Trusted Execution Environment (TEE)

Fengwei Zhang and Haoyang Huang

COMPASS Lab, SUSTech

2024.11.17

Background

Introduction to TEE

Arm TrustZone

Arm CCA

COMPASS Lab Publications

At the Beginning

Ensuring **data security** has become a critical need to prevent economic and personal losses.

- **Equifax Data Breach (2017)**: In 2017, Equifax, one of the largest credit reporting agencies, experienced a breach that exposed the personal data of 143 million individuals.
- **Facebook Data Breach (2019)**: In 2019, over 500 million Facebook user records, including phone numbers and email addresses, were exposed.

Categories of Data Security

Data Security efforts can be grouped into three main areas:

- **Data at Rest**: This refers to data stored on devices or backup media, such as SSDs and cloud storage.
- **Data in Transit**: This category includes data actively moving from one location to another, such as across the internet or within private networks.
- **Data in Use**: Data in use refers to data that is actively being processed by applications or systems.

Data at Rest

Data at Rest refers to data that is stored on a device or backup medium, such as SSDs and cloud storage. To secure data at rest, there are several methods that can be used:

- **Disk Encryption**: Encrypting entire disks or storage devices ensures that all data stored on them is protected.
- **File Encryption**: Encrypting individual files or folders ensures that sensitive files are protected even if they are moved or copied to different locations.
- **Database Access Control**: Implementing strict access control measures for databases ensures that only authorized users can access and modify sensitive data.

Data in Transit

Data in Transit refers to data actively moving from one location to another, such as across the internet or through a private network. Data in transit can be secured through various protocols:

- **HTTPS**: HTTPS is an extension of HTTP that uses SSL/TLS to encrypt data transmitted between a web server and a client.
- **IPSec**: IPSec encryption provides a secure way to transmit data over IP networks, ensuring that data is protected from interception and tampering.
- **MACsec**: MACsec is a security protocol that provides encryption and integrity protection for data transmitted over Ethernet networks.

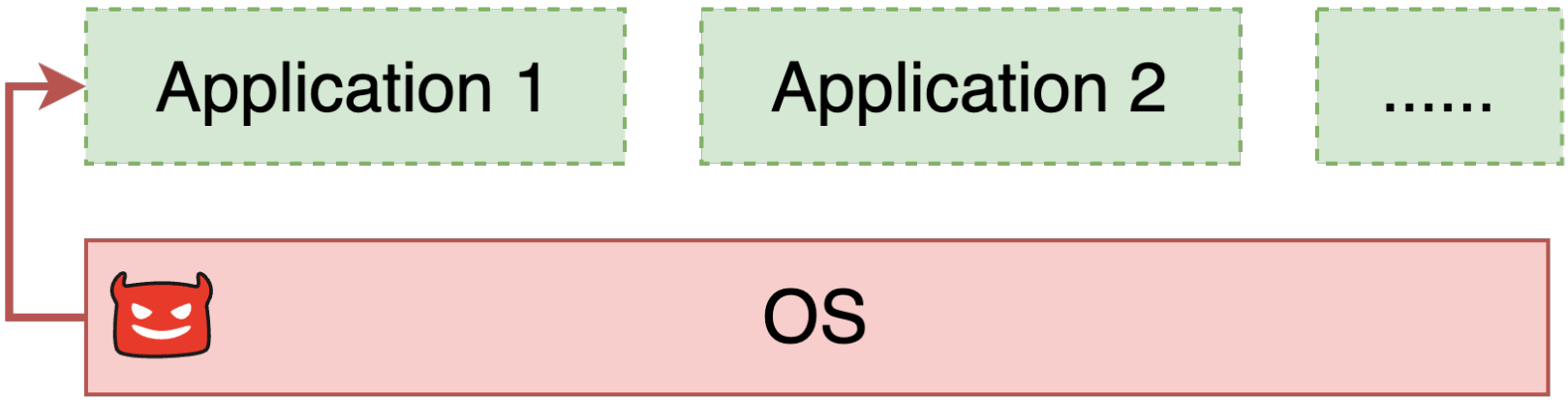
Data in Use

Data in Use refers to data that is actively being processed by applications or systems. To secure data in use, there are several methods that can be used:

- **ASLR**: Address Space Layout Randomization (ASLR) makes it more difficult for attackers to predict the location of specific code or data, reducing the effectiveness of certain types of attacks.
- **CFI**: Control Flow integrity (CFI) ensures the control flow of a program follows a predefined path, preventing attackers from hijacking the program's execution flow.
- **SFI**: Software Fault Isolation (SFI) isolates different software components within isolated memory regions, preventing them from interfering with each other.

Data in Use

Securing data in use presents unique challenges compared to data at rest or in transit. Approaches mentioned above are implemented at the user level, falling short in protecting data from privileged threats. Attacker can bypass these mechanism through vulnerabilities in OS (e.g., Linux Kernel).



Data in Use

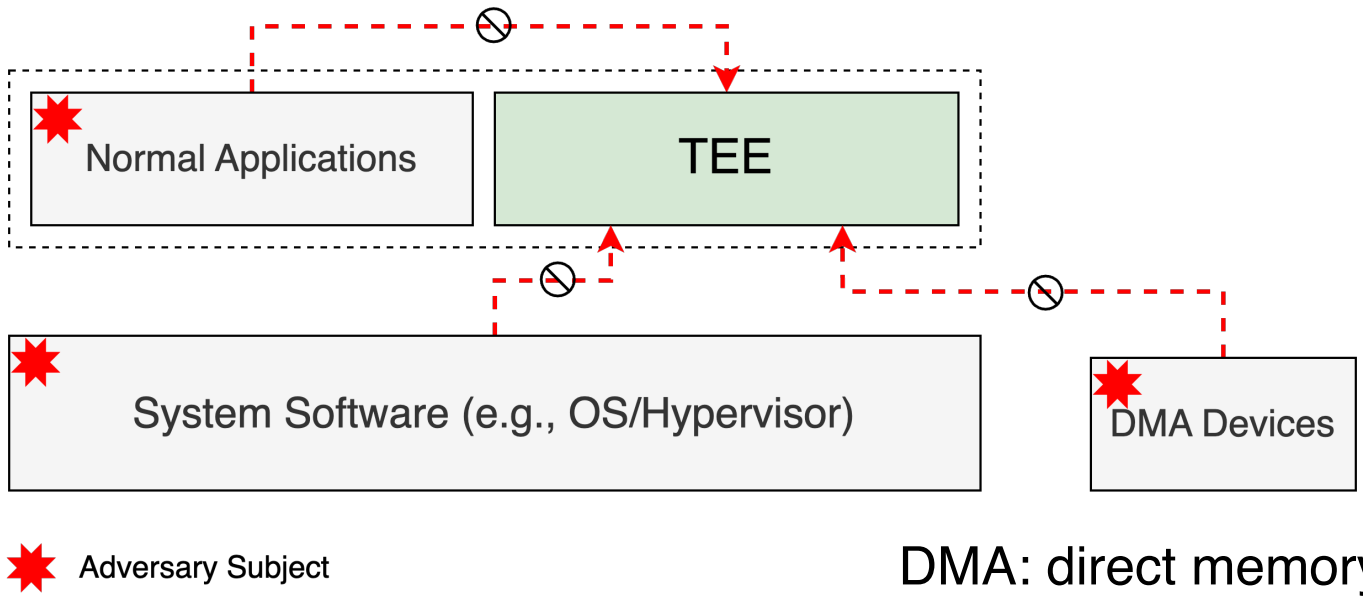
In recent years, the discovery of vulnerabilities in the Linux kernel has highlighted the ease with which attackers can exploit these to gain privileged access.
Therefore, we need stronger isolation mechanisms to safeguard data in use effectively.

Year	Overflow	Memory Corruption	Sql Injection	XSS	Directory Traversal	File Inclusion	CSRF	XXE	SSRF	Open Redirect	Input Validation
2014	18	31	0	0	0	0	0	0	0	0	22
2015	13	17	0	0	0	0	0	0	0	0	5
2016	36	76	0	0	0	0	0	0	0	0	17
2017	62	86	0	0	1	0	0	0	0	0	20
2018	32	70	0	0	0	0	0	0	0	0	11
2019	30	124	0	0	1	0	0	0	0	0	4
2020	10	40	0	0	1	0	0	0	0	0	2
2021	18	54	0	0	1	0	0	0	0	0	7
2022	41	149	0	0	0	0	0	0	0	0	2
2023	19	161	0	0	0	0	0	0	0	0	1
2024	44	758	0	0	1	0	0	0	0	0	0
Total	323	1566			5						91

What is TEE

protect the data in use when OS is malicious

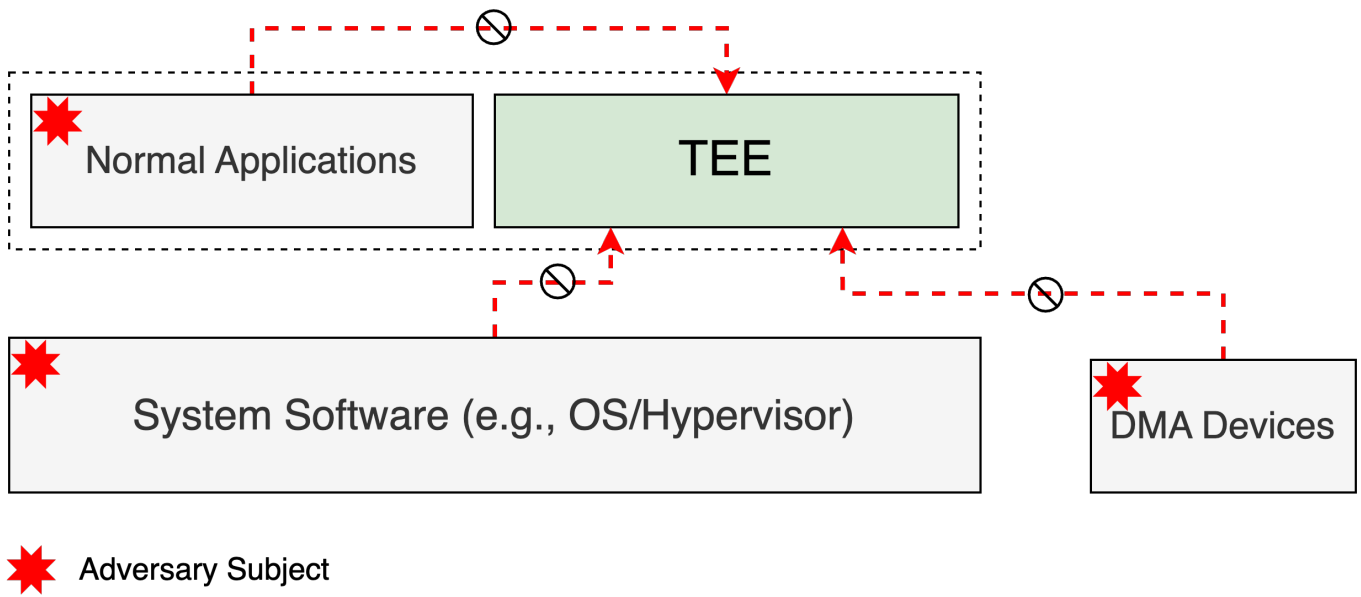
A **Trusted Execution Environment (TEE)** is a secure area within a system designed to isolate and protect sensitive data and computations. This isolation ensures that even high-privilege system components, such as the operating system or hypervisor, cannot access data or code within the TEE.



TEE Security Features

TEEs provide several key security features to protect sensitive data and operations:

- **integrity**: Data and code within the TEE are protected from unauthorized modification or tampering.
- **Confidentiality**: Data and code within the TEE are protected from unauthorized access, even from system software.
- **Attestation**: Remote parties can verify security states and integrity of the TEE.



TEE Use Cases

TEEs are now integrated into various devices and systems to provide secure services and protect sensitive data.

