



# **NVIDIA Secure AI with Blackwell and Hopper GPUs**

White Paper

# Document History

WP-12554-001\_v1.3

Version	Date	Authors	Description of Change
1.0	July 25, 2023	Rob Nertney	Initial Release – Early Access
1.2	February 23, 2024	Rob Nertney	General Access Release
1.3	August 14, 2025	Karthik Mandakolathur	Blackwell Update

# Table of Contents

Introduction .....	1
Drivers of the CC Paradigm.....	1
Data Privacy Concerns.....	1
Government Regulatory Regimes.....	1
Acceleration of Sensitive Workloads .....	2
How CC Secures Data in Use .....	2
The Confidential Computing Consortium (C3) .....	2
Security - Why Should You Care? A Study of Privacy-Preserving Technologies.....	3
Scope and Audience of this Document.....	4
A Description of Secure Systems.....	5
What Is a Trustworthy System? .....	5
Chain of Trust – Hardware Validity.....	6
Confidential Computing – A Feature for Secured Systems.....	7
How NVIDIA Hopper and Blackwell GPUs Integrate into a TEE .....	8
Bounce Buffers.....	8
Direct Access with Inline Encryption Using TDISP/IDE .....	9
Secure and Trusted Boot .....	9
Confidential Computing Features of NVIDIA Hopper and Blackwell .....	12
Goals for Confidential Computing.....	12
Secure AI Confidential Computing Modes.....	13
Single GPU Pass-Through.....	13
Multiple GPU Pass-Through.....	13
Protected PCIe in Hopper .....	13
Multiple GPU Pass-Through Mode with Blackwell.....	14
Threats and Mitigations .....	14
Confidentiality.....	15
Integrity .....	16
Modifying a Payload .....	16
Replaying a Message.....	17
Side-Channel Attacks.....	17
Performance Counters.....	18

Availability .....	18
Running a Confidential Compute Application on the GPU.....	20
Data Flow in CC Modes .....	20
Summary .....	22

---

# Introduction

NVIDIA GPUs have long been used as an extremely effective method of accelerating the processing of enormous amounts of data, from deep learning to high-performance computing. To meet the more stringent requirements for privacy and security that result from the processing of such enormous amounts of data, Confidential Computing (CC) was introduced with NVIDIA Hopper GPUs. Blackwell GPUs continue to support CC capabilities with enhanced security with performance.

## Drivers of the CC Paradigm

The CC paradigm is rapidly approaching because of data privacy concerns, government regulatory regimes, and the need to accelerate sensitive workloads.

### Data Privacy Concerns

Information is becoming increasingly valuable. Rapid digital transformation has led to an explosive increase in the generation, storage, and processing of sensitive data in on-premises datacenters, in the cloud, or at the edge.

By harnessing the power of AI, enterprises can take advantage of the opportunity that this wealth of data presents. From this data, enterprises can extract actionable insights, unlock new revenue streams, and improve customer experience to gain a competitive edge in today's data-driven business landscape.

As the world enters the era of exascale data collection and data brokering, control of your data is paramount. Furthermore, the data is generated, transferred, stored, and used across multiple platforms and deployment models in a highly distributed manner. These trends test the ability to retain confidence in and control over the data.

### Government Regulatory Regimes

Many government regulatory regimes are being imposed across markets and across continents, which requires a certain level of data protection that prevents access and use of private data. The complex and evolving nature of data privacy laws and regulations can pose significant challenges to organizations seeking to gain value from the use of AI on that sensitive data.

These data privacy laws and regulations include:

- > General Data Protection Regulation (GDPR) in Europe
- > Health Insurance Portability and Accountability Act (HIPAA) in the United States
- > Gramm-Leach-Bliley Act (GLBA) in the United States.

## Acceleration of Sensitive Workloads

The global imperative for nations to invest in sovereign AI capabilities has grown since the rise of [generative AI](#), which is reshaping markets, challenging governance models, and inspiring new industries while transforming other industries. Along with essential infrastructure for AI production, “AI factories”, where data comes in and intelligence comes out, the next-generation data centers that host advanced, full-stack accelerated computing platforms for the most computationally intensive tasks are being built with much valuable model IP utilizing those private data.

## How CC Secures Data in Use

Until recently, securing data was limited to data-in-motion (for example, moving a payload across the Internet with SSL/TLS), and data-at-rest (for example, encryption of storage media). Data-in-use, however, remained at risk.

Enter CC, with which users can close this gap and enable a more holistic end-to-end data protection model. CC provides the ability to process data and code in-use securely and prevent unauthorized users from both accessing and modifying the data. It enables users to have an end-to-end data protection model and strengthen the layered security approach.

Enabling CC involves:

1. Securing the hardware interfaces to the CPU or GPU (for example, DRAM, and PCIe)
2. Securing the software from the owner/operators of the systems from unauthorized access to end-user data in a Trusted Execution Environment (TEE)

## The Confidential Computing Consortium (C3)

C3, a project at the Linux Foundation, is the primary governing body of CC. C3 comprises the top hardware and software vendors in the world, all of whom have the unified goal of standardizing CC across hardware and software.

C3 has set forth in defining standards for attesting the hardware’s validity and setting up secure sessions to and from that hardware, while simultaneously working with hypervisor vendors to provide mechanisms to end-users that would ensure their data security, even from a ‘super user’ with direct access to the memory.

C3 defines CC as the following:

*Confidential Computing protects data in use by performing computation in a hardware-based attested Trusted Execution Environment. These secure and isolated environments prevent unauthorized access or modification of applications and data while in use, thereby increasing the security assurances for organizations that manage sensitive and regulated data. Today, data is often encrypted at rest in storage and in transit across the network, but not while in use in memory. Additionally, the ability to protect data and code while it is in use is limited in conventional computing infrastructure. Organizations that handle sensitive data such as Personally Identifiable Information (PII), financial data, or health information need to mitigate threats that target the confidentiality and integrity of either the application or the data in memory.*

## Security - Why Should You Care? A Study of Privacy-Preserving Technologies

Broad market agreement with many technology consultancies such as [Gartner](#), [Deloitte](#), [McKinsey](#), [BCG](#), or [Capgemini](#), International Data Corporation (IDC) highlight the need for broader data collaboration and sharing to extend competitive advantages in AI and data-centric solution development. For many market verticals, natural market competitiveness and regulatory and privacy concerns create barriers to collaboration (even in the same organization), which dilutes the value of data in the organization and increases market friction for cross-company and cross-industry data collaborations. Even government organizations such as the US DHS, US VA, US Census, and other large sources of data are challenged to realize their full potential due to regulatory and privacy barriers within the public and public-private spheres.

Increasingly, organizations are looking to privacy-enhancing technologies (PETs) to help navigate the complex regulatory, privacy, and market barriers to realize the full potential of the local and shared data opportunities. As noted by the analysts, PETs are increasingly a driving force of innovation and value creation in data ecosystems.

