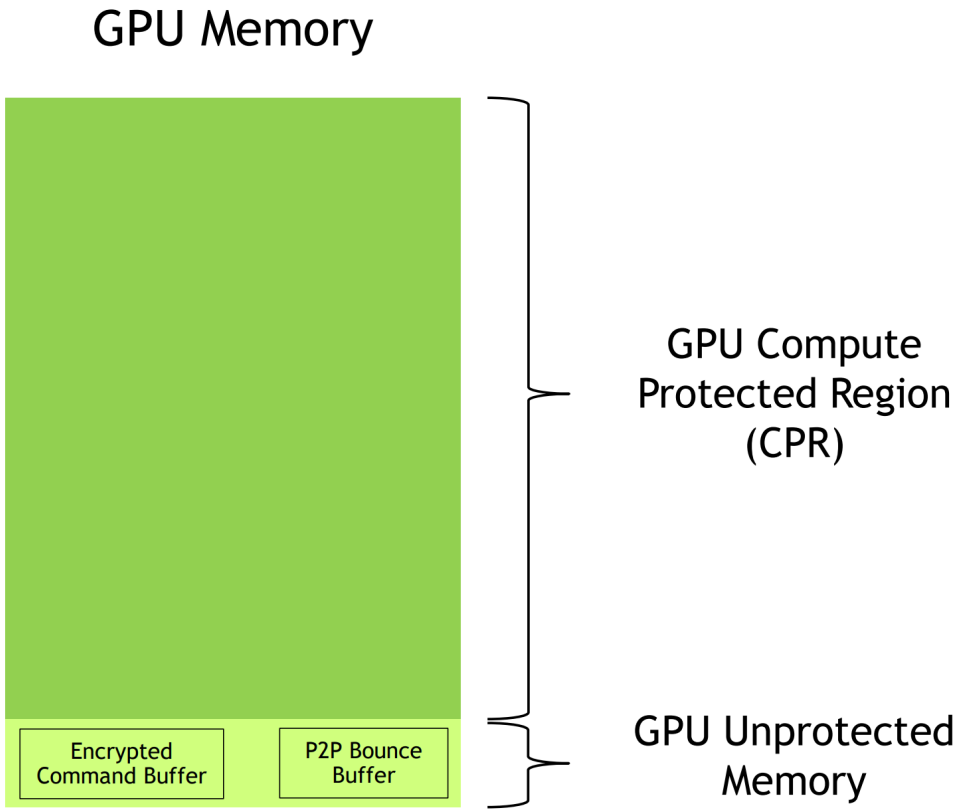
- Prerequisites:
  - CPU with support for a Virtualized-based TEE ("Confidential VM")
  - Supported variants are AMD Milan or later, or Intel SPR and later
- Capabilities:
  - Trusted Execution Environment
  - Virtualization-based
  - Secure Transfers
  - Hardware Root of Trust
    - Authenticated firmware; measurement & attestation for the GPU





How Memory is Managed in Confidential Mode

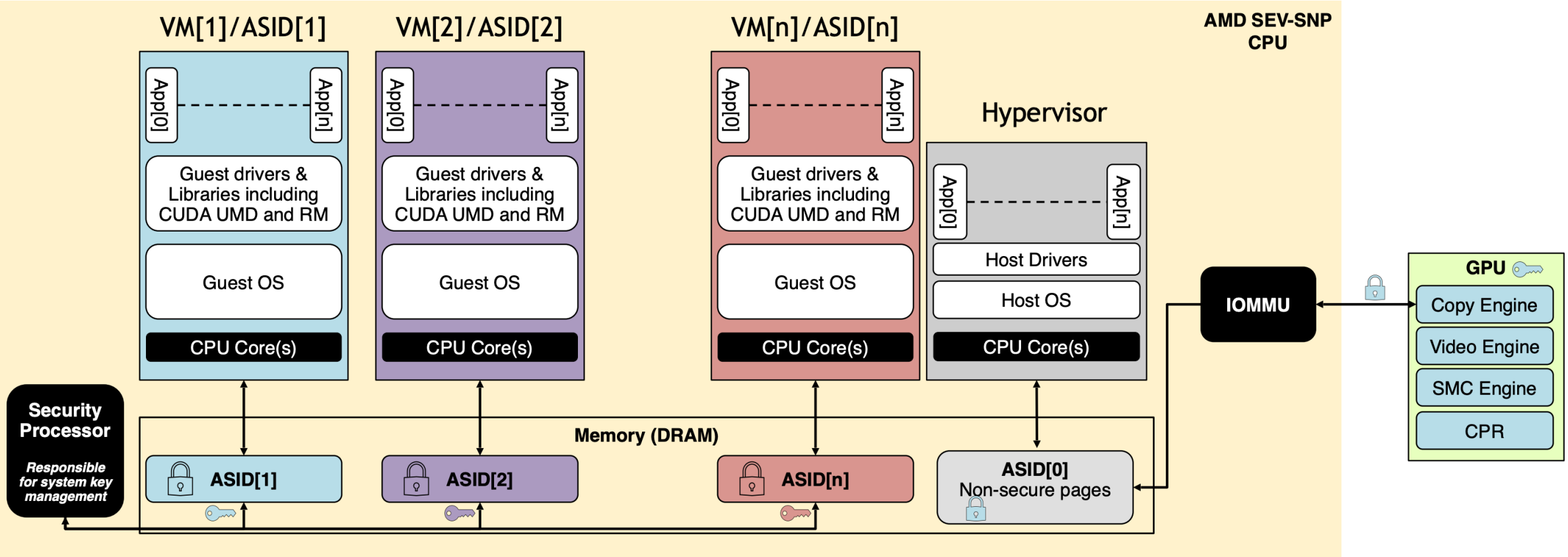




- CVM = Confidential Virtual Machine
- NVIDIA Driver allocates bounce buffers in the Shared Memory area and encrypts data in those buffers with the session key

- Compute Protected Region (CPR) is protected by hardware firewalls
- GPU memory outside of the CPR:
  - Encrypted CUDA Command Buffers
  - Encrypted Bounce Buffers for NVLINK Peer to Peer