Redhat
Enarx

Azure Confidential computing

Microsoft
OpenEnclave

Google Asylo
Google OAK

AWS Nitro Enclaves

Baidu MesaTEE

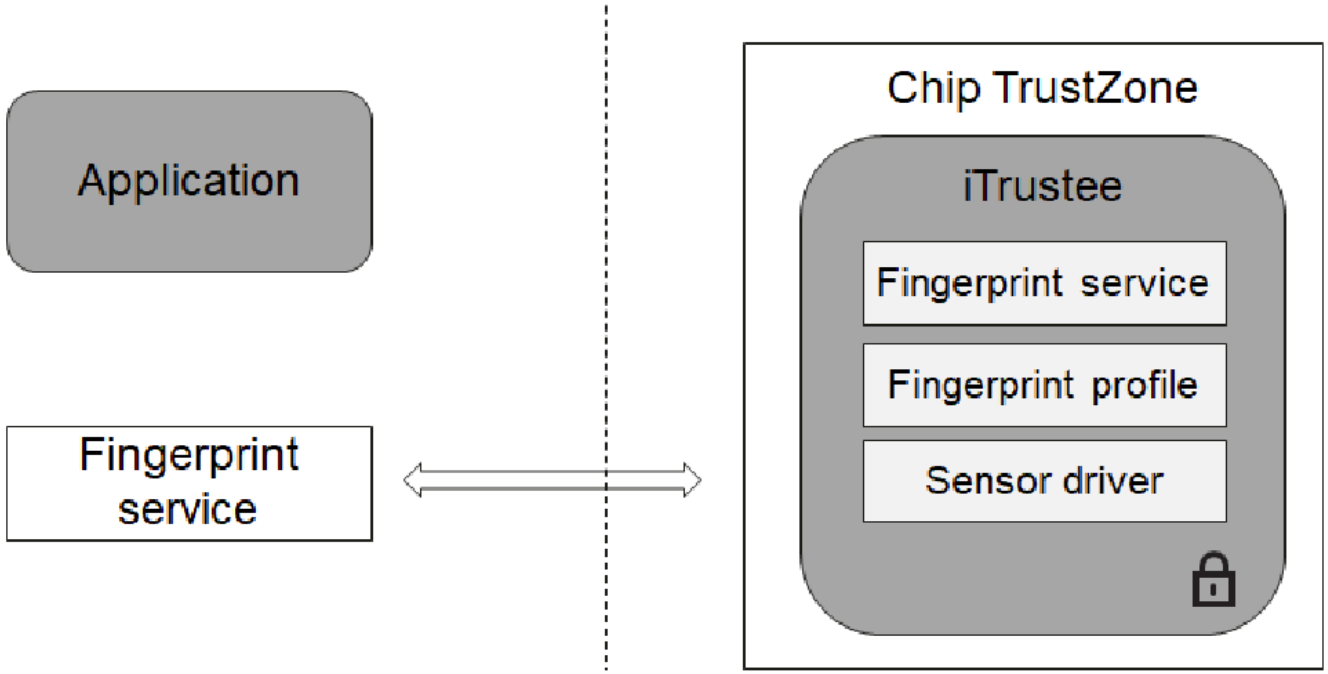
Apache Teaclave

Google
confidential VMs

Veracruz
Veraison

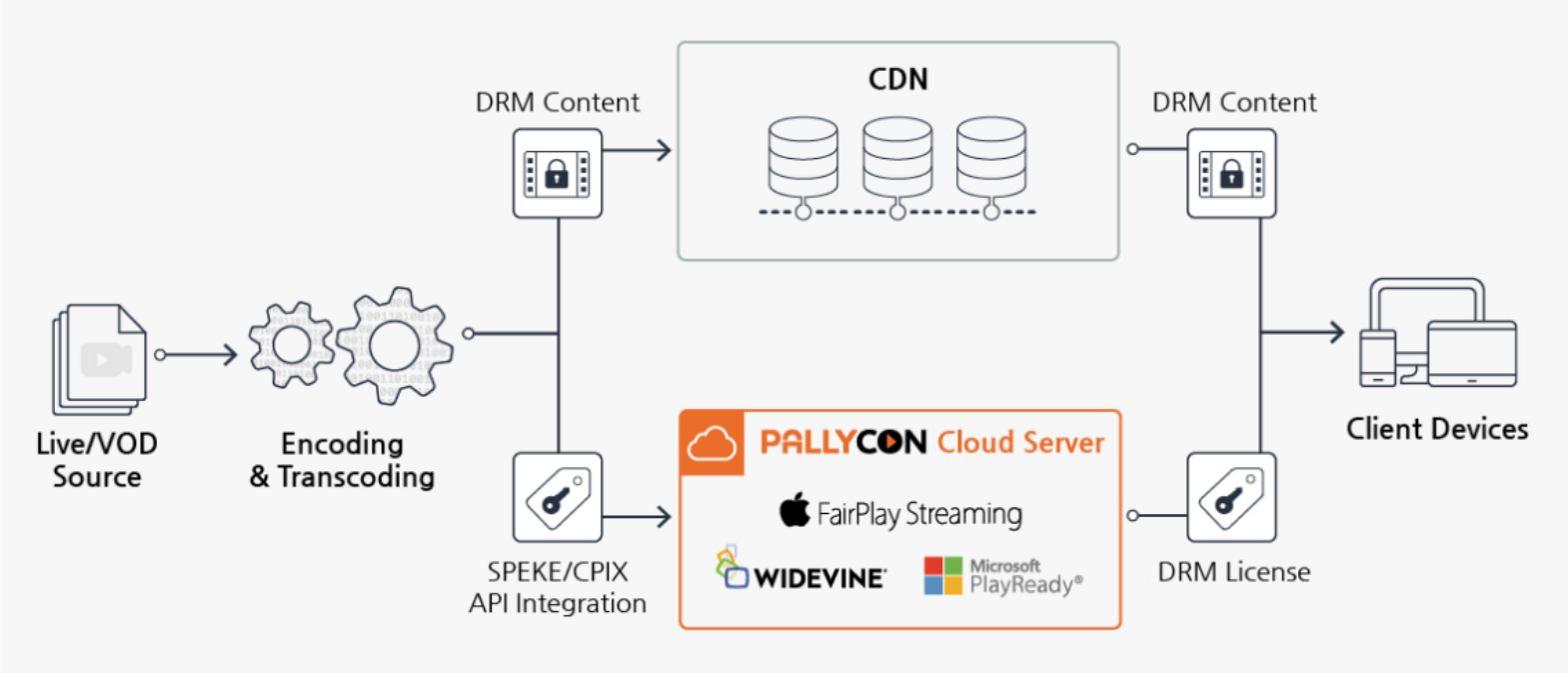
TEE Use Cases

In mobile devices, TEEs are commonly used to securely store biometric data and process authentication requests, ensuring that sensitive information is protected from unauthorized access.



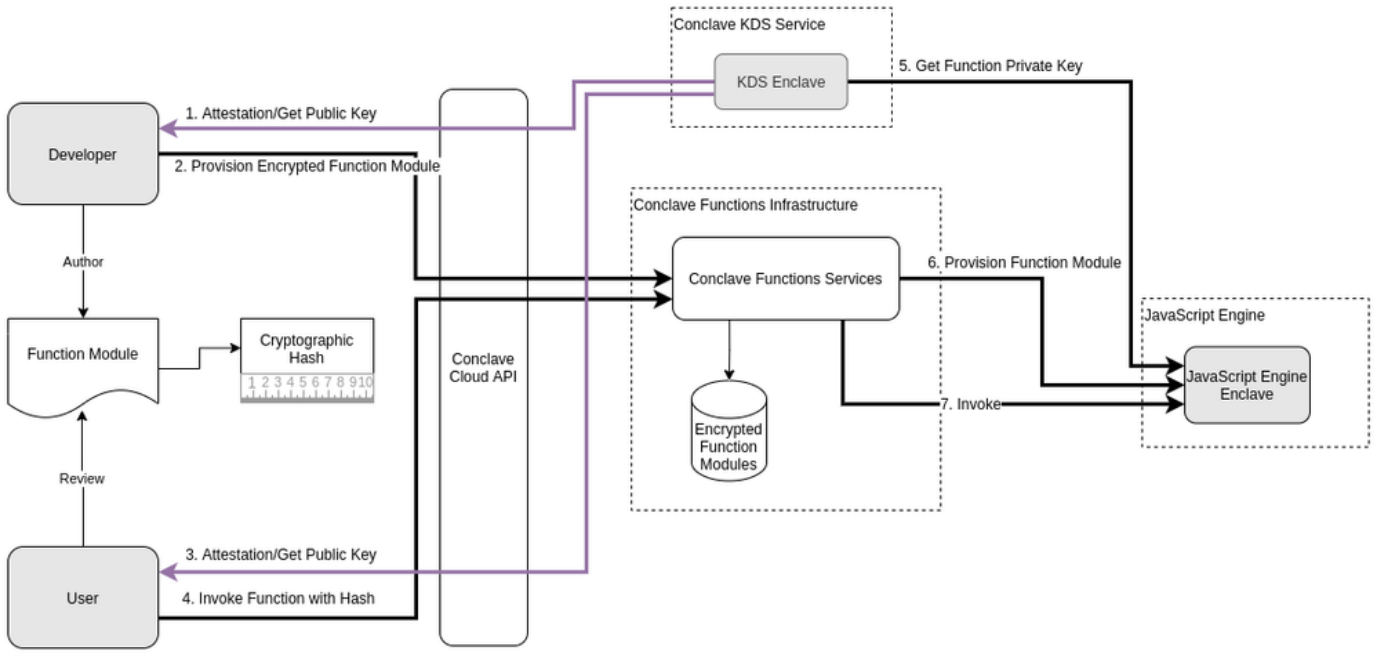
TEE Use Cases

Additionally, TEEs are used for Digital Rights Management (DRM) services, safeguarding digital content from unauthorized access and copying.

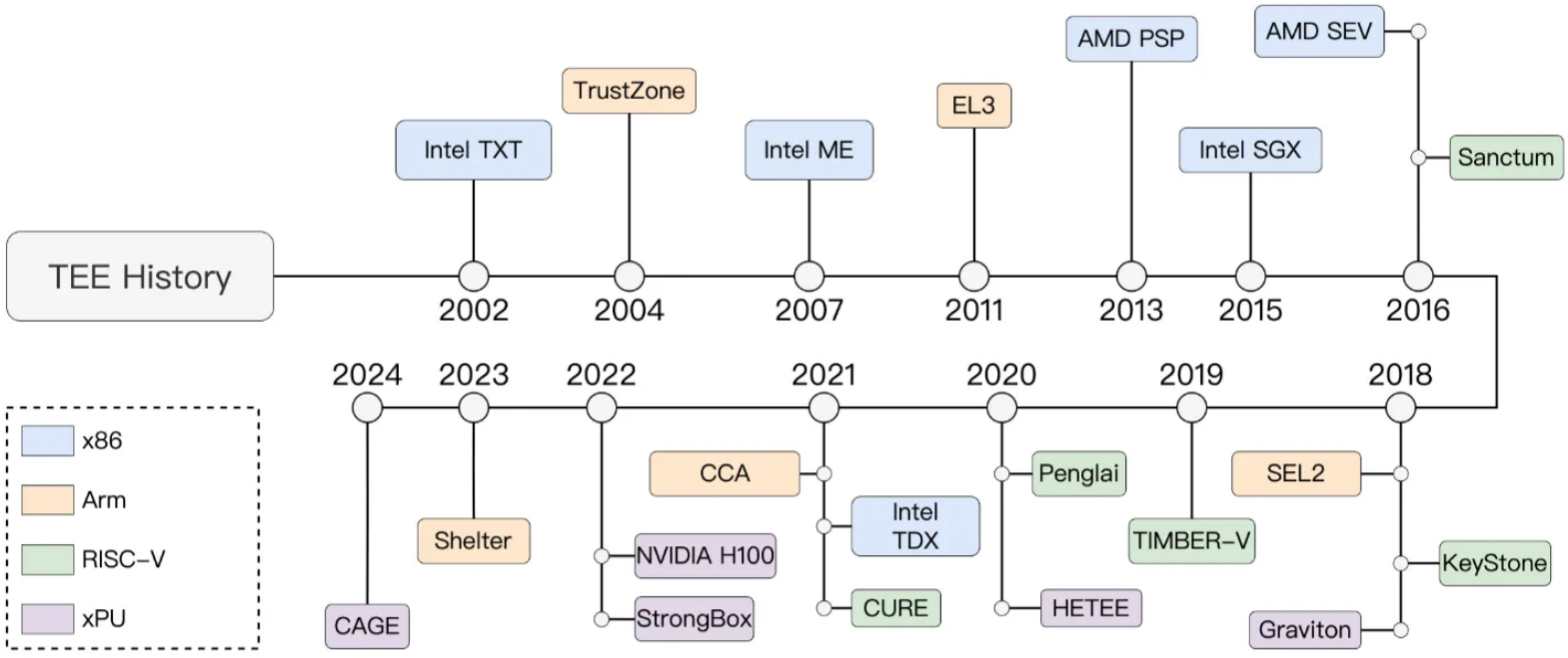


TEE Use Cases

With the rise of cloud computing, TEEs are also utilized to secure cloud services, protecting sensitive data and operations from unauthorized access or tampering.

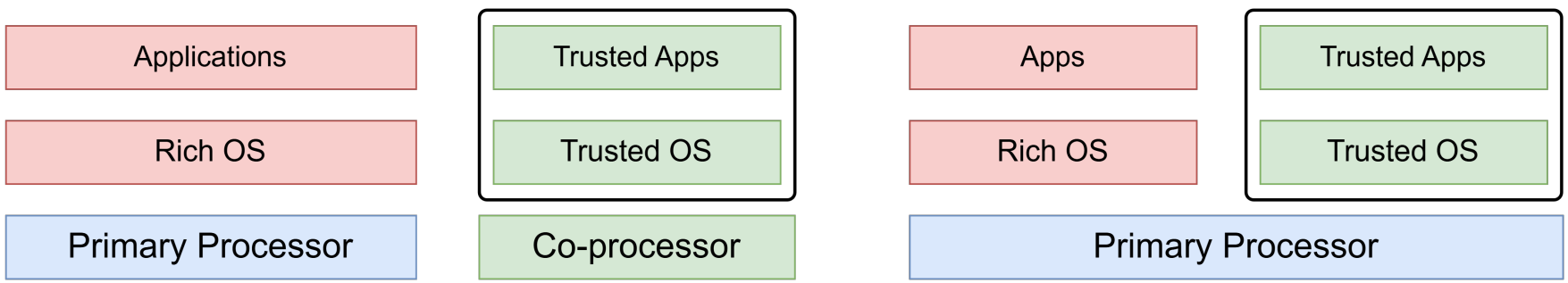


TEE History


TEE History
2002
2004
2007
2011
2013
2015
2016
2024
2023
2022
2021
2020
2019
2018
Intel TXT
TrustZone
Intel ME
EL3
AMD PSP
Intel SGX
AMD SEV
Sanctum
CAGE
Shelter
NVIDIA H100
StrongBox
CCA
Intel TDX
CURE
Penglai
HETEE
TIMBER-V
SEL2
KeyStone
Graviton
x86
Arm
RISC-V
xPU

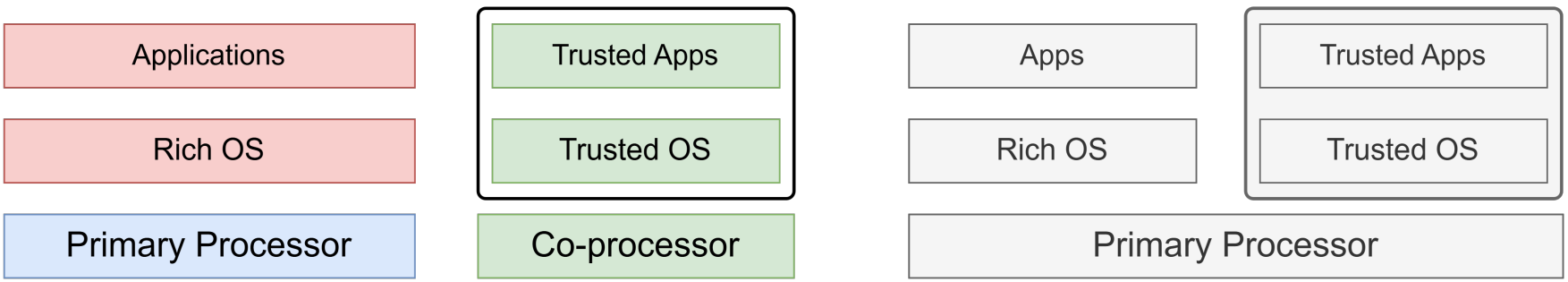
TEE History

Early TEE technologies focused on providing security services to platform designers and operating system vendors (OSVs), including Intel TXT, AMD PSP, and Intel ME. These technologies were designed to deliver security-sensitive services, such as power management and device integrity verification.
The approaches used in early TEE technologies can be categorized as **Secure Co-processor** and **Isolated Execution Mode**.