**Figure 1. Cost of Data Breaches <sup>1</sup>**



Source: [IBM Cost of Data Breaches report](#)

# Scope and Audience of this Document

This document is designed to provide a high-level overview of the CC capabilities of the NVIDIA Hopper and Blackwell GPUs. Specific details on industry standards such as encryption/authentication algorithms and certifications are beyond the scope of this document.

Other NVIDIA Secure AI and Confidential Computing documents may be found at [NVIDIA Trusted Computing Solutions](#) on the NVIDIA Docs Hub.

---

# A Description of Secure Systems

This section provides information about secure systems.

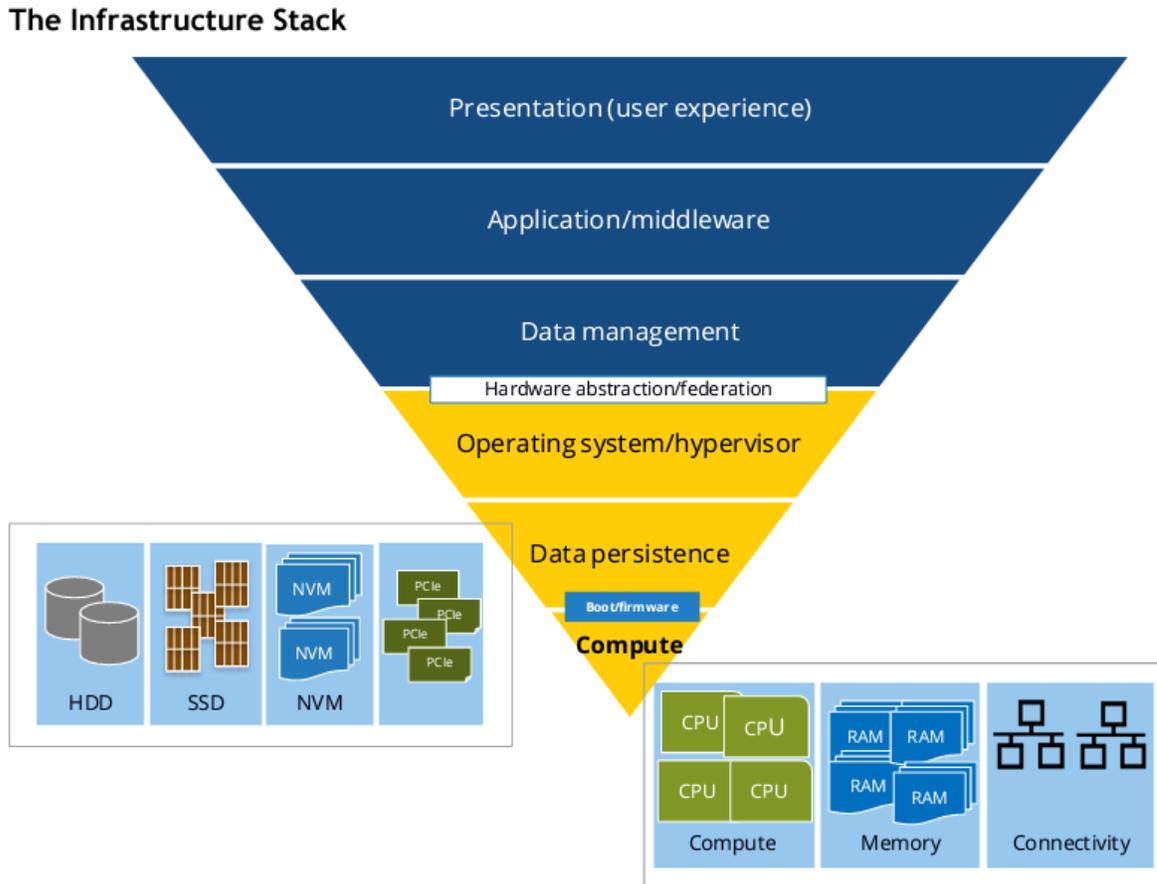
## What Is a Trustworthy System?

C3 is helping drive the industry towards a standardized flow to ensure confidentiality. However, confidentiality in a system is a multilayered stack. Achieving the highest level of trust involves more than physically securing nodes in a data center through measures such as restricted entry, vetted technicians, and isolated management networks. The process of securing a system actually begins even before the system is powered on for the first time. To achieve the highest level of trust, you must start with the base hardware of the system as shown in **Error! Reference source not found.**

A trustworthy system is made up of hardware and software that can be verified as trustworthy by using mechanisms to verify the authenticity of technologies that make up the system. In the context of using CC, users must verify the trustworthiness of the devices and firmware before they can run their workloads while protecting the data and code in use.

In the layers of technologies that make up systems, each layer relies on the technology in the layer below it in the infrastructure stack. To verify the trustworthiness of the system, it is important to start with verifying the trustworthiness of the lowest layer of technology of the infrastructure stack, which is the hardware compute layer, and work your way up.

Figure 2. A Secure System Stack



Source: IDC, 2019

## Chain of Trust – Hardware Validity

The *Chain of Trust* describes the process of validating the trustworthiness of a system from the end unit at every step until you reach the final authority, or “Root” that can be inherently trusted. This process establishes a verifiable paper trail where, at any point, you can inspect and ensure the validity of a system’s trustworthiness. As a developer seeking CC capabilities, you should be vigilant in scoping the trustworthiness of not just one part of a device or its firmware, but the overall system.

Many silicon providers have created their own methods of providing proof of authenticity. For example, the CPU vendor who provides proof of silicon authenticity or restricts its use to a specific OEM, the motherboard vendors who also provide similar proof for their coprocessors (for example, a BMC) and the associated firmware-based components such as the BIOS, UEFI, or BMC code. Paired together, owner-operators of standard compute nodes have built the foundation for Confidential Computing.

In a chain of trust, the trustworthiness of each layer of software that composes the chain is guaranteed by the previous layer, until it reaches the root of the chain (or Root of Trust (RoT)). Immutability and formal verification provide the foundation for a RoT. In CC, we use the On-Die RoT to verify the key fused onto the GPU to verify the trustworthiness of the GPU identity, and to verify the trustworthiness of the firmware used.

Combined with the RoT, Secure Boot verifies that the firmware of a GPU has been signed by NVIDIA and allows the execution only of signed and authenticated firmware during the GPU boot. It is a mechanism that is used to authenticate and load the GPU firmware modules. Hardware mechanisms ensure the firmware cannot be modified after the load process until a subsequent reset. The combination of these two mechanisms ensures the GPU will always run only authenticated and uncorrupted firmware.

CC is a feature of secure hardware and must rely on attested hardware before confidentiality can be expected. Hardware vendors provide methods to ensure that you have confidence that at no point in the (often global) shipping process was the hardware meaningfully modified without your being able to detect it.