H100 CC with AMD SEV-SNP



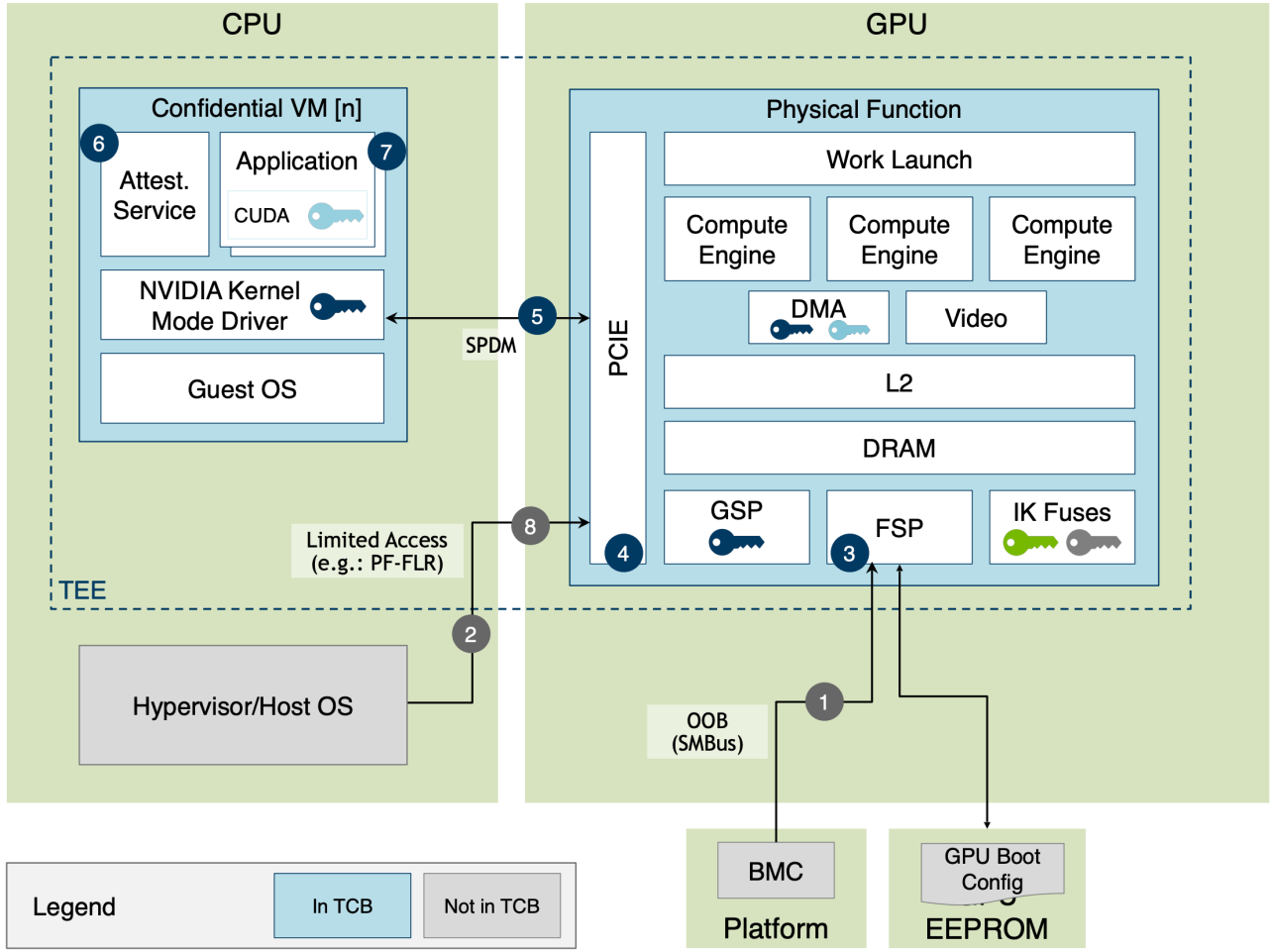
H100 CC on Mode Initialization Sequence

Mode Enable

Device Boot

Tenant Initialization

Tenant Shutdown



H100 CC on Mode Initialization Sequence

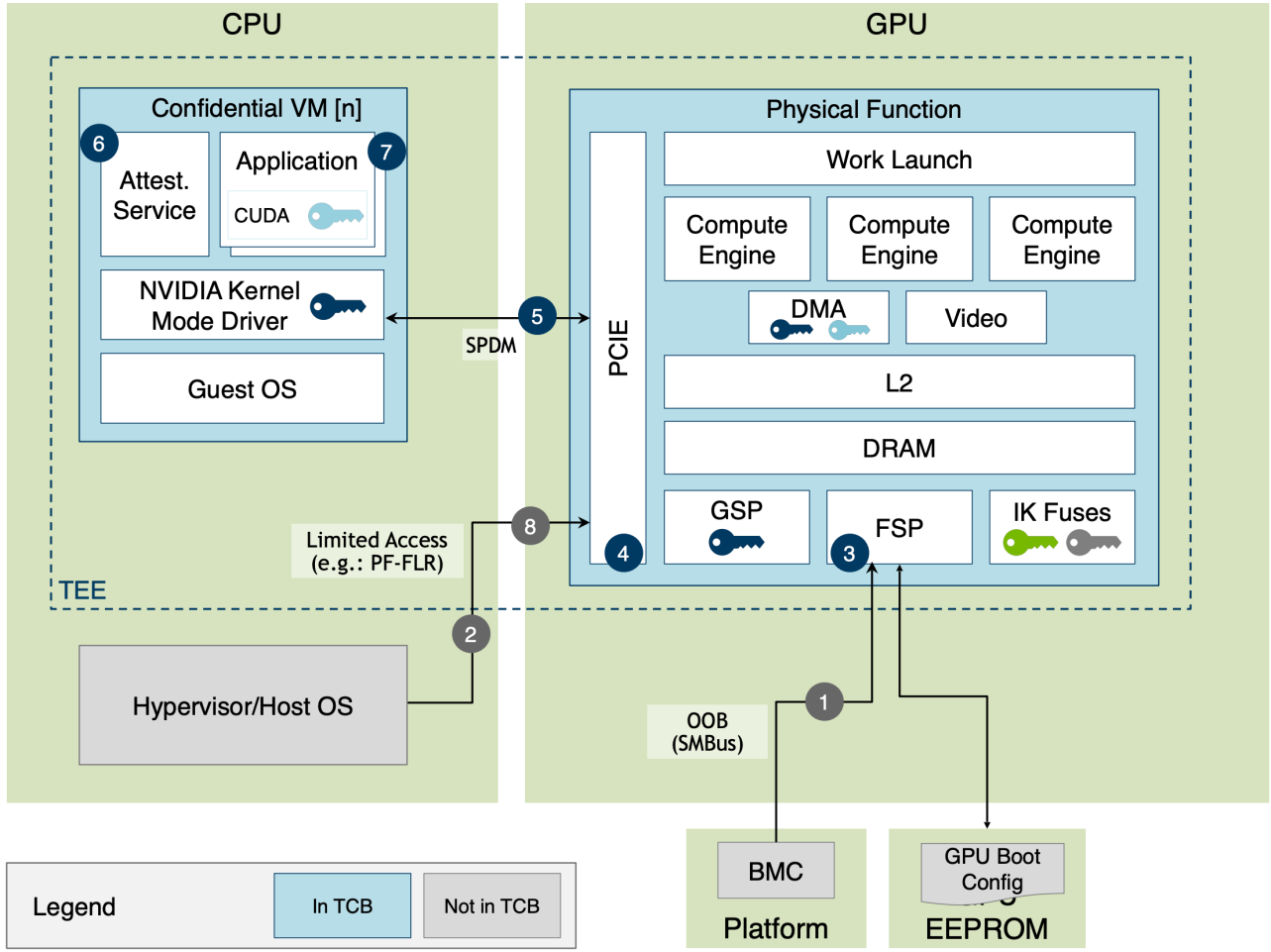
Mode Enable

1) BMC issues out-of-band request to persistently enable CC mode.
NVIDIA OOB Specification will provide APIs to integrate into customer tools and OpenBMC.

Device Boot

Tenant Initialization

Tenant Shutdown



H100 CC on Mode Initialization Sequence

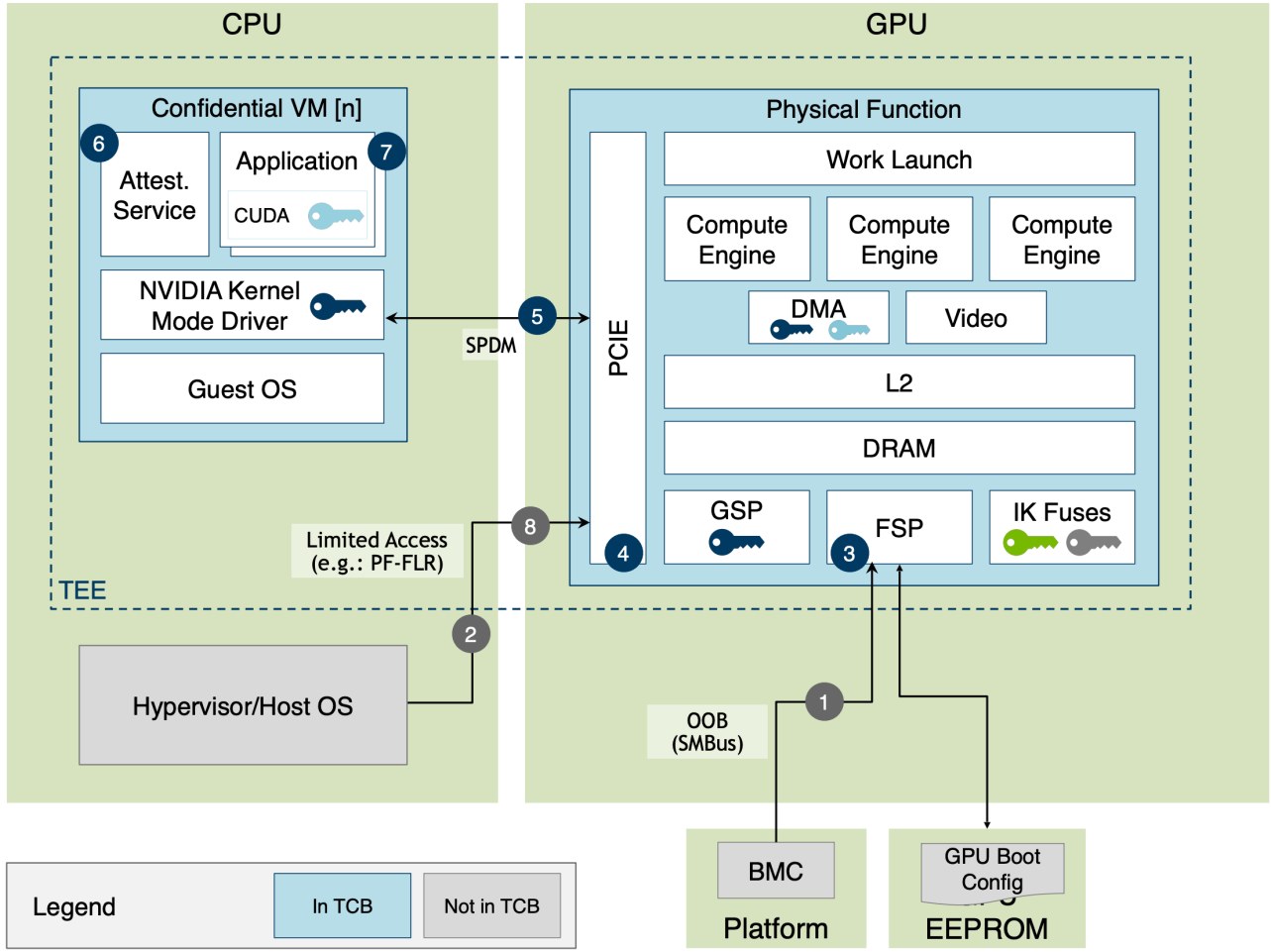
Mode Enable

1) BMC issues out-of-band request to persistently enable CC mode.
   NVIDIA OOB Specification will provide APIs to integrate into customer tools and OpenBMC.
2) Host triggers GPU reset for mode to take effect

