Discussion on Scalable AP-TEE for RISC-V

Ravi Sahita

Outline (initial discussion)

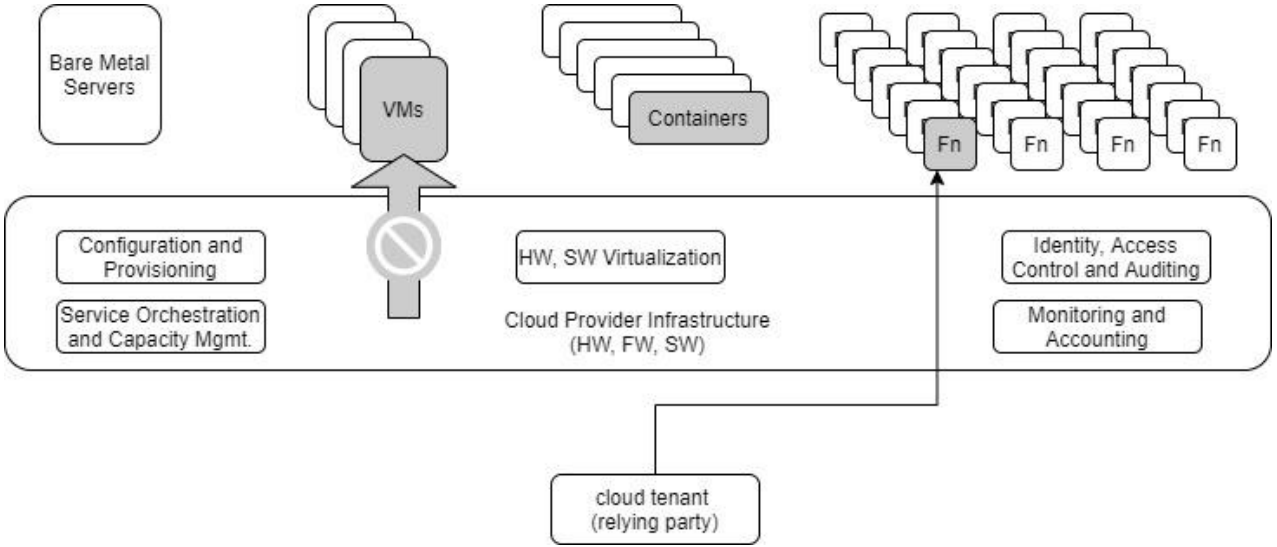
- **What is a TEE?**
- **What use cases do TEEs enable - what threats do they address?**
  - **Hosted cloud (multi-tenancy, operators)**
  - **Edge platform (untrusted physical environment)**
  - **Multi-party compute (privacy-oriented use cases)**
  - **Client platforms (IP-protection, remote workers)**
- **Security requirements of a TEE (vis-a-vis the above use cases)**
- **Other approaches (Commercial)**
- **Related**
  - **Hetero/Accelerator compute**
  - **Attestation Standards**

Trusted Execution Environment

A TEE enables isolated workloads on an application processor (or accelerator), where a hardware-attestable trusted computing base (TCB) enforces confidentiality and integrity of assets loaded within the trusted execution environment.