## Confidential Computing – A Feature for Secured Systems

So far, the descriptions have strictly ensured the validity of the hardware and have not yet focused specifically on CC. These features are new, and are constantly evolving to meet the performance demands of developers and confidentiality requirements set forth by organizational regimes, so you must have specific CPU hardware SKUs to enable CC with NVIDIA's Hopper and Blackwell GPUs:

- > Intel CPUs must support *Trusted Domain eXtensions* (TDX).
- > AMD CPUs must support Secure Encrypted Virtualization (SEV). NVIDIA further recommends that the AMD CPU also support Secure Nested Paging (SEV-SNP) to protect against memory integrity attacks.

These CPUs have hardware-based features that are required for CC, where vendors have similar methods to isolate virtual machines (VMs) that are to remain confidential (Confidential Virtual Machine (CVM)), with unique differences that achieve the goal of isolating a VM from outside sources.

These hosts have hardware features to support confidential compute, and each host can use a combination of techniques such as access control checks, paging control, address translation and memory encryption to provide protection of data while it is being used. Refer to [Common Terminology for Confidential Computing](#) for more information.

# How NVIDIA Hopper and Blackwell GPUs Integrate into a TEE

The first problem to solve is to provide the GPU access to the encrypted data in the existing framework of how the CPUs are handling memory pages in the system. The MMU in the processor is already configured to prevent unauthorized memory access between VMs, but the data is encrypted.

There are two broad ways to integrate GPUs into the TEE:

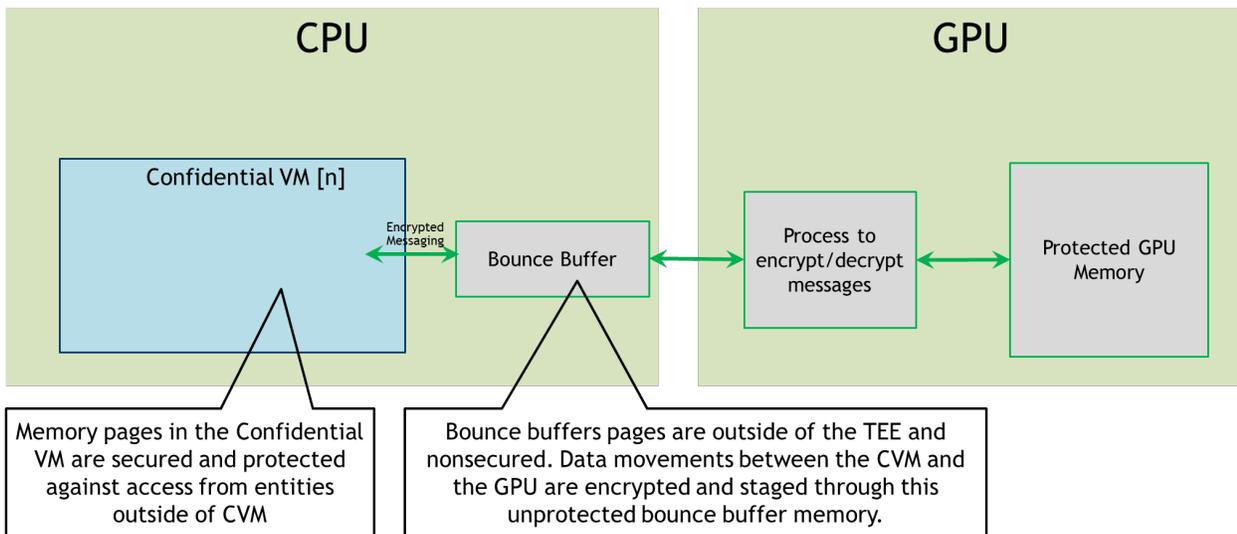
- > Use software to stage data transfers through Bounce Buffers.
- > Use the GPU's TEE-IO capability alongside IDE and TDISP protocol to natively extend the TEE to include the GPU.

## Bounce Buffers

CPU vendors have allowed for the allocation of nonsecure pages while in confidential mode, which means that anyone in the system can gain access to it. However, provided that the data in these nonsecure pages remains encrypted, confidentiality is maintained.

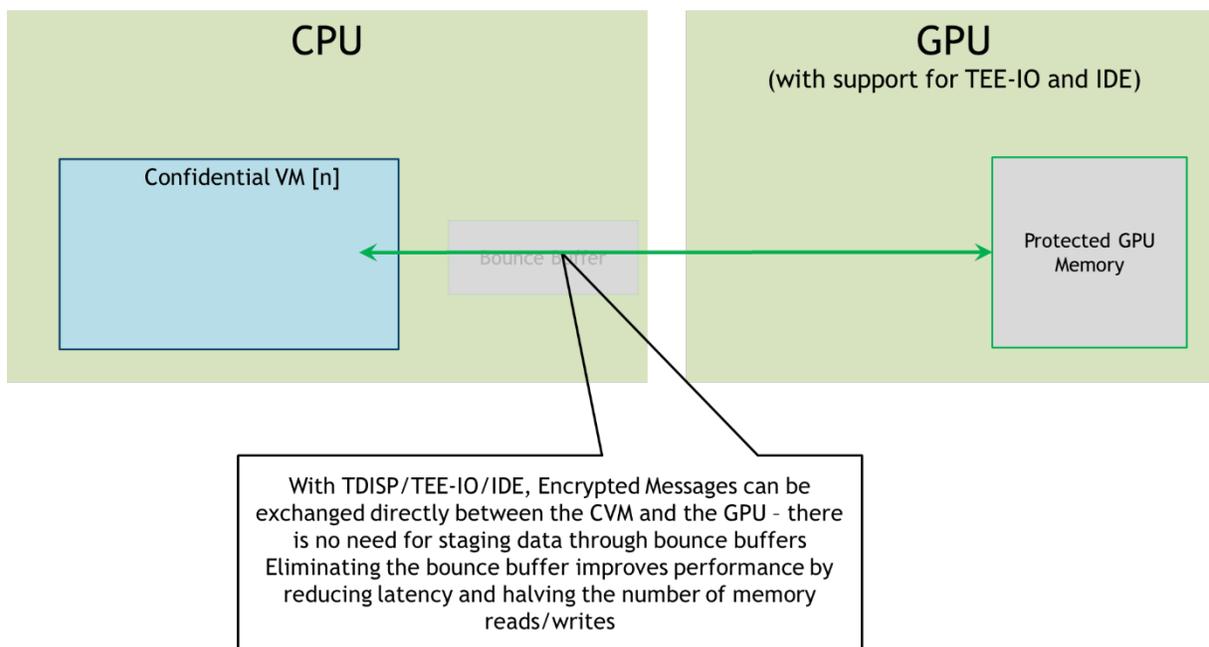
These nonsecure pages are visible to the GPU, which can transfer payloads from or to its internal protected storage after encrypting or decrypting them.

