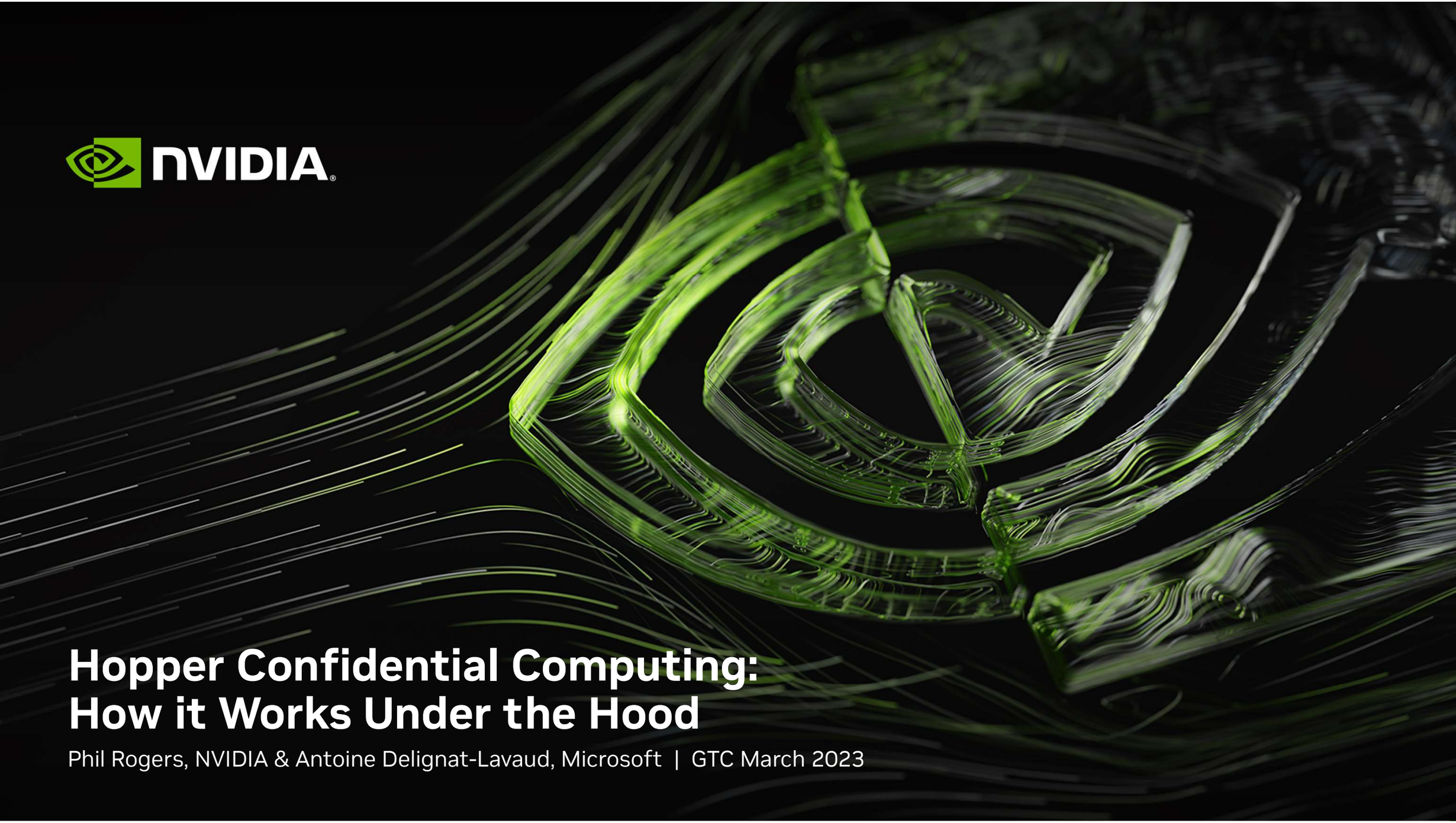


Hopper Confidential Computing: How it Works Under the Hood

Phil Rogers, NVIDIA & Antoine Delignat-Lavaud, Microsoft | GTC March 2023

Agenda

- Introduction to Confidential Computing
- How Your CUDA Application Runs in Confidential Mode
- GPU Features in Hopper for Confidential Computing
- Confidential Mode: Enable, Initialize, Teardown
- Attesting the GPU – Local Attestation
- Enabling Confidential GPUs on Encrypted and Integrity Protected Confidential VMs
- Full Attestation of the Trusted Hardware Platform (CPU+GPU)
- How Azure Exposes the GPU to the Guest VM while Protected from the Host.

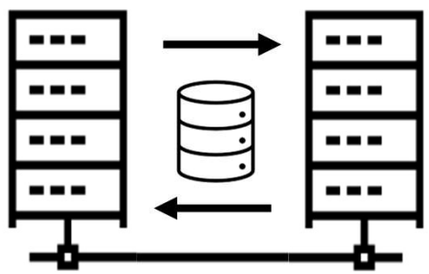
Traditional Computing Systems

Application Code and Data Not Protected in Use

Data at Rest
(In Storage)

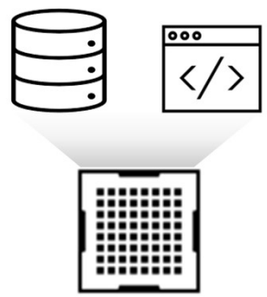
SECURE

Data in Transit
(Across a Network)



SECURE

Data in Use
(Being Processed)



NOT SECURE

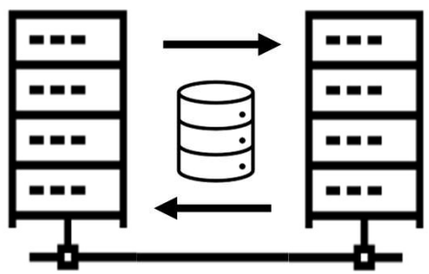
Confidential Computing

End to End Security – Even When Processing on Computers You Do Not Own

Data at Rest
(In Storage)

SECURE

Data in Transit
(Across a Network)



SECURE

Confidential Computing

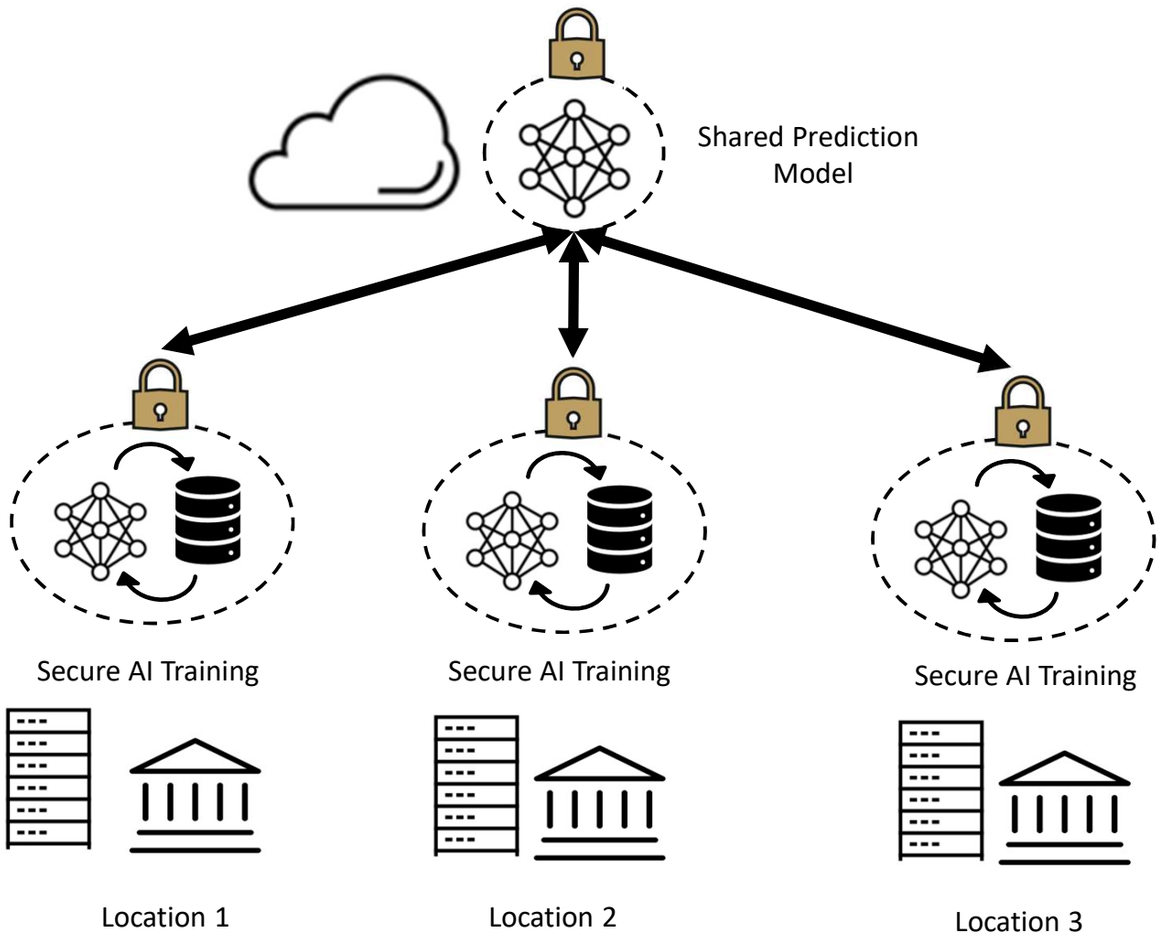
Multiple benefits from this model:

1. User is protected against the platform
2. Platform operator can guarantee to user that they can never see the data or code
3. ISV can prove to users that user data is confidential
4. ISV can craft their app to only run if confidential computing is enabled

NVIDIA

Secure Multi Party Computing

Example – Confidential Federated Learning



Confidentiality and integrity benefit at the Hub Server

Integrity benefit at the Training Spokes ensuring that code and data are not tampered, and the expected data is used

A consensus of trust occurs since all participants have the attestation reports of the others

Confidential Computing Approaches

Starting From the CPU

Process based

- Such as Intel SGX
- Security is strong and integrity is protected
- Application memory divided into trusted and untrusted
- Applications must be re-written to decide what data to protect or use middleware such as Open Enclave to manage the encrypted memory
- Application cannot make any system calls to the operating system – including to drivers

Virtual Machine based

- Such as AMD SEV-SNP, Intel TDX, Arm Realm
- Security is strong and integrity is protected
- All the Virtual Machine memory is encrypted
- Applications need no changes and are unaware of Confidential Computing (CC)
- NVIDIA driver can operate inside of the CPU TEE

Confidential Computing Approaches

Now Adding the GPU

- For GPU Confidential Computing we have implemented the Virtual Machine based approach
- Brings GPU level of performance to Confidential Computing.

Virtual Machine based

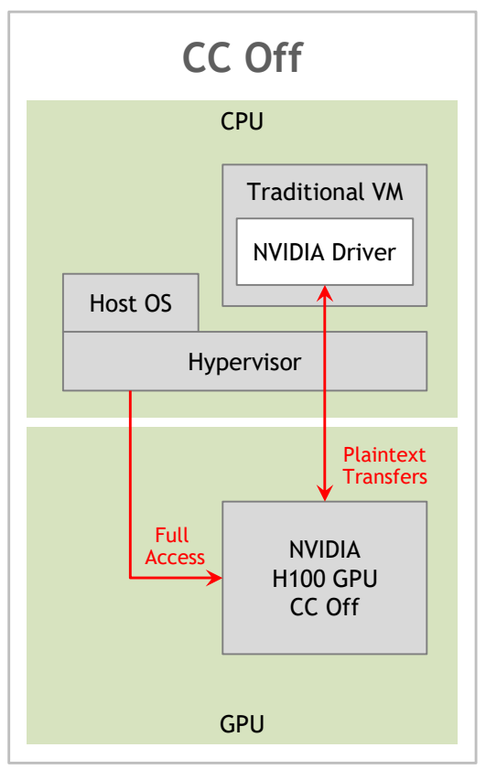
- Such as AMD SEV-SNP, Intel TDX, Arm Realm
- Security is strong and integrity is protected
- All the Virtual Machine memory is encrypted
- Applications need no changes and are unaware of CC
- NVIDIA driver can operate inside of the CPU TEE

Confidential Computing on the GPU

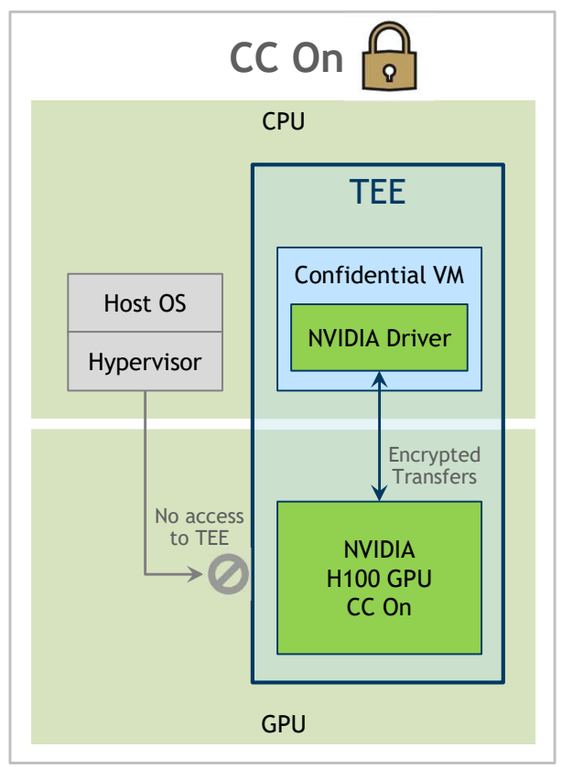
- Virtualization based
- Security is strong and integrity is protected
- CUDA applications can run unchanged
- NVIDIA H100 is the first Confidential GPU

GPU With VM Based Confidential Computing

Without confidential computing enabled, the hypervisor has full access to all of system memory and all of GPU memory



With confidential computing guarantees, the hypervisor is blocked from accessing the Confidential VM in system memory and blocked from reading GPU memory



Confidential Computing Protects VMs From Each Other

Can Mix Confidential and Non-Confidential VMs on the Same Server

Each GPU is passed through to just one VM

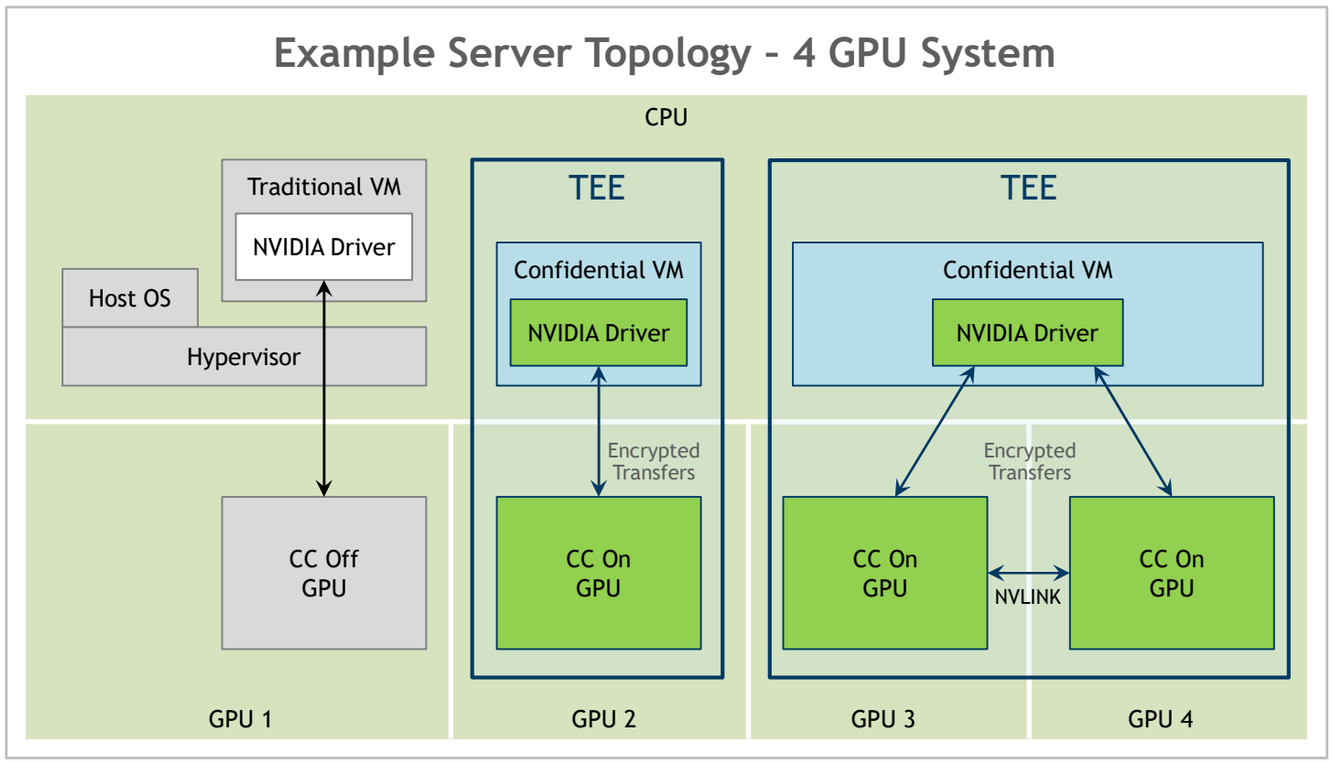
- VM may have multiple GPUs

VM-based TEE protects the entire workload

- NVIDIA drivers, libraries, APIs run in the VM
- No application changes are needed to run in this mode