Use Cases

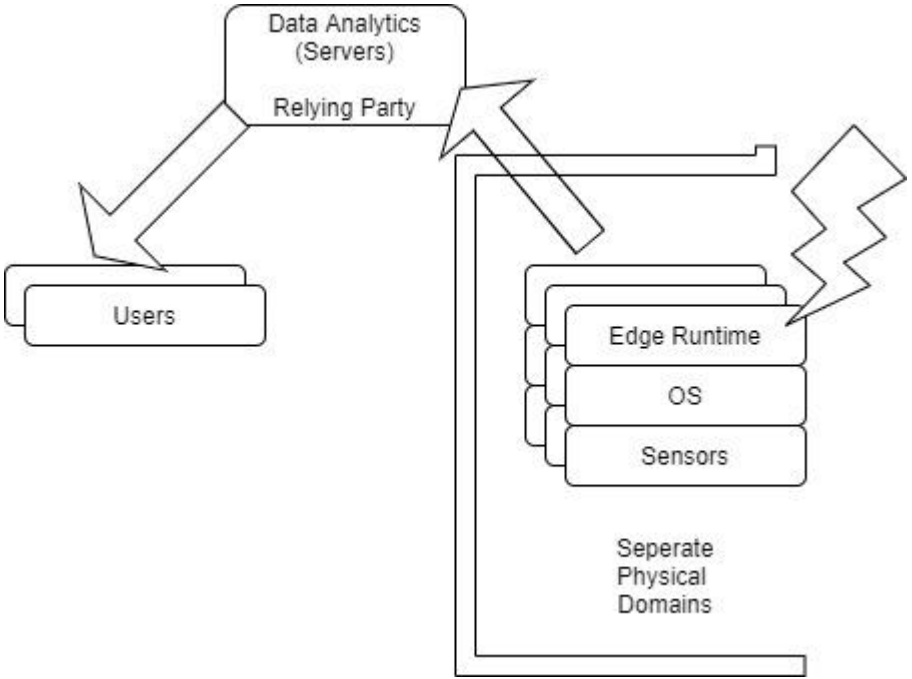
Multi-tenant hosted cloud environment



Unauthorized entities may include anyone with physical access to the hardware, including system administrators, the infrastructure owner, cloud service providers, the host firmware, operating system and hypervisor, other applications and devices on the host.

Use Cases

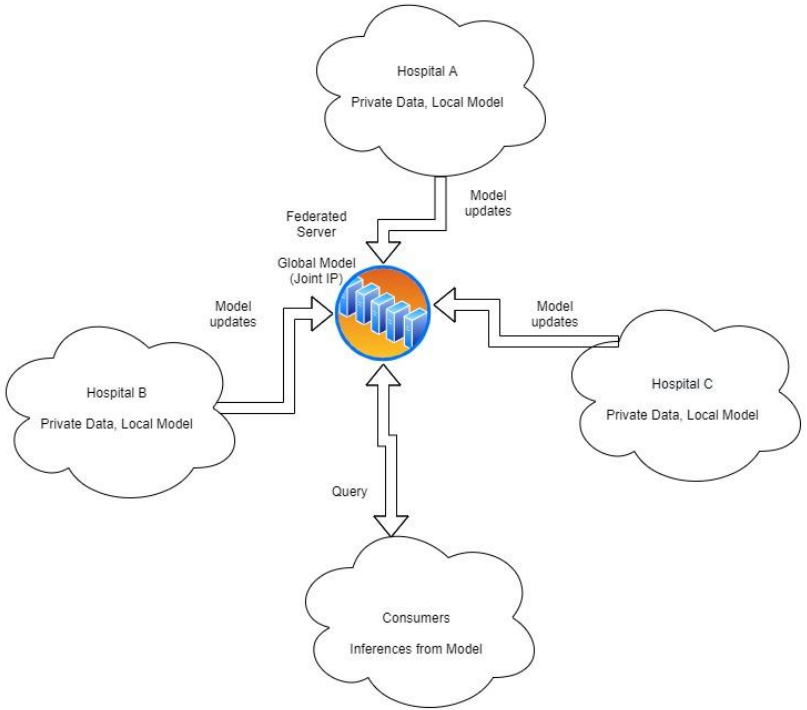
Edge cloud, IOT-based data analytics require TEE to preserve data integrity



Edge cloud and (IoT) devices are generally deemed to be under constant threat of malicious physical access.