Device Boot

Tenant Initialization

Tenant Shutdown



H100 CC on Mode Initialization Sequence

Mode Enable

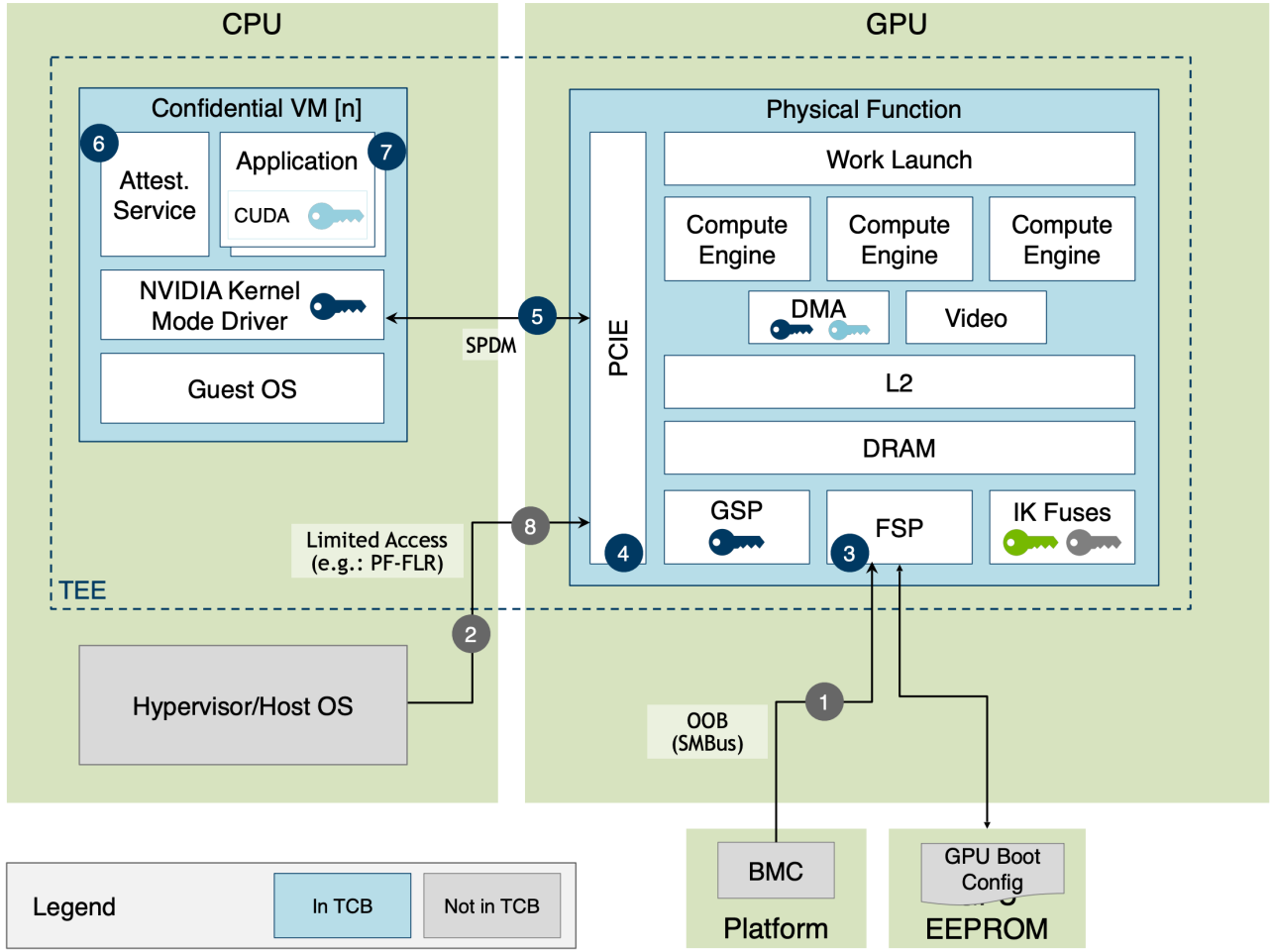
1) BMC issues out-of-band request to persistently enable CC mode.
   NVIDIA OOB Specification will provide APIs to integrate into customer tools and OpenBMC.
2) Host triggers GPU reset for mode to take effect

Device Boot

3) GPU firmware scrubs GPU state & memory

Tenant Initialization

Tenant Shutdown



H100 CC on Mode Initialization Sequence

Mode Enable

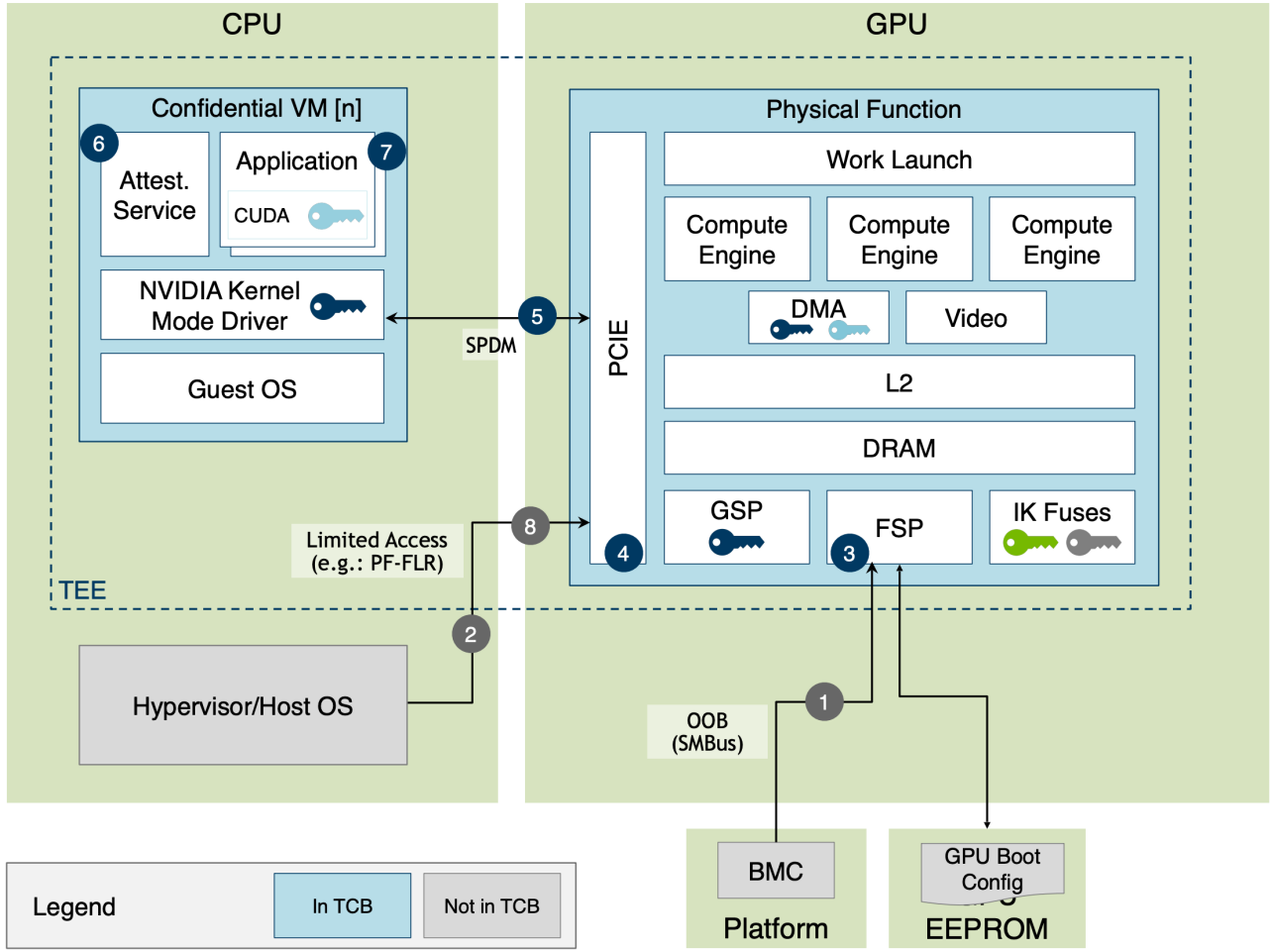
1) BMC issues out-of-band request to persistently enable CC mode.
   NVIDIA OOB Specification will provide APIs to integrate into customer tools and OpenBMC.
2) Host triggers GPU reset for mode to take effect

Device Boot

3) GPU firmware scrubs GPU state & memory
4) GPU firmware configures firewall to prevent unauthorized access, then enables PCIE

Tenant Initialization

Tenant Shutdown



H100 CC on Mode Initialization Sequence

Mode Enable

1) BMC issues out-of-band request to persistently enable CC mode.
   NVIDIA OOB Specification will provide APIs to integrate into customer tools and OpenBMC.
2) Host triggers GPU reset for mode to take effect

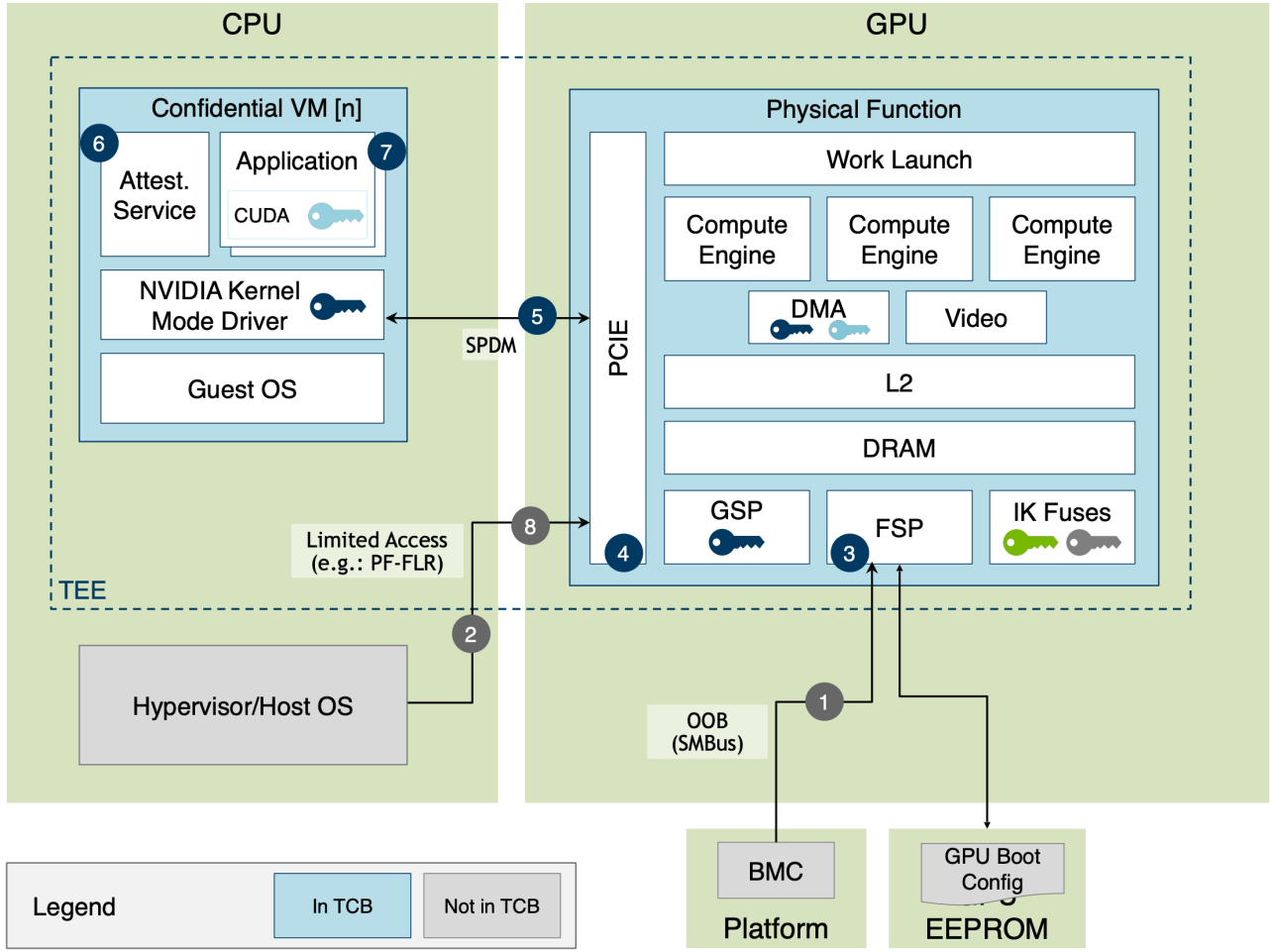
Device Boot

3) GPU firmware scrubs GPU state & memory
4) GPU firmware configures firewall to prevent unauthorized access, then enables PCIE

