Cisco Confidential Computing Overview

Version 1.0

November 2025

Contents

Intended use and audience

This document contains confidential material that is proprietary to Cisco Systems, Inc. The materials, ideas, and concepts contained herein are to be used exclusively to assist in the configuration of Cisco® software solutions.

Bias statement

The documentation set for this product strives to use bias-free language. For purposes of this documentation set, bias-free is defined as language that does not imply discrimination based on age, disability, gender, racial identity, ethnic identity, sexual orientation, socioeconomic status, and intersectionality. Exceptions may be present in the documentation due to language that is hardcoded in the user interfaces of the product software, language that is based on standards documentation, or language that is used by a referenced third-party product.

Legal notices

All information in this document is provided in confidence and shall not be published or disclosed, wholly or in part, to any other party without Cisco's written permission.

Prerequisites

We recommend reviewing Cisco UCS® release notes, installation guides, and user guides before proceeding with any configuration. Please contact Cisco support or your Cisco representative if you need assistance.

1. Executive summary

Confidential computing represents the next evolution in data security. It accomplishes this by protecting information not only at rest and in transit, but also while it is being actively processed. Using Trusted Execution Environments (TEEs) built into modern CPUs and GPUs, confidential computing creates secure environments that keep data encrypted and isolated. The data is isolated from hypervisors, from bad actors, and even from system administrators.

Cisco UCS servers provide a robust confidential computing foundation for enterprise data centers and cloud deployments. They combine multiple hardware roots of trust (mainboard, BMC, add-in cards, etc.), secure boot, and platform attestation with support for technologies such as Intel® SGX, Intel TDX, AMD SEV-SNP, ARM TrustZone, and NVIDIA GPU-CC (NVIDIA GPU Confidential Computing). This integration allows organizations to run sensitive workloads in a verifiable (attestable), tamper-resistant environment while maintaining centralized policy control through Cisco Intersight®.

Confidential computing is increasingly vital for businesses operating in regulated industries, pursuing zero-trust security strategies, or handling intellectual property and customer data in shared or cloud-hosted multitenant environments. It mitigates insider threats, strengthens compliance posture, and enables secure collaboration and privacy-preserving analytics.

By adopting confidential computing on Cisco UCS, organizations can protect their data through its entire lifecycle: at rest, in transit, and now, in use. This closes a longstanding gap in enterprise security and enables a secure, trusted platform for workloads and data.

2. Introduction

The modern data center operates in an environment where trust must be continuously verified rather than assumed. Cyberattacks have evolved beyond perimeter breaches to target firmware, runtime memory, and shared infrastructure. These are places where traditional encryption and access controls are no longer sufficient. To meet these challenges, Cisco has developed a trusted computing paradigm. This is a security framework that anchors every layer of the infrastructure in measurable and verifiable trust and integrity.

At the heart of this paradigm is the Cisco Unified Computing System™ (Cisco UCS), a platform designed from the ground up to integrate compute, networking, and security under a unified management model. Cisco UCS extends trusted computing principles beyond secure boot and firmware validation to encompass the entire data lifecycle, protecting data at rest, in transit, and now, with confidential computing, data in use.

Confidential computing leverages **Trusted Execution Environments (TEEs)** built into modern CPUs and GPUS from Intel, AMD, and NVIDIA. TEEs isolate sensitive workloads and cryptographically protect data during active processing. This ensures that even privileged software, hypervisors, or system administrators cannot access protected data.

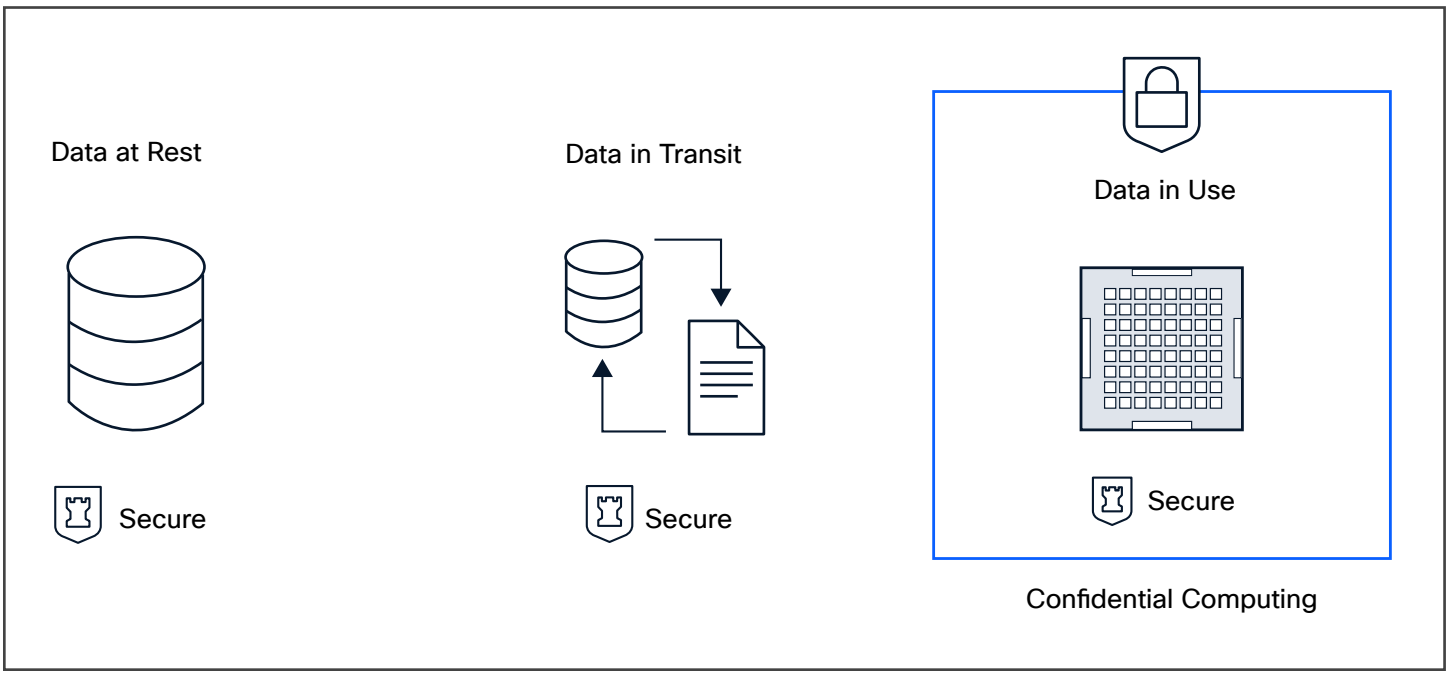

Figure 1. Confidential computing and data in use

By combining the Cisco Trusted Computing Paradigm with confidential computing technologies, Cisco UCS platforms deliver hardware-based assurance of workload integrity and data privacy. This approach enables organizations to safely process sensitive information in hybrid- and multicloud environments, meet stringent regulatory compliance requirements, and ensure secure operation in a verifiable way.

Cisco UCS confidential computing extends beyond hardware protection—it integrates with Cisco Intersight and complementary technologies such as Cisco Hyperfabric™ and Cisco Hypershield™, along with platform attestation services, to deliver end-to-end visibility, automation, and enforcement. Together, these capabilities form a trusted foundation for applications in sectors such as finance, healthcare, and government, where data protection and verifiability are required.

What is a trustworthy system?

A trustworthy system is set of products and solutions with multilayered security that is provably secure. Confidential computing is a core feature of this at the application execution level.

By now, most enterprises are familiar with securing Data-at-Rest (D@RE) using tried and true hardware technologies such as Self-Encrypting Drives (SEDs) or software technologies such as transparent encryption clients (for example, BitLocker). Most are also familiar with using secure transport methods such as IPsec-based border connections, encrypted management sessions using HTTPS, and secure client access with VPNs. These represent protected data in transit. This is also combined with robust RBAC user roles and controls to ensure the right people and processes are doing the right things with the right resources. Confidential computing rounds out this offering by eliminating the trustworthy system's execution vulnerabilities.

Confidential computing with Cisco UCS is complementary with our other trustworthy technologies:

- Secure supply chain
- Counterfeit protection
- Image signing and secure boot
  - Cisco Trust Anchor module (TAm)
  - TPM
- Runtime Defenses (RTDs)
- Cisco Hyperfabric
- Cisco Hypershield

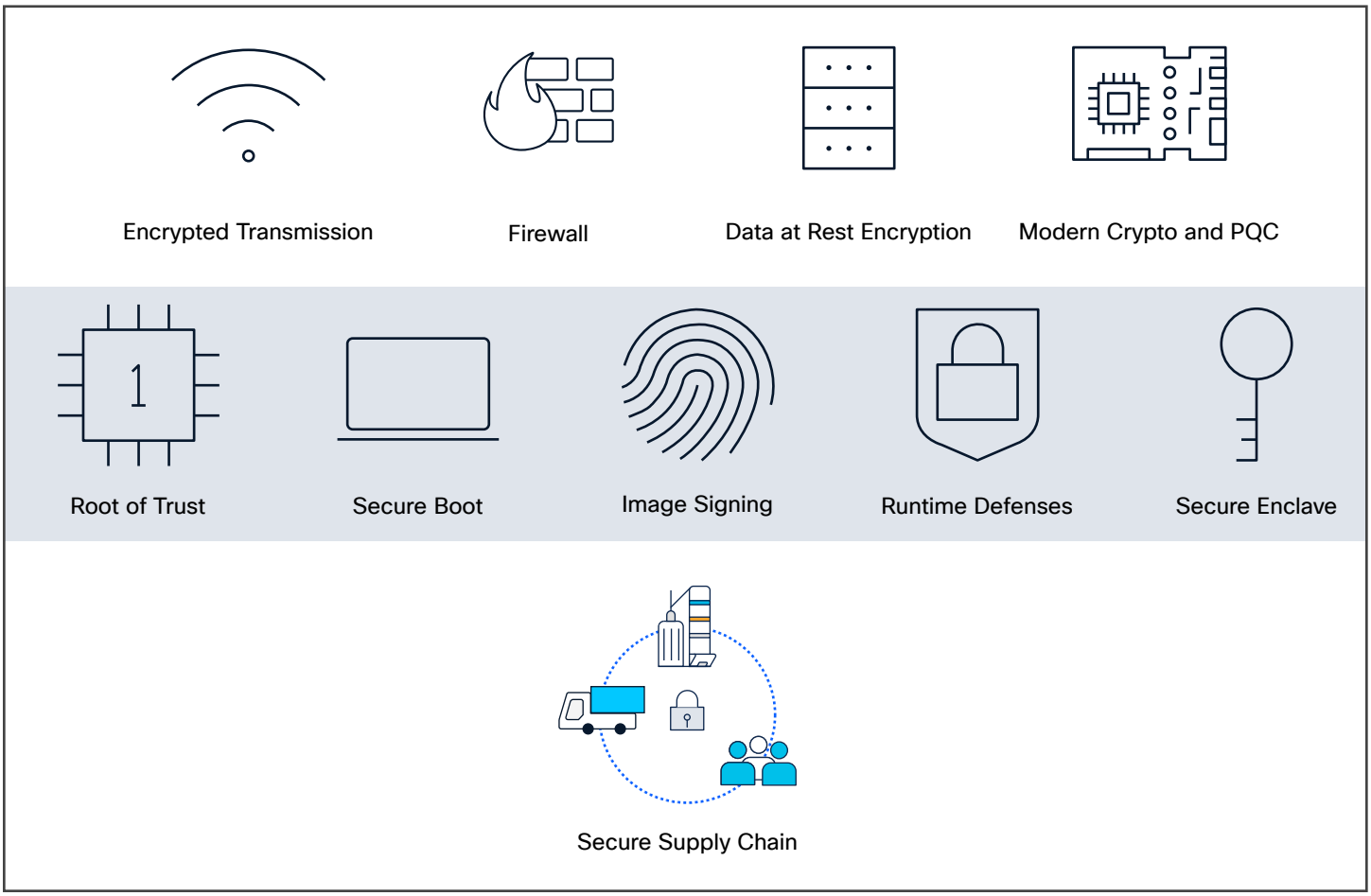

Figure 2. The components of Cisco UCS trustworthy computing

These establish a chain of trust before application execution even begins.