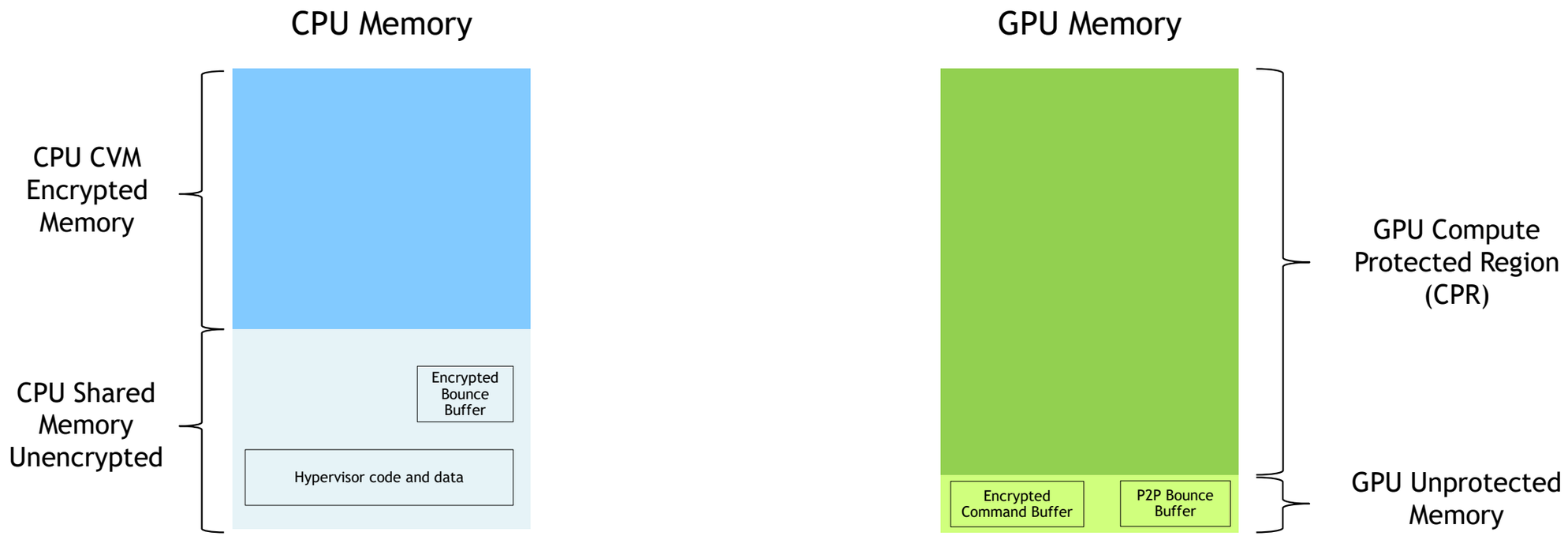
Isolation from host for confidentiality and integrity

- All data transfers are encrypted over PCIE
- Encryption transparent to applications



How Memory is Managed in Confidential Mode

Protected vs Unprotected Memory and How Each is Used

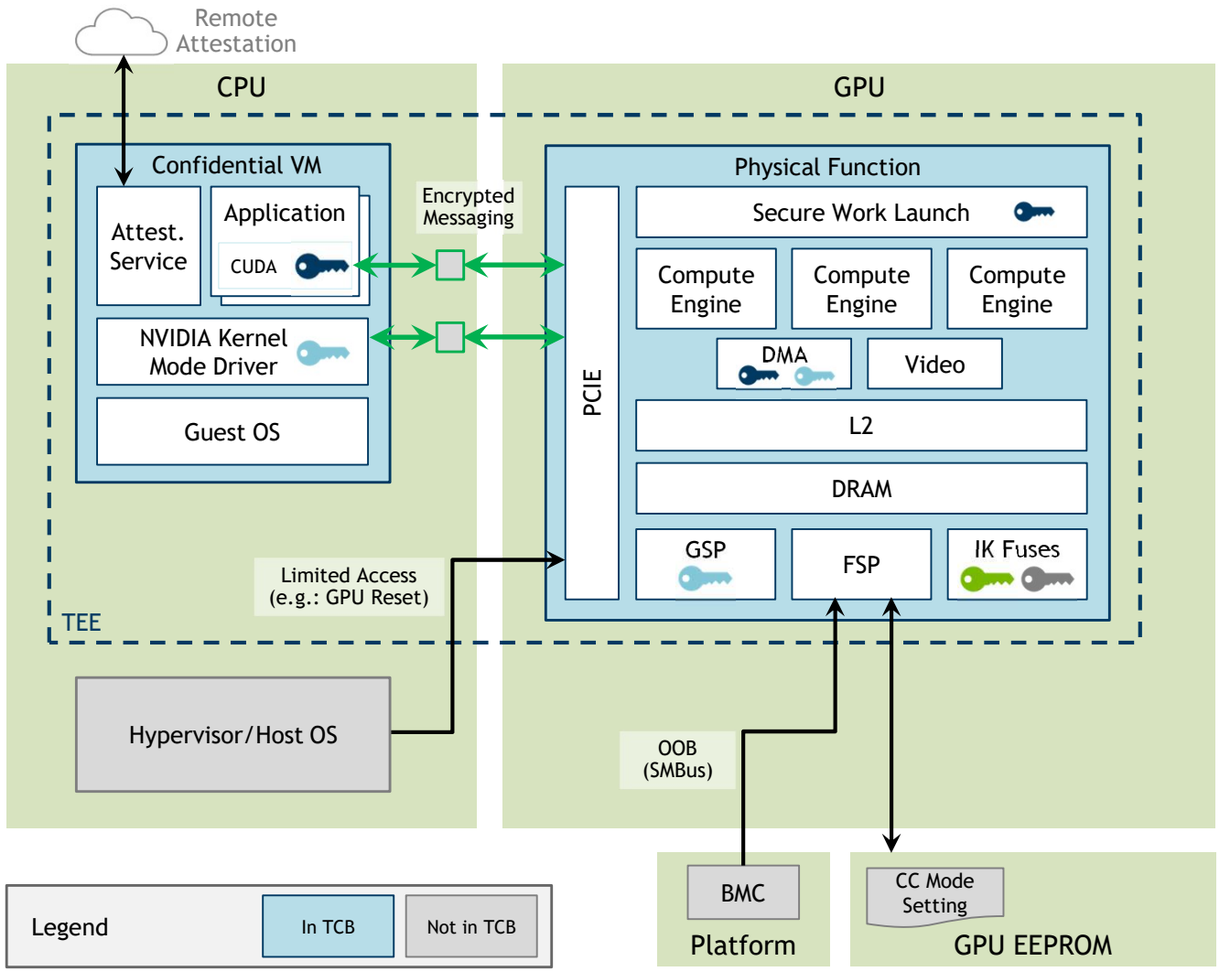


- CVM = Confidential Virtual Machine
- By default, all Guest VM memory is encrypted by the CVM private key stored in a CPU register
- NVIDIA Driver allocates bounce buffers in the Shared Memory area and encrypts data in those buffers with the session key

- Most of the GPU memory is configured as Compute Protected Region (CPR), protected by hardware firewalls
- A small portion of GPU memory is outside of the CPR and is used for:
  - Encrypted CUDA Command Buffers
  - Encrypted Bounce Buffers for NVLINK Peer to Peer

How Your CUDA Program Runs In Confidential Mode

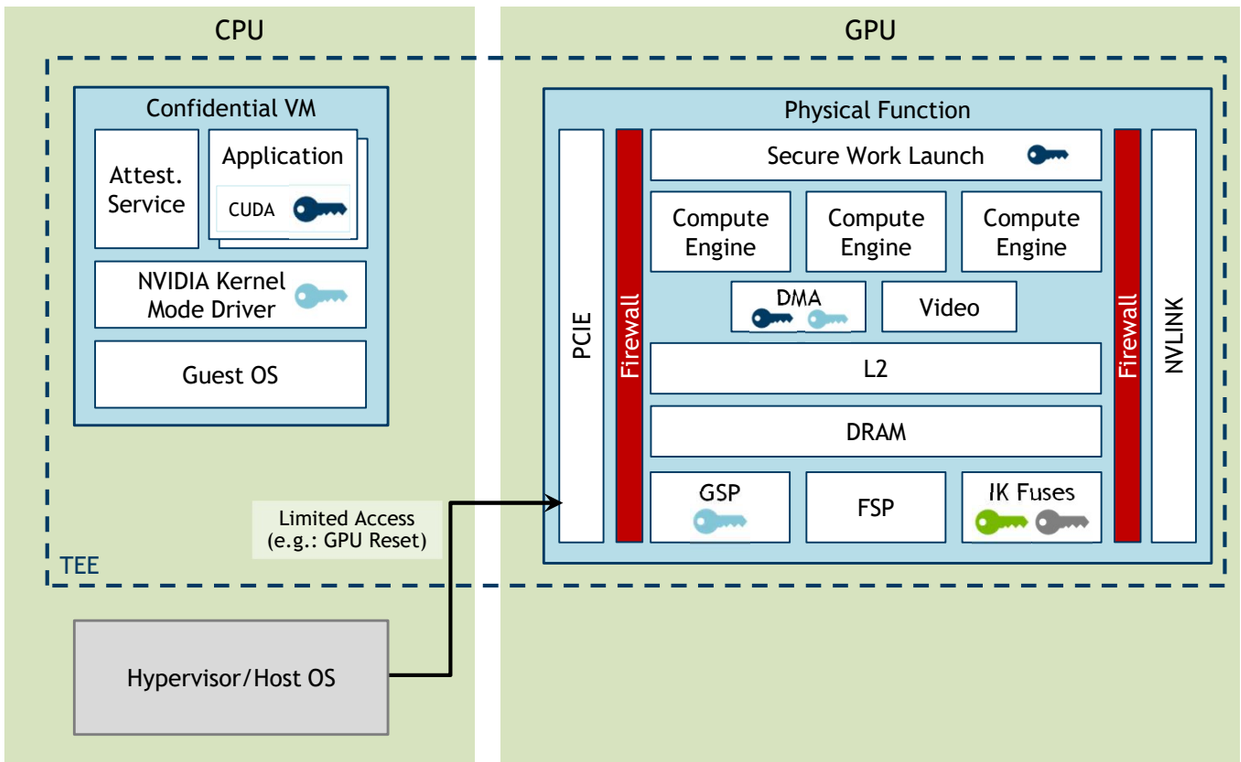
User Data and Programs are Protected in Use