Tenant Initialization

5) GPU PF driver uses SPDM for session establishment & attestation report

Tenant Shutdown



H100 CC on Mode Initialization Sequence

Mode Enable

1) BMC issues out-of-band request to persistently enable CC mode.
   NVIDIA OOB Specification will provide APIs to integrate into customer tools and OpenBMC.
2) Host triggers GPU reset for mode to take effect

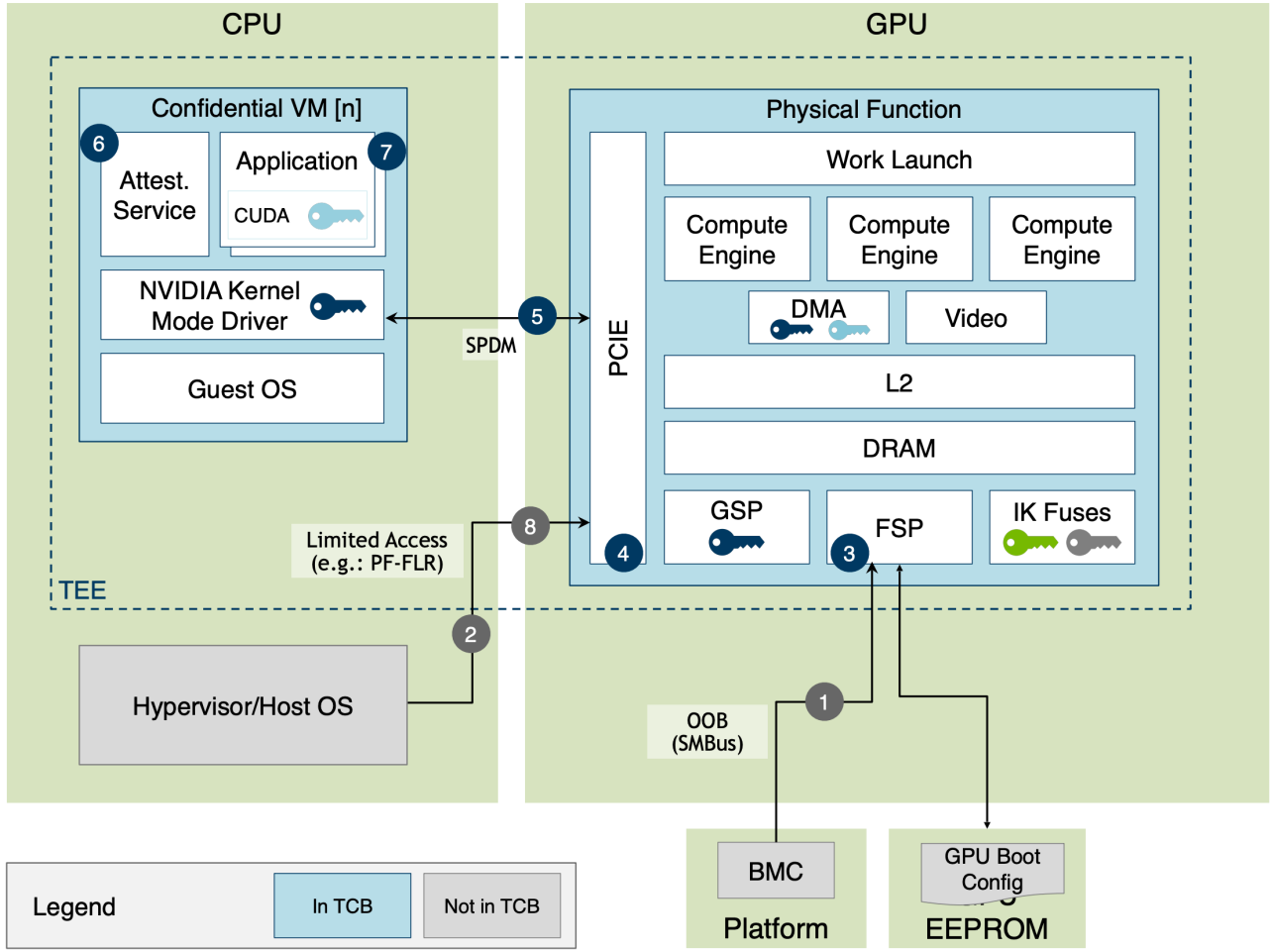
Device Boot

3) GPU firmware scrubs GPU state & memory
4) GPU firmware configures firewall to prevent unauthorized access, then enables PCIE

Tenant Initialization

5) GPU PF driver uses SPDM for session establishment & attestation report
6) Tenant attestation service gathers measurements, device certificate using NVML APIs.
   Verification done locally or transmitted to remote service

Tenant Shutdown



H100 CC on Mode Initialization Sequence

Mode Enable

1) BMC issues out-of-band request to persistently enable CC mode.
   NVIDIA OOB Specification will provide APIs to integrate into customer tools and OpenBMC.
2) Host triggers GPU reset for mode to take effect

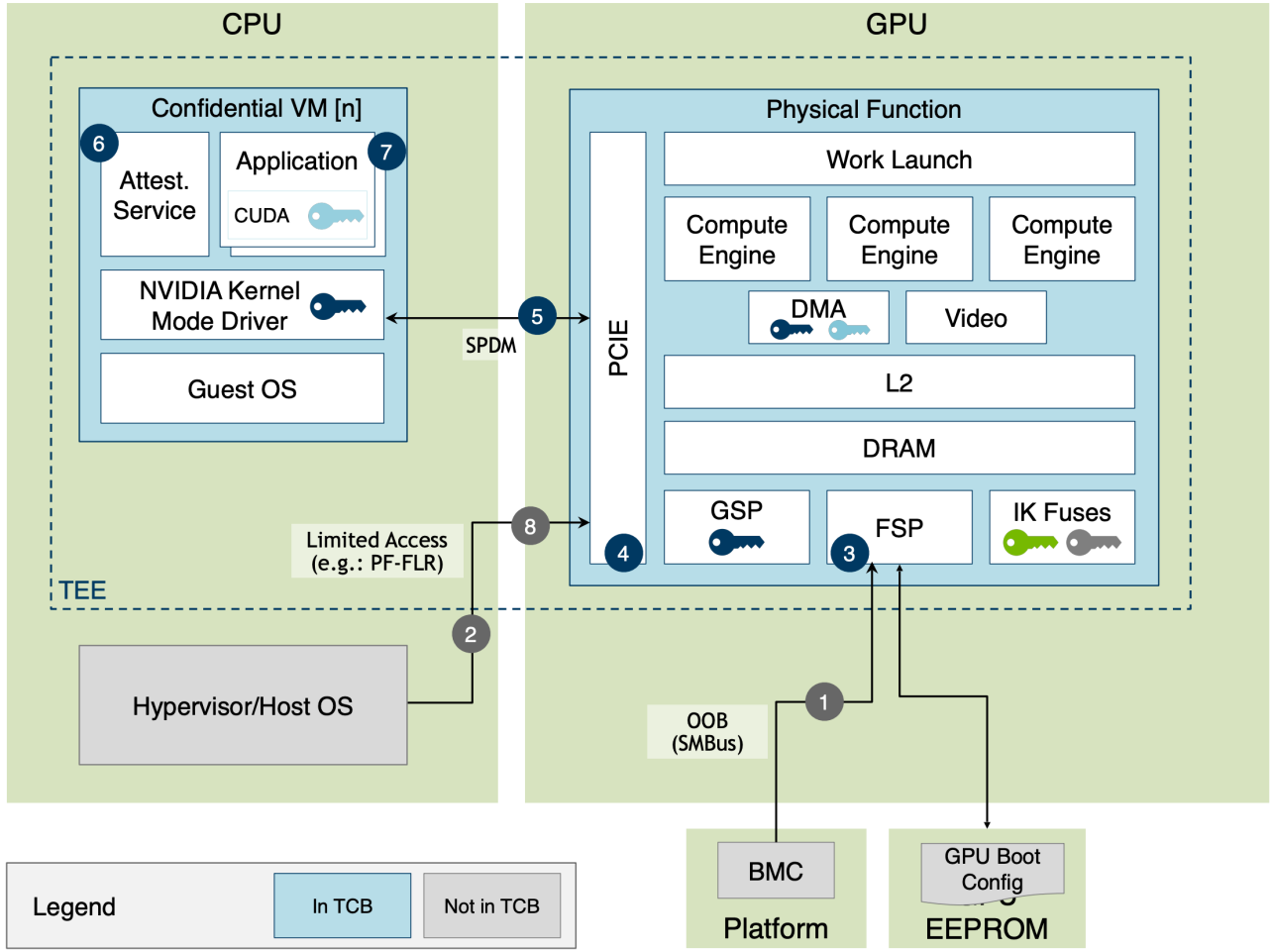
Device Boot

3) GPU firmware scrubs GPU state & memory
4) GPU firmware configures firewall to prevent unauthorized access, then enables PCIE