Use Cases

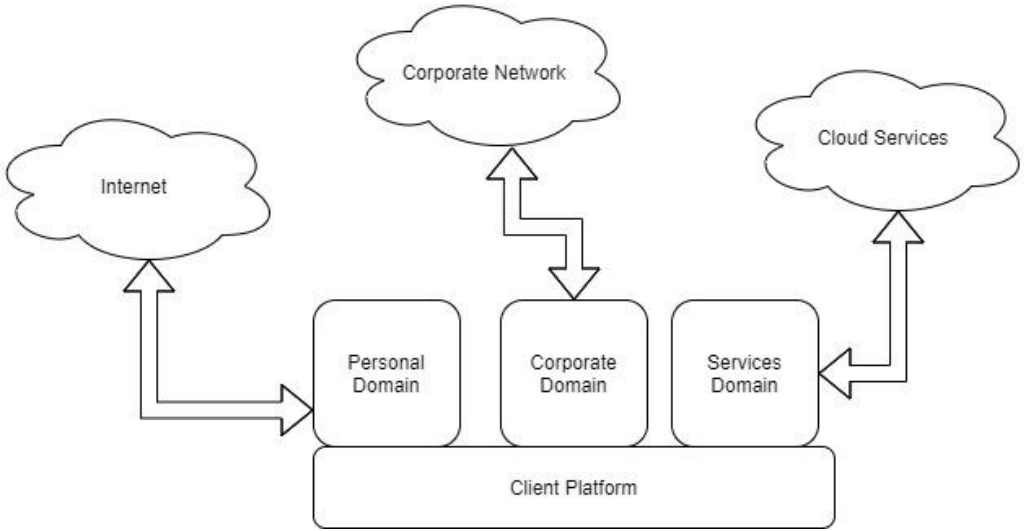
Multi-party computation on CPU/Accelerators



Multiple entities aggregating their proprietary privacy-sensitive data and collaboratively analyzing it to gain new insights (also privacy-sensitive)

Use Cases

Client Platforms



Personal computing devices used to separate personal/corporate roles (trust domains);

Analyze data and build models on the device to reduce the need for off-device processing

Security Requirements (high-level)

- Isolation
- Confidentiality
- Integrity
- Authentication
- HW-rooted TCB
  - Attestation
  - Recovery
  - Updates
- No Dos from tenants

Non-security requirements - DoS from platform host, availability -> RAS/functional goal?

Other (commercial) approaches

AMD SEV-SNP

Provides - Isolation, Confidentiality, Logical Integrity, TCB (A, R, U)

Intel SGX, TDX

Provides - Isolation, Confidentiality, Crypto Integrity, TCB (A, R, U)

ARM CCA

Provides - Isolation, Confidentiality (IP-specific), Logical Integrity, TCB (A, R, U)

Standards, Interoperability

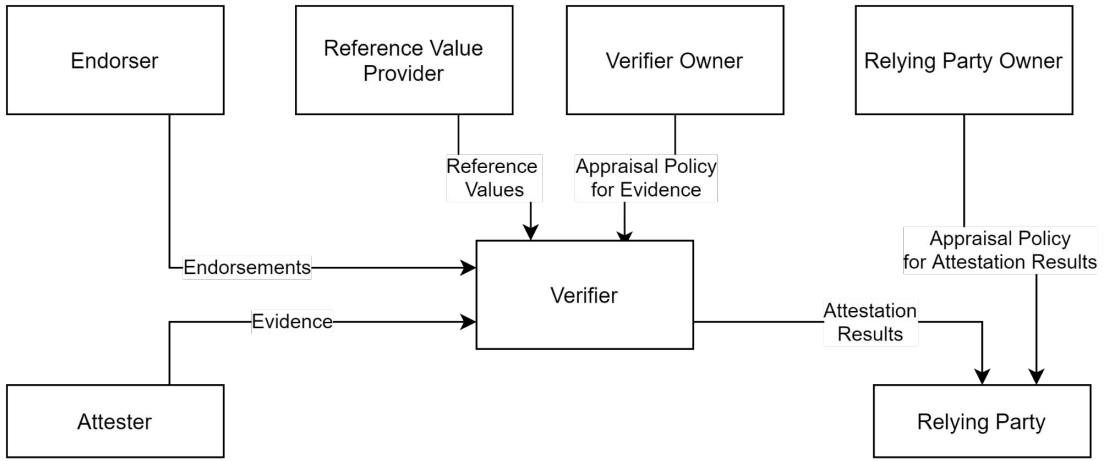
IETF Remote AttesTation procedures (RATS)