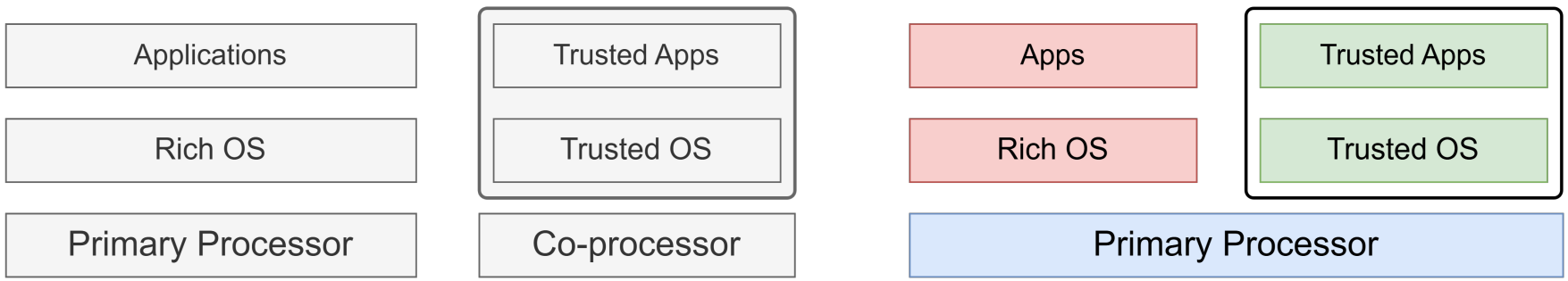
TEE History

Secure Co-processor provides a dedicated, separate environment to handle sensitive computations and data, ensuring their protection from the main processor and operating system.
For example, Intel Management Engine (ME) operates as a micro-controller within the system, handling low-level security and power management functions, which are isolated from the main CPU to enhance security.



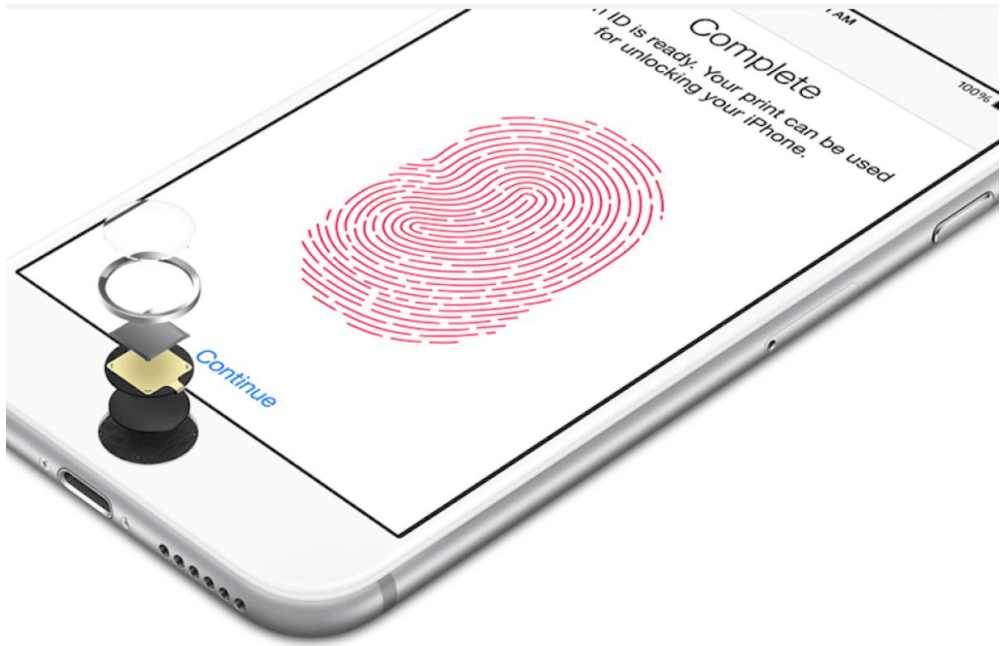
TEE History

Isolated Execution Mode creates a dedicated environment within the primary processor, allowing secure code execution separate from the main system.
For example, Arm TrustZone extends processors with a new execution mode, allowing applications to run sensitive code in the Secure world, which is isolated from the Normal world.



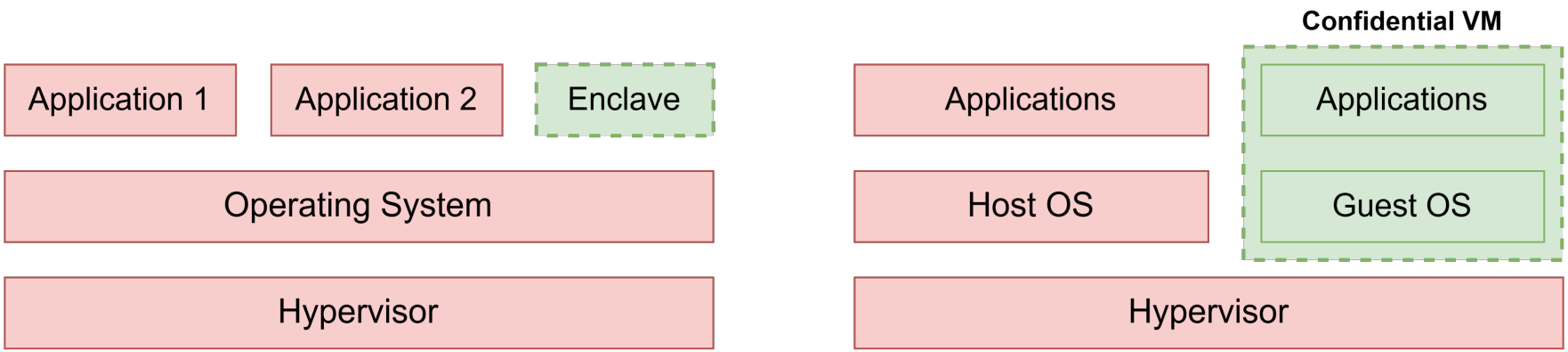
TEE History

With the development of smartphones, the need to protect growing volumes of personal data became more critical. Consequently, more applications began to leverage TEEs to enhance security. For example, in 2013, Apple uses the TEE to protect Touch ID, securely storing and processing fingerprint data.



TEE History

Due to security requirements, these trusted applications were initially limited to specific functions and could only be deployed by Original Equipment Manufacturers (OEMs).
In the following years, new TEE designs have been introduced to unlock the potential of confidential computing, allowing a broader range of developers to benefit from TEEs: **Enclave** and **Confidential Virtual Machine (VM)**.



TEE History

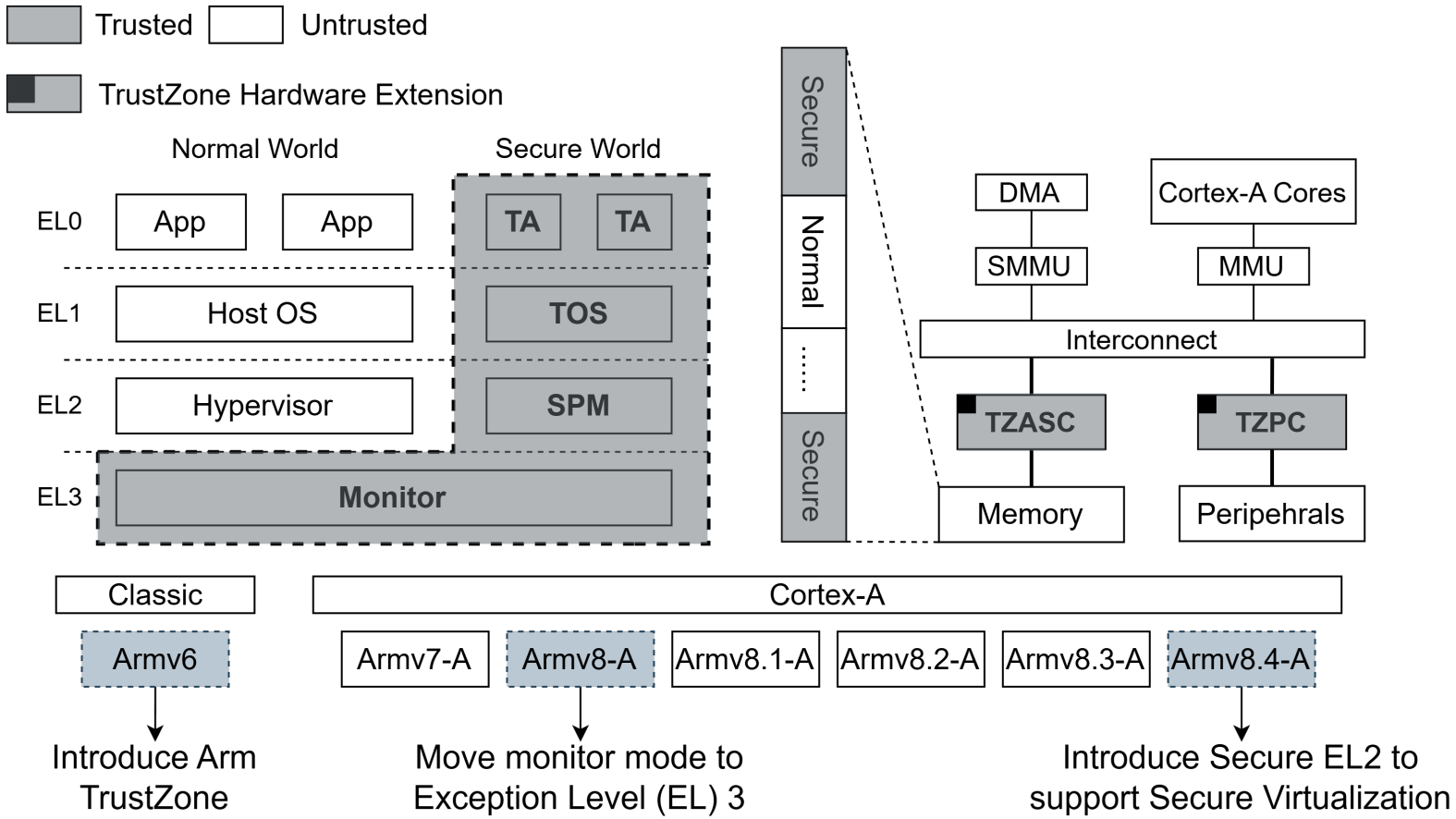
Enclave creates secure regions within a processor where applications can run sensitive code and manage data in isolation. Intel Software Guard Extensions (SGX) is a notable example, enabling developers to partition applications to protect specific code and data within enclaves.

TEE History

Confidential VM extends the concept of isolation to virtualized environments, allowing entire virtual machines to operate in a secure, encrypted state.
To date, most vendors have published their VM-based TEE solutions, such as AMD SEV, Intel TDX, and Arm CCA.