**Figure 3. How Bounce Buffers Work**



## Direct Access with Inline Encryption Using TDISP/IDE

**Figure 4. How TDISP/IDE Eliminates Bounce Buffers**



The TEE Device Interface Security Protocol (TDISP) establishes a framework for extending the trusted execution environment to include PCIe devices, such as GPUs. TDISP allows for secure communication between GPUs and confidential virtual machines (VMs) over a secure channel, eliminating the necessity of intermediate bounce buffers after a trust relationship has been formed. Additionally, Integrity and Data Encryption (IDE) is employed to safeguard against physical attacks on link layers, with TDISP enhancing security by ensuring the confidentiality and integrity of the data exchanged. To enable TDISP/IDE end-to-end, both the GPU and CPU should support it. If there is a PCI Express switch between the CPU and GPU, the switch should support IDE flow through to enable TDISP.

Like CC capabilities in the CPU, CC capabilities in the GPU require many new and innovative hardware features. Before the specifics of these internal features, modes of operation, and how developers can start to use them are discussed, one last topic in a secure system must be considered: Secure Boot.

## Secure and Trusted Boot

After the system hardware has arrived and been taken into your data center, the components from the various hardware vendors are connected and the process of

enabling a confidential system can truly begin. However, many activities begin long before the operating system is ready to provision VMs with GPUs attached.

When the system is turned on, many components will begin their self-checks (their RoT; and in some cases, components might communicate). If all the tests pass, the system begins the process of booting into a host operating system. In Secure Boot, the motherboard is configured to store a set of certificates in non-volatile memory that can be used to authenticate the validity of software binaries that try to load for execution. Many OEMs have the certificates preloaded from the main OS vendors (for example, Canonical, Red Hat, and Microsoft). As the UEFI firmware fetches the OS boot-loader binaries, their signatures are checked against the preloaded certificates. If the binary is unsigned or fails the check (for example, for a signature without a matching certificate, or with a revoked certificate), the system halts its boot process.

If the first bootloader binary passes the signature/certificate check, the UEFI firmware loads and passes execution off to it. This authenticated bootloader can then load the final kernel. Because this kernel still has full access to the system firmware, it must also be validated against certificates in the system. After they are loaded, these validated kernels disable access to the system firmware as the system continues to boot.

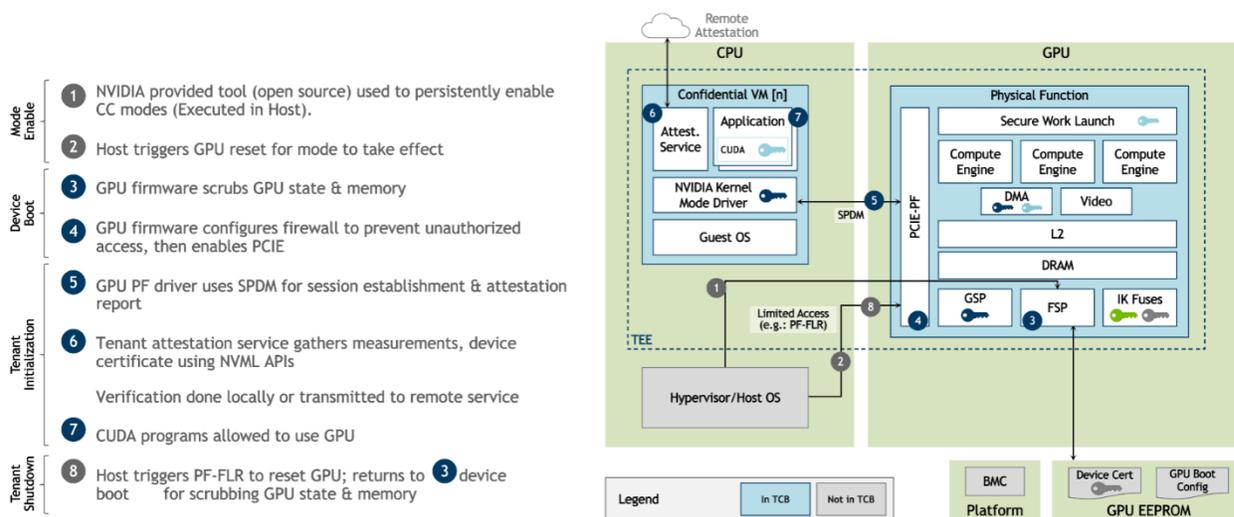
The kernel drivers still have access to the boot firmware, and, therefore, must ensure that I/O devices in the system are allowed to operate by checking the signatures on the hardware drivers. The kernel begins its own query of the hardware in the system, and through a handshake to the motherboard, it requests to determine whether the certificate is stored, is valid, and matches the signature of the drivers that are loading.

At any point, a failure can stop the boot process or disable the driver without halting the boot process.

As mentioned in Confidential Computing – A Feature for Secured Systems, CC with NVIDIA Hopper and Blackwell GPUs relies on a CPU with AMD SEV-SNP or Intel-TDX. These are virtual-machine-based solutions and, as such, must repeat most of the steps performed by the underlying physical host. At this point in the boot flow, the host (owner/operator) OS has completed a base validation of the hardware in the system but has implicitly trusted the hardware vendors that they “did the right thing.” To confirm that the hardware vendors “did the right thing,” an extra step called “attestation” is required. In the attestation step, the host might request certain hardware to provide evidence that it “is what it claims to be,” that it is running the correct levels of firmware, that it is configured correctly, and so forth.

Attestation of running hardware is paramount to the adage “trust but verify.” The host should run attestation on all available hardware before launching its hypervisor and creating VMs for their final confidential workloads. The host, after doing its own attestation checks, passes appropriate hardware to its hypervisor and provisions a Confidential Virtual Machine (CVM). In this CVM, as the kernel is launched, it must again revalidate the signatures of any drivers attempting to gain direct access to system

hardware as described previously. After the system has booted, the CVM is ready to be transferred to the user. The user might (and should) use additional attestation for the CVM in the infrastructure of the host's owner-operator.



The preceding figure shows the steps involved in setting up and tearing down a Confidential Virtual Machine with an NVIDIA GPU.

- > Steps 1 and 2 describe how to persistently enable Secure AI modes in the system and make them take effect.
- > Steps 3 and 4 are automatically performed by the firmware in the confidential computing mode when the device is booted.
- > As part of Step 5, the driver establishes an SPDM session with the firmware running in the GPU. In this step, a Diffie-Hellman key exchange takes place for setting up a shared symmetric session key between the driver running in the CPU and firmware running in the GPU. With the keys in place, encrypted messages can be exchanged between the CPU and the GPU.
- > Step 6 briefly describes the attestation process.
- > CUDA programs are allowed to run as part of Step 7 only after the attestation is completed successfully.
- > Step 8 describes how the GPU state is cleaned up in the system before the CVM is torn down.

---

# Confidential Computing Features of NVIDIA Hopper and Blackwell

We have established a basic understanding of what makes a secure system, how the market has begun to leverage CC, and how an NVIDIA GPU can integrate into the existing flows. We have covered hardware checking when the power is turned on, ensuring that the devices can do some initial cross-checks for supported and authorized hardware and ready states. After the CVM begins to load, the NVIDIA GPU driver begins to initiate many processes to prepare the GPU for use in a confidential system.