- All communications to and from CPU to GPU memory are delivered encrypted including data transfers, commands and CUDA kernels.
- Bounce buffer encryption is performed by the NVIDIA driver inside the VM, transparent to applications
- For Host to Device data transfers:
  - CPU encrypt by driver to bounce buffer in shared memory
  - DMA engine reads from bounce buffer and decrypts to protected GPU memory
- For Device to Host data transfers:
  - DMA engine encrypts and writes across PCIe to bounce buffer
  - CPU decrypt by driver into confidential VM memory
- Secure Work Launch path for command buffers and CUDA kernels

Protections in Accelerated Confidential Computing

CPU VM TEE and GPU with CC=On



When the Hopper GPU boots in confidential mode, it blocks ingress and egress for the Compute Protected Region (CPR) of GPU Memory

- The PCIe Firewall blocks access by the CPU to most registers and all of the GPU CPR Memory
- NVLINK Firewall blocks access by NVLINK peer GPUs to GPU CPR Memory.
- DMA engines can only read or write outside of CPR with encryption enabled
- All other engines (e.g. Compute SMs) are blocked from reading or writing outside of CPR.

In this way, the Compute Protected Region of memory is secured so that the GPU can process data at full speed in its High Bandwidth Memory

With CC=On, all GPU performance counters are disabled, to protect against side channel attacks.

CC Modes and How to Switch Them

CC Mode is Immutable for a Running GPU

CC Mode	Meaning
Off	Regular GPU operation
On	Confidential Computing Protections are ON
Devtools	Confidential Computing Protections are OFF, but the driver uses the same paths as for CC On. Performance counters are on, so NSIGHT profiling is possible

- CC Mode must be set prior to GPU boot – in a cloud this is done by the hypervisor
  - To set the CC Mode for the next boot of a GPU, write the CC Mode bits of the GPU EEPROM
  - Once the CC Mode has been set in EEPROM, the GPU must be reset to boot in that mode.
  - It is not possible to change the CC Mode without a GPU reset.
- GPU attestation report records which of the three modes the GPU is in
  - This gives the user assurance that the hypervisor did set up the GPU correctly
- A GPU reset initiates a scrub of all GPU memory and state, including all session keys and secrets
  - This happens regardless of previous or new CC state

GPU Initialization in CC mode

CC Mode Enablement and Session Establishment

Mode Enable

1. BMC issues out-of-band request to select CC mode and write to EEPROM

Or

1. Host issues in-band request to select CC mode and write to EEPROM
2. Host triggers GPU reset for mode to take effect

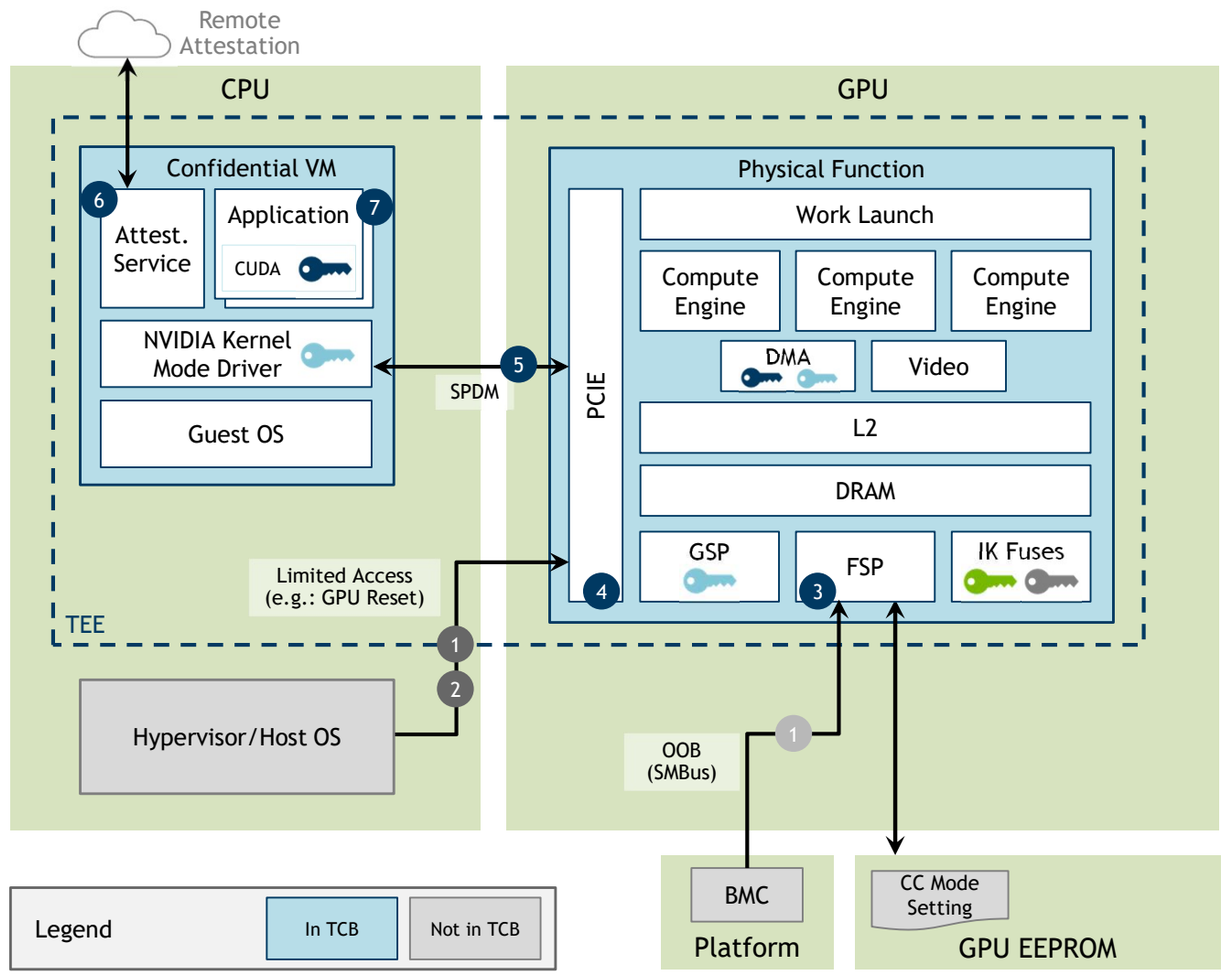
GPU Boot

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

Tenant Initialization

5. GPU driver uses SPDM for session establishment & attestation report
6. Tenant attestation service gathers the GPU attestation report and device certificate using NVML APIs

   Verification done locally or transmitted to remote service
7. CUDA programs allowed to use GPU