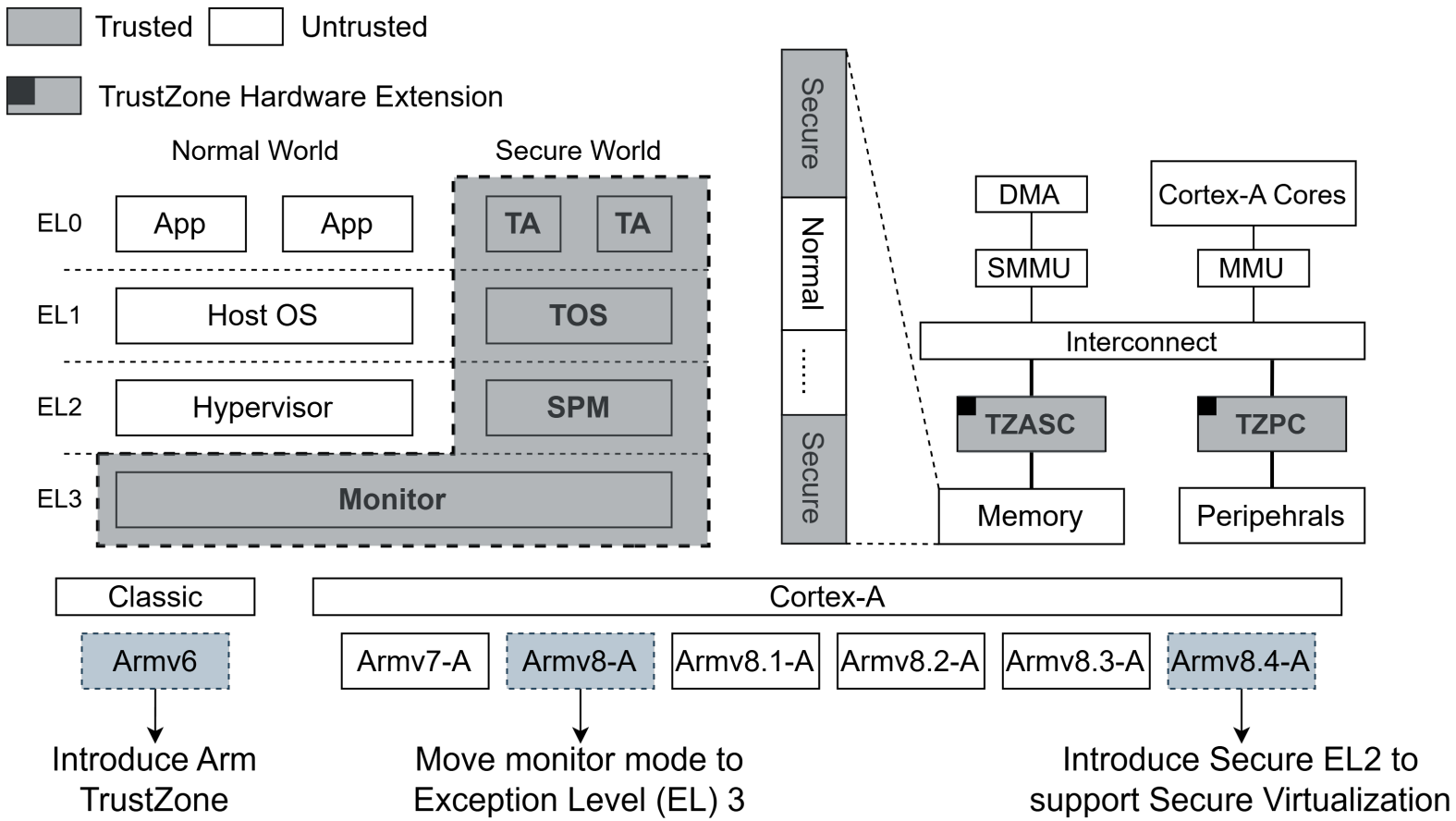
Overview

TrustZone was first introduced in ARMv6 and provides a hardware-based isolation of two execution environments. To date, TrustZone has be update for several times.



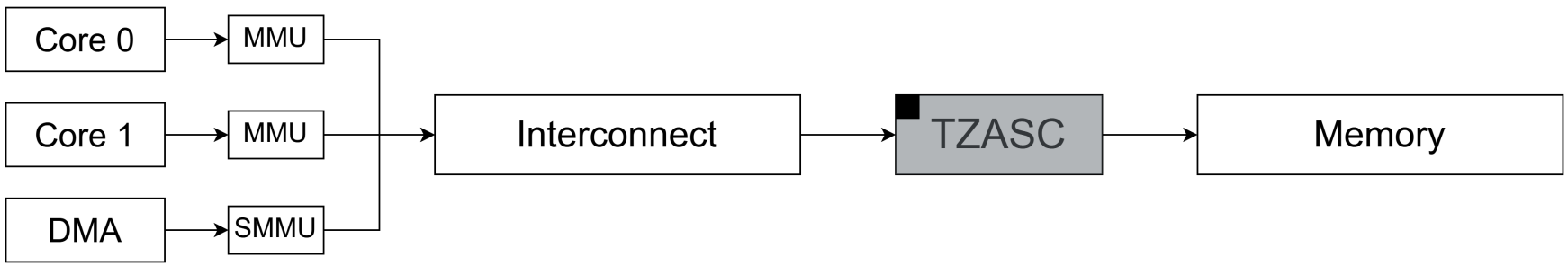
Overview

The whole system with TrustZone enabled is divided into two parts: Normal world and Secure world. **Normal world** hosts complex software like OS in REE. **Secure world** hosts trusted applications (TAs) and a lightweight Trusted OS (TOS).



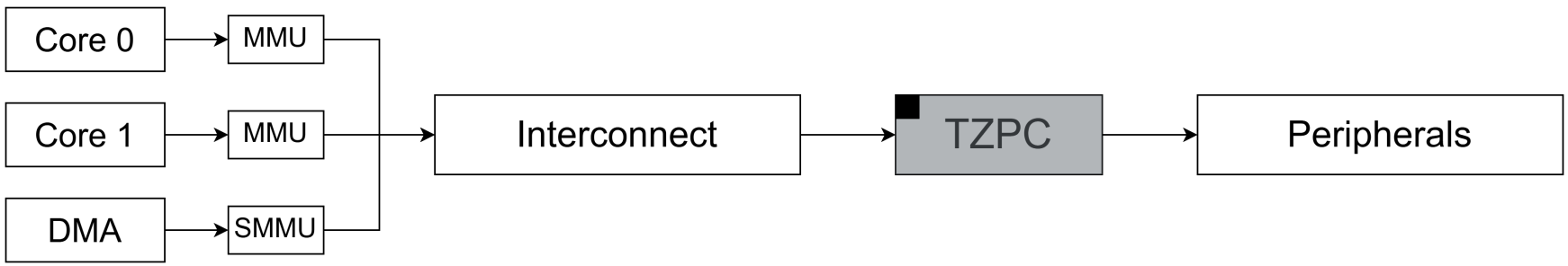
Memory Isolation

TrustZone achieves memory isolation using **TrustZone Address Space Controller (TZASC)**:
TZASC is responsible for enforcing memory access controls between the Secure and Non-Secure Worlds. It defines specific memory regions that are either accessible or restricted based on security requirements.



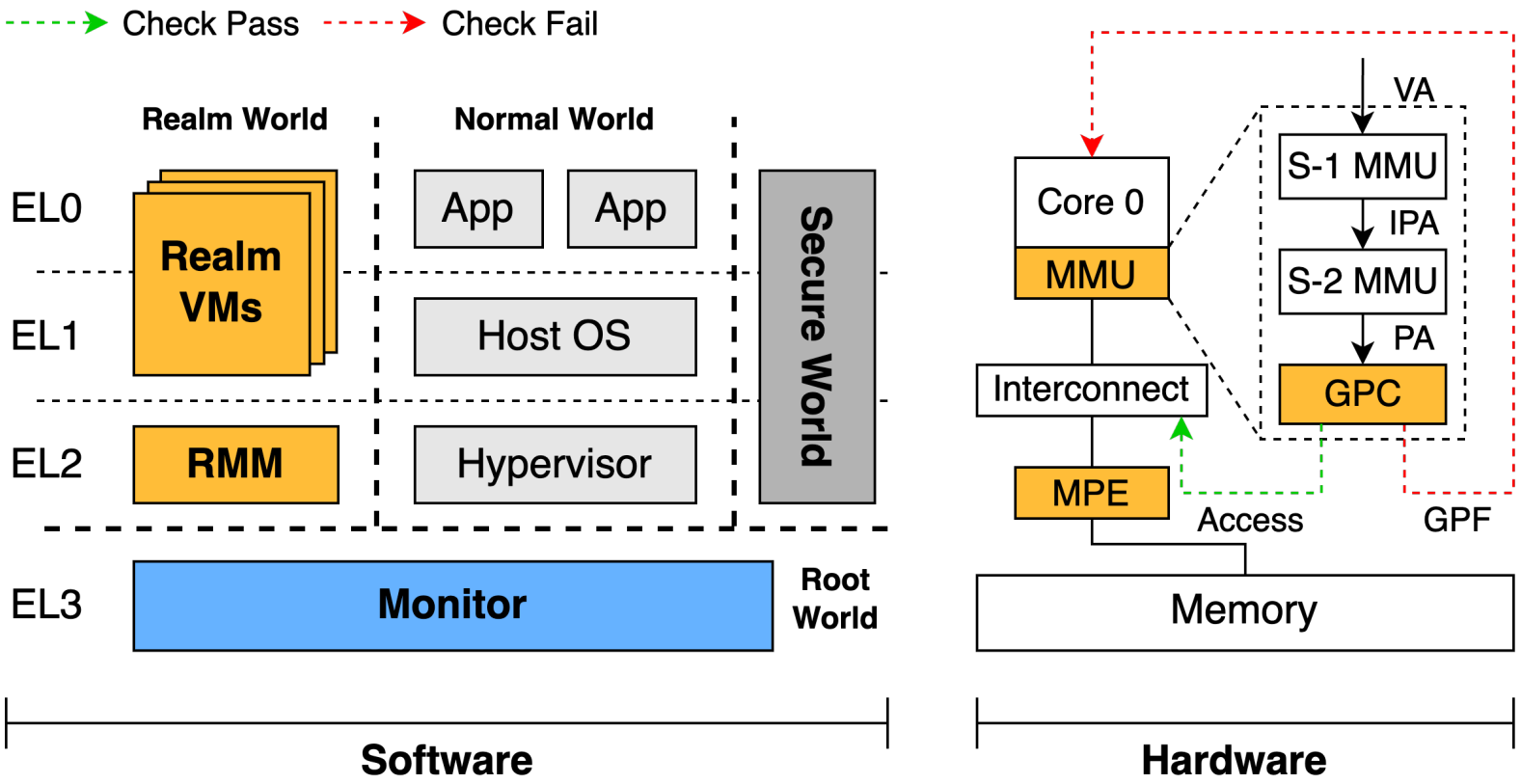
Peripheral Isolation

TrustZone achieves peripheral isolation using **TrustZone Protection Controller (TZPC)**.
TZPC extends TrustZone's isolation capabilities to system peripherals. It allows dynamic configuration of the security state of each peripheral, ensuring that only approved worlds can interact with designated devices.



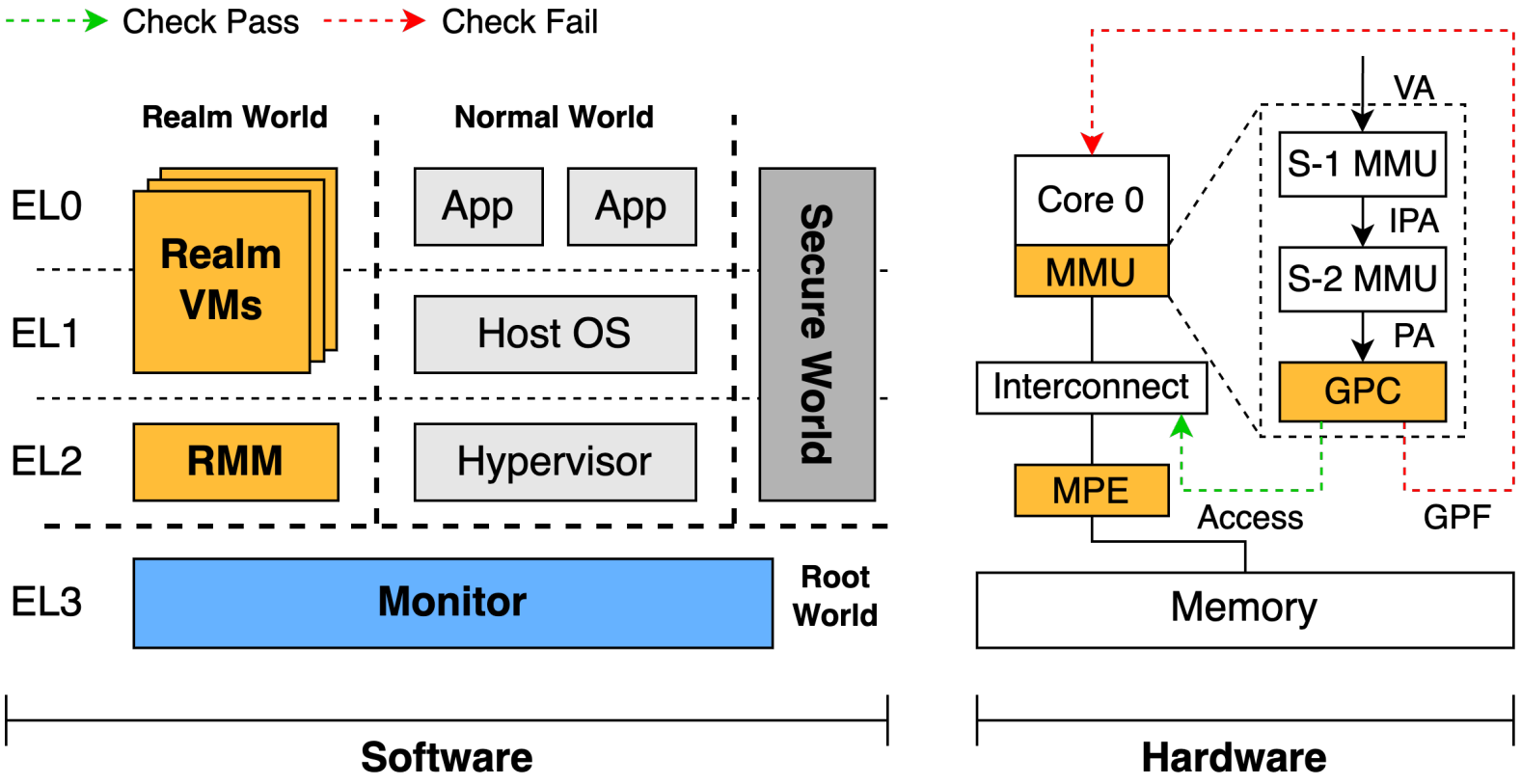
Overview

Confidential Compute Architecture (CCA) was announced in 2021 and introduced as supplement to Armv9.2-A. Compared to TrustZone, CCA introduces two additional worlds: **Root World** and **Realm World**.