To have a true production-ready confidential accelerator, NVIDIA needs robust firmware and software stacks with attestation flows to provide a complete CC solution that includes protection and integrity of both code and data. The Hopper and Blackwell GPUs have several new hardware-based features added to their silicon that enable this level of confidentiality.

## Goals for Confidential Computing

The main goals that NVIDIA set for enabling GPUs with CC, aligned to the charter of the C3, are as follows:

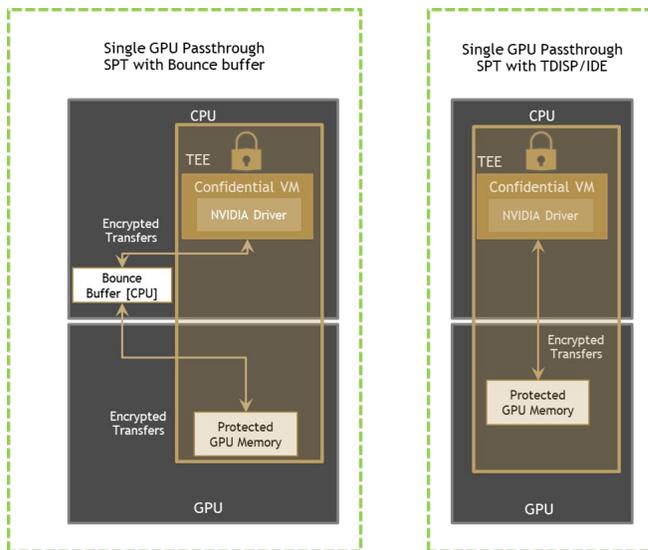
- > Data and code confidentiality  
Protect all application code and data in the VM instance from being read by the host.
- > Data and code integrity  
Protect all application code and data in the VM instance from being altered by the host.
- > Protection against basic physical attacks  
Ensure that interposers on buses such as PCIe and DDR memory cannot leak data or code.

# Secure AI Confidential Computing Modes

Depending on the workload requirements, users might want to use a single GPU or multiple GPUs in their Confidential Computing environments. NVIDIA currently supports single-GPU and multiple-GPU pass-through<sup>1</sup> modes with Hopper and Blackwell GPUs:

## Single GPU Pass-Through

This mode permits the allocation of a maximum of one GPU to a single Confidential Virtual Machine (CVM). Data movement between the CPU and GPU is encrypted using software data encryption with bounce buffers. Alternatively, encryption can be achieved through TDISP/IDE when coupled with a compatible CPU, Blackwell GPU, and supported firmware and driver releases.

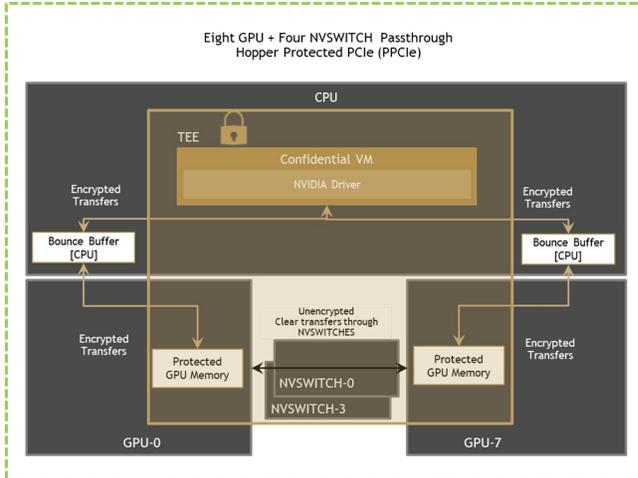


## Multiple GPU Pass-Through

### Protected PCIe in Hopper

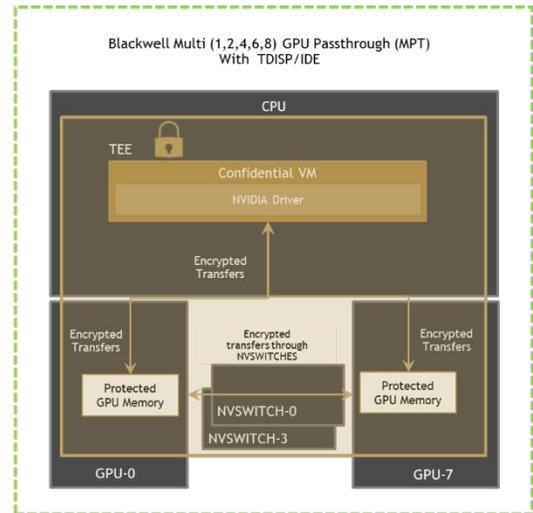
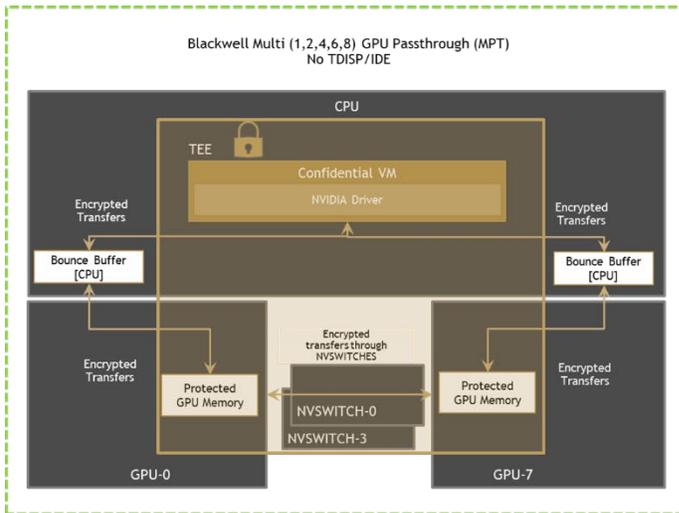
This mode is exclusively supported in eight-GPU four-NVSWITCH HGX Hopper air-cooled systems. All eight GPUs in the node are passed through to a single CVM. Data traffic between the CPU and GPU is encrypted by using bounce buffers, while data traffic between GPUs over NVLINK remains unencrypted.

<sup>1</sup> GPU pass-through directly assigns one or more entire physical GPUs to one VM. In this mode of operation, each GPU is accessed exclusively by the NVIDIA driver running in the VM to which it is assigned. The GPUs are not shared among VMs.



### Multiple GPU Pass-Through Mode with Blackwell

In this mode, the NVLINK pathway is also encrypted, and up to eight GPUs can be passed through to a CVM. Data movement between the CPU and GPU is encrypted by using software data encryption with bounce buffers. Alternatively, encryption can be achieved through TDISP/IDE when coupled with a compatible CPU, Blackwell GPU, and supported firmware and driver releases.



## Threats and Mitigations

The in-scope and out-of-scope threat vectors for CC mode are as follows:

- > In-scope threat vectors:
  - Software attacks
  - Basic physical attacks
  - Software rollback attacks
  - Cryptographical attacks
  - Data rollback and replay attacks

- > Out-of-scope threat vectors
  - Sophisticated physical attacks
  - Denial of service attacks

The goals for CC mode can be divided into the categories shown in **Error! Reference source not found.** The following sections describe how the H100 mitigates a threat at a high level for these categories (confidentiality, integrity, availability, and general).