GPU Teardown and Scrubbing

Clean up after a confidential compute session

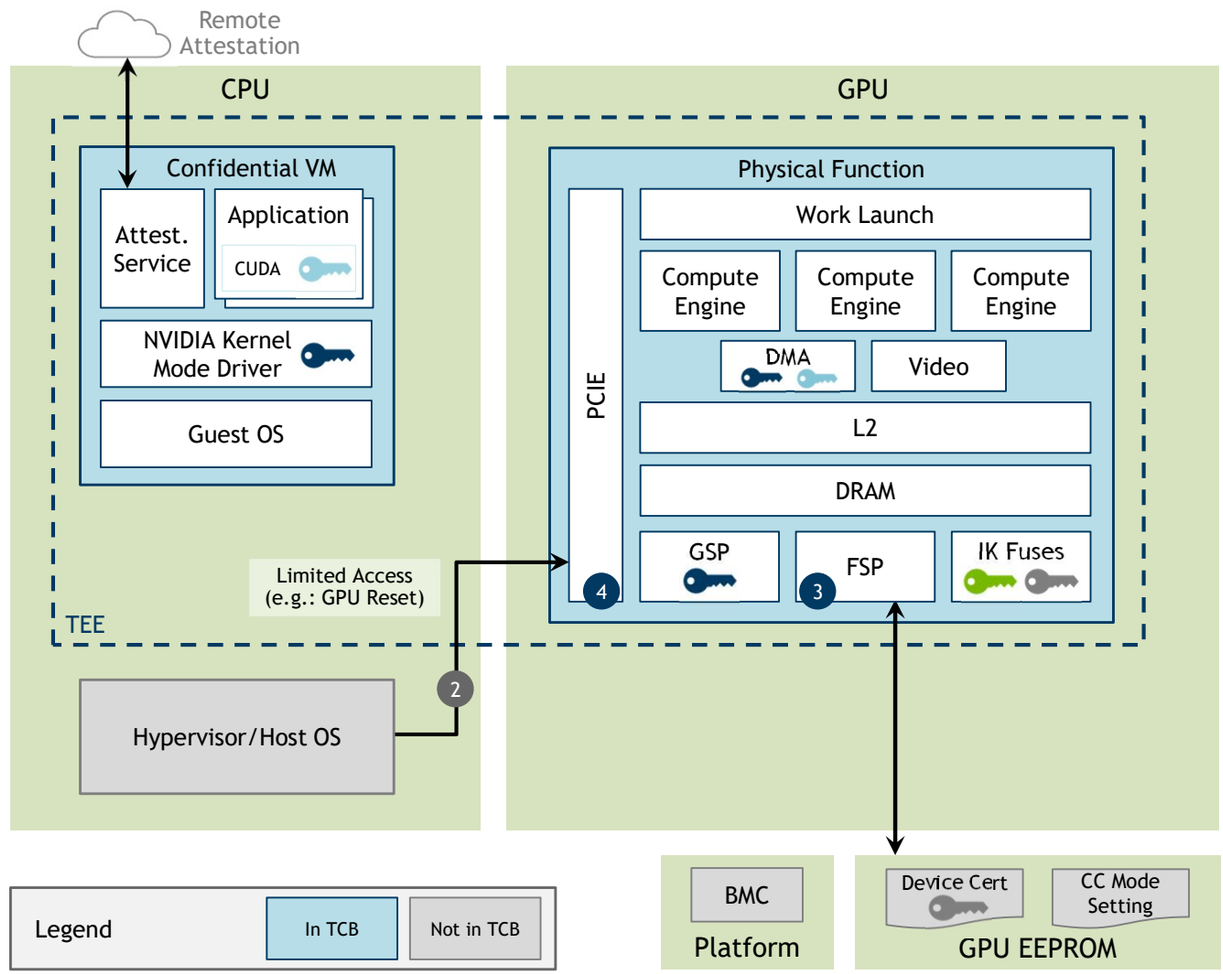
After the VM and the NVIDIA driver exit, the GPU is locked until next reset. In CC Mode the driver is run in persistence mode, since only one driver load per GPU boot is permitted.

Scrubbing of GPU is done on the next boot and follows these steps 2,3,4 of the previous slide

GPU Boot

2. Host triggers GPU reset for mode to take effect
3. GPU firmware scrubs GPU state, memory and secrets
4. GPU firmware configures firewall to prevent unauthorized access, then enables PCIE

Note that PCIE BARs are completely blocked until GPU scrubbing is complete, independent of CC mode



Attesting the GPU from Within the VM

Local Attestation

A GPU attestation report is generated while GPU is booting in confidential mode and the driver is loading. The user from within the VM can request a GPU attestation report at any time

- Some measurements are static; others are dynamic and can change after boot

Multiple ways for a user to request the GPU attestation report:

- NVIDIA Verifier, NVIDIA-SMI, NVML API

The NVIDIA Verifier performs multiple tasks on behalf of the user:

- Fetches the GPU device certificate from the driver, which builds it from IK Public in EEPROM
- Verifies the GPU certificate chain via the NVIDIA OCSP service
- Requests the attestation report from the GPU and authenticates it based on GPU certificate chain
- AND compares it to an expected “Golden RIM” result to generate a pass/fail report for correct CC configuration

GPU attestation adheres to the Trusted Computing Group RIM specification for Golden Measurements

- RIM = Reference Integrity Manifest

H100 GPU Attestation Measurements

	Measurement Group	Details	States captured (Examples)
Static - Time Domain: Life of Session	**Static HW Configurations**	• States configured at manufacturing process that defines device personality and identity	• Fuses responsible for security settings such as debug enablement, microcode revocation, CC enablement/features
	Firmware/VBIOS	• Software components flashed to EEPROM	• Signature of all firmware loaded from VBIOS and their execution environment • Device initialization data tables
	Driver microcodes	• Microcodes loaded from driver package	• Signature & execution environment of micro code for engines such as security enclaves
	HW Initialization states	• Initialization done by VBIOS and physical function driver during boot, primarily by security enclaves • Configurations required to establish and maintain GPU TEE	• Boot time initialization such as PCI-E firewalls, state of debug interfaces etc.
	Runtime states	• HW States configured by trusted GPU SW at runtime • HW and SW states programmed based on CC configuration	• CC policies (e.g., Production vs Development) • Secure/Insecure mode of HW engines • Resource isolation between TEEs
Dynamic	**Dynamic states**	• SW Engine states programmed by drivers or other software	
Static	**Reported Information**	• Signed plaintext metadata to assist attestation report verification	• CC VM configuration like secure memory size • Device configuration like SKU type, MIG state • SW version (Driver version, VBIOS version) • Event Log

Multi-GPU Confidential Computing

With NVLINK Connected GPUs

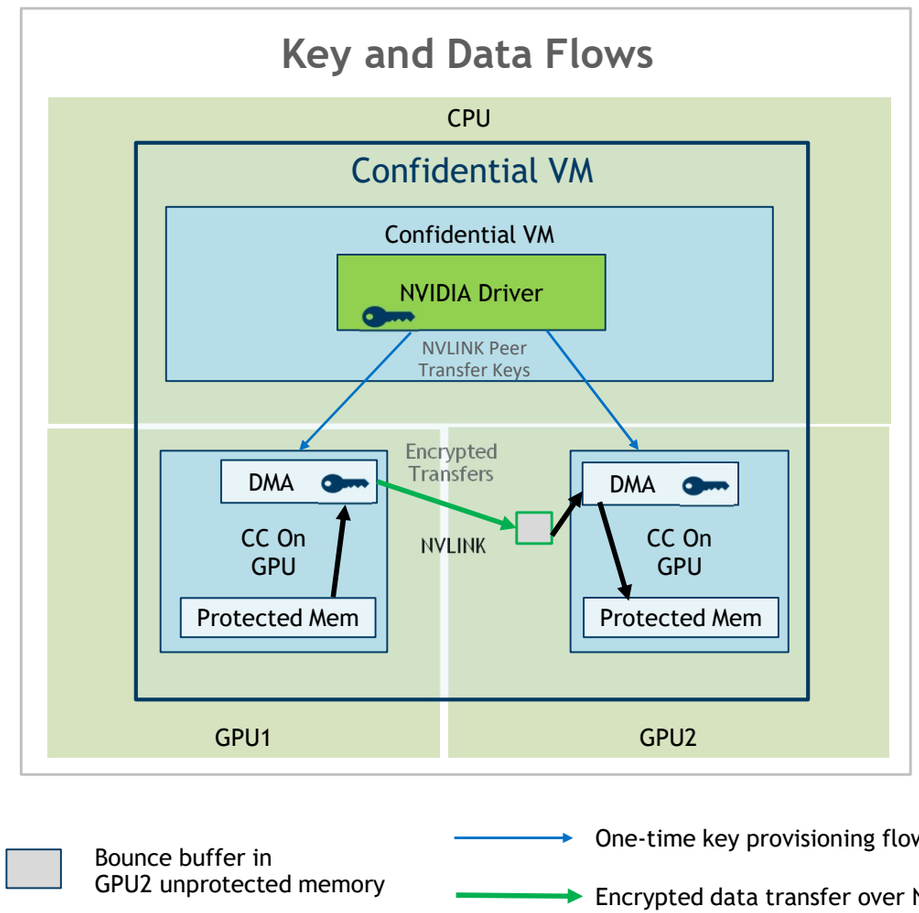
NVLINK is required between all GPUs in a multi-GPU Confidential GPU VM Instance

- PCIe P2P is not supported in Confidential Mode
- CUDA APIs and hardware firewall disallow direct pointer dereference of peer GPU memory

GPU DMA engines are provided with a shared session key for NVLINK peer transfers

cudaMemcpyDeviceToDevice() calls result in encrypted transfers via a bounce buffer