3. Drivers for confidential computing

The challenge: protecting data in use

Organizations have made significant investments in encrypting data at rest (for example, on disk) and in transit (for example, TLS/SSL). However, data must be decrypted when being processed by applications, creating a window of vulnerability where:

- **Insiders or privileged administrators** can access sensitive data in memory.
- **Malware or rootkits** can scrape memory contents or inject malicious code.
- **Cloud hypervisors or other tenants** in shared infrastructure can exploit vulnerabilities to view another customer's data.

This "last mile" of data security has historically been difficult to address without impacting performance or operational flexibility.

Rising threat landscape

Advanced Persistent Threats (APTs) and nation-state actors increasingly target runtime memory to exfiltrate secrets, cryptographic keys, or customer data. Confidential computing offers a hardware boundary that even a fully compromised host OS cannot bypass.

Cloud and multitenant adoption

As more organizations shift sensitive workloads to public clouds, concerns about shared infrastructure risks grow. Confidential computing provides cryptographic assurance, through attestation services, that data and workloads are isolated from the cloud provider and other tenants.

Secure AI workloads

As dedicated AI systems for inference and RAG (Retrieval Augmented Generation) become widely available and adopted, ensuring privacy for the data you analyze using open AI resources becomes more important. This enables secure processing of sensitive datasets, protects proprietary AI models, and facilitates collaboration on data without compromising privacy, which is vital for regulated industries such as healthcare and finance. This is particularly relevant and effective when you run AI applications and datasets in confidential VMs or containers.

Regulatory pressure and compliance

Industries such as finance, healthcare, and government face strict requirements to prove that data is protected throughout its lifecycle. Confidential computing strengthens compliance with such standards as HIPAA, PCI-DSS, GDPR, and FedRAMP, making it easier to demonstrate control during audits.

Secure collaboration and data sharing

In scenarios where multiple parties must share and process data collaboratively, such as multi-institution research or financial-crime analysis, confidential computing enables privacy-preserving analytics. Partners can compute on shared data without exposing their raw datasets to one another.

4. Benefits of confidential computing across key industries

Finance

Financial services organizations face some of the most stringent data-protection and regulatory requirements. Confidential computing helps banks, insurers, and fintech companies process sensitive data such as transactions, customer records, and fraud-detection models inside TEE's. This reduces the risk of insider threats, memory scraping attacks, and malicious code injection that could expose confidential data. It also enables privacy-preserving analytics—allowing financial institutions to share risk data or AML (Anti-Money Laundering) intelligence with partners securely, without disclosing raw customer information. By protecting data in use, financial institutions can meet regulatory demands such as PCI-DSS, SOX, and PSD2 while maintaining trust with customers and auditors.

Healthcare

In healthcare, patient privacy and compliance with regulations such as HIPAA are paramount. Confidential computing provides a trusted environment for processing electronic health records, genomic data, and AI-driven diagnostic models without risk of unauthorized access by administrators, cloud providers, or attackers. It enables multi-institutional research collaborations, allowing hospitals, pharmaceutical companies, and research labs to run analytics on shared datasets securely. This makes it possible to accelerate medical discoveries and improve patient outcomes while keeping sensitive health data private and compliant with legal requirements.

Cloud computing and multitenant environments

Cloud adoption continues to accelerate, but many organizations remain cautious about moving sensitive workloads, because of shared infrastructure risks. Confidential computing addresses this challenge by isolating workloads at the hardware level, ensuring that even a compromised hypervisor or rogue cloud operator cannot access customer data. This significantly strengthens the security posture for multitenant environments and makes the cloud a viable option for workloads previously kept on premises. For managed service providers and SaaS vendors, it also enables stronger guarantees to their customers regarding data privacy and regulatory compliance, which can be a competitive differentiator.

5. Trusted Execution Environments (TEEs) vs. traditional security models

Traditional security models

Organizations have long relied on two primary mechanisms to protect data:

- **Encryption at rest** – protects stored data (on disk or in databases) by encrypting it so that only authorized systems can decrypt it. This mitigates risks from physical theft, lost devices, or unauthorized access to storage systems.
- **Encryption in transit** – protects data as it moves between systems (for example, over networks) using protocols such as TLS/SSL or IPsec. This defends against eavesdropping, man-in-the-middle attacks, and interception of sensitive traffic.

Together, these techniques secure data in two of its three states. But there's a critical third state: when data is actively being processed by applications.

The gap: data in use

When data is loaded into memory for processing, it must be decrypted. At this moment, data becomes vulnerable to:

- **Memory scraping malware or rootkits** that can read application memory
- **Compromised operating systems or hypervisors** with privileged access
- **Malicious insiders** with administrative credentials
- **Side-channel attacks** targeting CPU cache or speculative execution vulnerabilities

Traditional security models do not address this exposure. Encryption stops at the application boundary, leaving data in plaintext inside system memory.

Trusted Execution Environments (TEEs)

A Trusted Execution Environment (TEE) is a secure, isolated area of a processor where code and data can be executed with hardware-level protection. “Trusted execution environment” is a general term that includes a secure enclaves and its various capabilities. TEE’s and Secure Enclaves are casually used interchangeably, but Secure Enclaves are strictly an Apple technology for confidential computing. This paper will stick with TEE’s for the execution environment.

Applications process data, and to do this, they interface with a computer’s memory. Before an application can process (encrypted) data, it goes through decryption in memory. Because the data is, for a moment, unencrypted, it is left exposed. It can be accessed, encryption-free, right before, during, and right after it has been processed. This leaves it exposed to such threats as memory dump attacks, which involve capturing and using Random Access Memory (RAM) put on a storage drive in the event of an unrecoverable error.

The attacker triggers this error as part of the attack, forcing the data to be exposed. Data is also exposed to root user compromises, which occur when the wrong person gains access to administrator privileges and can therefore access data before, during, and after it has been processed.

Confidential computing fixes this issue by using a hardware-based architecture TEE. The hardware portion is a secure coprocessor inside a CPU and GPU. Embedded encryption keys are used to secure the TEE. To make sure the TEEs are only accessible to the application code authorized for it, the coprocessor uses attestation mechanisms that are embedded within the application. If the system comes under attack by malware or unauthorized code as it tries to access the encryption keys, the TEE will deny the attempt at access and cancel the computation.

This allows sensitive data to stay protected while in memory. When the application tells the TEE to decrypt it, the data is released for processing. While the data is decrypted and being processed by the computer, it is invisible to everything and everyone else. This includes the cloud provider, other computer resources, hypervisors, virtual machines, and even the operating system.

TEEs ensure:

- **Isolation:** Memory inside the TEE is encrypted and inaccessible to the OS, hypervisor, or other workloads.
- **Integrity:** The hardware verifies that only trusted code is loaded, preventing tampering.
- **Attestation:** TEEs can produce cryptographic proof (attestation reports) that they are running genuine code in a secure state.

Examples include Intel SGX, Intel TDX, AMD SEV-SNP, NVIDIA GPU-CC, and ARM TrustZone.

Intel and AMD offer different technologies and approaches to achieve confidential computing TEEs within their respective processor architectures. Follow-up sections in this paper detail a comparison between Intel's technologies (SGX, TDX, and TME) and AMD's features (SEV and SME) in terms of their approaches, functionalities, and key characteristics.

How TEEs complement traditional security

By protecting data in use, TEEs complete the "full data lifecycle" security model.

Table 1. The full data lifecycle

State of Data	Traditional Protection	TEE Protection
At rest	Disk/database encryption	N/A (unchanged)
In transit	TLS, IPsec, VPNs	N/A (unchanged)
In use	None (vulnerable)	Encrypted memory, isolated execution, attestation

This creates a three-layered defense model where data is protected end-to-end—from storage, through network transfer, to active processing.

The concept of the TEE is primarily focused on maintaining the confidentiality, integrity, and privacy of sensitive information, especially when dealing with critical data or executing sensitive operations. These environments use hardware-based security mechanisms to create isolated and trusted spaces within the system's memory or processing units, offering a high level of protection against various types of attacks, including those attempting to access or manipulate the TEE's contents. The contents of the TEE, the data being processed and the techniques used to process it, are accessible only to authorized programming code and are invisible and unknowable to anything or anyone else.

The TEE's isolation can generally take one of two forms, and sometimes both. A VM or container can be run in an isolated TEE, basically as an entire self-contained system, or an application can be wholly run inside a TEE provisioned by a server or VM. The illustration below shows both of these cases.

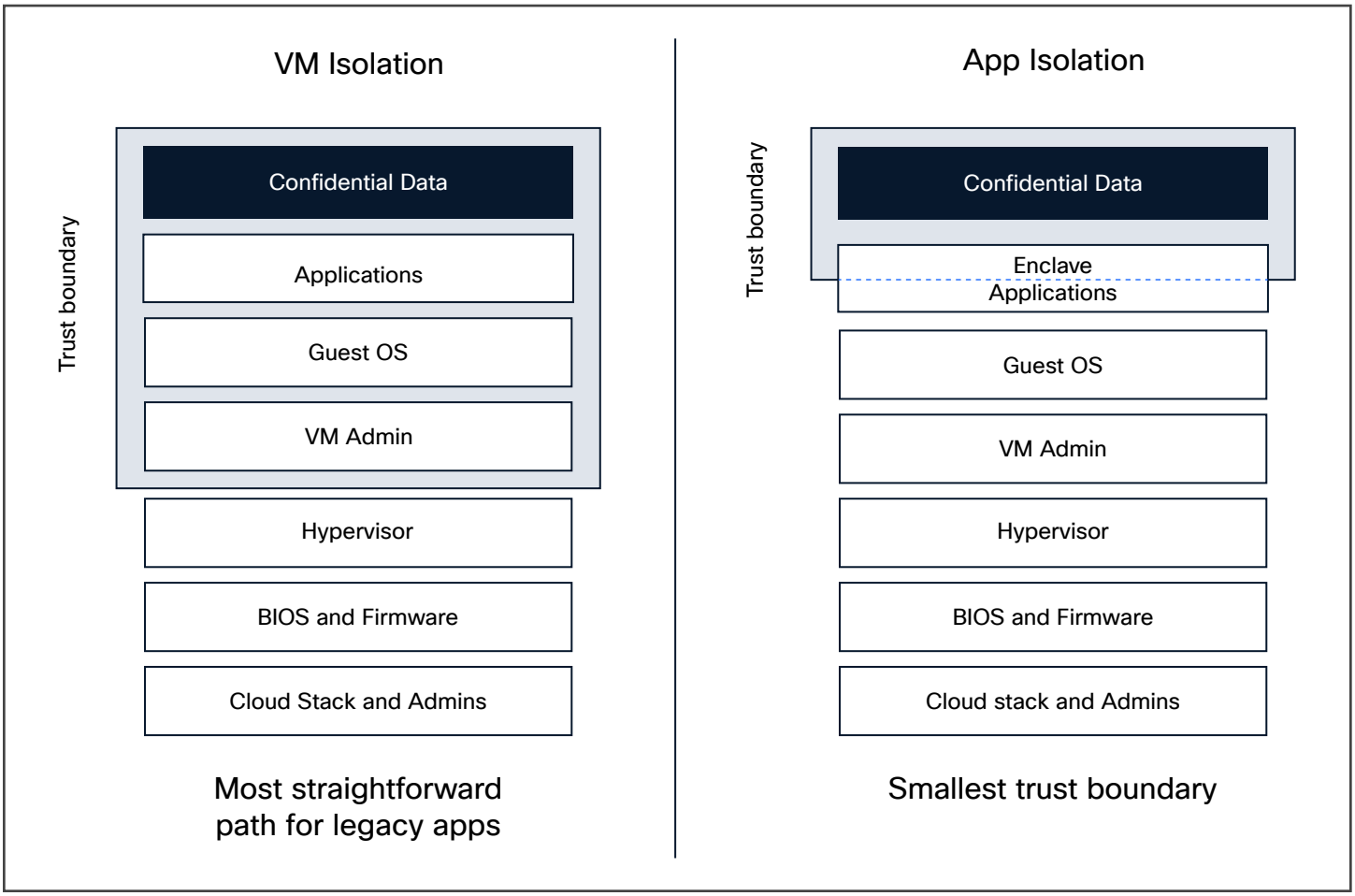

Figure 3. TEE isolation

Intel technologies

Intel Software Guard Extensions (Intel SGX)