TCG Device Identifier Composition Engine (DICE)

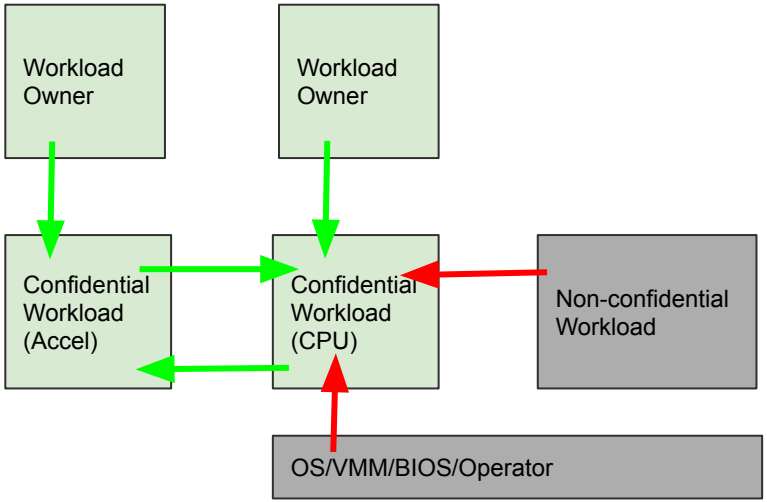
DMTF Security Protocol and Data Model (SPDM)

OpenEnclave SDK

ARM Veraison



Extend to Accelerators



Multi-party computing - two or more different workloads computing collaboratively

- Insurance provider getting patient records and doing data analytics on the data
- Healthcare provider assured of security and compliance to privacy laws
- Insurance provider can assert data protection (insider threats, bugs, etc.)

Outline (follow-on discussion)

Discussion on:

- Goals
- Threat Model
- Approach
  - Runtime Isolation
  - Attestation
  - SW Implications
- Interfaces of interest
- Implementation aspects

Goals

- Primary: Meet a high security bar for workload confidentiality
  - See adversary and threat model on next slides

- Accommodate App, VM, container, other SW deployment models within TEEs
- Minimize software refactoring (for workloads)
- Avoid (new) ISA complexity
- Be able to accommodate future ISA extensions
- Leverage attestation standards, frameworks
- Provide line of sight to confidential IO, migration, snapshot, TCB updates
- Ensure requirements are met for Data-Center, Edge, IOT and other use cases

Threat Model Discussion

Adversary Model

System Software adversary - This includes system software executing in M-mode as well as S- and HS-modes. Such an adversary can access privileged CSRs, all of system memory, CPU registers and IO devices that can be programmed to access system resources (memory and other devices).

Simple Hardware adversary - This includes adversaries that can use hardware attacks such as bus interposers to snoop on memory/device interfaces, which may give the adversary the ability to tamper with data in memory.

Advanced Hardware adversary - This includes adversaries that can use advanced hardware attacks, with unlimited physical access to the devices, and use mechanisms to tamper with the hardware TCB e.g., extract keys from hardware, using capabilities such as scanning electron microscopes, fib attacks, glitching attacks etc.

Threats — *Terminology - TVM: TEE VM (a confidential workload example); TSM: TEE Security Monitor (a TCB element enforcing the confidentiality of TVMs)*

T1: Loss of confidentiality of TVM and TSM memory via in-scope adversaries that may **read TSM/TVM memory via CPU accesses**

T2: Tamper/content-injection to TVM and TSM memory from in-scope adversaries that may **modify TSM/TVM memory via CPU side accesses**

T3: Tamper of TVM/TSM memory from in-scope adversaries via **software-induced row-hammer attacks on memory**

T4: Malicious injection of content into TSM/TVM execution context using **physical memory aliasing attacks via system firmware adverary**

T5: Information leakage of workload data **via read of CPU registers, CSRs** via in-scope adversaries