- DMA Engine in the source GPU encrypts data, transfers over untrusted NVLINK to unprotected memory of the destination GPU
- DMA Engine in the destination GPU decrypts data into protected GPU memory



Confidential Computing and MIG

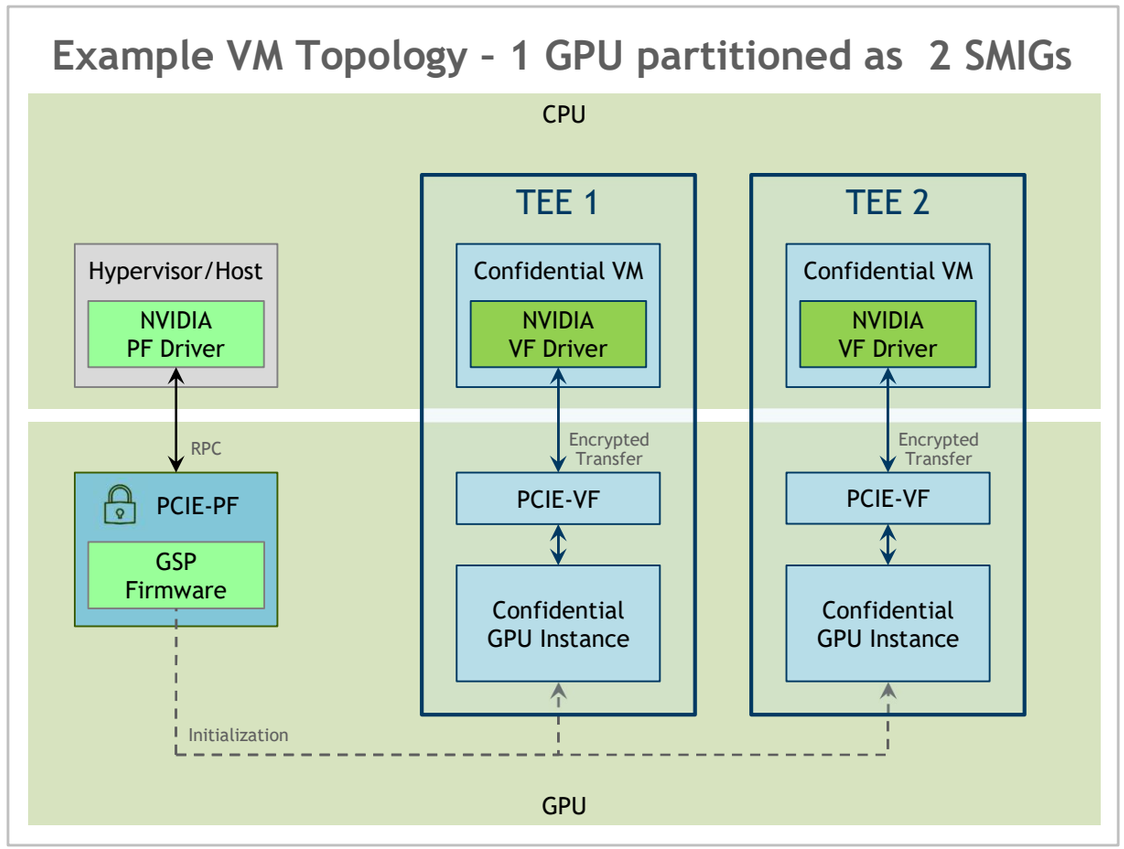
Multiple Confidential Tenants per GPU

CC with vGPU enables confidential MIG partitions:

- TEE = Confidential GPU Instance + Confidential VM
- Each TEE isolated from hypervisor/host and other tenants
- Hypervisor creates TEE instances but cannot access state
- With CC=On there are additional hardware protections between MIG partitions preventing memory attacks

Confidential Computing on MIG is based on:

- NVIDIA vGPU – multiple VMs sharing a single NVIDIA GPU
- Multi-Instance GPU (MIG) – partitioning of GPU into instances
- SR-IOV – PCIE devices expose VF for direct control by VM



Security Features in Hopper

Foundation on Which the Confidential GPU is Built

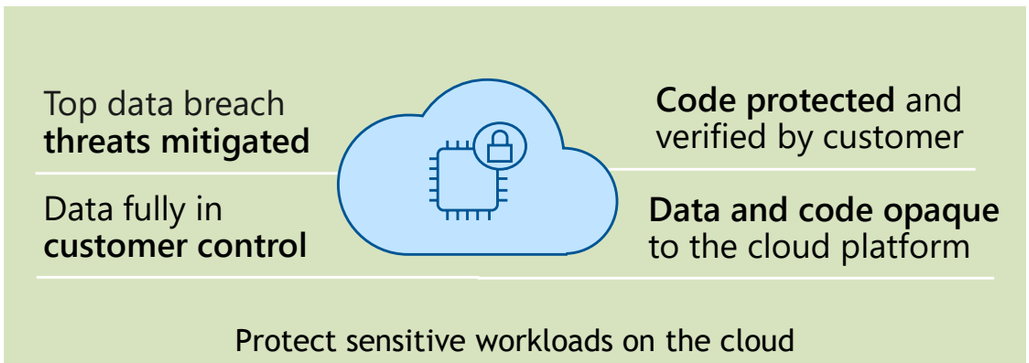
- On-Die Root of Trust
- FIPS 140-3 Level 2 Cryptography
- NVIDIA RISCV microcontrollers
- Unique Identity Key Pair
- Device Certificate
- Hardware Fault Injection Counter Measures
- PKC Firmware Authentication
- Encrypted Firmware
- Secure Boot
- Measured Boot
- Firmware Revocation
- SR-IOV for Secure MIG

Azure Confidential Computing

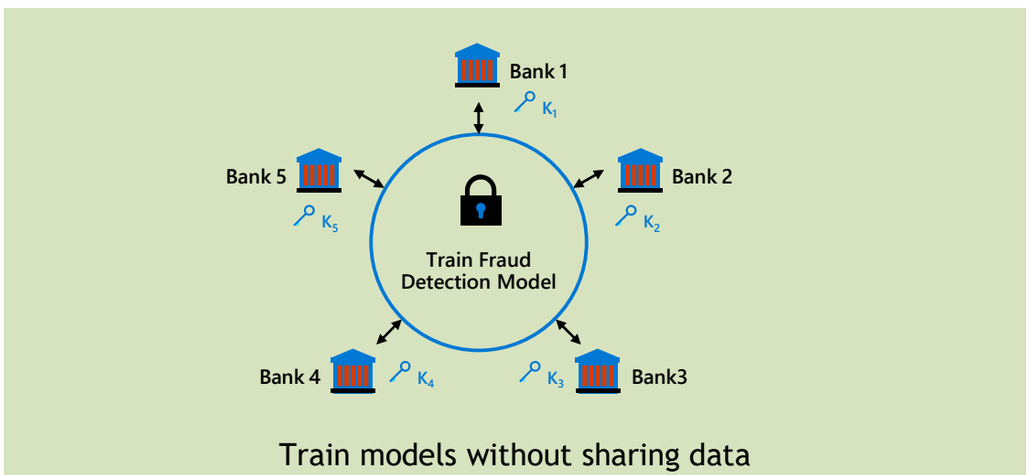
Enabling Confidential GPUs on encrypted and integrity protected Confidential VMs

Confidential GPU: Unlocking AI's Potential

Unlock access to sensitive data to securely train and run large models



Protect sensitive workloads on the cloud

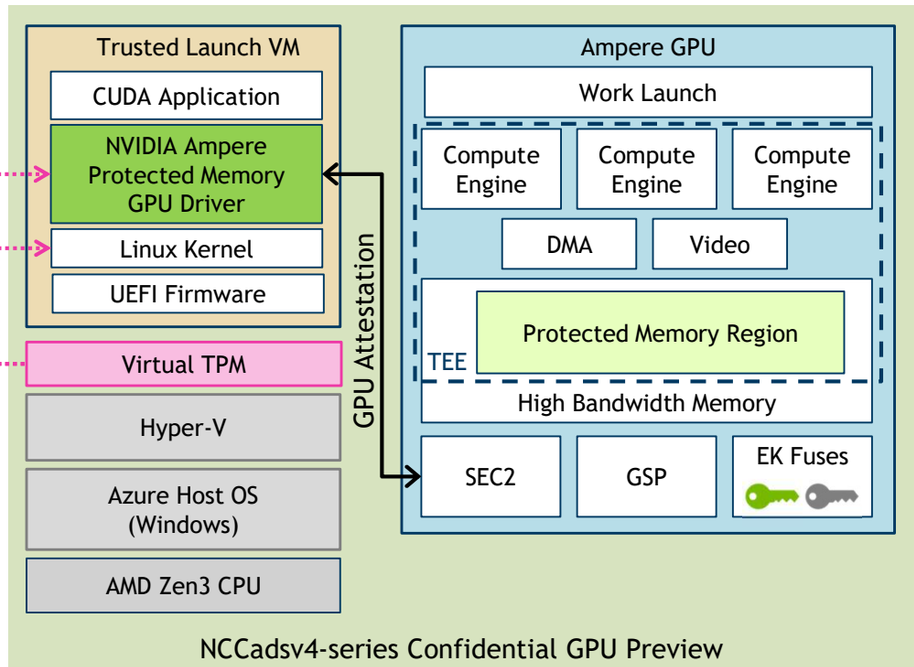


Train models without sharing data

- The full value of AI is only realized with access to data and computing power.
- Confidential Computing makes data **more accessible**: it is possible to train models without sharing data, even when multiple parties contribute their dataset.
- So far, it could only be used for small models. With GPUs, training and inference performance is **multiplied by 10 to 50** compared to current confidential CPUs.
- We showcase Azure's efforts to integrate NVIDIA's GPU with Azure's Confidential Computing products to create a fully trustworthy AI platform.