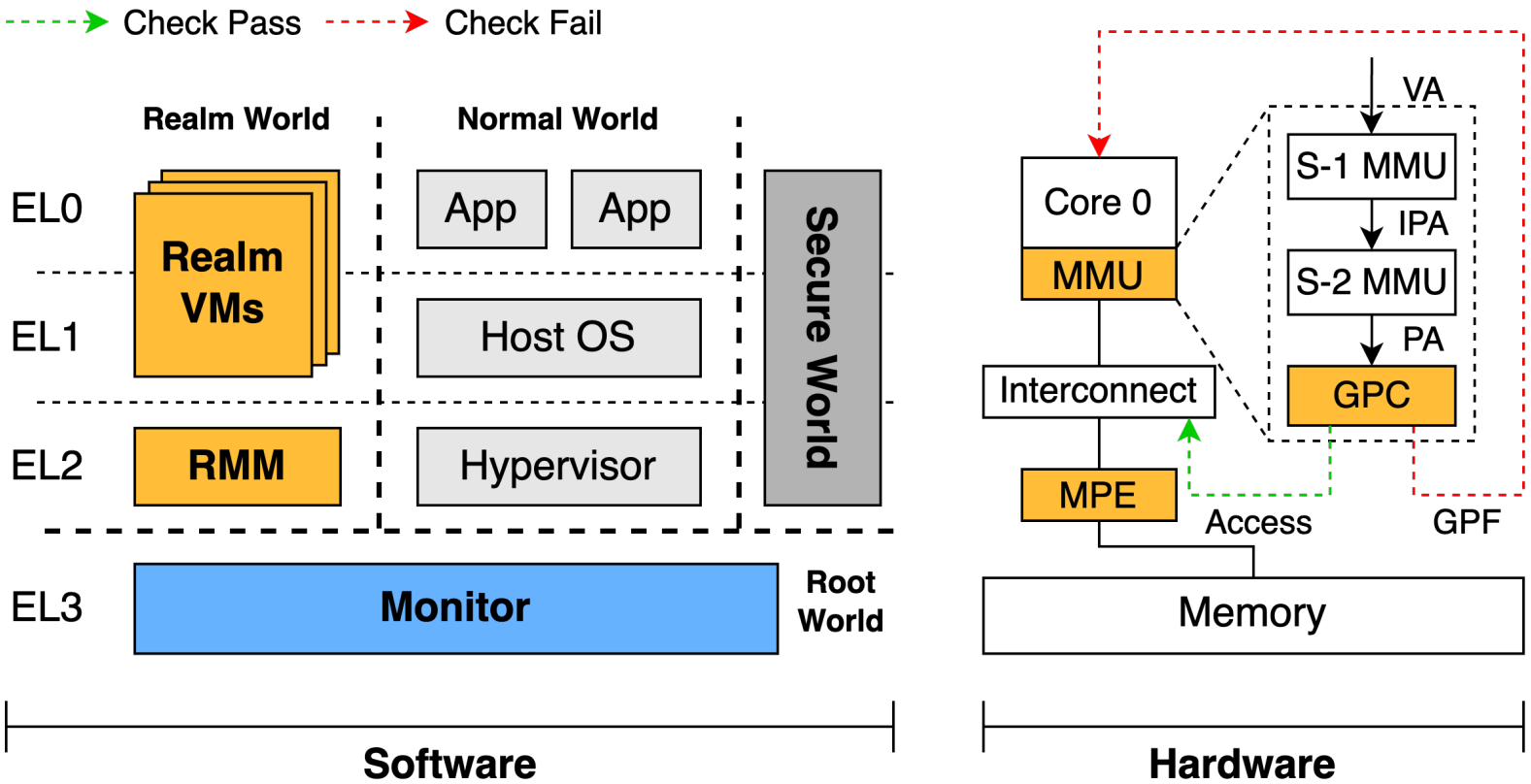
Overview

In TrustZone, code and data in the **highest privilege level** belong to Secure World. In CCA, they are moved to an individual execution environment called **Root World**.



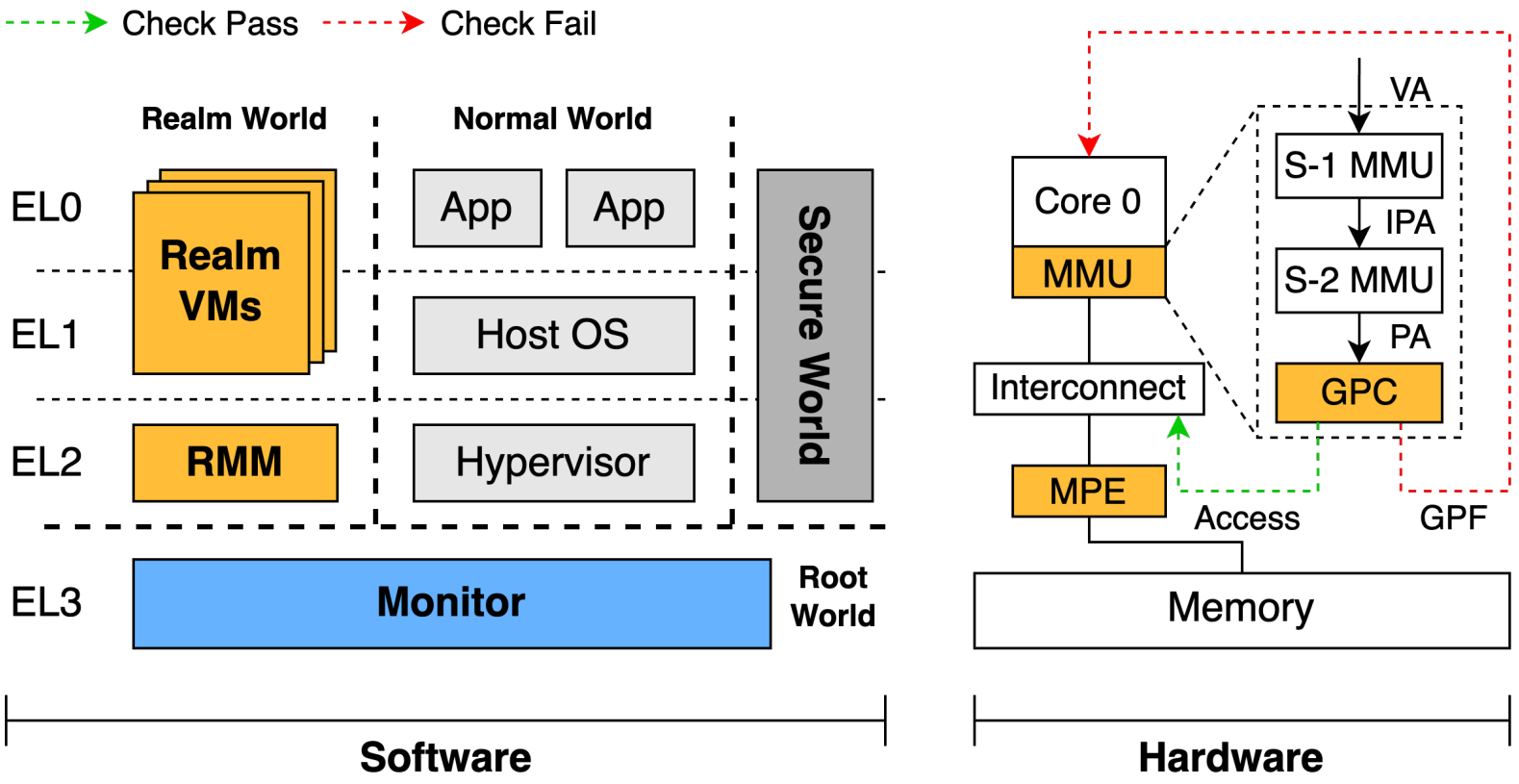
Overview

Realm World is similar to Secure World. However, it is designed to run third-party applications and is isolated from the Normal world and the Secure world. Developers can deploy their applications to Realm VMs for secure execution.



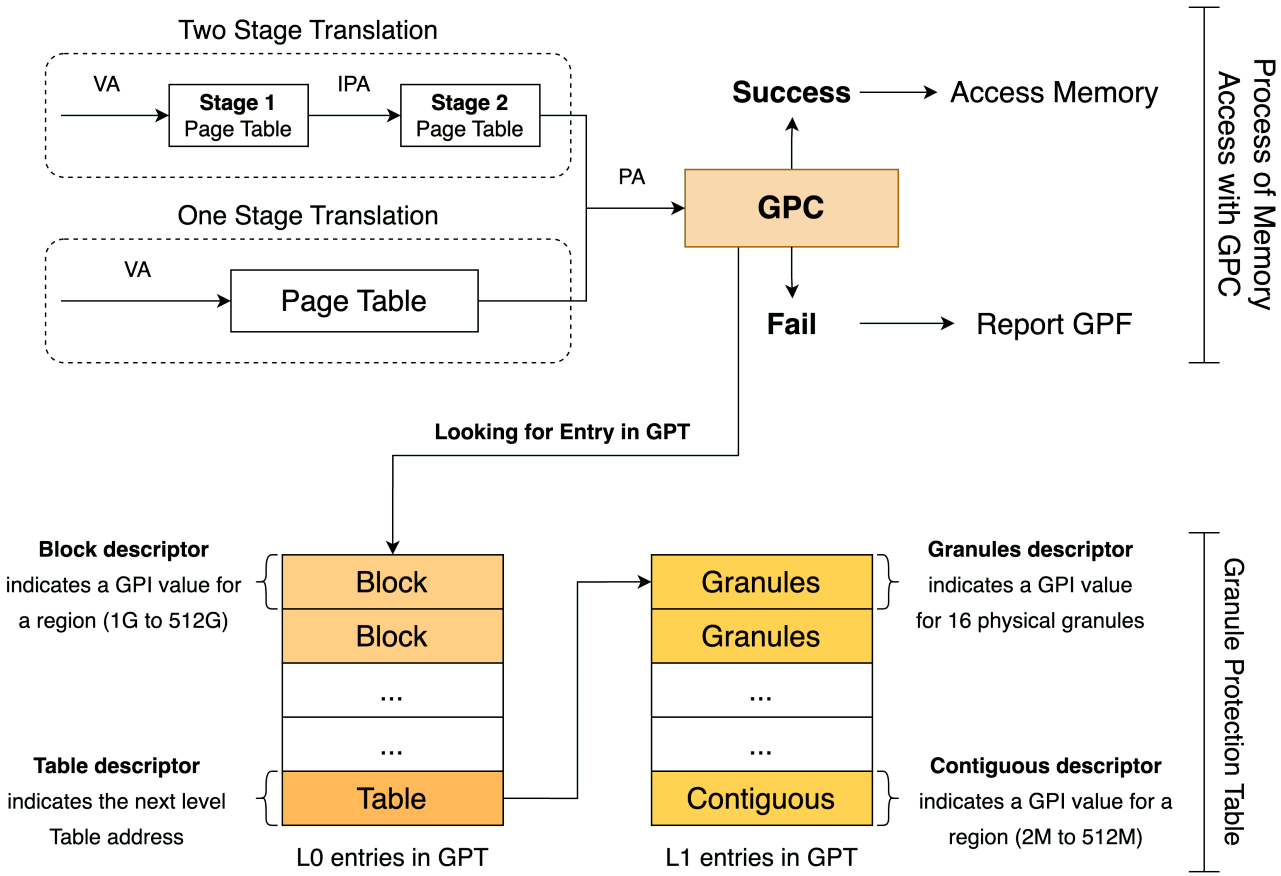
Overview

In Realm World, CCA introduces **Realm Management Monitor (RMM)** to manage Realm VMs. Different from hypervisors, RMM is only responsible for security-sensitive operations, such as memory mapping. The other operations are delegated to the hypervisor running in Normal World.