T6: Incorrect execution of workload via **runtime modification of CPU registers**, CSRs, mode switches via in-scope adversaries

T7: Invalid code execution or data injection/replacement via **second-level paging remap attacks** via system software adversary

T8: **Malicious asynchronous interrupt injection** or denied leading to information leakage or incorrect execution of the TEE

T9: **Malicious hardware mtime register manipulation** or manipulation of time read from the time CSR causing invalid execution of TVM to lead to information loss

T10: Loss of Confidentiality **via DMA access from devices under adversary control** e.g. via manipulation of IOMMU programming

T11: Loss of Confidentiality **via DMA access from devices assigned to a TVM**. Devices bound to a TVM must enforce similar properties as the TEE on the SOC.

T12: Content injection, exfiltration or replay (within and across TEE memory) **via hardware approaches, including via exposed interface/links** to other CPU sockets, memory and/or devices assigned to a TVM

T13: **Downgrading TEE TCB elements** (example M-mode firmware, TSM) to older versions or loading Invalid TEE TCB elements on the platform to enable confidentiality, integrity attacks

T14: **Leveraging transient execution side-channel attacks** to leak confidential data e.g. via shared caches, branch predictor poisoning, page-faults.

T15: **Leveraging architectural side-channel attacks** due to shared cache and other shared resources e.g. via prime/probe, flush/reload approaches

T16: **Malicious access to ciphertext with known plaintext** to launch a dictionary attack on a TVM to extract confidential data.

T17: **Tamper of TVM state during migration** of a TEE workload from one platform to another.

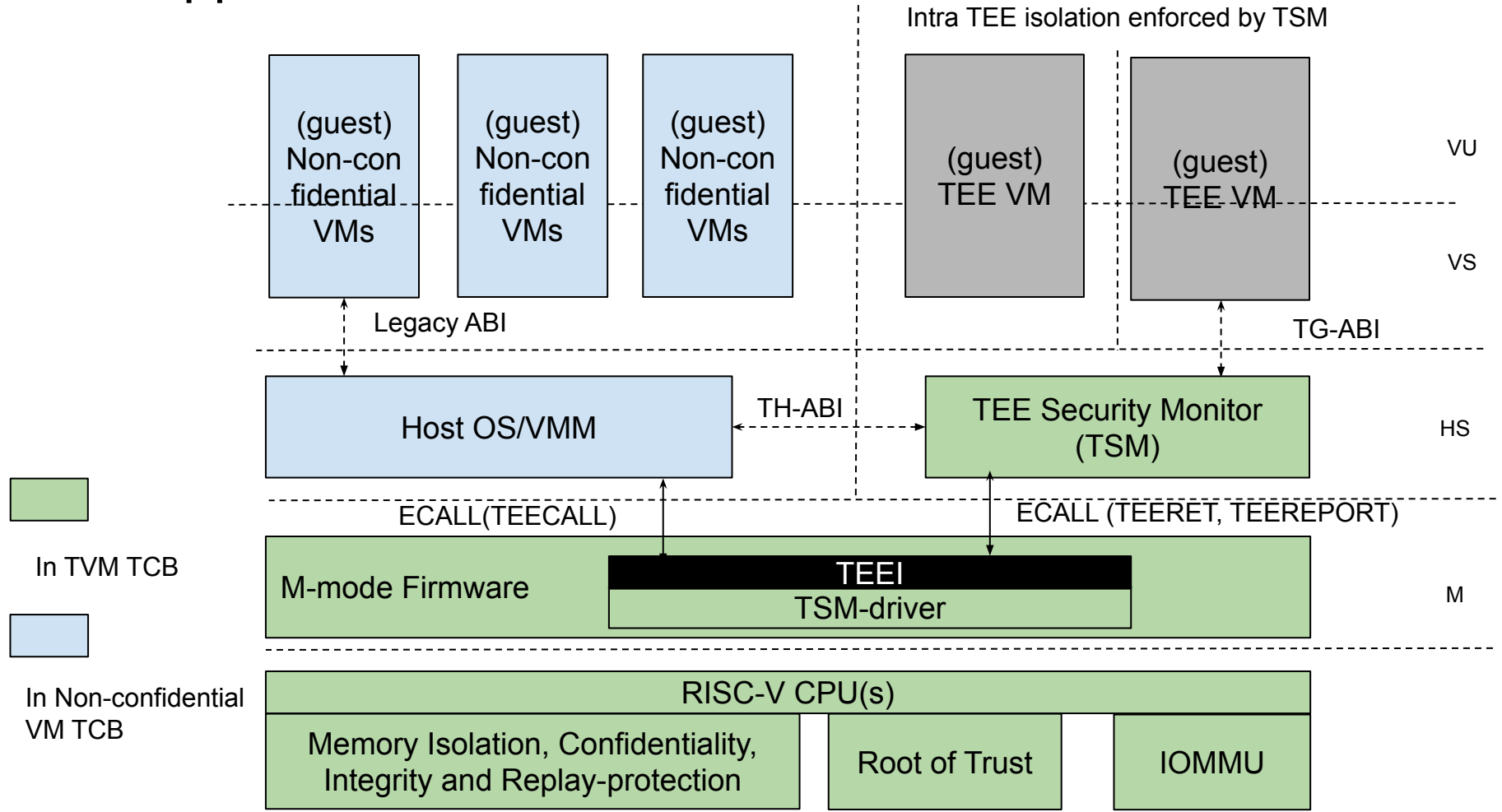
T18: **Forging attestation reports** from the RoT