Tenant Initialization

5) GPU PF driver uses SPDM for session establishment & attestation report
6) Tenant attestation service gathers measurements, device certificate using NVML APIs.
   Verification done locally or transmitted to remote service
7) CUDA programs allowed to use GPU

Tenant Shutdown



H100 CC on Mode Initialization Sequence

Mode Enable

1) BMC issues out-of-band request to persistently enable CC mode.
   NVIDIA OOB Specification will provide APIs to integrate into customer tools and OpenBMC.
2) Host triggers GPU reset for mode to take effect

Device Boot

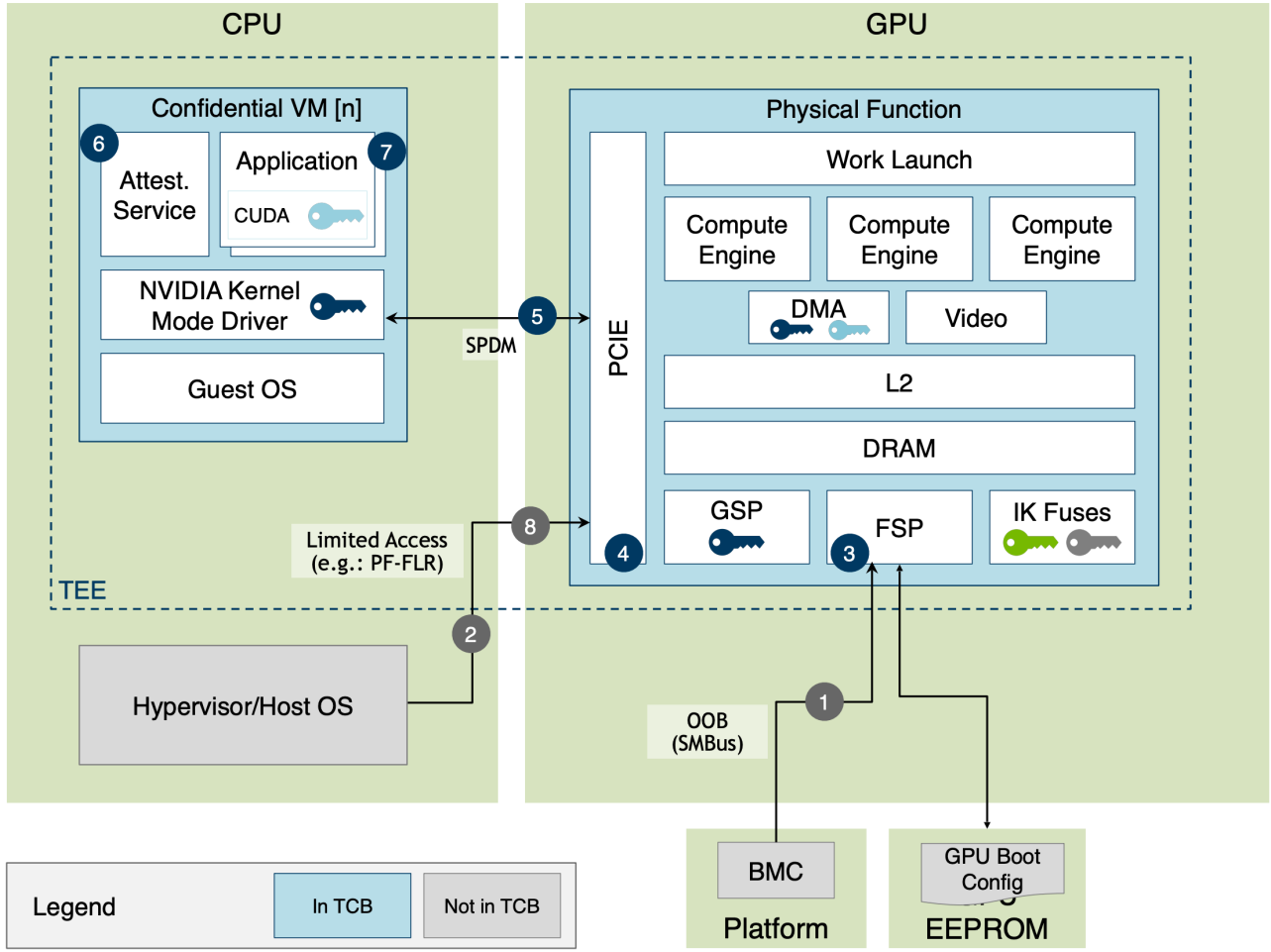
3) GPU firmware scrubs GPU state & memory
4) GPU firmware configures firewall to prevent unauthorized access, then enables PCIE

Tenant Initialization

5) GPU PF driver uses SPDM for session establishment & attestation report
6) Tenant attestation service gathers measurements, device certificate using NVML APIs.
   Verification done locally or transmitted to remote service
7) CUDA programs allowed to use GPU

Tenant Shutdown

8) Host triggers PF-FLR to reset GPU; returns 3) device boot for scrubbing GPU state & memory



H100 CC on Mode Initialization Sequence

Mode Enable

1) BMC issues out-of-band request to persistently enable CC mode.
   NVIDIA OOB Specification will provide APIs to integrate into customer tools and OpenBMC.
2) Host triggers GPU reset for mode to take effect

Device Boot

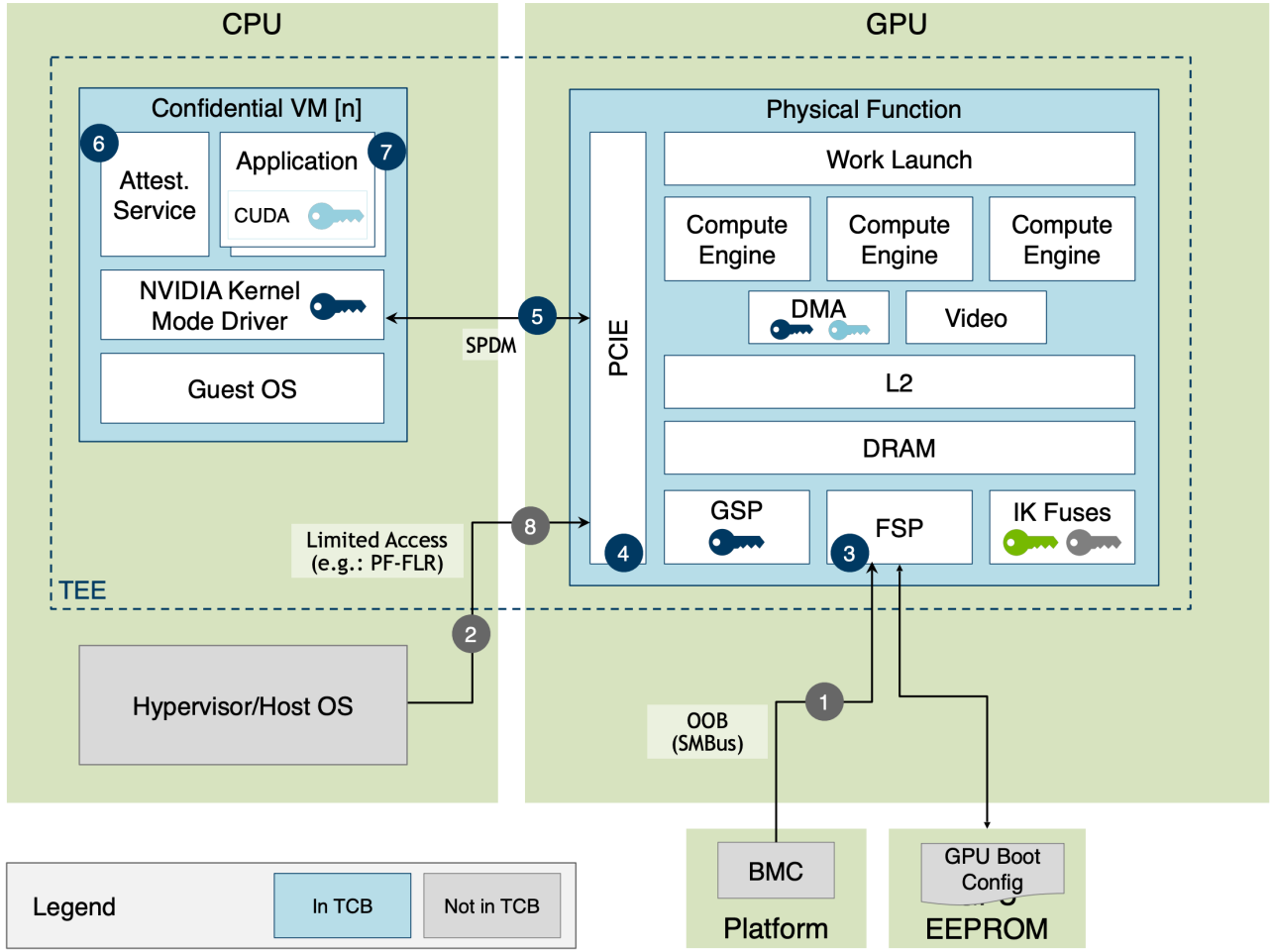
3) GPU firmware scrubs GPU state & memory
4) GPU firmware configures firewall to prevent unauthorized access, then enables PCIE