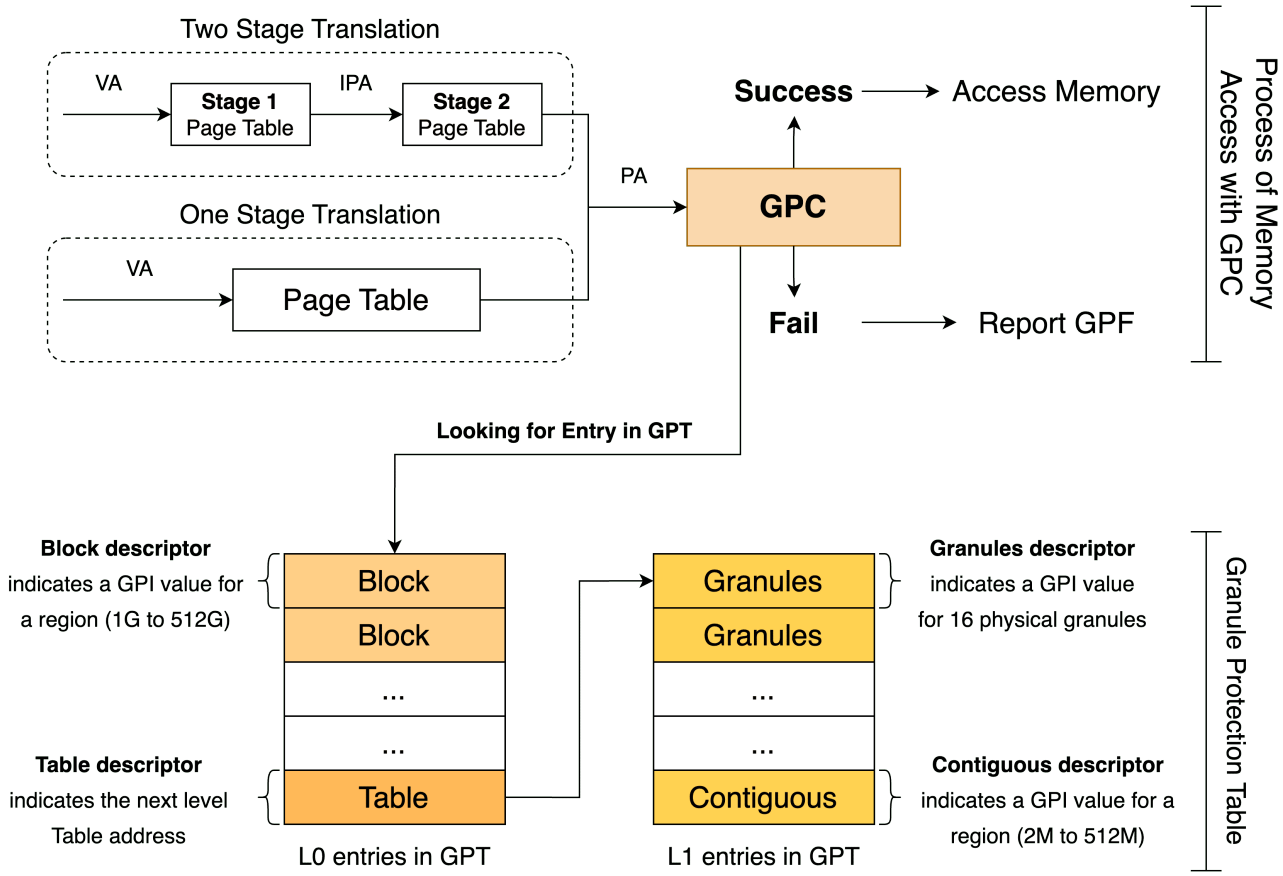
Granule Protection Check

CCA ensures memory and peripheral isolation by extending Memory Management Unit (MMU) with **Granule Protection Check (GPC)**.



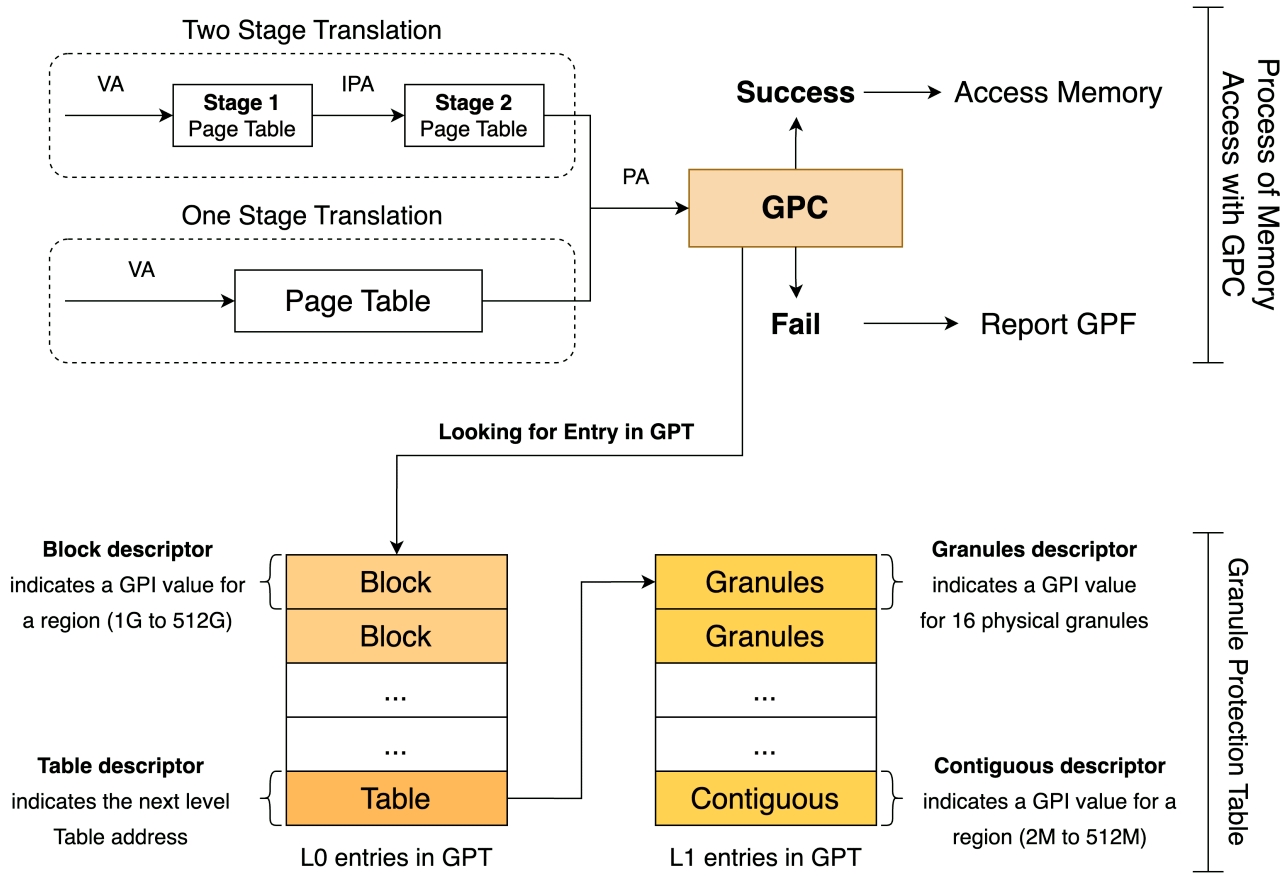
Granule Protection Check

When translating VA to PA, GPC in MMU checks whether the current security state can access the target physical address. When the processor fails in GPC, it will report granule protection check fault (GPF).



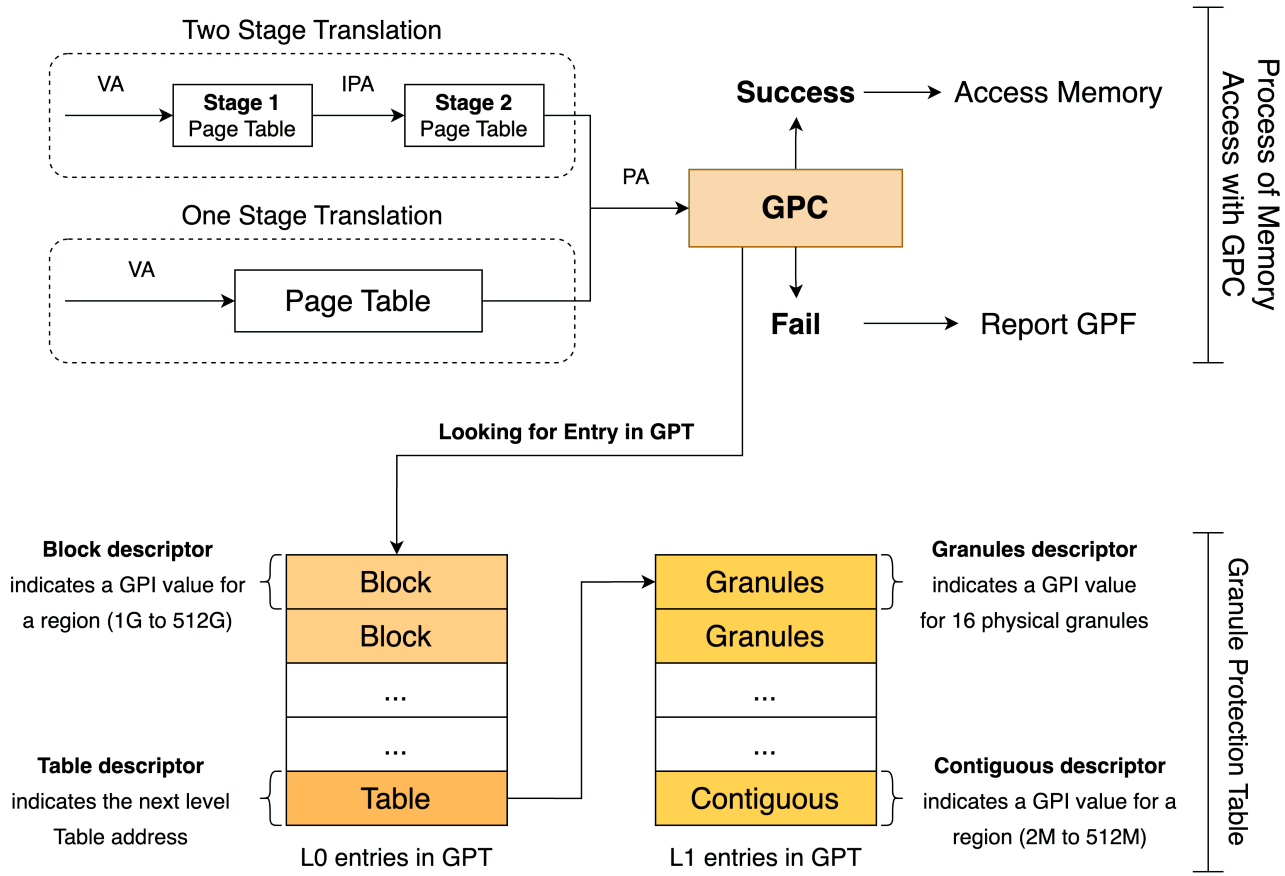
Granule Protection Check

MMU performs GPC based on the **Granule Protection Table (GPT)**, an in-memory structure which is similar to the page table. GPT has two levels to look up, and the entries in the GPT at different levels have different descriptors.



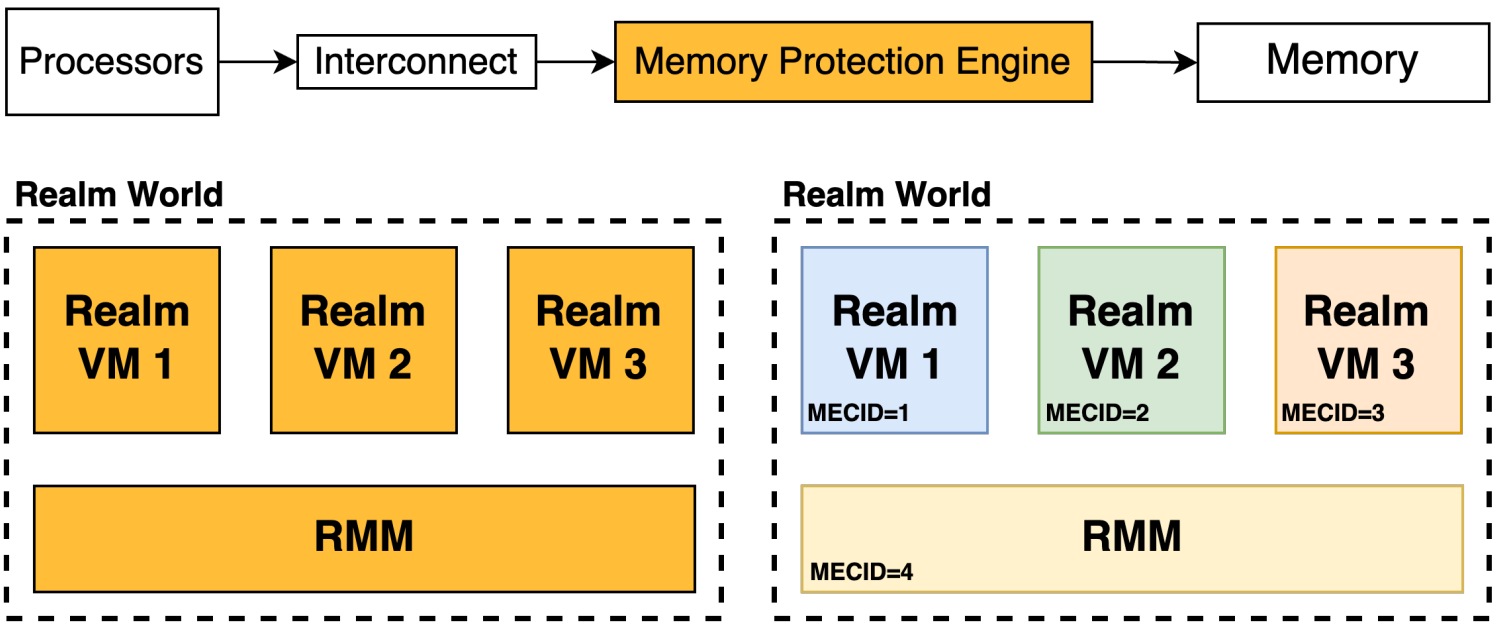
Granule Protection Check

In GPT, **Granule Protection Information (GPI)** indicates the associated world of granules. GPC can block illegal access by checking whether the current security state has rights to access memory based on the corresponding GPI.