The Confidential GPU Journey

From Ampere Protected Memory to Hopper Confidential Computing



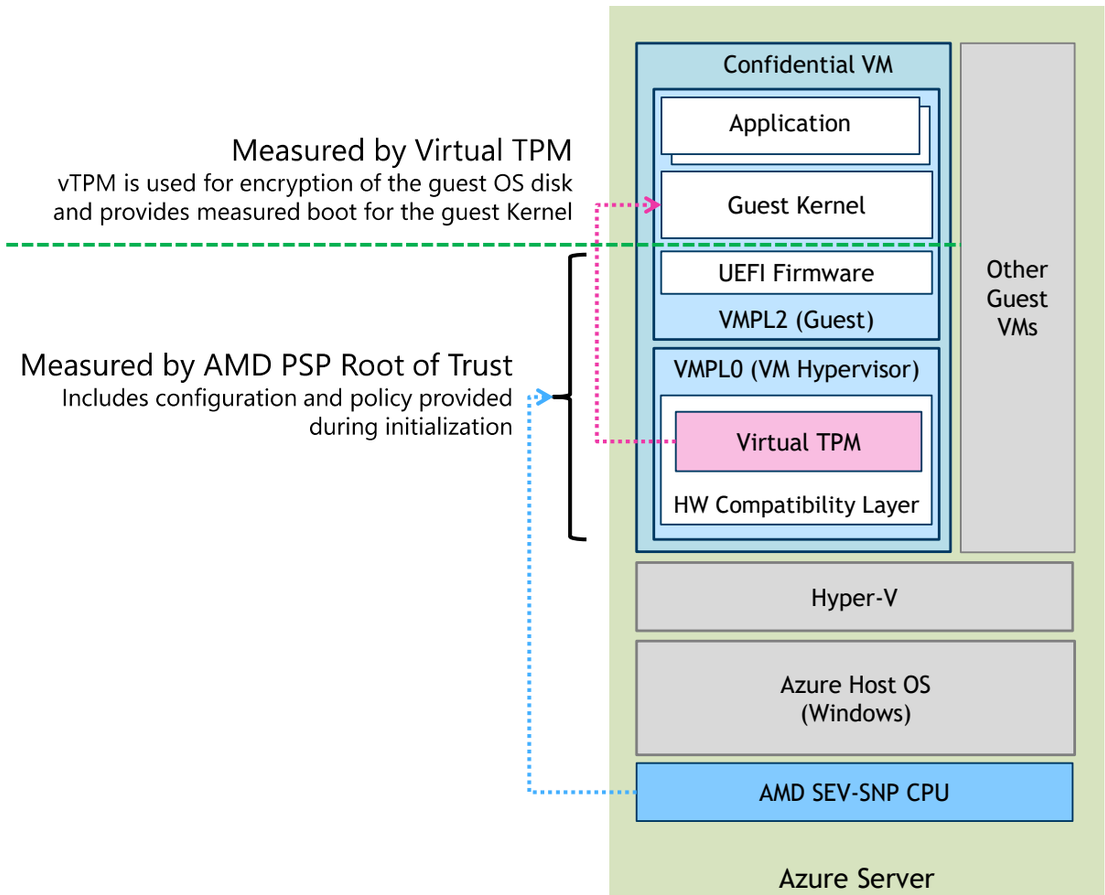
Limited preview sill ongoing, sign up at:
https://aka.ms/accgpusignup

- GTC Spring 2022: limited preview of confidential GPUs with Ampere Protected Memory and Trusted Launch VM.
- Strong interest from a broad variety of companies from many industries and research groups.
- Several startups specialized in ML privacy are eager to launch PaaS and SaaS solutions built on confidential GPU
- We received valuable feedback about the known limitations of the preview offer and worked with NVIDIA to offer the strongest possible guarantees.

	A100 APM	H100 CC
VM isolation technology	Trusted Launch (TPM)	CVM (AMD SEV-SNP)
Data protection (GPU)	✓	✓
Data protection (VM)	×	✓
Code protection (GPU)	×	✓
Code protection (VM)	×	* (via disk encryption)
Attestation (GPU)	* (partial)	✓
Attestation (VM)	Measured Boot (TPM) Signed GPU Driver	See later slides

Overview of Azure Confidential VMs

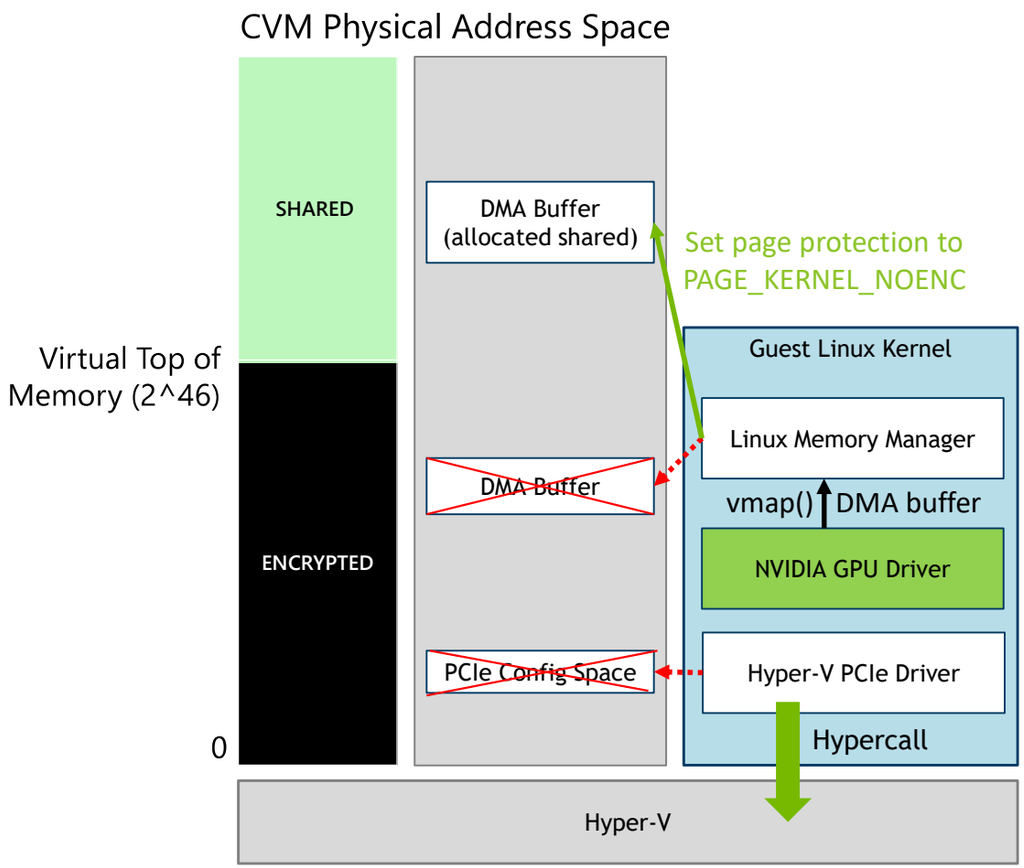
Available today on Azure (DCasv5-series, DCadsv5-series)



- Azure Confidential VMs can run unmodified Linux 5.15+ distributions (Full Disk Encryption needs a supported image)
- Guest kernel is unaware of SEV memory encryption and SNP reverse map table for memory integrity
- We re-use standard TCG security features to avoid guest changes: measured boot (attestation), full disk encryption with TPM keys
- Why should the guest trust the CVM's virtual TPM?
  - The TPM is implemented in a small hypervisor that is measured by the AMD PSP Root of Trust
  - We use the VM Privilege Level of AMD SEV-SNP to offer transparent devices (TPM, disks) to the guest
  - The AMD PSP attestation of the CVM firmware components (HCL, TPM, UEFI) is exposed via the TPM

Assigning PCIe Devices to Confidential VMs

Hyper-V is the first VMM to support transparent assignment of GPUs to SEV-SNP guests