Intel Software Guard Extensions (Intel SGX) creates isolated and TEE's within the CPU's memory, allowing applications to protect sensitive code and data. This enables developers to create isolated execution environments for applications, protecting data and code even from higher-privileged software layers. It provides memory encryption, secure execution, remote attestation, and isolation.

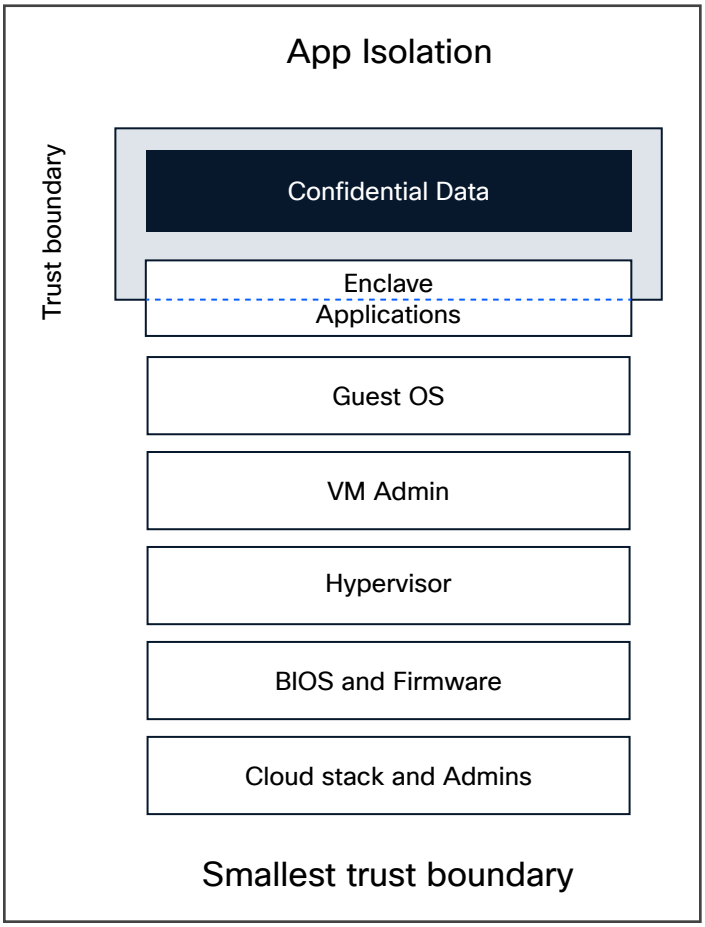

Figure 4. Intel SGX application isolation

How Intel SGX works:

- **TEE creation:** Developers mark sensitive code and data to be executed or stored inside the TEE.
- **Hardware protection:** The CPU ensures that memory inside the TEE is:
  - **Encrypted** and stored in a reserved area called the **EPC (Enclave Page Cache)**
  - **Inaccessible** to any code outside the TEE—including OS, hypervisor, BIOS, or other TEE's
- **Remote attestation:** Before you trust a TEE, you can verify it through **remote attestation** to check it's running the right code.
- **Secure execution:** TEE's run in isolation. Even if the OS is compromised, it can't read or modify TEE memory.

Use cases:

- Password management
- Digital Rights Management (DRM)
- Secure computation on untrusted infrastructure (for example, cloud, AI workloads)
- Blockchain wallets

Figure 5 depicts an attacker unable to penetrate a confidential computing application and its data in a VM after compromising the parent hypervisor.

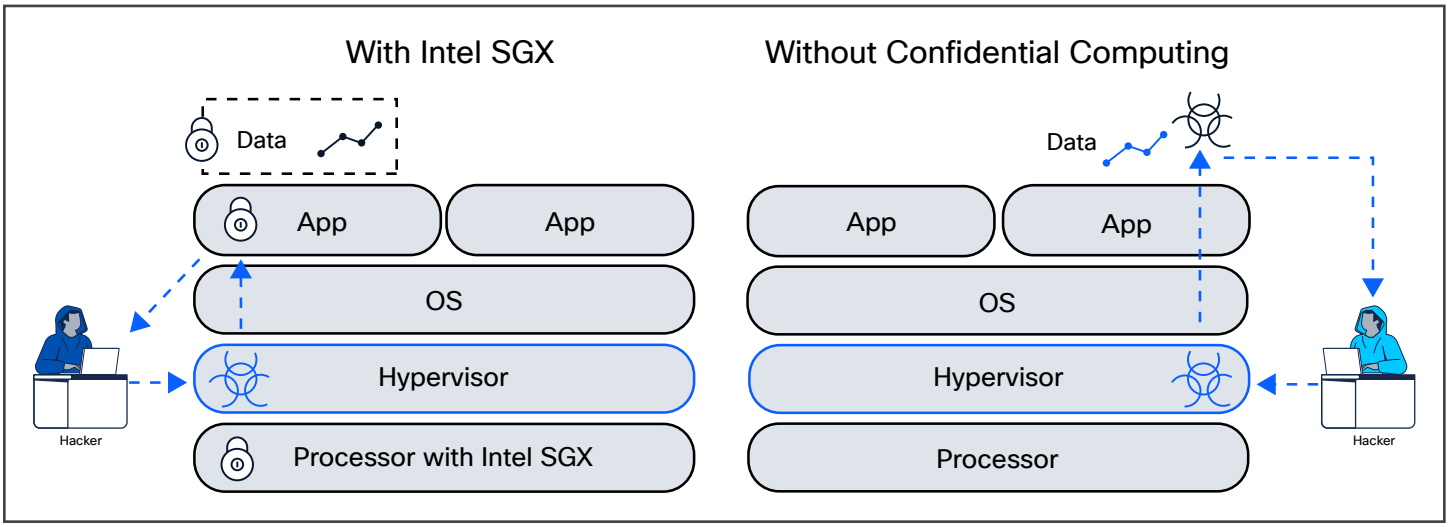

Figure 5. Intel SGX application exfiltration defense

Enabling Intel SGX support on Cisco UCS is a matter of setting the appropriate BIOS tokens in a system policy. Figure 6 shows the relevant tokens in a policy creation using Cisco Intersight for an Intersight Managed Mode (IMM) Cisco UCS server.

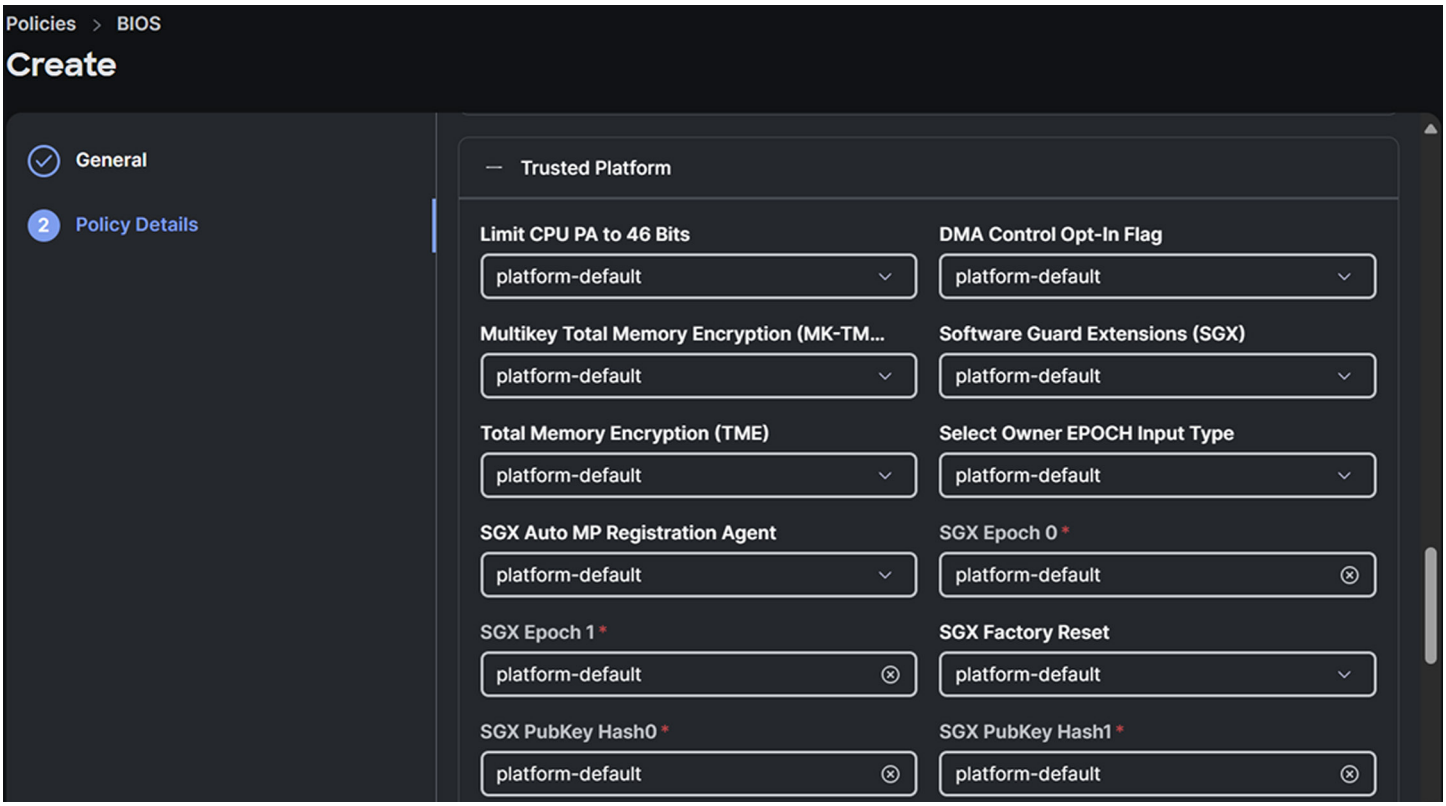

Figure 6. Intel SGX BIOS tokens screen in Cisco Intersight

Intel Total Memory Encryption, Intel TDX, and Intel Trusted Execution Technology (Intel TXT)

Intel Total Memory Encryption (Intel TME) focuses on enhancing security in virtualized environments by providing memory encryption and secure execution environments for virtual machines. It provides total memory encryption, secure boot processes, and hardware-based isolation to protect against attacks in virtualized environments.