Memory Protection Engine

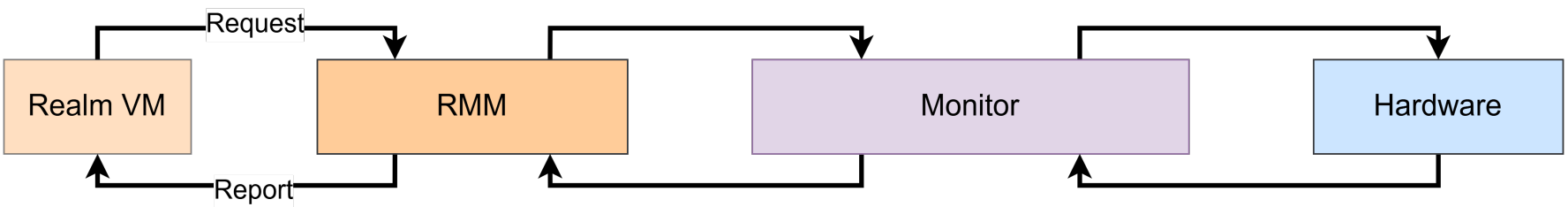
Memory Protection Engine (MPE) is a hardware component that enhances memory security in CCA by providing encryption and integrity for data in the Realm World. However, MPE encrypts the entire memory space with a single key.
To enhance security, CCA introduces **Memory Encryption Context (MEC)**, which allows each memory region in the Realm World to have a unique encryption key.



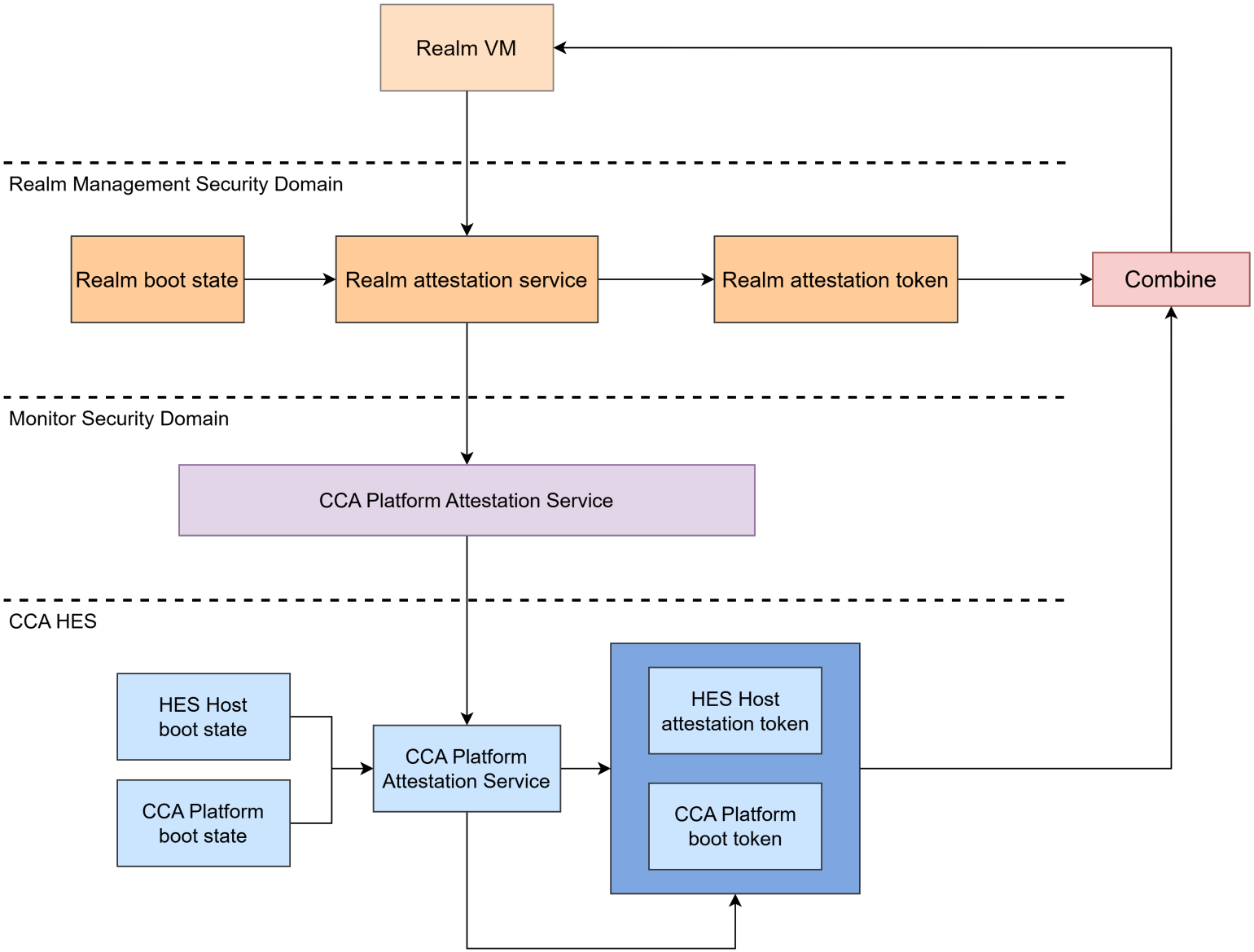
Hardware-assisted Attestation

The **hardware-assisted attestation** brings significant benefits to system security and integrity. It measures the system's state and provides assurance that the software running on the system has not been tampered with or modified since its initial trusted state.

CCA offers hardware-assisted attestation for Realm VMs. The hardware in CCA implements this feature by being bound to a unique identity and supporting the measurement of essential firmware. Users can request a signed attestation report from the hardware to verify the system's integrity and security states.



Hardware-assisted Attestation



COMPASS Lab Publications on Arm

TEE Attacks:

- Understanding the Security of ARM Debugging Features (S&P 2019)

TEE Applications:

- Ninja: Towards Transparent Tracing and Debugging on ARM (USENIX Security 2017)
- Scrutinizer: Towards Secure Forensics on Compromised TrustZone (NDSS 2025)

TEE Designs:

- StrongBox: A GPU TEE on Arm Endpoints (CCS 2022)
- Shelter: Extending Arm CCA with Isolation in User Space (USENIX Security 2023)
- CAGE: Complementing Arm CCA with GPU Extensions (NDSS 2024)

COMPASS Lab Publications on Arm

TEE Attacks:

- **Understanding the Security of ARM Debugging Features** (S&P 2019)

TEE Applications:

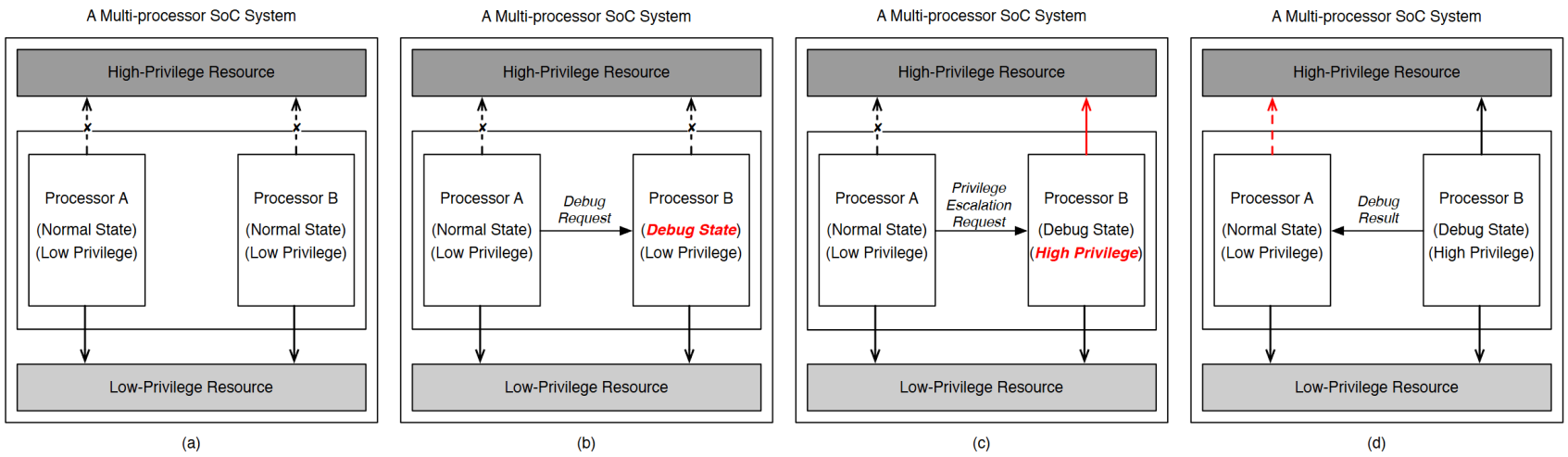
- Ninja: Towards Transparent Tracing and Debugging on ARM (USENIX Security 2017)
- Scrutinizer: Towards Secure Forensics on Compromised TrustZone (NDSS 2025)

TEE Designs:

- StrongBox: A GPU TEE on Arm Endpoints (CCS 2022)
- Shelter: Extending Arm CCA with Isolation in User Space (USENIX Security 2023)
- CAGE: Complementing Arm CCA with GPU Extensions (NDSS 2024)

Understanding the Security of ARM Debugging Features

Nailgun attack[1] obtains sensitive information (e.g., AES encryption key and fingerprint image) and achieves arbitrary payload execution in a high-privilege mode from a low-privilege mode via misusing the debugging features.



[1] S&P 2019.

COMPASS Lab Publications on Arm

TEE Attacks:

- Understanding the Security of ARM Debugging Features (S&P 2019)

TEE Applications:

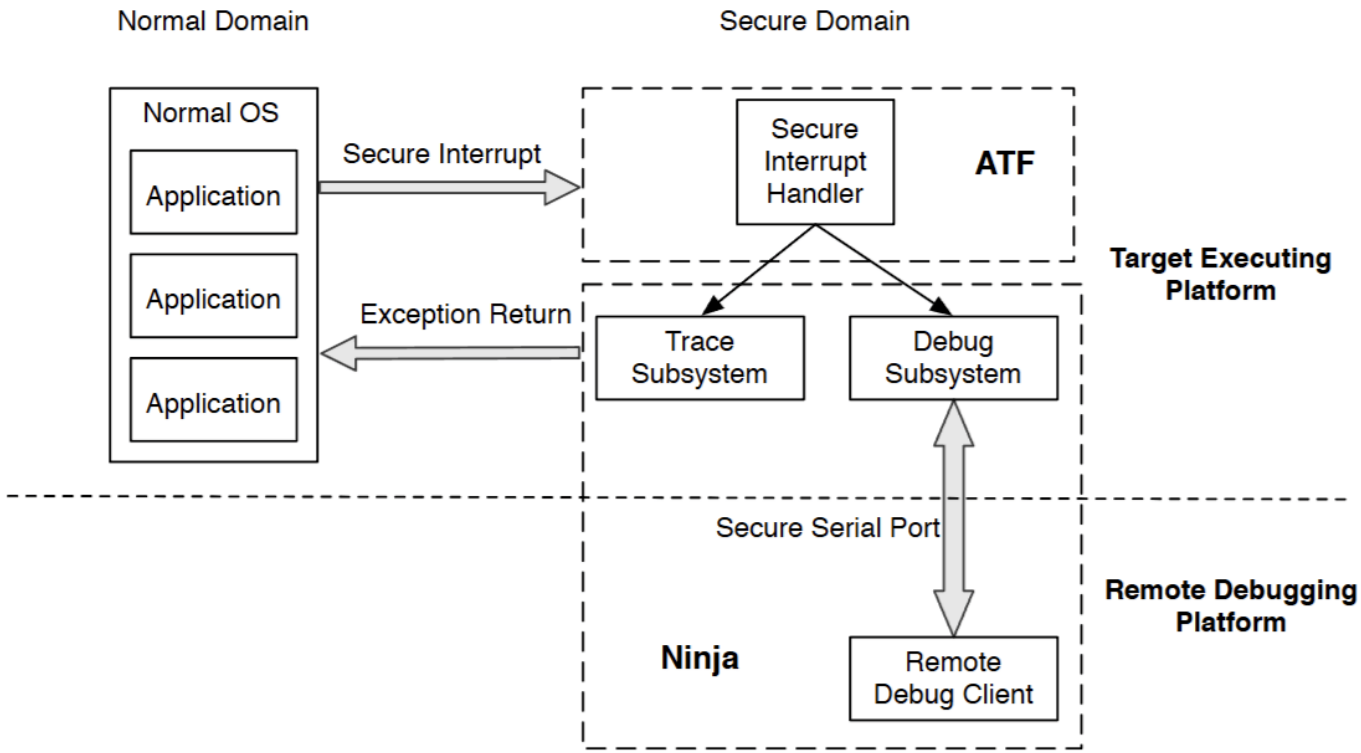
- **Ninja: Towards Transparent Tracing and Debugging on ARM** (USENIX Security 2017)
- **Scrutinizer: Towards Secure Forensics on Compromised TrustZone** (NDSS 2025)

TEE Designs:

- StrongBox: A GPU TEE on Arm Endpoints (CCS 2022)
- Shelter: Extending Arm CCA with Isolation in User Space (USENIX Security 2023)
- CAGE: Complementing Arm CCA with GPU Extensions (NDSS 2024)