Figure 5. Threats and Mitigations of Confidential Computing Modes

## THREATS & MITIGATIONS

✓=Mitigation, ✗✓=Partial Mitigation and ✗=Not Mitigated

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 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	✗

## Confidentiality

The confidentiality category describes how your data is kept secret from malicious or otherwise unauthorized actors. There are many actions that you can consider as snooping or otherwise obtaining confidential data, from hardware-based snooping of the physical interfaces to software-based attempts to access memory or GPUs not assigned to the bad actor. Some mitigations for these actions are inherited from the CPU vendors’ efforts to keep a traditional CVM secure, while others are novel from NVIDIA and specific to securing a GPU.

The easiest action to visualize is using the physical PCIe connections to read tenant data. PCIe bus analyzers are quite common throughout the industry, particularly for debugging new silicon or testing for PCI SIG compliance. Traditionally, these analyzers are physically connected between the host and the target device, quietly recording all activity that goes across the copper lines of the connector, and then refactoring the data into a human-readable format. NVIDIA has built-in encryption and decryption

engines that use 256-bit AES-GCM encryption across all its ingress and egress paths. Any request or response that enters or exits the GPU must be encrypted.

However, without changing the key, sending the same plain-text message multiple times results in the same encrypted data on the bus. This result is unacceptable because it might provide an attacker with knowledge of the system, payloads, and so forth. A solution is to introduce an additional layer of protection by changing the Initialization Vector (IV), which can be used to effectively randomize each message. The base key does not change, but is instead mathematically modified by the IV, which changes with each payload.

The benefits of this solution are as follows:

- > Multiple plain-text messages produce entirely different ciphertexts with the same key.
- > It introduces a strongly ordered behavior to the payloads, which mitigates a commonly known attack vector called Replay Attacks.

In AES-GCM, a 96-bit IV is required.

Even with the addition of IVs, AES-GCM is still somewhat vulnerable to reusing the (Key, IV) pair for encrypting different data blocks. The total number of “uniqueness” is limited to  $(2^{96})-1$ , after which an IV is considered exhausted, and the encryption key must be replaced. The NVIDIA software stack automatically supports the replacement of the encryption key, with options for user-modification to the default policy.

All data being transferred out of the developers’ TVM or out of their provisioned GPUs is encrypted in such a manner. The “bounce buffer” mentioned in [How NVIDIA Hopper and Blackwell GPUs Integrate into a TVM’s TEE](#) contains only encrypted data. Data in the VM is stored in host memory and is encrypted and secured with facilities provided by the CPU vendors.

## Integrity

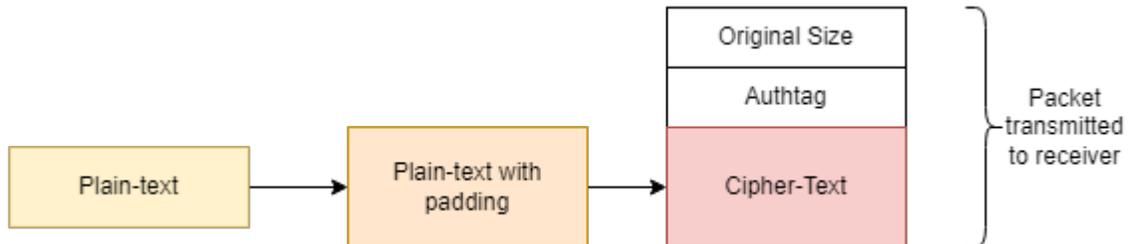
### Modifying a Payload

In a similar vein to Confidentiality, where the mitigations used to prevent an adversary from reading your data were discussed, modifying data in flight cannot be mitigated by encryption alone. It is true, especially with rolling IVs, that it is highly improbable that modifying encrypted traffic could result in a determinable outcome, but it can absolutely introduce silent errors or discrepancies that might go entirely unnoticed.

The AES-GCM algorithm also provisions the creation of a digital signature, called an AuthTag, with the ciphertext. The algorithm uses a key to sign the ciphertext and create a unique fingerprint of the payload. This method is a cryptographically secure method because it is pragmatically improbable for an adversary to modify the ciphertext, the associated AuthTag, or both items.

After receiving the payload, the receiver performs the same operation using its copy of the key to calculate its own version of the AuthTag. If both versions of the AuthTag do not match, the decryption fails, and an error occurs.

**Figure 6. The Secured Payload**



## Replaying a Message

The next logical place for an attacker to attempt to interfere with your code is something called a *Replay Attack*. This is an attack where a *man in the middle* intercepts an otherwise valid payload, and at some point, sends it again. An analogy is where an adversary has identified a payload that transmitted q, **Password accepted, continue onward and trust me**. If that payload was captured, it can be replayed at an opportune moment, which gives the adversary access to the user's password.

Again, the AES-GCM rolling IV feature comes to mitigate this threat. Before, we mentioned how a rolling IV had a secondary feature to prevent Replay-Attacks. When both endpoints of the secure channel are related, each can keep a counter of messages as the IV increments. The receiving endpoint performing the decryption does not use the IV sent by the encrypting transmitter, instead it uses its own.

Like the IV increment after encryption, the decrypting end point will increment the IV after decryption. If an adversary or malicious user tries to capture and replay, since IV has already been incremented, the decryption and AuthTag comparison will fail.

## Side-Channel Attacks

In many enterprise and hyperscale datacenters, where the number of nodes can reach into the millions, the need for remote management to access, upgrade, or debug hardware is paramount. Normally, the motherboards in these servers have a secondary light-weight coprocessor called a Baseboard Management Controller (BMC).

The BMC has its own network interface, memory, network, and other components, and through a System Management Bus (SMBus), technicians can log in to this separate BMC and obtain full system access to everything in the node:

- > Virtual desktop access, similar to plugging in a local monitor and keyboard, to the main CPU complex
- > Node hardware access and information (UEFI/BIOS updates, temperature, inventory, power control, and so forth)
- > Peripheral access and information (GPU inventory, firmware updates, temperature, performance counters, and so forth)

This side channel is extremely powerful and if access falls into the wrong hands, the effects can be catastrophic. NVIDIA Hopper and Blackwell GPUs lock out access to any paths that can access tenant data and reduce access on remaining paths to an anonymous *card health*-only level of detail.

Joint Test Action Group (JTAG ) access, which is commonly used as another method of debugging electronic components, is also disabled when the GPU is operating in Confidential Mode.

## Performance Counters

When you use NVIDIA Developer Tools (DevTools) to profile the performance and behavior of your application, special device-side hardware counters are used to help measure on-device code. However, performance counters could be used to infer the behavior of the device when in use and provide an avenue for side-channel attacks. Therefore, these performance counters are disabled in full CC-On mode. However, because profiling is critical to optimizing application performance, NVIDIA Hopper and Blackwell GPUs support a development mode called CC-Dev Tools. While the GPU is in CC-DevTools mode, all encryption paths, bounce buffers, and so forth, are enabled but access to performance counters is unblocked for developer use.