T19: **Stale TLB translations** (for U/HS mode or for VU/VS) created during TSM or TVM operations are used to execute malicious code in the TVM (or consume stale/invalid data)

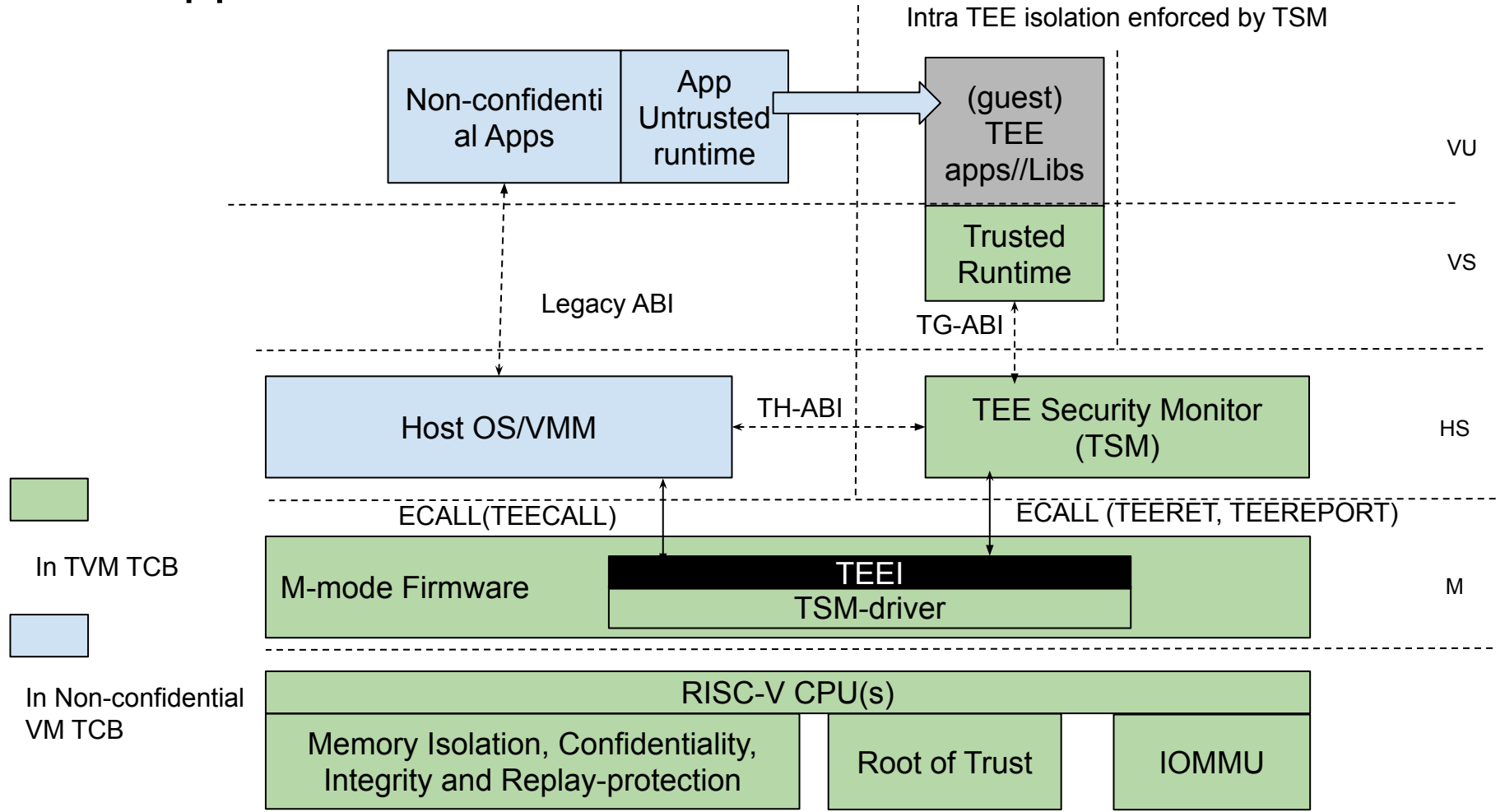
T20: **Unexpected enabling of performance monitoring and/or debug** on a TVM leading to information loss via performance monitoring events/counters and debug mode accessible information.