TDX protects VMs from unauthorized access and tampering, ensures secure migrations, and provides a trusted execution environment.

Intel Trusted Execution Technology (TXT) is a set of hardware extensions designed to enhance security in computing systems. It provides a hardware-based foundation for security, helping to validate platform trustworthiness during boot and launch, and addressing evolving security threats across physical and virtual infrastructures. Intel TXT aims to ensure the authenticity of the platform and its operating system, thereby enabling reliable evaluation of the computing environment. Intel TXT is part of a larger platform security capability and is beyond the scope of this discussion.

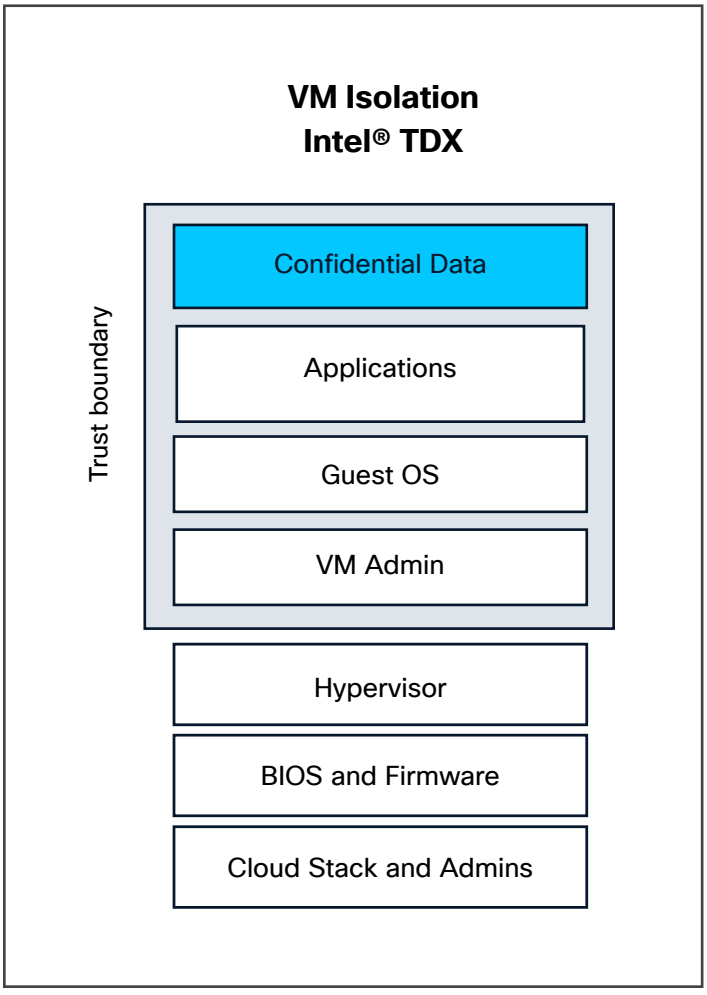

Figure 7. Intel TDX VM isolation

How Intel TDX works:

- **TD creation:** A virtual machine launches as a **Trust Domain (TD)**.
- **Isolation by hardware:** Intel TDX hardware provides a **TD-scope memory encryption and access control**:
  - The **hypervisor can't see inside the TD's memory**.
  - Only the **TD's CPU context** can access its own memory.
- **Secure boot and attestation:** Like Intel SGX, Intel TDX supports **TD attestation**, allowing remote parties to verify that the VM is in a trusted state.
- **Intel TDX module:** A special firmware (running below the hypervisor) manages the creation and lifecycle of trust domains.

Use cases:

- Running VMs securely on cloud infrastructure you don't fully trust
- Confidential multitenant workloads
- Secure containers and VMs in zero-trust environments

Table 2 shows the key differences between Intel SGX and Intel TDX.

Table 2. Intel SGX and Intel TDX comparison

Feature	Intel SGX	Intel TDX
Scope	Specific application components	Full Virtual Machines (VMs)
OS/hypervisor access	No access to TEE memory	No access to TD memory
Developer involvement	High (code changes needed)	Low (runs regular OS/VMs)
Memory model	Limited EPC (~128MB usable)	Much larger VM memory supported
Typical use case	Application-level secrets	Cloud VM confidentiality

Intel Total Memory Encryption (Intel TME)

Intel Total Memory Encryption (Intel TME) encrypts system memory to safeguard against unauthorized access, ensuring data confidentiality even if an attacker gains physical access to the memory. It protects system memory contents through encryption, ensuring data confidentiality and integrity. Intel TME aims to prevent data breaches and unauthorized access to memory contents.

Intel TME encrypts all data passing between the CPU and RAM using a hardware-generated single key, thus preventing unauthorized access to memory contents even if the memory is physically removed. This is achieved by using a dedicated AES-XTS encryption engine on the memory controller to encrypt data on external buses and decrypt it inside the CPU. The key is transient, never exposed to software, and generated by a hardware random number generator during boot.

How Intel TME works:

- **Key generation:** During the boot process, the CPU's internal hardware generates a unique, transient encryption key using a hardened random number generator. This key is never exposed to software.
- **Encryption/decryption:** All data moving to and from the CPU is encrypted by hardware on the memory bus before it leaves the CPU and decrypted after it returns. Data is only in plaintext while it is inside the CPU's internal caches.
- **Encryption algorithm:** The encryption is done using the NIST-standard AES-XTS algorithm, with the key being unique for each cache block, because the physical address is part of the encryption process.
- **Protection against attacks:** This process protects against physical attacks such as cold boot attacks, where an attacker removes a running system's memory to read its contents. Since the data is encrypted, it appears as garbage without the correct key.
- **No software modification:** Because the encryption/decryption is handled in the hardware, existing operating systems and applications can run without modification.

Intel Multi-Key Total Memory Encryption (Intel MKTME)

A more advanced version of Intel TME, Intel Multi-Key Total Memory Encryption (Intel MKTME) allows for multiple keys to be used at page granularity, giving software the capability to use different keys for different memory regions, though it still uses the primary Intel TME key by default unless software specifies otherwise.

AMD technologies

AMD Secure Encrypted Virtualization (AMD SEV)

AMD Secure Encrypted Virtualization (AMD SEV) focuses on enhancing security in virtualized environments by providing hardware-based memory encryption for VMs. It offers memory encryption for each VM, isolating VMs from each other and from the hypervisor and protecting VMs from attacks in cloud environments. AMD SEV provides memory encryption and isolation and facilitates secure VM migrations between physical hosts.

AMD SEV provides the following benefits:

- Hardware-based security features designed to protect VMs include:
  - Protection from untrusted hypervisor(s)
  - Protection from other VMs running on the same physical
  - Protection from access to data
    - Ensures data privacy
    - Major use case: multitenant cloud environments

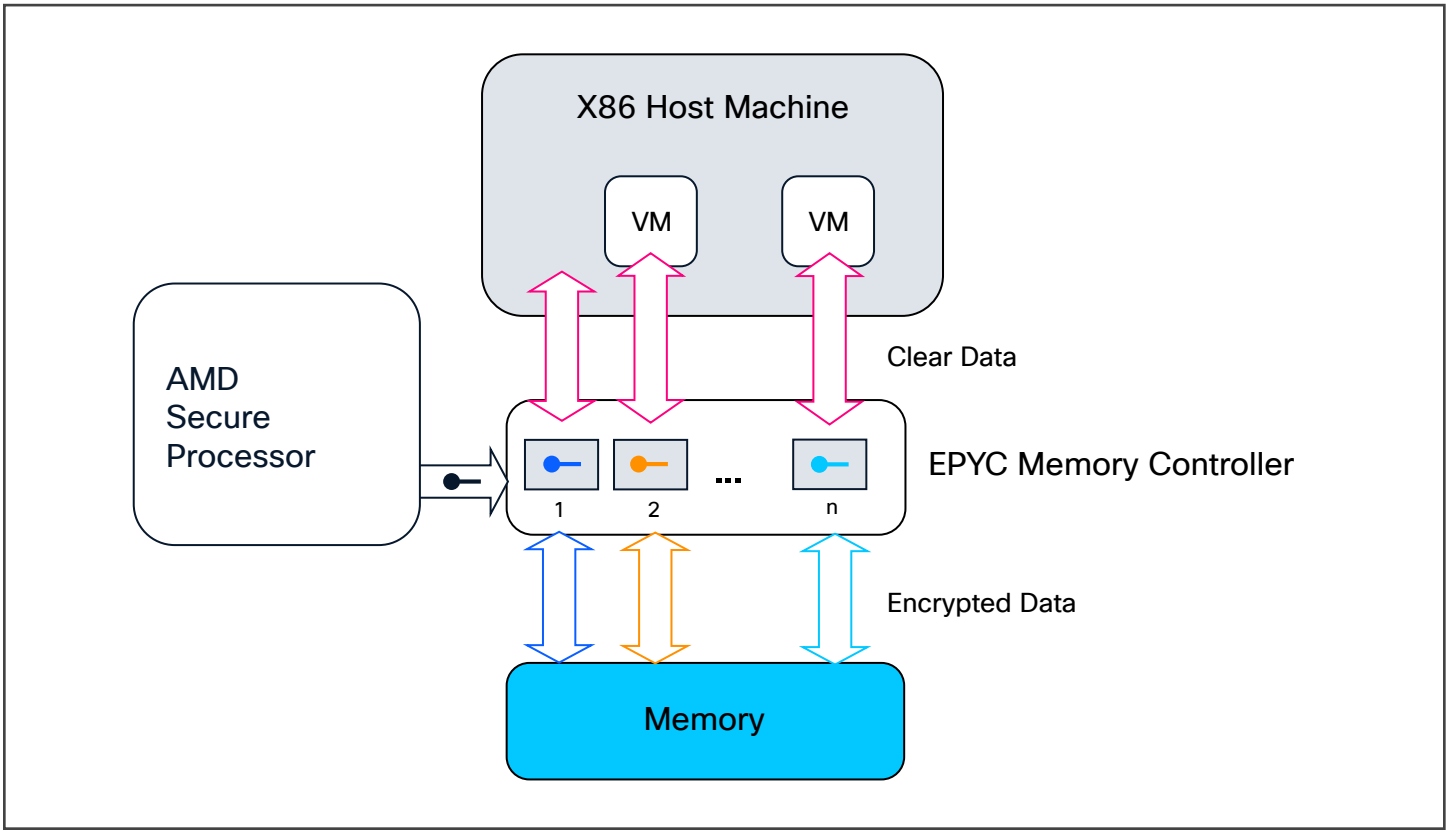

Figure 8. AMD EPYC processor confidential computing encryption workflow

Programs and OSes running inside AMD SEV-encrypted VMs do **not** need to be modified or recompiled. The encryption is handled entirely by the hardware and hypervisor and is transparent to the guest.

AMD SEV is a hardware-based security feature that encrypts Virtual Machine (VM) memory, making it inaccessible to the host operating system or other VMs. This isolation protects data during processing and ensures data privacy, especially in multitenant cloud environments.