Ninja: Towards Transparent Tracing and Debugging on ARM

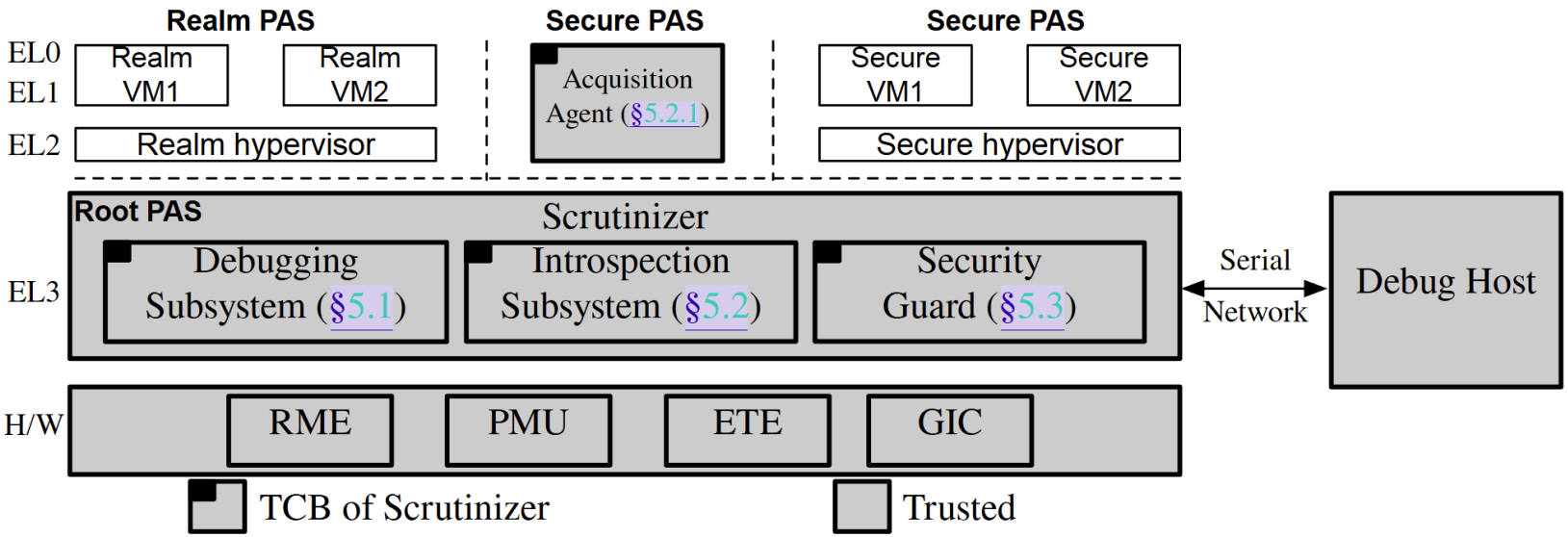
Ninja[2] is a hardware-assisted analysis framework on ARM platforms. It resides in a hardware isolation execution environment, and thus is transparent to the analyzed malware.



[2]USENIX Security 2017.

Scrutinizer: Towards Secure Forensics on Compromised TrustZone

Scrutinizer[3] is the first software-hardware co-design framework that offers debugging and introspection with security guarantees, specifically designed for potentially compromised-TrustZone systems.



[3] NDSS 2025.

COMPASS Lab Publications on Arm

TEE Attacks:

- Understanding the Security of ARM Debugging Features (S&P 2019)

TEE Applications:

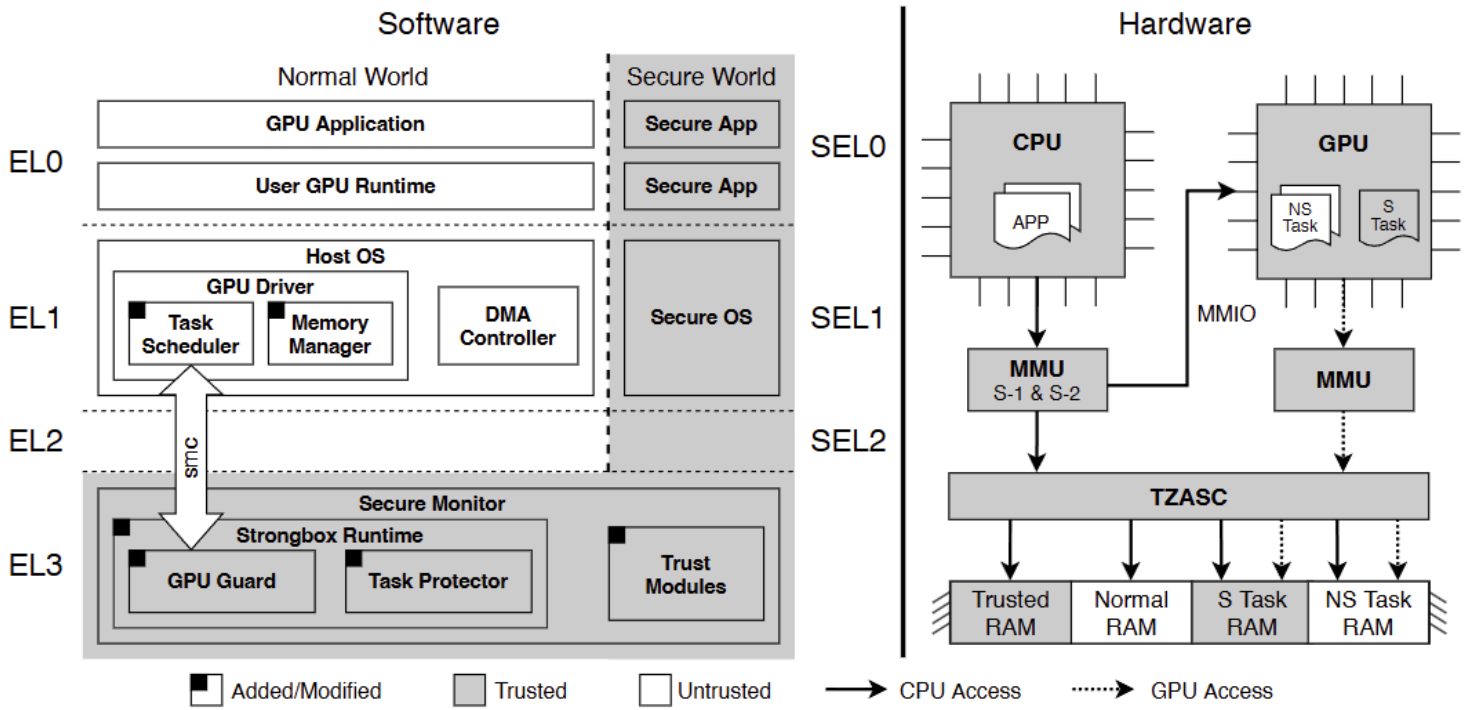
- Ninja: Towards Transparent Tracing and Debugging on ARM (USENIX Security 2017)
- Scrutinizer: Towards Secure Forensics on Compromised TrustZone (NDSS 2025)

TEE Designs:

- **StrongBox: A GPU TEE on Arm Endpoints** (CCS 2022)
- **Shelter: Extending Arm CCA with Isolation in User Space** (USENIX Security 2023)
- **CAGE: Complementing Arm CCA with GPU Extensions** (NDSS 2024)

StrongBox: A GPU TEE on Arm Endpoints

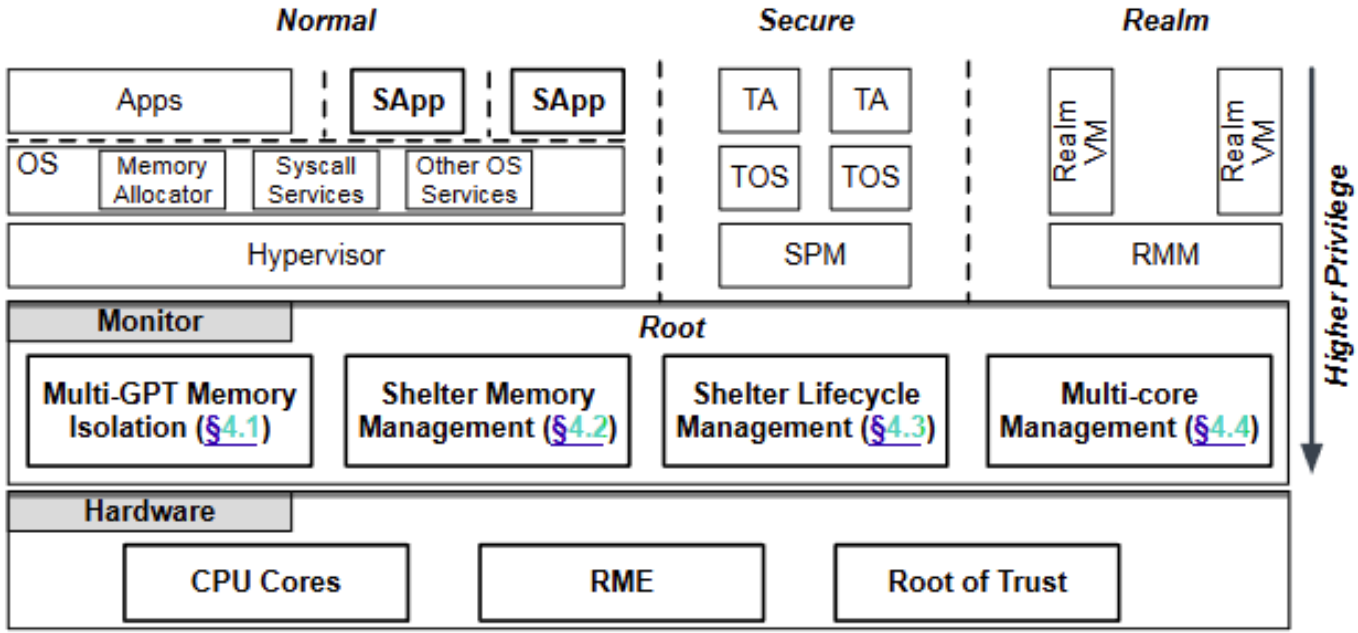
StrongBox[4] is the first unified-memory GPU TEE that runs on Arm endpoints. StrongBox provides an isolated execution environment for secure tasks and protects sensitive data and code from a compromised kernel.



[4] CCS 2022.

Shelter: Extending Arm CCA with Isolation in User Space

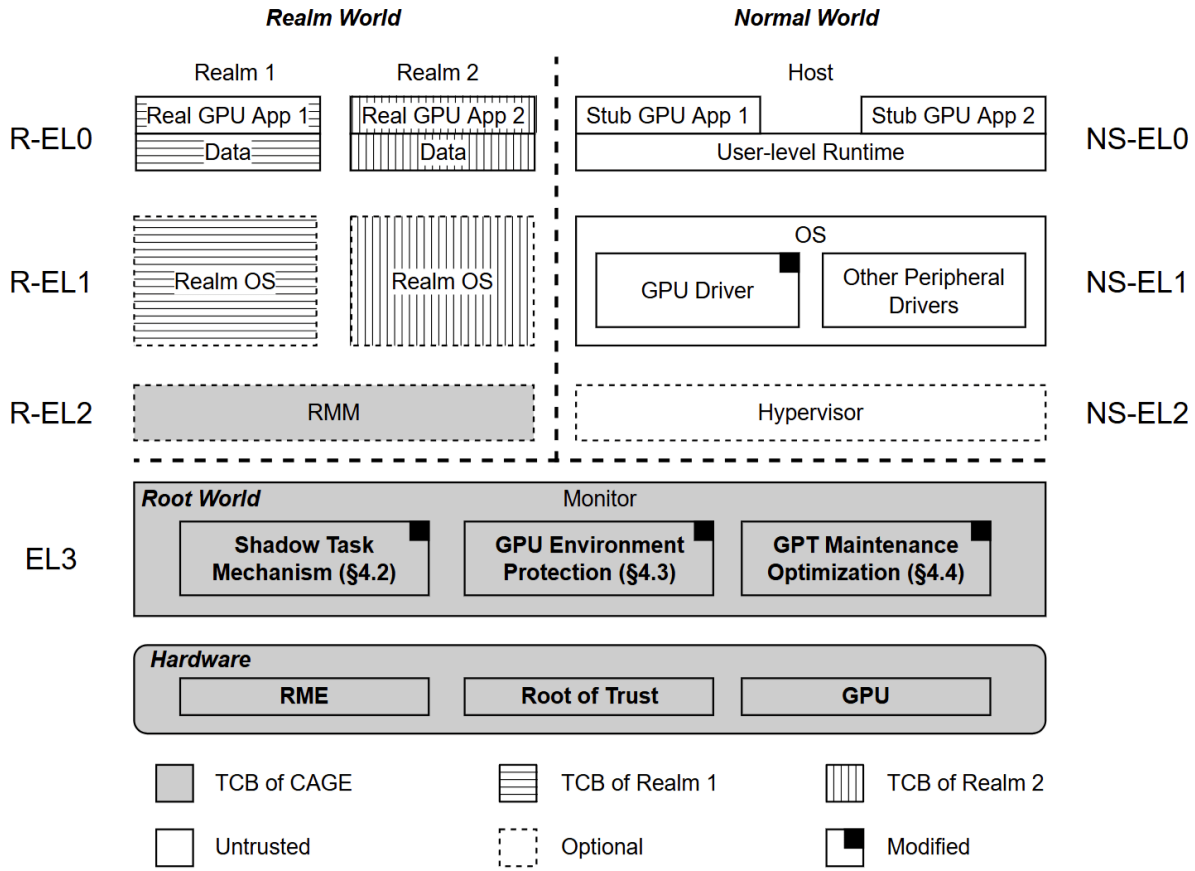
Shelter[5] is an isolated environment in the Normal world userspace via the Arm CCA hardware primitive with a minimal TCB. It utilizes a novel isolation mechanism that deploys multi-GPTs cooperating with Arm RME available in CCA to securely and efficiently protect applications.



[5]USENIX Security 2023.

CAGE: Complementing Arm CCA with GPU Extensions

CAGE[6] provides GPU acceleration support for Arm CCA. In confidential GPU computing, CAGE secures the sensitive data from the strong adversary assumed in Arm CCA.



[6] NDSS 2024.

Thanks!