T21: A **TVM causes a denial of service** on the platform

Approach


TEE/non-TEE isolation provided by CPU
Intra TEE isolation enforced by TSM
(guest) Non-con fidential VMs
(guest) Non-con fidential VMs
(guest) Non-con fidential VMs
(guest) TEE VM
(guest) TEE VM
VU
VS
Legacy ABI
TG-ABI
Host OS/VMM
TH-ABI
TEE Security Monitor (TSM)
HS
ECALL(TEECALL)
ECALL (TEERET, TEEREPORT)
M-mode Firmware
TEEI
TSM-driver
M
In TVM TCB
In Non-confidential VM TCB
RISC-V CPU(s)
Memory Isolation, Confidentiality, Integrity and Replay-protection
Root of Trust
IOMMU

App workloads



Role of M-mode

- M-mode firmware programs HW to provide the isolation of the TSM memory via following mechanisms:
  - Cryptographic mechanisms (confidentiality, integrity) and/or
  - Access-control mechanisms (page ownership tracking or sequestered memory)
  - Performs context switch from host OS/VMM to TSM and vice versa (TEE ABI functions invoked via ECALL to M-mode (TEECALL, TEERET, TEEREPORT)
    - Activates “TEE mode” on the hart
- M-mode programs HW engines for access-control, memory confidentiality, and RoT for TSM reporting.

Role of the TSM

TSM enforces isolation of TVM memory (amongst TVMs) via 2nd stage translation

TSM enforces isolation of TVM hart state when invoked by the host VMM to schedule TVM virtual-hart on a physical hart.