## Availability

Availability involves the overall health of the system in relation to a bad actor attempting to deny or interfere with the operation or access of another tenant, or the hypervisor itself. The NVIDIA Hopper and Blackwell GPUs and their software stacks do not require any special hooks into the hypervisor and cannot interfere with the operation of the host. The traditional methods for isolation and protection of the hypervisor from the VM are supplemented with the new CPU features (TDX and SEV-SNP) designed to isolate the VM from a potentially malicious hypervisor.

Traditionally, a GPU can access only memory locations owned by its assigned VM and addresses configured by the hypervisor into the IOMMU. TDX and SEV-SNP go a step further and permit only IOMMU-allowed transactions to access the memory pages marked as **Shared** within the CVM.

This purpose of the combined security is as follows:

- > It provides the standard protections of multiple tenants from denying service to other tenants in the system (a malicious GPU that tries to access another VM, or a malicious VM that tries to access other valid VMs or their hardware).
- > It prevents a malicious GPU from compromising the CVM's memory space that it should not access.

NVIDIA has long been integrated into the top public cloud service providers (CSPs) who provide semi-autonomous, direct access to hardware within their infrastructure. It is impractical to have a complete vetting of every potential user who might access these systems. Therefore, NVIDIA partners with these CSPs to ensure that any malicious user who uses their services cannot leave the GPU in an otherwise unusable state for the next tenant, through host crashes, PCIe-bus lockups, or physically damaging the GPU or other parts of the infrastructure. Any permanent administrative access to the GPU is locked out of in-band control.

The only out of scope availability attack is one where a malicious hypervisor prevents access to a CVM. These attacks could include removing tenant network access or disconnecting power to the GPU. While these attacks cannot have software blocks, all the private keys used to encrypt and decrypt the virtual machine and GPU memories are in key slots that cannot be accessed by the hypervisor. These keys are ephemeral and are destroyed when a VM is torn down or a GPU is reset.

# Running a Confidential Compute Application on the GPU

After the CPU TEE's trust has been extended to the GPU, running compute applications is identical to running the applications on a regular GPU. NVIDIA has worked extensively to ensure that your code works. After attestation of the GPU is completed with the correct hardware, drivers, and a passing attestation report, executing your kernels should be transparent.

## Data Flow in CC Modes

After a CVM with the GPUs has been correctly configured, booted, and attested to, you can start securely processing data on your GPU. NVIDIA has worked to ensure a *lift and shift* style of coding as far as possible. The goal is to have the existing code and kernels from users work without changes when CC modes are enabled.

By default, devices are blocked from interacting with the CVM and cannot directly access CVM memory. The remainder of this section describes how the driver enables GPUs to securely communicate with the CVM in CC mode.

Figure 7. Confidential GPU with an AMD SEV-SNP TEE

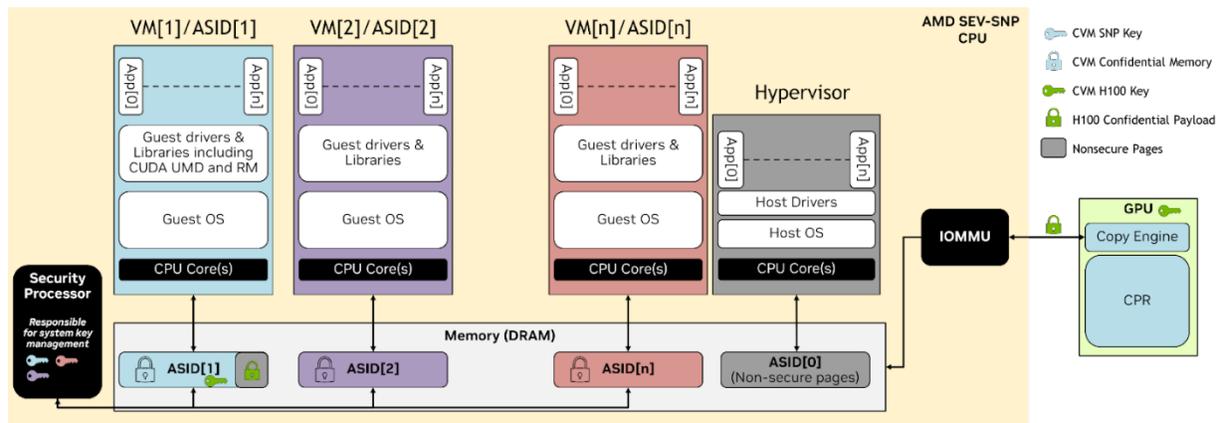
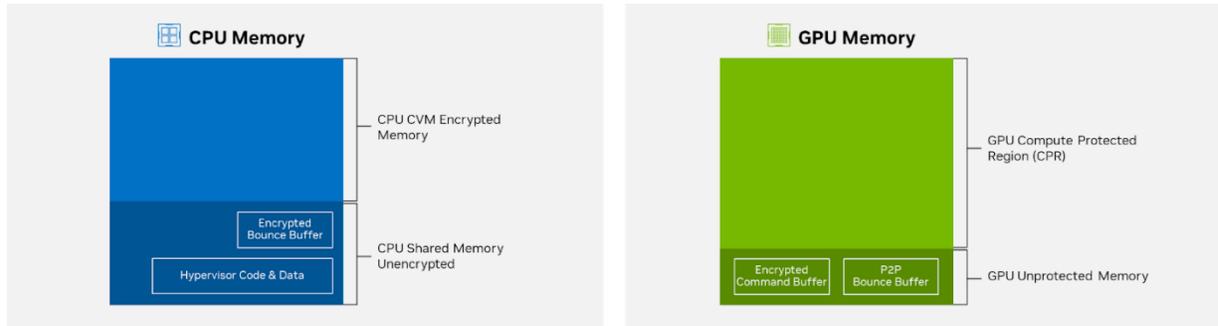


Figure 7 shows an AMD CPU with SEV-SNP, running  $n$  CVMs, one of which (VM[1]) has a GPU passed through. Because the GPU cannot directly access the memory of

CVM ASID[1] shown in light blue, all communication between the CPU and the GPU must go through a bounce buffer allocated in shared memory (represented by the gray box shown in ASID[1]).

**Figure 8. Protected Memory Regions for CPU and GPU**



NVIDIA Hopper and Blackwell GPUs have DMA engines with encryption and decryption capabilities, which are responsible for the movement of data to and from the CPU's memory. In a confidential environment, DMA engines are allowed to access shared memory pages only to retrieve and place data. To ensure the confidentiality and integrity of the payloads, models, and data, the data in these pages is encrypted and signed. These shared memory regions are called *bounce buffers* because they are used to stage the secured data before the data is transferred into the secured memory enclaves, decrypted and authenticated, and then processed.