How AMD SEV works:

- AMD Secure Processor (AMD-SP):
  - A dedicated hardware component (part of AMD's Platform Security Processor) that generates and handles keys
- Memory encryption engine:
  - Built into memory controller, the memory encryption engine encrypts/decrypts memory on-the-fly as data moves between RAM and the CPU.
- VM launch:
  - When a VM is started, the hypervisor requests the AMD-SP to create a unique encryption key for that VM.
  - The key is stored only in the secure processor and is never exposed to the hypervisor or guest OS.
- Encrypted memory:
  - The memory pages allocated to the VM are encrypted using AES-128 with the VM-specific key.
  - Only the CPU can decrypt and execute the memory contents.
  - Any attempt to read VM memory is useless
- Register state encryption (AMD SEV-ES):
  - AMD SEV-ES encrypts the CPU register state when a VM stops running, preventing leakage of sensitive information to the hypervisor.

Use case:

- Lift and shift existing workloads into AMD SEV-enabled cloud environments such as Azure Confidential VMs without code changes.

AMD Secure Encrypted Virtualization – Secure Nested Paging (AMD SEV-SNP)

AMD Secure Encrypted Virtualization – Secure Nest Paging (AMD SEV-SNP) builds upon AMD SEV by adding memory-integrity protection, further hardening VM isolation against hypervisor-based attacks.

AMD SEV-SNP:

- Adds memory-integrity protections to AMD SEV
- Protects entire VMs, not user-level TEE's
- Provides confidentiality and integrity
- Supports remote attestation, similar to Intel SGX
- Great for cloud-based workloads needing isolation from the hypervisor

Use case:

- A cloud tenant wants to run a secure VM on an untrusted host (for example, Azure Confidential VMs with AMD SEV-SNP).

AMD SEV and its variants are enabled on Cisco UCS by selecting the appropriate BIOS tokens and setting them in a BIOS policy. These policies are then assigned to a service profile and applied to the server. Table 3 shows the relevant tokens.

Table 3. AMD BIOS tokens to enable confidential computing memory protections

Memory		
BME DMA Mitigation	Disable	Allows you to disable the PCI BME bit to mitigate the threat from an unauthorized external DMA.
CPU SMEE	Enable	Whether the processor uses the Secure Memory Encryption Enable (SMEE) function, which provides memory encryption support.
DRAM Scrub Time	24 hours	The value that represents the number of hours to scrub the whole memory.
SNP Memory Coverage	Enable	This option selects the operating mode of the Secured Nested Paging (SNP) Memory and the Reverse Map Table (RMP). The RMP is used to ensure a one-to-one mapping between system physical addresses and guest physical addresses.
SNP Memory Size to Cover in MiB	8192	Allows you to configre SNP memory size.
SEV-SNP Support	Enable	Allows you to enable the Secure Nested Paging feature.

Memory		
Secured Encrypted Virtualization	Enable	Enables running encypted Virtual Machines (VMs) in which the code and data of the VM are isolated.
SMEE	Enable	Whether the processor uses the Secure Memory Encryption Enable (SMEE) function, which provides memory encryption support.
Transparent Secure Memory Encryption	Enable	Provides transparent hardware memory encryption of all data stored on system memory.

Table 4. AMD BIOS tokens to enable confidential computing CPU protections

Processor		
Transparent Secure Memory Encryption (TSME)	Enable	Provides hardware memory encryption of all the data stored on system DIMMs that is invisible to the OS and slightly increases the memory latency. C245 M8 only.
SVM Mode	Enable	Whether the processor uses AMD Seucre Virtual Machine Technology. C225 M6, C245 M6, C245 M8 only.

Benefits of AMD SEV in confidential computing

AMD SEV ensures that sensitive data is protected even while being processed within a VM. The isolation provided by SEV prevents unauthorized access to data by cloud operators, malicious administrators, and even privileged software. This makes AMD SEV and its memory encryption mechanisms ideal for running sensitive workloads that require a high level of confidentiality, such as those in healthcare, financial services, and data analytics. These safeguards within AMD SEV help organizations comply with strict data privacy regulations by protecting data during processing.

How AMD SEV is used in the cloud

Various cloud-computing providers, such as Google Cloud, Microsoft Azure, and Oracle Cloud Infrastructure (OCI), offer confidential VMs powered by AMD SEV technology. AMD SEV is used in these instances as Infrastructure as a Service (IaaS) deployments, enabling users to run confidential VMs in the cloud. AMD SEV can be used to protect databases and their associated workloads, ensuring data confidentiality even while the database is in use. AMD SEV is also used in other cloud services, such as confidential container instances, and in confidential Kubernetes clusters.

AMD SEV is a crucial technology for confidential computing in AMD environments, providing a hardware-based foundation for isolating and protecting sensitive data during processing in virtualized environments.

AMD Secure Memory Encryption (AMD SME)

AMD Secure Memory Encryption (AMD SME) encrypts the system's memory, protecting against unauthorized access and physical attacks by encrypting memory contents. It encrypts system memory contents transparently without requiring specific software modifications, thus protecting against memory snooping attacks. AMD SME protects memory contents through encryption, enhancing security against physical attacks.

Intel and AMD TEEs compared

Both Intel and AMD technologies aim to provide hardware-based security mechanisms to protect sensitive data and create secure TEE's. Intel SGX and AMD SEV focus on creating isolated execution environments for applications or virtual machines, whereas Intel TME and AMD SME concentrate on encrypting system memory to protect against unauthorized access.

Intel TDX is tailored more for virtualized environments, offering features for VM security, while AMD SEV is similarly focused on enhancing security in virtualized environments. Each technology has its unique characteristics, such as Intel SGX's focus on secure execution or AMD SEV's capabilities for secure VM migrations.

Overall, both Intel and AMD technologies contribute significantly to confidential computing by offering hardware-based security features, encryption mechanisms, and isolation to protect against various threats and attacks targeting sensitive data and applications. The choice between these technologies often depends on specific use cases, system requirements, and compatibility with the existing infrastructure.

How to choose CPU technology:

Table 5. Intel and AMD confidential computing feature matrix

Tech	Full VM protection	Recompilation needed?	Use case
AMD SEV	✓ Yes	✕ No	Confidential VMs with attestation
Intel TDX	✓ Yes	✕ No (for applications) or Maybe (for OS/libs)	Confidential VMs with attestation
Intel SGX	✕ No (at application level)	✓ Yes	Isolated, secure application components

Challenges and limitations of enabling TEEs on CPUs

Running confidential computing on CPUs, regardless of vendor, presents several challenges and limitations. These issues generally relate to performance overhead, application compatibility, hardware constraints, and security complexity.

Performance overhead

The isolation mechanisms of confidential computing, such as encrypting memory and managing TEE's, introduce computational overhead that can increase latency and decrease throughput. For typical workloads, this is on the order of 10 percent. For applications that perform a lot of Input/Output (I/O) operations (for example, reading from disk or sending data over a network), this overhead can be more severe. This is because data must be encrypted and decrypted as it moves in and out of the TEE's. The performance costs can be particularly prohibitive in High-Performance Computing (HPC) environments where every microsecond and every cycle of parallelism is critical.

Hardware and ecosystem limitations

Confidential computing relies on specialized CPU hardware, such as Intel Software Guard Extensions (Intel SGX) and Intel Trusted Domain Extensions (Intel TDX), and AMD Secure Encrypted Virtualization (AMD SEV). This can lead to vendor lock-in, limited availability, and potential compatibility issues between different hardware platforms.

CPUs and hardware-based Trusted Execution Environments (TEEs) often place limitations on resources, such as the number of confidential VMs, the amount of encrypted memory, and available hardware encryption keys. This restricts large-scale and data-intensive confidential deployments. It is critical to have an understanding of these limitations before designing a solution that utilizes TEE hardware.

Development and management complexity

To take full advantage of some confidential computing features, particularly older TEE architectures such as Intel SGX, applications must be rewritten to work with specific APIs. Managing secure TEE's also adds operational complexity, including handling encryption keys and attestation protocols. This can lead to the proliferation of instances and new data silos that are difficult for IT teams to manage and monitor. For example, the inherent opacity of TEEs, which intentionally limit external observability for security purposes, can create issues for developers. Standard security tools such as intrusion detection systems and log aggregators may not be able to verify what is happening inside the TEE.

Finally, careful planning must be undertaken when developing a comprehensive solution to avoid ecosystem fragmentation. The variety of confidential computing architectures and isolated programming frameworks can make widespread adoption difficult.

Security vulnerabilities

CPUs have historically been vulnerable to side-channel attacks (for example, Spectre and Meltdown) that attempt to extract secrets based on a processor's operational behavior. While confidential computing makes these attacks harder, it does not prevent them entirely, making ongoing mitigation efforts necessary. As confidential computing systems become more complex, incorporating such features as multi-TEE orchestration and shared memory, they may introduce new vulnerabilities and expand the overall attack surface. As with any technology, care, maintenance, and due diligence are still required.

Finally, attestation may raise centralized trust concerns. A growing reliance on vendor-managed attestation services raises potential concerns about centralized trust, supply chain risks, and possible backdoors. The simple fact of the matter, however, is that something needs to be trusted at some point to establish a baseline from which to work.

7. Attestation

Attestation is the process of providing evidence or measurements of a TEE's origin and current state. The evidence can be verified by another party either programmatically or manually. This party can then decide whether to trust the code that is running in the TEE. It is, typically, important that such evidence is signed by hardware that can be vouched for by a manufacturer so the party checking the evidence has strong assurances it was not generated by malware or other unauthorized parties.

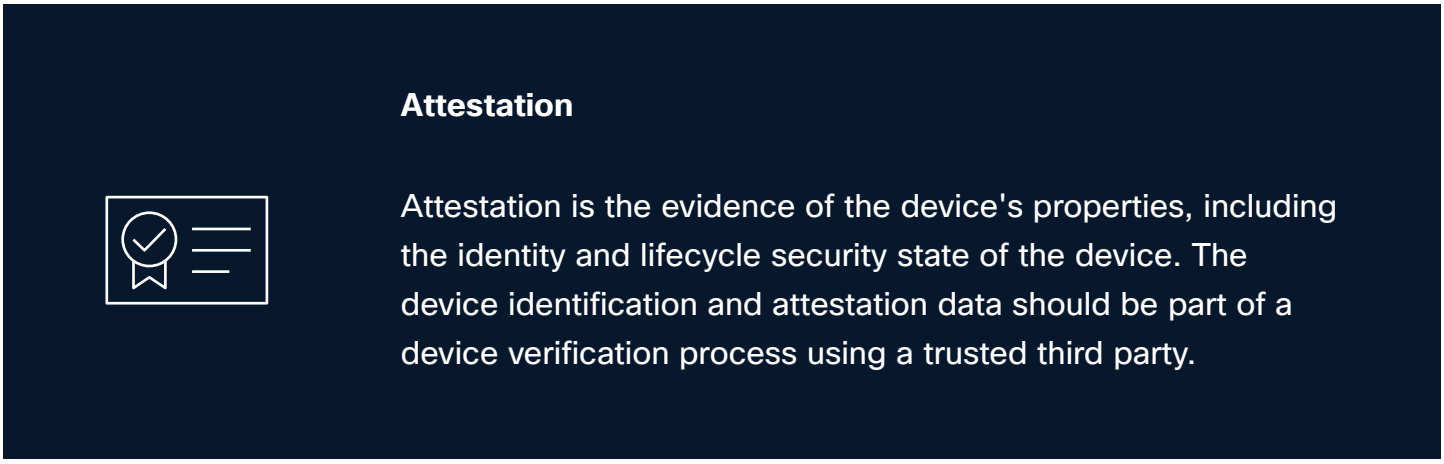

Figure 9. Attestation definition

There is a distinct difference between the type of attestation that occurs for, say, secure boot and the type of attestation demanded by TEEs in a confidential computing context. One is attestation “after the fact” (for example, the system has booted securely); the other is attestation “before the fact” (for example, this TEE is verified as secure to run your application[s]).