Tenant Initialization

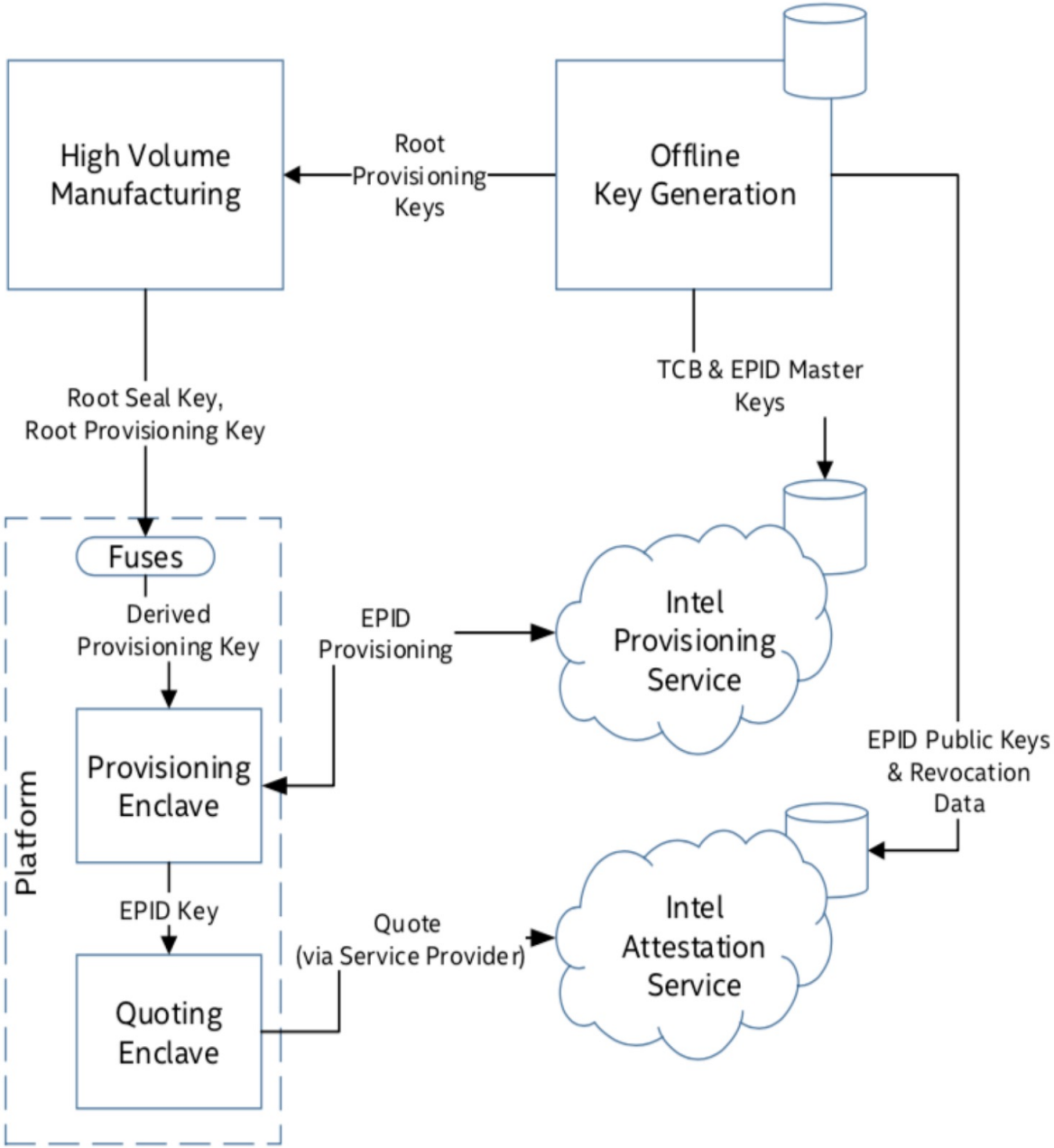
5) GPU PF driver uses SPDM for session establishment & attestation report
6) **Tenant attestation service gathers measurements, device certificate using NVML APIs. Verification done locally or transmitted to remote service**
7) CUDA programs allowed to use GPU

Tenant Shutdown

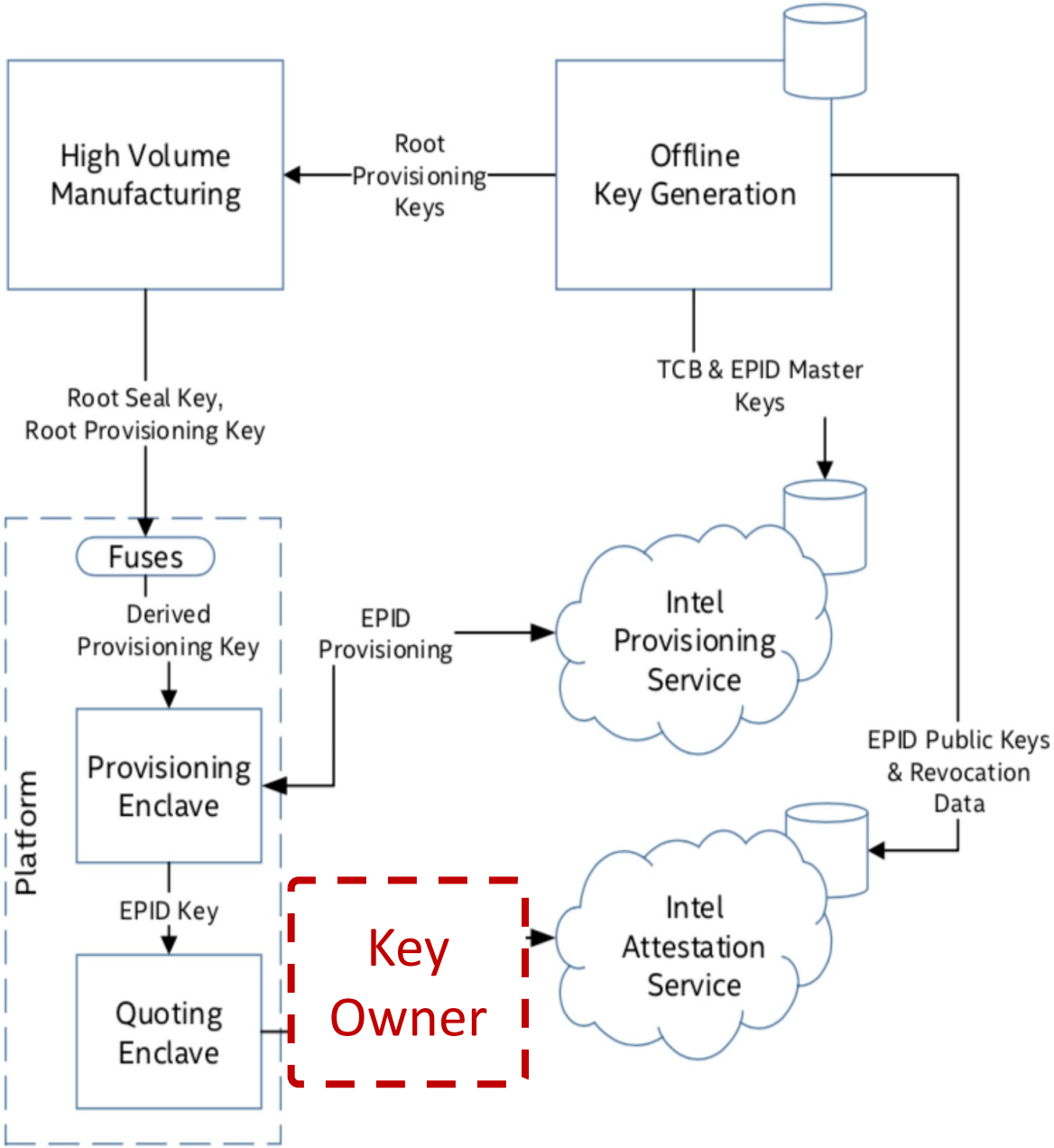
8) Host triggers PF-FLR to reset GPU; returns 3) device boot for scrubbing GPU state & memory



Where Should Users be?



Where Should Users be?



Cloud Native Computing Foundation: CoCo

Cloud Native Computing Foundation: CoCo

CONFIDENTIAL
CONTAINERS

Cloud Native Computing Foundation: CoCo

CONFIDENTIAL CONTAINERS

- What are Confidential Containers?