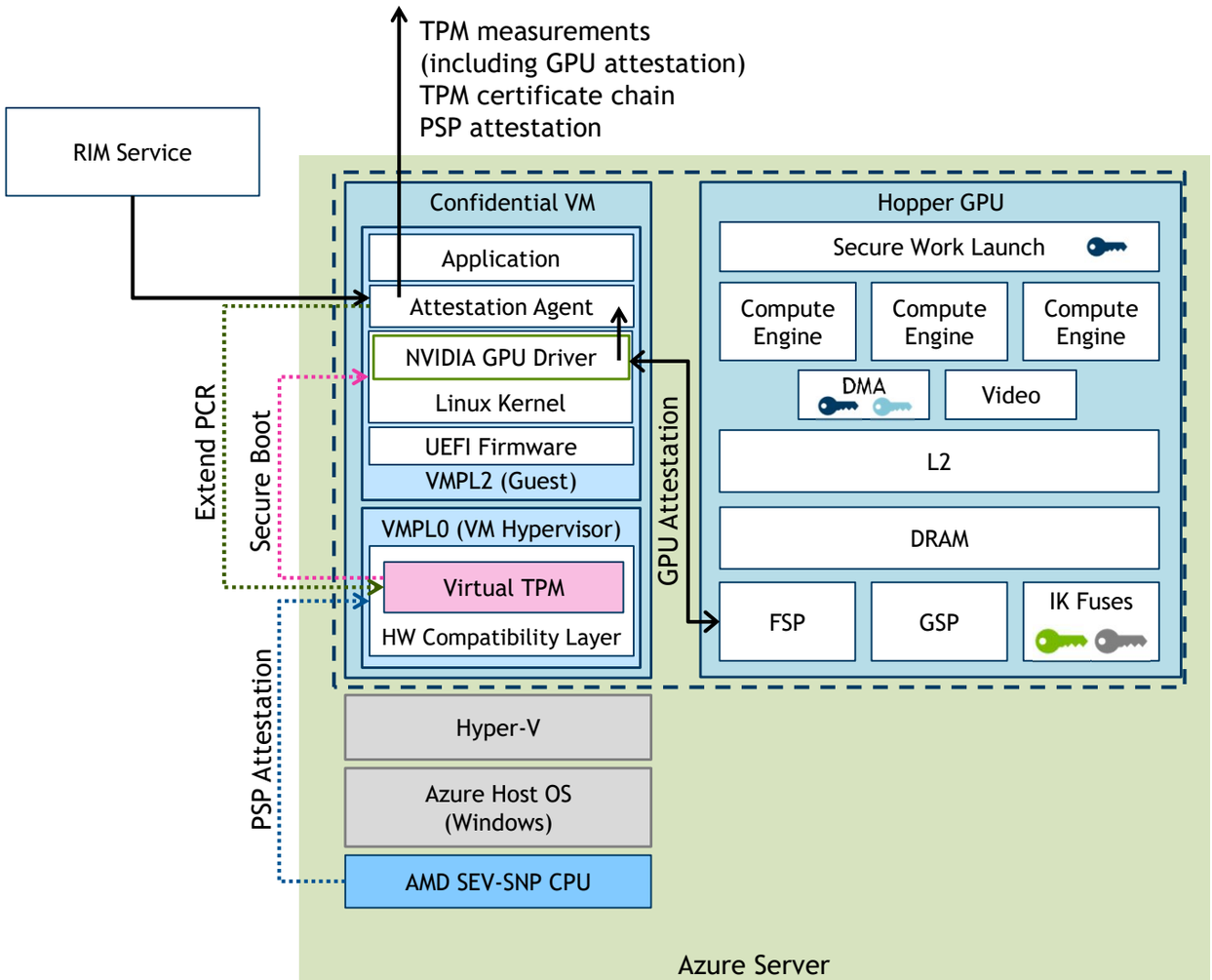
CVMs split the physical address space between encrypted pages (address < vTOM) and shared pages (address >= vTOM).

1. Only shared pages can be accessed outside the CVM. DMA buffers must be mapped above vTOM.
   - **Normal DMA operations automatically use bounce buffers mapped above vTOM (and thus 'shared', not encrypted)**
   - **NVIDIA driver also accesses buffers using `vmap()`. Those buffers must use the 'NOENC' protection flag to map as 'shared'**
2. PCIe config space access (used to enumerate devices) requires Hyper-V emulation
   - **We updated the Linux PCIe driver and Hyper-V to access config space via hypercalls; no change to individual drivers**

For drivers: no change when using standard Linux DMA APIs. Other APIs like vmap() may require the 'NOENC' flag.

Linux patches are being up-streamed to the 6.3 kernel

Extending CVM Attestation with Hopper CC Attestation



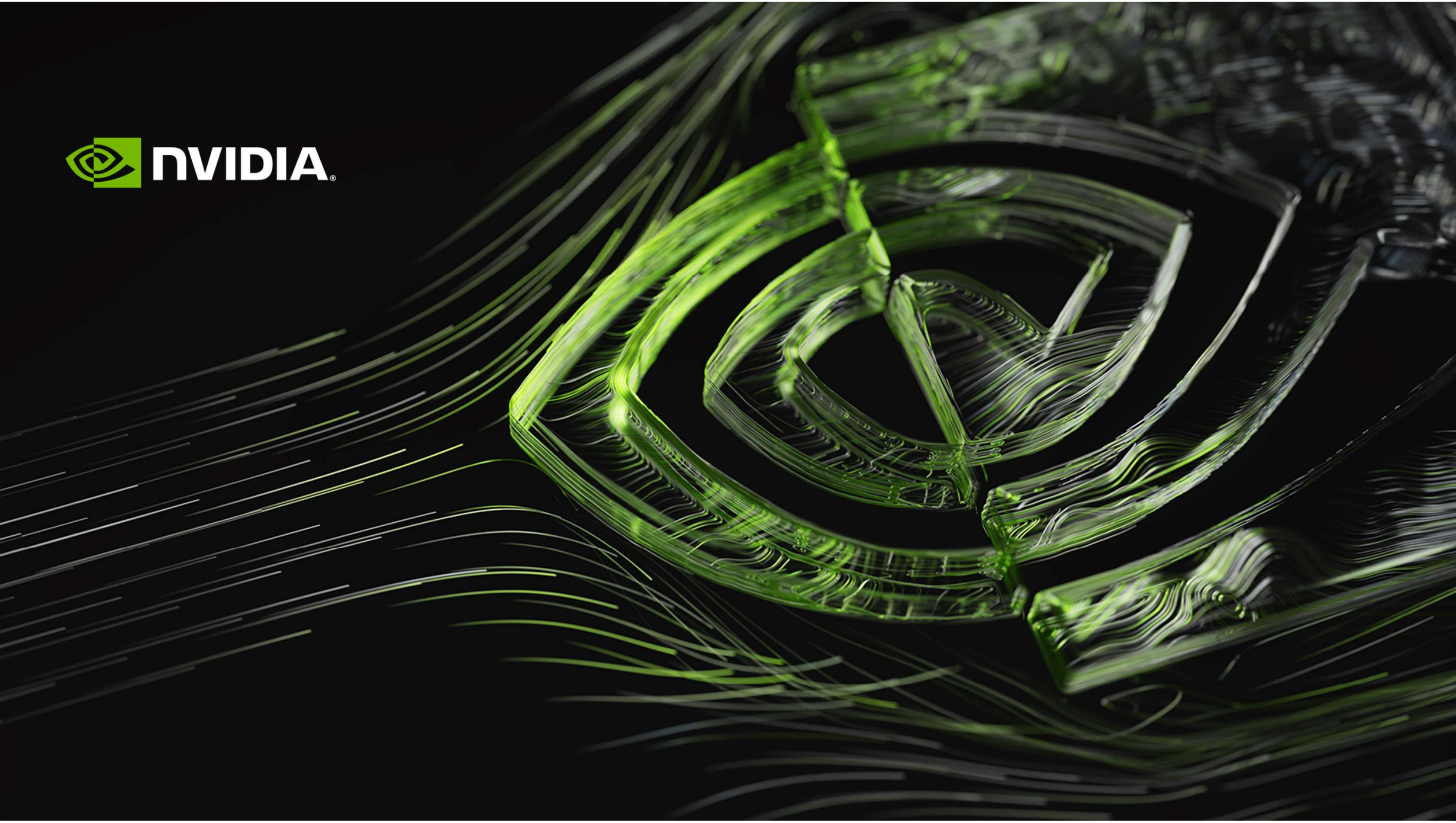
- After the secure boot of the guest, the GPU driver gets the GPU's attestation and may check some measurements (based on Golden RIM)
- An attestation agent can extend the Platform Configuration Registers (PCR) of the TPM with the GPU attestation measurements
- Developers can use the attestation evidence (TPM measurements, PSP attestation of TPM root key) to remotely attest the CVM+GPU

If you enjoyed this talk on Confidential Computing …

… here is another CC talk you should add to your GTC schedule

Tomorrow at 6am PDT:

- **S51684**: “The Developer’s View to Secure an Application and Data on H100” by Rob Nertney of NVIDIA


NVIDIA