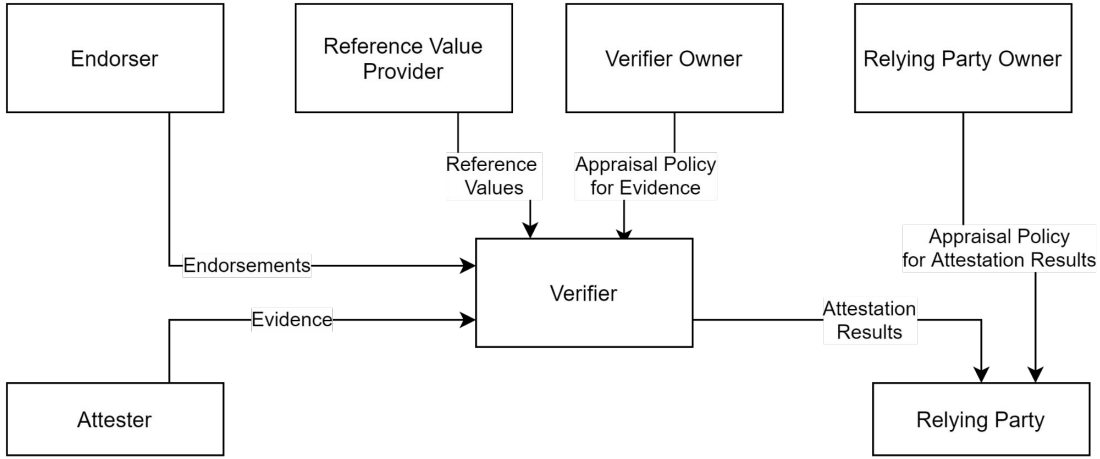
TSM maintains TVM measurement for attestation

Important TSM attributes

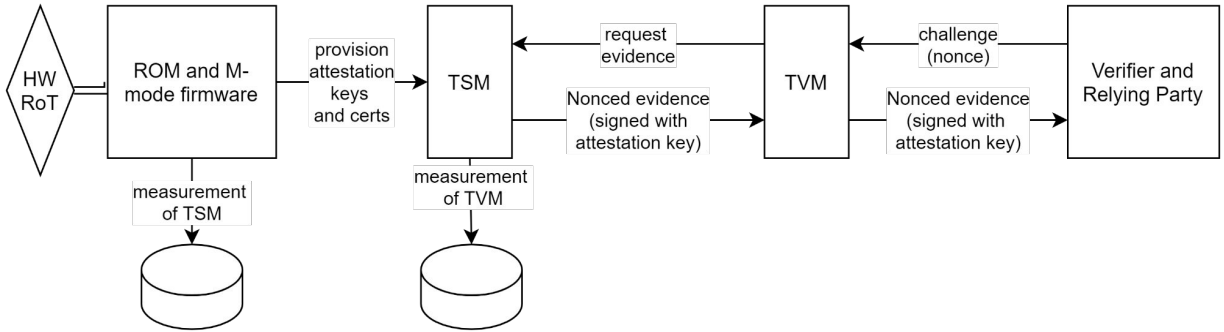
- The TSM is stateless across TEECALL invocations.
- The TSM does not perform any dynamic resource management, scheduling, interrupt handling etc.
  - All memory is assigned by the OS/VMM but access-controlled by the TCB once assigned to the TSM/ TVMs.
  - Expect TSM to be much smaller than OS/VMM in terms of size and complexity
- TEE and TVM address spaces are identified by an additional hart **AP-TEE mode qualifier** to maintain isolation during access and in internal caches, e.g. Hart TLB lookup may be extended with the AP-TEE mode qualifier.

Attestation

Remote Attestation Procedures:



TSM-based Attestation of a TVM:



DICE Attestation Architecture describes evidence as X.509 Certificate with TCB Info Evidence Extension - other formats possible as well

AP-TEE TEEI must cover two aspects

TH Interface:

- TSM lifecycle management
- TVM global - create, destroy
- TVM static measurement
- TVM memory management
- TVM virtual-hart management
- TVM context switch

TG Interface

- TVM measurement extension
- TVM memory management
- TVM report generation
- TVM explicit host service invocation
- TVM performance monitoring and debug

Next Steps

- Discuss and ratify adversary, threat model and high level approach/requirements in this Trusted Computing SIG

- Form an AP-TEE TG to build upon directions - specifically,
  - Discuss and review concrete proposals in this TG
    - Formalize impact on ISA and SW ABIs
    - Interaction with other ISA, platform components such as IOMMU, SCA
  - Form and review a TEEI ABI within AP-TEE TG
  - Review proposed ABI within Software, Hypervisor and Trusted Computing SIGs