- **Attestation after:** The operation has already occurred; is it safe? (Secure boot)
- **Attestation before:** The operation is about to occur; is it safe? (Confidential execution environment)

Confidential VMs and the TEE’s in general can be verified with attestation. Throughout the attestation process, information is kept confidential, and data leakage is prevented. Moving forward in this document, we will be discussing only attestation as it pertains to confidential computing.

Types of attestation

There are two distinct types of attestation in confidential computing: remote and local.

Remote attestation

In remote attestation, one peer (the attester) produces believable information about itself (the evidence) to enable a remote peer (the relying party) to decide whether to consider that attester as a trustworthy peer. Another vital party (the verifier) carries out the remote attestation procedures. In short, it is evaluating a remote TEE (for example, on a cloud server) to assess whether it is in a trustworthy state **before** performing confidential operations. APIs are provided by the verifier service to handle these tasks.

Local attestation

Used to verify the integrity of a local TEE. This is typically conducted for container type deployments that rely on local integrity with a local service or resource to manage the verification, such as a Trusted Platform Module (TPM).

Remote attestation decouples the generation of evidence from its verification, allowing, for example, an attestation server to dynamically respond to situations such as the discovery of new vulnerabilities and start rejecting a previously accepted configuration. A physical chip such as a Trusted Platform Module (TPM) can only do very limited policy enforcement. Using a remote server allows for a much richer policy verification, as well as near real-time updates for new vulnerabilities.

In a typical remote attestation deployment, the attestation is handled either by the passport or background check model as described in the Remote ATtestation procedureS (RATS) Architecture.

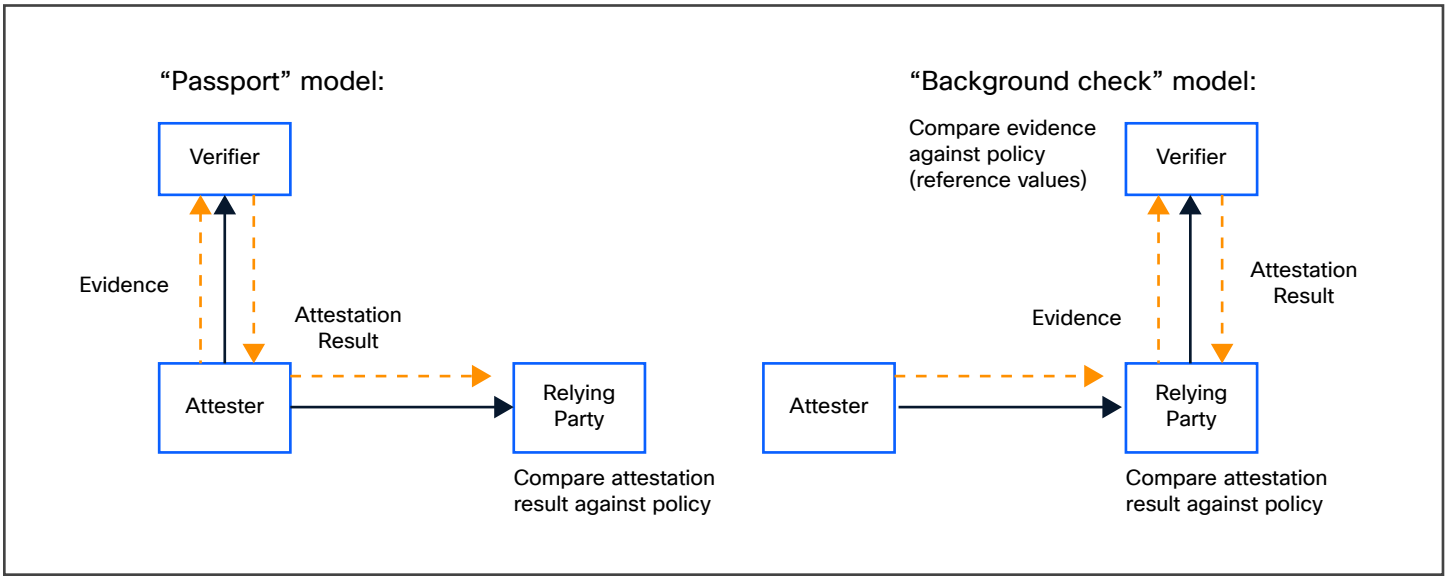

Figure 10. Attestation models

Attestation technologies

In this section we will cover the remote attestation services and capabilities for Intel, AMD, and NVIDIA.

Intel Trust Authority

Intel Trust Authority is a zero-trust attestation service that provides customers with assurance that their applications and data are protected on the platform of their choice, including multicloud, sovereign clouds, edge, and on-premises environments.

Intel Trust Authority verifies the trustworthiness of compute assets, including infrastructure, data, applications, endpoints, and AI/ML workloads. This attests to the validity of Intel confidential computing environments running on CPUs and GPUs.

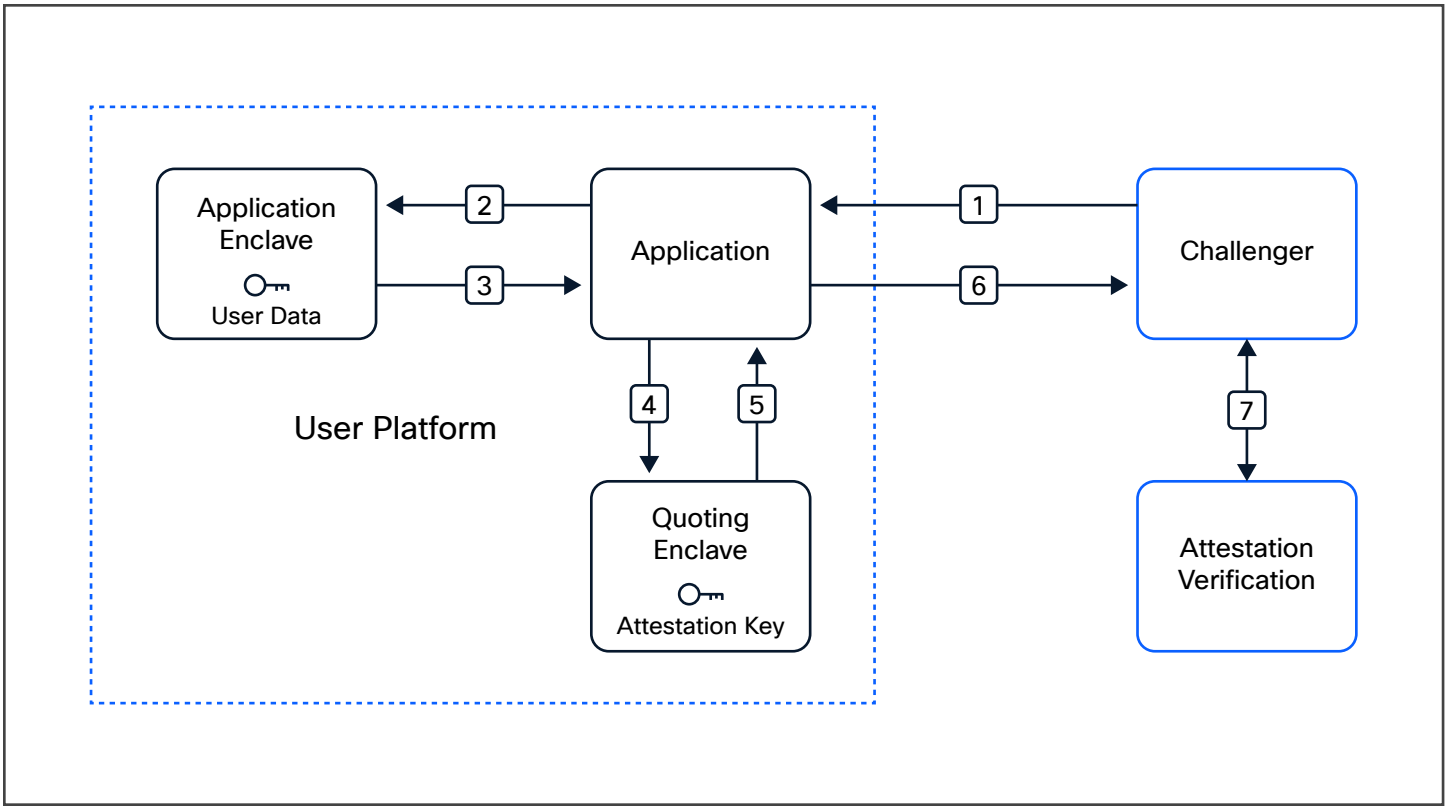

Figure 11. Intel attestation high-level workflow

The validity of Intel confidential computing environments includes attestation of Trusted Execution Environments (TEEs), Graphical Processing Units (GPUs), and Trusted Platform Modules (TPMs).

How Intel Trust Authority works

Intel Trust Authority operates as a Software as a Service (SaaS) that provides remote attestation to the authenticity and integrity of a confidential computing environment and related workloads. It is designed to work across on-premises, hybrid-cloud, and multicloud environments, providing a consistent security policy across different deployment models.

The attestation process involves initializing TEE environments to contain attestation signing assets and other secrets, with all token generation processing occurring within these TEEs. Protected certificates and keys are released only to code that has been verified as genuine. The attestation evidence for each microservice is stored in a blockchain-backed ledger and referenced using a unique ID in every attestation token generated. These references can be used to validate the specific microservice instances that produced the token and retrieve their TEE attestation information.

Intel Trust Authority provides cryptographic evidence that the code generating attestations is authentic and unaltered, ensuring that the code generating attestation is itself secure and unaltered. This service is particularly useful for organizations that handle sensitive data or intricate AI models, because it provides a robust security solution to demonstrate the trustworthiness of their confidential computing environments.

Intel Trust Authority supports multiple platforms, including on-premises hardware and cloud service providers such as Microsoft Azure and Google Cloud. Intel Trust Authority's attestation process involves three main steps:

1. The confidential computing workload attests its identity and fidelity by providing TEE measurements and other cryptographic evidence, known as a quote.
2. A verifier evaluates the quote against reference values and endorsers to determine if the quote is valid and if certain claims match stored values and policies.
3. The relying party uses the attestation token to decide if it should trust the attester.

The platform also provides a client and Intel TDX CLI to simplify the task of obtaining a quote from the attesting workload. The attester is responsible for collecting evidence for a quote, using an Intel Trust Authority library or other compatible method. The quote is then forwarded directly to Intel Trust Authority or sent to a relying party that relays the quote to Intel Trust Authority. It also supports attestation for Google Cloud Confidential Spaces and NVIDIA H100 GPUs. To do this, the platform's attestation tokens include additional claims to enhance security and trust verification.

AMD attestation

AMD SEV-SNP supports local attestation with APIs available for remote attestation.