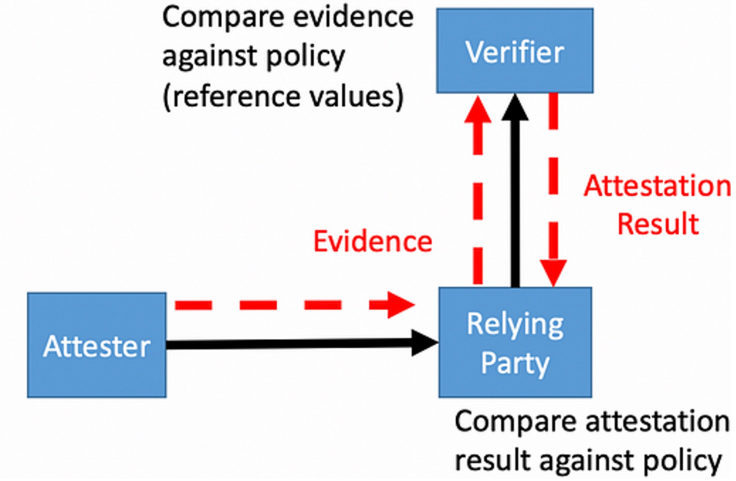
Cloud Native Computing Foundation: CoCo

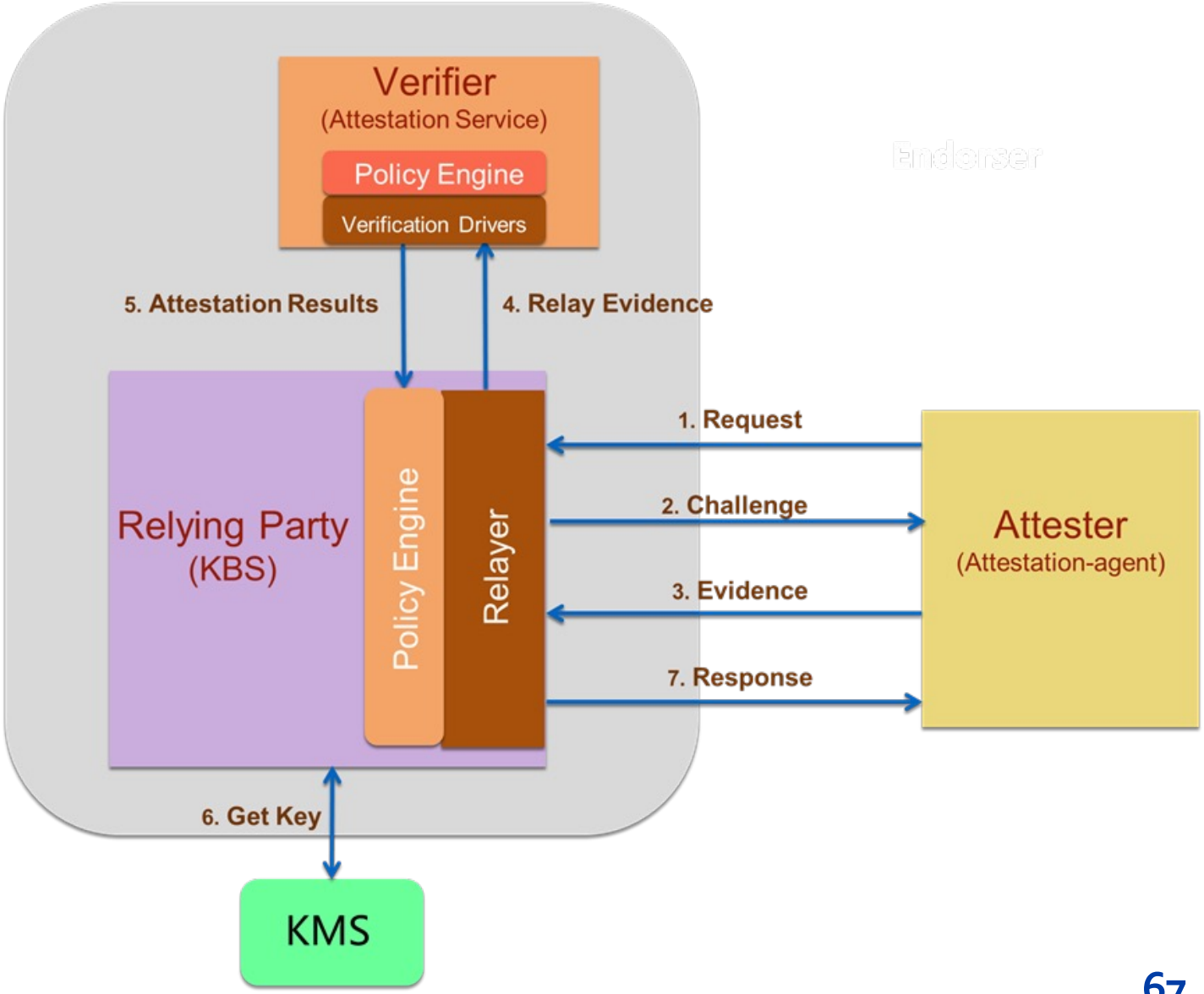
CONFIDENTIAL CONTAINERS

- **What are Confidential Containers?**
- **Confidential Containers** (CoCo) is a new sandbox project of the Cloud Native Computing Foundation (CNCF) that enables cloud-native confidential computing by taking advantage of a variety of hardware platforms and technologies. The project brings together software and hardware companies including Alibaba-cloud, AMD, ARM, IBM, Intel, Microsoft, Red Hat, Rivos and others.

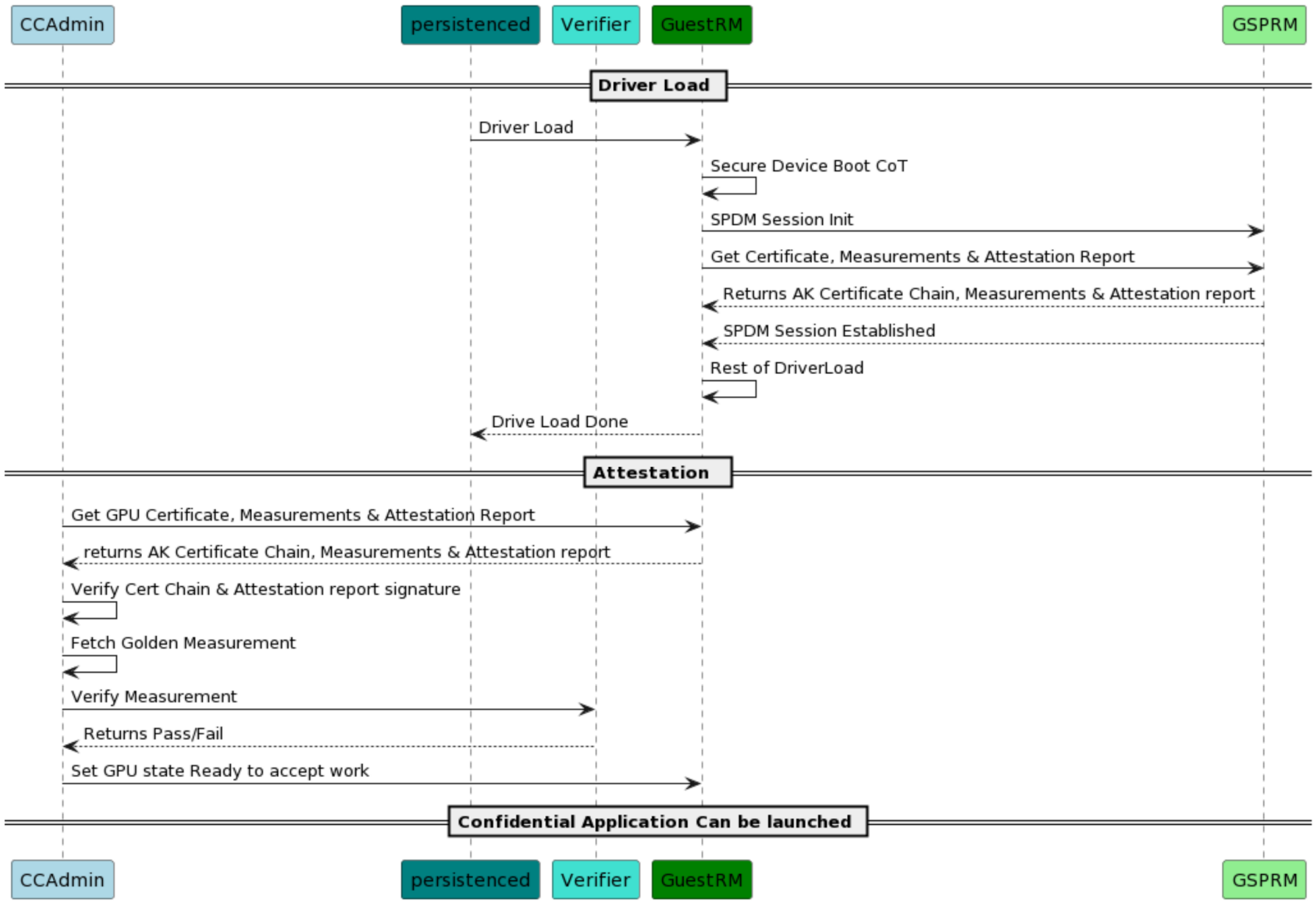
High Level of CoCo Key Broker Service (CoCo-KBS)

Background check" model:





H100 Tenant Attestation



GPU Attestation in CoCo-KBS

GPU Attestation in CoCo-KBS

- CoCo-KBS (Rust-based)

GPU Attestation in CoCo-KBS

- CoCo-KBS (Rust-based)
- nvTrust (Python-based)

GPU Attestation in CoCo-KBS

- CoCo-KBS (Rust-based)
- nvTrust (Python-based)
- POC sample code:

```
pyo3::prepare_freethreaded_python();

let gil = Python::acquire_gil();
let py = gil.python();

// Create a global dictionary containing __file__
let globals = [("__file__", "./LocalGPUTest.py")]
    .into_py_dict(py);

// Read the content of the Python script
let code = fs::read_to_string("./LocalGPUTest.py")
    .expect("Could not read file");

// Execute the Python script
py.run(&code, Some(globals), None)?;

Ok(())
```