NVIDIA has long provided developers with a solution called Unified Virtual Memory (UVM), which automatically handles page migrations between GPU memory and CPU memory based on a memory allocation API called `cudaMallocManaged()`. When the CPU accesses the data, UVM migrates the pages to CPU system memory. When the data is needed on the GPU, UVM migrates it to GPU memory. For CC, UVM is extended to employ encrypted and authenticated paging through bounce buffers in shared memory.

In the future, as an alternative to the *bounce buffer method*, inline encryption can be used with systems that have Blackwell B100/B200 GPUs and TDISP/IDE-compatible CPUs. The inline encryption method has higher performance than the bounce buffer method because it eliminates the need for the shared memory space and copying of data into it and from it.

---

# Summary

CC is a paradigm that is rapidly approaching because of regulatory regimes, privacy concerns, or the desire to accelerate other sensitive workloads. The entire computing industry recognizes the need to modify traditional thinking and security measures when operating on data. NVIDIA is at the tip of this spear, collaborating with CPU partners, cloud service providers, and ISVs to ensure that the change from traditional accelerated workloads to confidential accelerated workloads is as inevitable and transparent as the change from HTTP to HTTPS. The need to accelerate high-performance workloads will only continue to grow in line with the equivalent need to ensure that those workloads remain secure.

For more information, reach out to your NVIDIA contact, or visit the [NVIDIA Confidential Computing Forum](#).

## Notice

This document is provided for information purposes only and shall not be regarded as a warranty of a certain functionality, condition, or quality of a product. NVIDIA Corporation ("NVIDIA") makes no representations or warranties, expressed or implied, as to the accuracy or completeness of the information contained in this document and assumes no responsibility for any errors contained herein. NVIDIA shall have no liability for the consequences or use of such information or for any infringement of patents or other rights of third parties that may result from its use. This document is not a commitment to develop, release, or deliver any Material (defined below), code, or functionality.

NVIDIA reserves the right to make corrections, modifications, enhancements, improvements, and any other changes to this document, at any time without notice.

Customer should obtain the latest relevant information before placing orders and should verify that such information is current and complete.

NVIDIA products are sold subject to the NVIDIA standard terms and conditions of sale supplied at the time of order acknowledgement, unless otherwise agreed in an individual sales agreement signed by authorized representatives of NVIDIA and customer ("Terms of Sale"). NVIDIA hereby expressly objects to applying any customer general terms and conditions with regards to the purchase of the NVIDIA product referenced in this document. No contractual obligations are formed either directly or indirectly by this document.

NVIDIA products are not designed, authorized, or warranted to be suitable for use in medical, military, aircraft, space, or life support equipment, nor in applications where failure or malfunction of the NVIDIA product can reasonably be expected to result in personal injury, death, or property or environmental damage. NVIDIA accepts no liability for inclusion and/or use of NVIDIA products in such equipment or applications and therefore such inclusion and/or use is at customer's own risk.

NVIDIA makes no representation or warranty that products based on this document will be suitable for any specified use. Testing of all parameters of each product is not necessarily performed by NVIDIA. It is customer's sole responsibility to evaluate and determine the applicability of any information contained in this document, ensure the product is suitable and fit for the application planned by customer, and perform the necessary testing for the application in order to avoid a default of the application or the product. Weaknesses in customer's product designs may affect the quality and reliability of the NVIDIA product and may result in additional or different conditions and/or requirements beyond those contained in this document. NVIDIA accepts no liability related to any default, damage, costs, or problem which may be based on or attributable to: (i) the use of the NVIDIA product in any manner that is contrary to this document or (ii) customer product designs.

No license, either expressed or implied, is granted under any NVIDIA patent right, copyright, or other NVIDIA intellectual property right under this document. Information published by NVIDIA regarding third-party products or services does not constitute a license from NVIDIA to use such products or services or a warranty or endorsement thereof. Use of such information may require a license from a third party under the patents or other intellectual property rights of the third party, or a license from NVIDIA under the patents or other intellectual property rights of NVIDIA.

Reproduction of information in this document is permissible only if approved in advance by NVIDIA in writing, reproduced without alteration and in full compliance with all applicable export laws and regulations, and accompanied by all associated conditions, limitations, and notices.

THIS DOCUMENT AND ALL NVIDIA DESIGN SPECIFICATIONS, REFERENCE BOARDS, FILES, DRAWINGS, DIAGNOSTICS, LISTS, AND OTHER DOCUMENTS (TOGETHER AND SEPARATELY, "MATERIALS") ARE BEING PROVIDED "AS IS." NVIDIA MAKES NO WARRANTIES, EXPRESSED, IMPLIED, STATUTORY, OR OTHERWISE WITH RESPECT TO THE MATERIALS, AND EXPRESSLY DISCLAIMS ALL IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY, AND FITNESS FOR A PARTICULAR PURPOSE. TO THE EXTENT NOT PROHIBITED BY LAW, IN NO EVENT WILL NVIDIA BE LIABLE FOR ANY DAMAGES, INCLUDING WITHOUT LIMITATION ANY DIRECT, INDIRECT, SPECIAL, INCIDENTAL, PUNITIVE, OR CONSEQUENTIAL DAMAGES, HOWEVER CAUSED AND REGARDLESS OF THE THEORY OF LIABILITY, ARISING OUT OF ANY USE OF THIS DOCUMENT, EVEN IF NVIDIA HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. Notwithstanding any damages that customer might incur for any reason whatsoever, NVIDIA's aggregate and cumulative liability towards customer for the products described herein shall be limited in accordance with the Terms of Sale for the product.

## Trademarks

NVIDIA, the NVIDIA logo, are trademarks and/or registered trademarks of NVIDIA Corporation in the U.S. and other countries. Other company and product names may be trademarks of the respective companies with which they are associated.

## VESA DisplayPort

DisplayPort and DisplayPort Compliance Logo, DisplayPort Compliance Logo for Dual-mode Sources, and DisplayPort Compliance Logo for Active Cables are trademarks owned by the Video Electronics Standards Association in the United States and other countries.

## HDMI

HDMI, the HDMI logo, and High-Definition Multimedia Interface are trademarks or registered trademarks of HDMI Licensing LLC.

## Arm

Arm, AMBA, and ARM Powered are registered trademarks of Arm Limited. Cortex, MPCore, and Mali are trademarks of Arm Limited. All other brands or product names are the property of their respective holders. "Arm" is used to represent ARM Holdings plc; its operating company Arm Limited; and the regional subsidiaries Arm Inc.; Arm KK; Arm Korea Limited.; Arm Taiwan Limited; Arm France SAS; Arm Consulting (Shanghai) Co. Ltd.; Arm Germany GmbH; Arm Embedded Technologies Pvt. Ltd.; Arm Norway, AS, and Arm Sweden AB.

## OpenCL

OpenCL is a trademark of Apple Inc. used under license to the Khronos Group Inc.

## Copyright

© 2025 NVIDIA Corporation & Affiliates. All rights reserved.