AMD SEV attestation provides a secure and trusted environment for virtual machines, ensuring that sensitive data is protected and that the VMs are running on an authentic AMD platform. This attestation involves several key steps to ensure the trustworthiness of Virtual Machines (VMs) running on AMD hardware. Here's a simplified overview of the process:

- **Launch measurement:** The hypervisor requests the AMD Secure Encrypted Virtualization (AMD SEV) service to measure the initial guest memory contents and vCPU state. This process generates a cryptographic hash, known as the launch measurement, which is signed with a Virtualization-Code Key (VCEK) and stored in the host.
- **Attestation report:** The attestation report contains the launch measurement and is signed with a Virtualization-Leaf Key (VLEK) signature that chains back to an AMD root of trust. This report is used to validate that the VM is running in a genuine AMD environment and that the initial boot code was used to launch the VM.
- **Attestation process:** The attestation process can be initiated by the guest owner at any time, allowing them to request attestation reports, cryptographic keys, and other information as needed.
- **Validation:** The attestation report is signed by the AMD-SP firmware using the VCEK.
- **Secure communication:** During VM launch, a set of private communication keys is created by the AMD-SP, enabling the guest to communicate directly with the AMD-SP. This path allows the guest to request attestation reports, cryptographic keys, and other information as needed.

The attestation report includes the following:

- Platform measurements
  - Platform versioning
  - Platform runtime configuration
  - Chip identification
- Guest measurements
  - Owner identification
  - Guest-image identification
  - Initialization of image measurement
  - AMD SEV-SNP guest policy
  - Migration agents

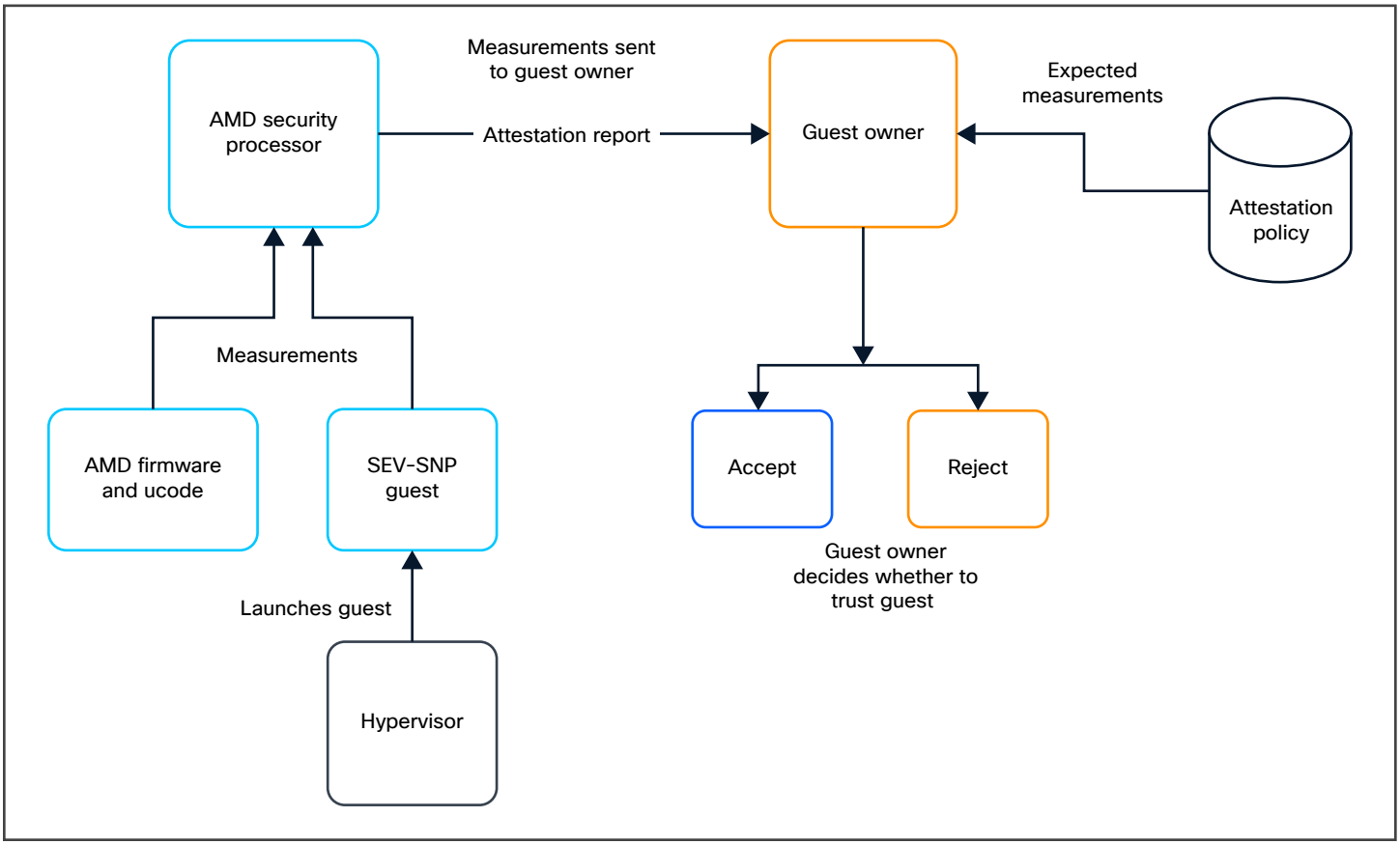

Figure 12. AMD attestation high-level workflow

NVIDIA attestation

NVIDIA GPU attestation verifies the integrity and authenticity of the GPU's hardware and software components to ensure a secure environment for confidential computing. This verification process can be performed in-band or out-of-band.

In-band attestation

In-band attestation uses the host system's standard communication channels, such as the Peripheral Component Interconnect express (PCIe) bus, to perform verification. This method is suitable for verifying components within the same system.

How NVIDIA in-band attestation works

1. A trusted part of the host system, typically a CPU's Trusted Execution Environment (TEE), initiates the attestation process by sending commands to the GPU over the PCIe bus.
2. The GPU responds with an attestation report, which is a cryptographically signed set of measurements detailing its hardware and software state.
3. The TEE on the host then sends this report to a remote attestation service for verification.
4. If the attestation is successful, the host can establish a secure, encrypted communication channel with the GPU using a session key to transfer data and workloads securely.

This approach is primarily used in confidential virtual machines where the GPU is part of the trusted boundary, and often with AI-based workloads that are aimed at GPUs. This allows a user to verify the integrity of the GPU before running a confidential workload, even in a cloud environment where the host provider cannot be fully trusted.

Out-of-band attestation

Out-of-Band (OOB) attestation uses a separate, independent channel for verification, bypassing the host system's main communication pathways. This is often done through a dedicated management interface, such as a Baseboard Management Controller (BMC).

How NVIDIA out-of-band attestation works

1. An external management service or BMC initiates the attestation process with the GPU over a separate, secure channel.
2. The GPU uses a Hardware-based Root of Trust (HRoT) to generate the attestation report. This HRoT is a secure, immutable component within the GPU that stores cryptographic keys.
3. The BMC or management service sends the report to a remote attestation service, such as the NVIDIA Remote Attestation Service (NRAS), for validation against known "golden measurements" stored in Reference Integrity Manifest (RIM) files.
4. Successful verification confirms the authenticity and integrity of the GPU and its firmware without relying on the host CPU or its software.

This method is particularly valuable for securing infrastructure in data centers and cloud environments. It allows the data-center operator to verify the security state of the GPU, independent of the server's operating system, to detect tampering or unauthorized modifications. For example, the confidential computing mode on an NVIDIA H100 GPU can be set using OOB commands from a BMC.

A summary of the differences between NVIDIA in-band and out-of-band attestation methods is given in Table 6.

Table 6. NVIDIA in-band vs. out-of-band attestation mechanisms

Feature	In-band attestation	Out-of-band attestation
Communication channel	Standard host-to-device interface (for example, PCIe)	Separate, independent channel (for example, BMC)
Trust reliance	Relies on a trusted element within the host system, such as a CPU TEE, to initiate the process	Bypasses the host system, relying on an external management controller to initiate attestation
Primary use case	Establishing trust for confidential workloads running on a specific virtual machine or tenant	Securing the underlying hardware infrastructure for data-center operators and cloud providers
Key advantage	Extends the trust boundary of a CPU-based TEE to include the GPU	Provides a higher level of assurance and resilience against a compromised host operating system

Services in the NVIDIA Attestation Suite

The NVIDIA Attestation Suite provides a unified solution for verifying the authenticity and validity of a document, record, identity, or other information. The goal of the NVIDIA Attestation Suite is to establish trust between parties by providing a reliable and independent verification of claims made by one party to another. The diagram in Figure 15 shows the suite's workflow and its three main features.

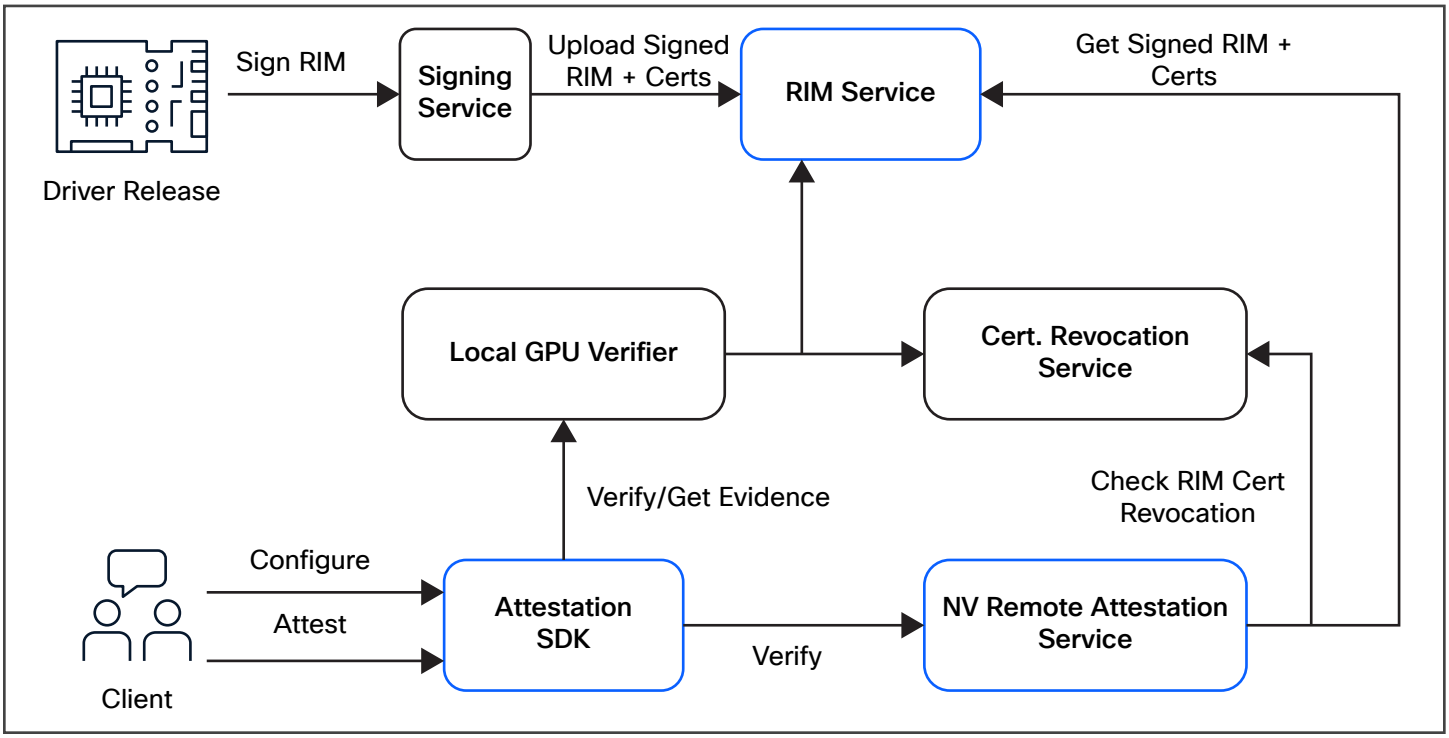

Figure 13. NVIDIA Attestation Suite services

The three leading components of the NVIDIA Attestation Suite are:

- Attestation SDK
  - NVTrust tools and resources
- RIM Service
  - The Reference Integrity Manifest (RIM) Service and the NVIDIA OCSP Service provide a unified solution for verifying the authenticity and validity of documents, records, identities, or other information. The NVIDIA RIM service is a file-hosting service that facilitates secure transfers of RIM bundles by attestation platforms for GPU attestation. It provides a mechanism for secure transfers of requested RIM bundles by attestation platforms for GPU attestation. The service is used to validate Reference Integrity Manifest (RIM) structures generated as part of the driver and VBIOS builds against actual values collected from a GPU at runtime. NVIDIA collects IP addresses and information about the GPU, including device certificates, for security, debugging, and troubleshooting purposes. Data collected is deleted when it is no longer needed for these purposes.