GPU Attestation in CoCo-KBS

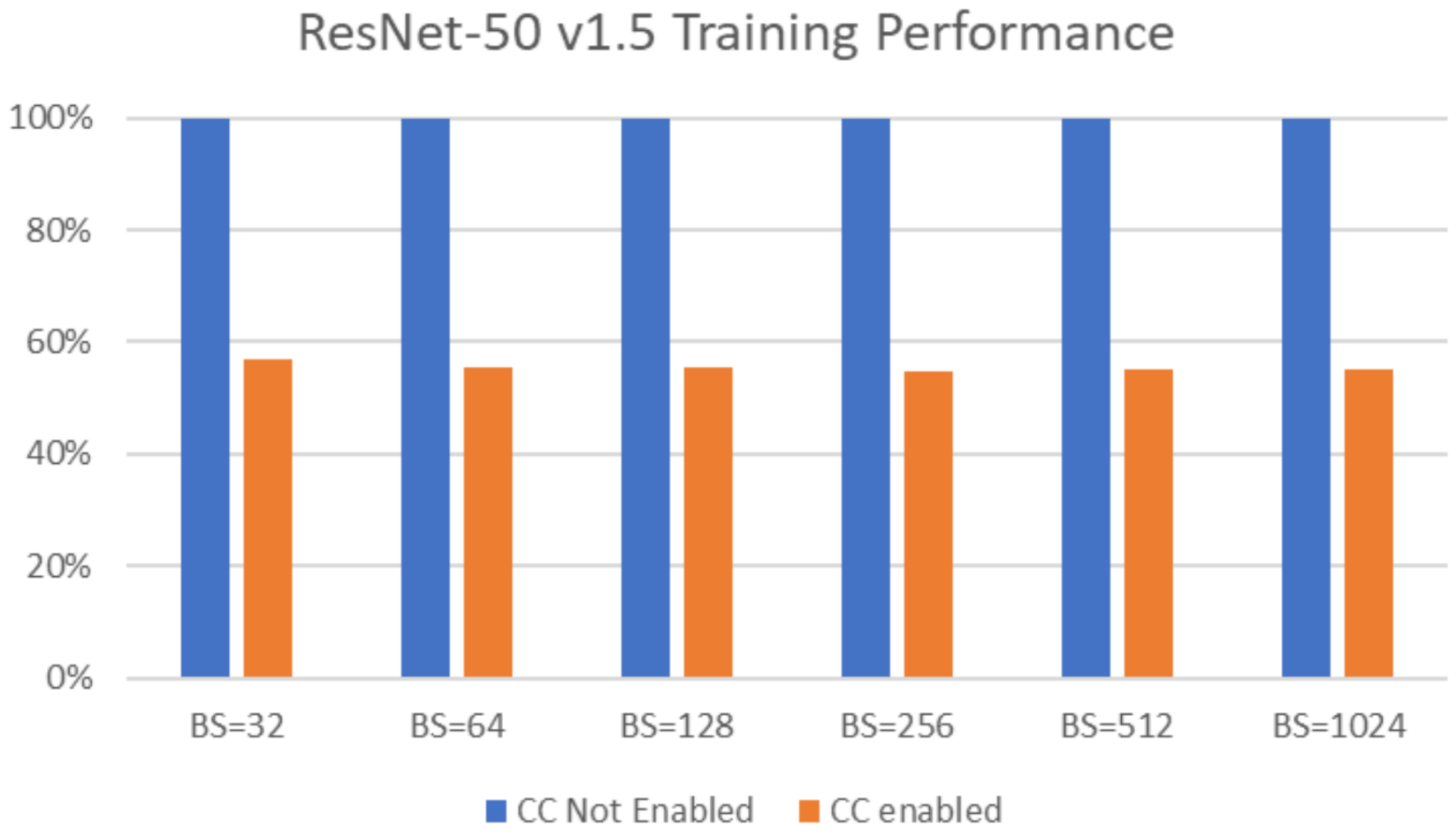
- CoCo-KBS (Rust-based)
- nvTrust (Python-based)
- POC sample code:
- **Soon to release!**

```
pyo3::prepare_freethreaded_python();

let gil = Python::acquire_gil();
let py = gil.python();

// Create a global dictionary containing __file__
let globals = [("__file__", "./LocalGPUTest.py")]
    .into_py_dict(py);

// Read the content of the Python script
let code = fs::read_to_string("./LocalGPUTest.py")
    .expect("Could not read file");

// Execute the Python script
py.run(&code, Some(globals), None)?;

Ok(())
```

Performance of H100 CC

Performance of H100 CC

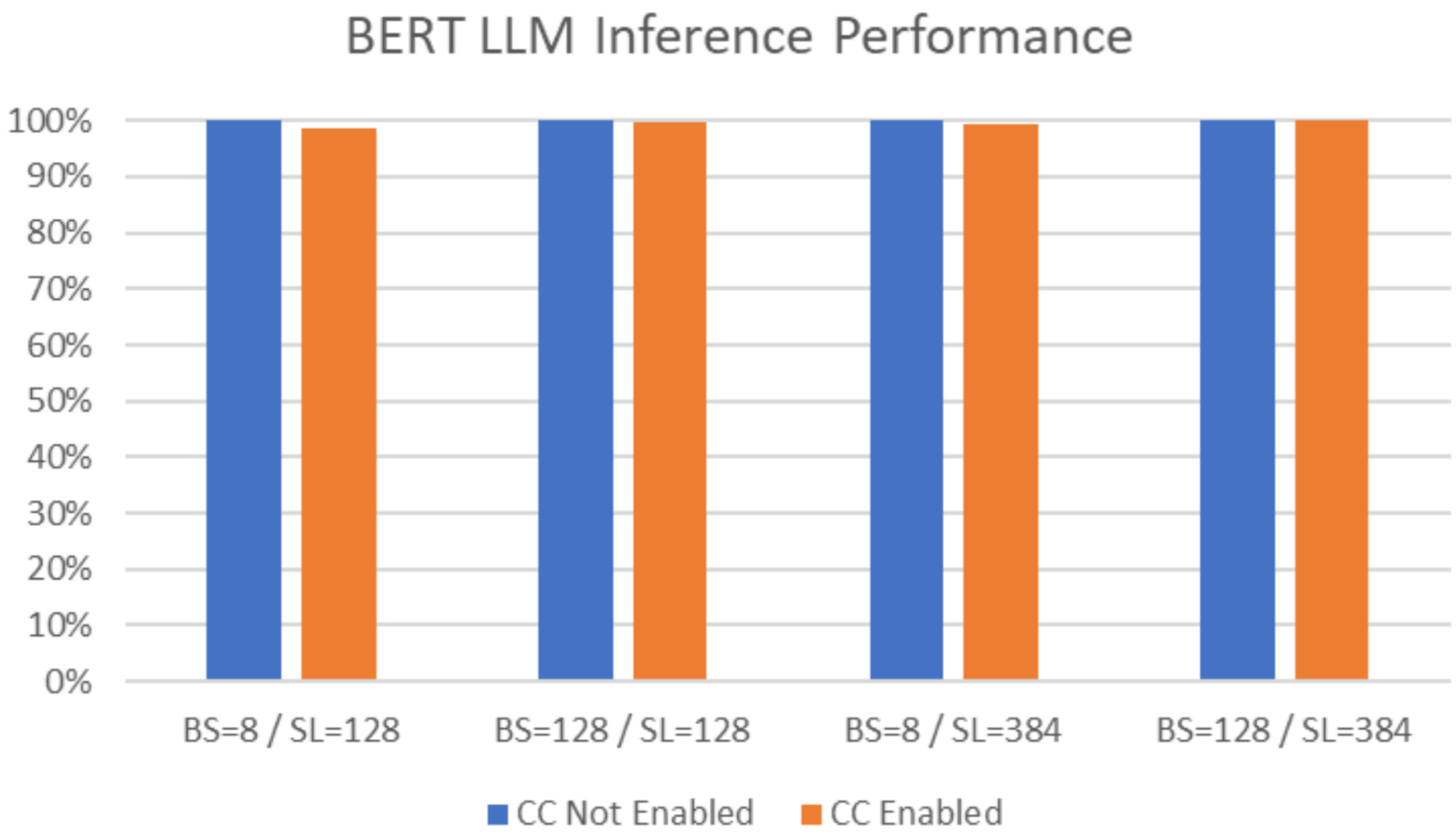
- Example of a Workload with a Low Compute to I/O Ratio
  - BS is the batch size



Performance of H100 CC

Performance of H100 CC

- Example of a Workload with High Compute to I/O Ratio
  - BS is the batch size, and SL is the sequence length



Summary

- How does typical Confidential Computing (Intel SGX) works
- Design tradeoffs between TCB size, flexibility, perf overhead, cost, etc.
  - Intel SGX, AMD SEV, ARM CCA
  - Keystone, Sanctum, Penglai, etc.

Summary

- How does typical Confidential Computing (Intel SGX) works
- Design tradeoffs between TCB size, flexibility, perf overhead, cost, etc.
  - Intel SGX, AMD SEV, ARM CCA
  - Keystone, Sanctum, Penglai, etc.
- What is GPU TEE

Summary

- How does typical Confidential Computing (Intel SGX) works
- Design tradeoffs between TCB size, flexibility, perf overhead, cost, etc.
  - Intel SGX, AMD SEV, ARM CCA
  - Keystone, Sanctum, Penglai, etc.
- What is GPU TEE
- How does NVIDIA H100 Confidential Computing work

Summary

- How does typical Confidential Computing (Intel SGX) works
- Design tradeoffs between TCB size, flexibility, perf overhead, cost, etc.
  - Intel SGX, AMD SEV, ARM CCA
  - Keystone, Sanctum, Penglai, etc.
- What is GPU TEE
- How does NVIDIA H100 Confidential Computing work
- How do we apply H100 CC to open-source project CoCo

Summary

- How does typical Confidential Computing (Intel SGX) works
- Design tradeoffs between TCB size, flexibility, perf overhead, cost, etc.
  - Intel SGX, AMD SEV, ARM CCA
  - Keystone, Sanctum, Penglai, etc.
- What is GPU TEE
- How does NVIDIA H100 Confidential Computing work
- How do we apply H100 CC to open-source project CoCo
- Performance of H100 CC

Q&A

Backup

GPU Attestation in CoCo-KBS

```python
from nv_attestation_sdk import attestation
import os
import json

client = attestation.Attestation()
client.set_name("thisNode1")
print ("[LocalGPUTest] node name :", client.get_name())
file = "NVGPULocalPolicyExample.json"

client.add_verifier(attestation.Devices.GPU, attestation.Environment.LOCAL, "", "")
with open(os.path.join(os.path.dirname(__file__), file)) as json_file:
    json_data = json.load(json_file)
    att_result_policy = json.dumps(json_data)

print(client.get_verifiers())

print ("[LocalGPUTest] call attest() - expecting True")
print(client.attest())

print ("[LocalGPUTest] token : "+str(client.get_token()))

print ("[LocalGPUTest] call validate_token() - expecting True")
print(client.validate_token(att_result_policy))
```

	HW TEE	Homomorphic Encryption	TPM
Data integrity	Y	Y (subject to code integrity)	Keys only
Data confidentiality	Y	Y	Keys only
Code integrity	Y	No	Y
Code confidentiality	Y (may require work)	No	Y
Authenticated Launch	Varies	No	No
Programmability	Y	Partial (“circuits”)	No
Attestability	Y	No	Y
Recoverability	Y	No	Y

	Native	HW Tee	Homomorphic Encryption
Data size limits	High	Medium	Low
Computation Speed	High	High-Medium	Low
Scale out across machines	Yes	More work	Yes
Ability to combine data across sets (MPC)	Yes	Yes	Very limited