- NVIDIA Remote Attestation Service
  - The NVIDIA Remote Attestation Service (NRAS) allows for centralized and standardized verification logic while ensuring that attestation results are trusted and consistently applied. It operates through a client-server model where the client collects evidence from the GPU and sends it to the NVIDIA Remote Attestation Service for verification. The service returns a signed JWT token containing attestation claims, which can be decoded and validated against a policy to ensure that specific security requirements are met. For troubleshooting, detailed error information is logged, and NVIDIA's Attestation Troubleshooting Guide provides comprehensive support.

8. Cisco UCS Runtime Defenses (RTDs)

Runtime defenses are complementary to hardware-based solutions for confidential computing. Protecting data in use requires the implementation of runtime defenses to safeguard against rogue activities from the applications or users running within the TEE. Many of these need to be implemented by the application or OS developer or both. Systems that run a Baseboard Management Controller (BMC) such as the Cisco Integrated Management Controller (CIMC) and other embedded software need to take this into consideration as well. To this end, Runtime Defenses (RTDs) are implemented on Cisco UCS platforms.

Runtime defenses are security measures applied while applications are running to protect against attacks during execution. This is largely accomplished by targeting injection attacks of malicious code in running software and randomizing otherwise traditionally deterministic operations such as memory assignment.

Types of runtime defenses

There are several types of RTDs that can be implemented:

- **Memory protection:** mitigating risks such as buffer overflows and memory corruption
- **Control Flow Integrity (CFI):** ensuring that the program flow is not tampered with
- **Stack canaries and Data Execution Prevention (DEP):** preventing specific types of exploits
- **Runtime threat detection:** techniques to detect anomalous behaviors during execution (for example, intrusion-detection systems and heuristic-based defenses)

Cisco UCS runtime defenses include Address Space Layout Randomization (ASLR), Built-in Object Size Checking (BOSC), and X-Space.

Address Space Layout Randomization (ASLR) is a defense that initiates before the operating system even loads. It randomizes portions of a device's memory layout using fuzzy logic algorithms, preventing an attacker from designing malware that depends on a known memory arrangement. For example, every Cisco Catalyst® switch (of the same type) is configured with the same factory memory layout. When powering up, ASLR kicks in and randomizes a portion of memory, resulting in each switch having a different running memory configuration. An attack designed to use the factory memory layout, or a memory layout on an attacker's device, will be thwarted by the ASLR reconfiguration. Since memory layout changes whenever a device boots, creating a persistent attack on an ASLR device becomes very difficult.

Built-in Object Size Checking (BOSC) and Safe C are two runtime defenses that work together to manage how incoming data is copied into memory. An attack that relies on forcing a larger amount of malicious code, through a routing packet or a CLI command, into a smaller chunk of memory to cause a buffer overflow error is stopped by implementing BOSC in combination with compatible compiler technology. In cases where the compiler cannot predict the destination buffer size, Safe C libraries are implemented to provide additional protection from buffer overflow attacks.

X-Space (sometimes called NX bit), or executable space protection, is frequently supported by a CPU-maker's architecture to prevent executable code that is hidden in data streams from executing in memory spaces. Enforcing X-Space in network devices marks memory regions as nonexecutable so that malicious code hidden in packets cannot be executed in those memory regions, adding another layer of protection to make it difficult for attackers to exploit buffer overruns.

Each of these runtime defenses must be implemented by design, since they are not deployed by default.

9. Cisco UCS: Integrating confidential computing in enterprise systems

Confidential computing enhancements specific to Cisco – Cisco Hyperfabric

Secure platform foundation

Cisco Hyperfabric integrates with Cisco UCS security mechanisms, including secure boot, platform root of trust (Intel Boot Guard, AMD PSB), and Cisco TPM 2.0 attestation. These capabilities ensure that servers are in a trusted state before workloads are launched. Hyperfabric provides a secure transport mechanism for attestation data, extending trust from hardware to the TEE and ultimately to the applications processing sensitive data.

Workload isolation and segmentation

Confidential computing depends on strict workload isolation to prevent data leakage or lateral movement attacks. Hyperfabric delivers:

- **Microsegmentation:** logical isolation at the fabric level, ensuring workloads cannot communicate outside defined security zones
- **Policy-based pinning:** secure service profiles that bind workloads to specific hosts and network paths, reducing the attack surface

This segmentation guarantees that sensitive workloads remain isolated, even in shared infrastructure environments.

High-performance encrypted data paths

Performance is critical for real-time analytics and machine-learning workloads. Hyperfabric provides hardware-accelerated encryption offload, SR-IOV for direct device assignment, and RDMA over Converged Ethernet (RoCE) for low-latency east/west communication. These features enable encrypted, high-throughput communication between TEEs with negligible performance impact.

Centralized management and compliance

Cisco Intersight provides centralized lifecycle management, policy enforcement, and visibility for Hyperfabric-connected systems. This integration allows organizations to:

- Automate secure provisioning of TEEs
- Enforce consistent security policies across Cisco UCS domains
- Generate compliance reports to prove infrastructure trustworthiness

This is particularly valuable in regulated industries, where proving the security state of the infrastructure is essential for audits and certifications.

Secure multicloud connectivity

Many confidential computing deployments extend across hybrid or multicloud environments. Hyperfabric provides encrypted interconnects and consistent policy enforcement across on-premises Cisco UCS systems and cloud resources. This ensures that data remains protected and workload integrity is preserved, even when crossing security domains.

Confidential computing enhancements specific to Cisco – Cisco Hypershield

Distributed microsegmentation

Cisco Hypershield delivers fine-grained segmentation at the kernel level using eBPF. This enforces least-privilege communication between workloads and prevents unauthorized lateral movement—closing a common attack vector that TEEs alone cannot address.

Runtime threat detection and enforcement

Through AI-driven analytics and behavioral monitoring, Hypershield can identify anomalous or malicious behavior in real time. Suspicious activity can be automatically blocked, isolating compromised workloads before they can impact sensitive data in TEEs.

Policy consistency across hybrid environments

Hypershield integrates with Cisco Intersight to provide centralized policy management and attestation-driven enforcement. This ensures that workloads running in Cisco UCS servers, public clouds, or edge environments follow the same security standards, even as they scale or migrate dynamically.

Attestation-driven security decisions

By consuming attestation data from confidential computing workloads, Hypershield enforces policies based on the cryptographic proof of workload integrity. Only workloads verified as trusted can communicate or access protected resources.

10. Future of confidential computing

Confidential computing is poised for wide adoption across industries as demands for tighter data security increasingly come to the forefront. This is also driven by the need to remain both competitive and innovative while minimizing the risk of compromising intellectual property and end-user data.

- **Emerging trends:**
  - Integration of confidential computing with AI and machine learning for secure AI model training
  - Decentralized and blockchain-based approaches to trust and security
  - Quantum computing's potential impact on confidentiality and encryption
  - Containerization
  - New attestation capabilities

11. Conclusion

In summary, confidential computing is critically important in server operations. Cisco UCS is uniquely positioned to offer a full set of confidential-computing capabilities in hardware along with a robust set of complementary technologies such as RTDs, Cisco Hypershield, and Cisco Hyperfabric. This breadth of capabilities provides a complete solution to data and execution protection throughout all aspects of the information lifecycle, from transport, to storage at rest, to, now also, data in use. The advantages of implementing data-in-use protections are wide reaching and provide new opportunities to expand the ability of enterprises to operate securely. These advantages include capabilities:

- To protect sensitive data, even while in use — and to extend the benefits of cloud computing to sensitive workloads. When used together with data encryption at rest and in transit with exclusive control of keys, confidential computing eliminates the single largest barrier to moving sensitive or highly regulated data sets and application workloads from an inflexible, expensive on-premises IT infrastructure to a more flexible and modern public-cloud platform.
- To protect intellectual property. Confidential computing isn't just for data protection. The TEE can also be used to protect proprietary business logic, analytics functions, machine-learning algorithms, or entire applications. This is particularly relevant to AI models.
- To collaborate securely with partners on new cloud solutions. For example, one company's team can combine its sensitive data with another company's proprietary calculations to create new solutions – without either company sharing any data or intellectual property that it doesn't want to share.

- To eliminate concerns when choosing cloud providers. Confidential computing lets a company leader choose the cloud-computing services that best meet the organization's technical and business requirements, without worrying about storing and processing customer data, proprietary technology, and other sensitive assets. This approach also helps alleviate any additional competitive concerns if the cloud provider also provides competing business services.
- To protect data processed at the edge. Edge computing is a distributed computing framework that brings enterprise applications closer to data sources such as IoT devices or local edge servers. When this framework is used as part of distributed cloud patterns, the data and application at edge nodes can be protected with confidential computing.

12. References

- What Is Confidential Computing? | NVIDIA Blogs
- Confidential VMs on Intel CPUs: Your data's new intelligent defense | Google Cloud Blog
- TDX: An In-Depth Exploration of Intel Trust Domain Extensions | OpenMetal IaaS
- Documentation for Intel Trust Domain Extensions
- Runtime Integrity Measurement and Attestation in a Trust Domain
- Intel Software Guard Extensions (Intel SGX)
- Hardware Setup - Intel TDX Enabling Guide
- Host OS Setup - Intel TDX Enabling Guide
- Guest OS Setup - Intel TDX Enabling Guide
- How to enable Intel TDX function in Windows Server 2022 - Microsoft Q&A
- Confidential Computing Performance with AMD SEV-SNP Google Cloud N2D VM Instances
- Confidential computing platform-specific details
- AMD SEV-SNP Attestation: Establishing Trust in Guests (PowerPoint presentation)
- Deployment models in confidential computing | Microsoft Ignite
- Confidential containers on Azure | Microsoft Ignite
- SGX Enclaves | Microsoft Ignite
- Zero trust starts here: Validated patterns for confidential container deployment
- NVIDIA Attestation Suite: Overview

- GitHub - NVIDIA/nvtrust: Ancillary open source software to support confidential computing on NVIDIA GPUs
- Attestation and Confidential Computing – a technical introduction
- Attestation in confidential computing
- Attestation types and scenarios | Microsoft Ignite
- RFC 9334: Remote ATtestation procedureS (RATS) Architecture
- Introduction | Intel Trust Authority
- Intel Trust Authority
- Seamless Attestation with Intel Trust Authority

Document information

Document summary	Prepared for	Prepared by
V1.0	Cisco Field	Aaron Kapacinskas
Key Contributors	Mike Isaia, Abdel-moniem Rezk, Eric Voit	
Changes		
N/A		