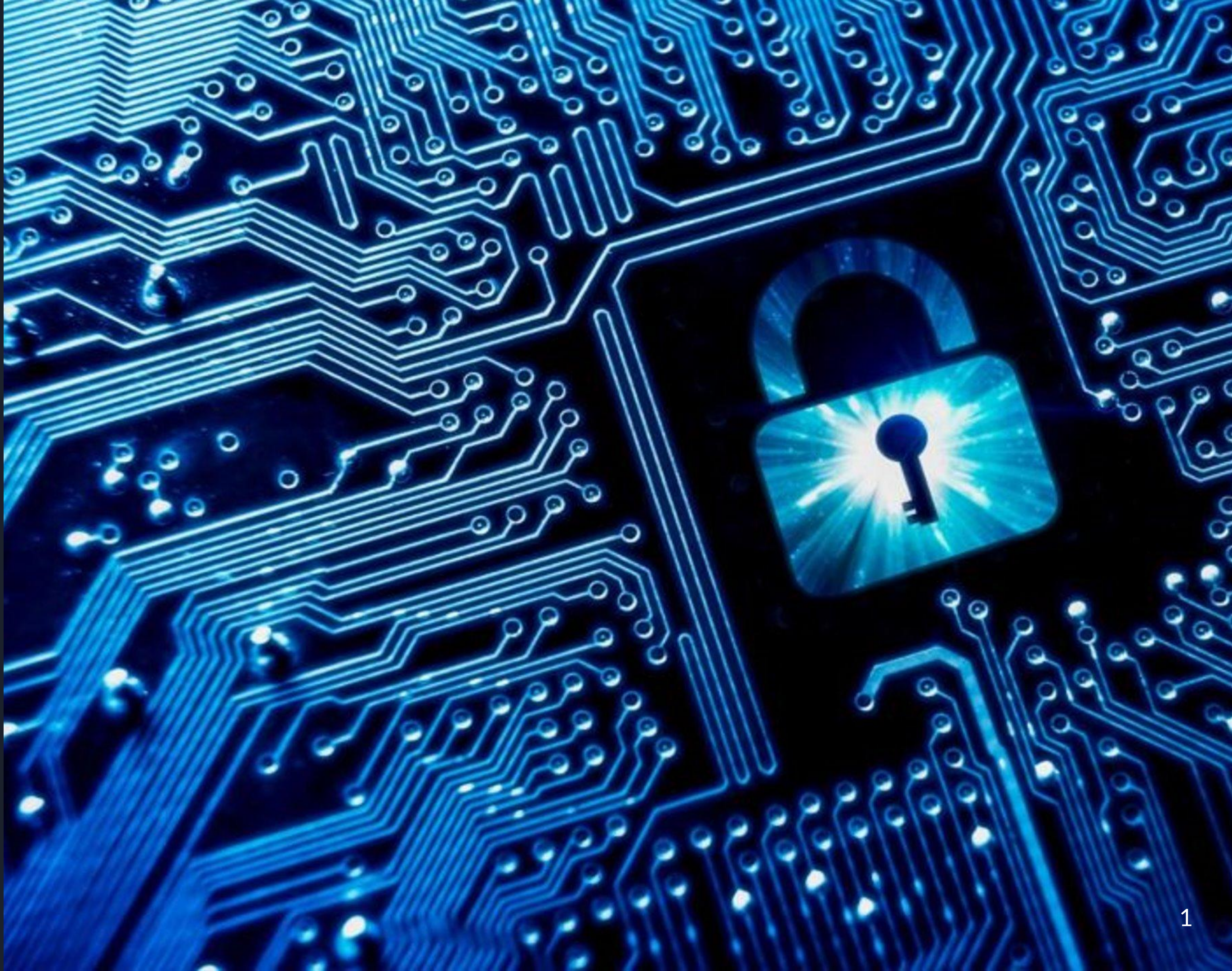
Confidential Computing for RISC-V Platforms

Ravi Sahita

For RVI Security HC
Trusted Computing SIG
AP-TEE TG

10/5/22



Outline

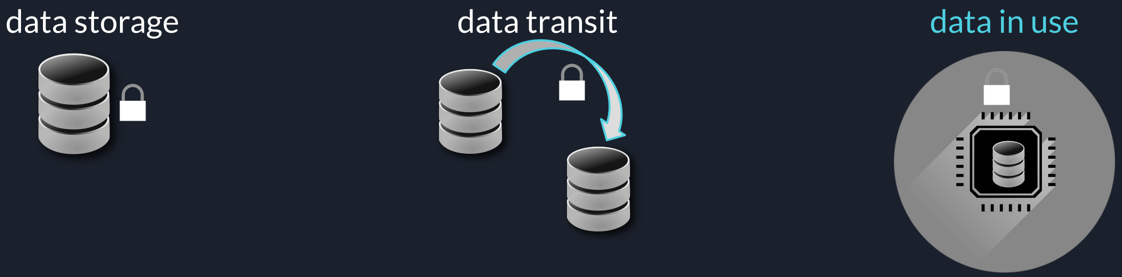
Introduction to Confidential Computing

Use Cases and Threat Model

Implications on RISC-V ISA and non-ISA

RVI approach and status

Confidential Computing



Confidential Computing is the protection of data in use by performing computation in a Hardware-based Trusted Execution Environment (TEE).

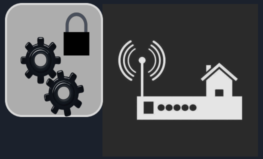
- *This definition is independent of topological location, which processor does it, and whether encryption or some other isolation technique is used.*
- *The protection of data in use is against a well-defined adversary.*

https://confidentialcomputing.io

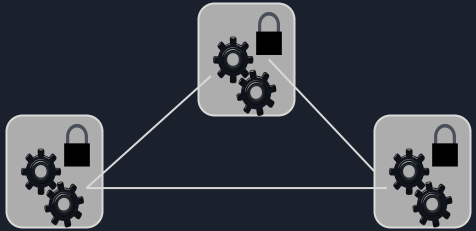
Confidential Computing applies to many RISC-V deployment scenarios

Multi-tenant hosted cloud platforms

Multi-domain Platforms - Client, Automotive



Edge processing & On-Prem deployment



Multi-party computation on CPU / Accelerators

Key properties of a HW-based TEE for Confidential Computing

A Trusted Execution Environment (TEE) is an environment that provides a level of assurance of three key properties:

- Data confidentiality
- Data integrity
- Code integrity

Additional desirable characteristics:

- Code confidentiality
- Authenticated Launch
- Programmability
- Attestability -- *This is a required from the RISC-V Trusted Computing SIG perspective*
- Recoverability

https://confidentialcomputing.io

Additional Goals and Principles

- Primary: Meet a high security bar for workload confidentiality
  - Hardware-based Attestable Security Properties - Confidentiality, Integrity, [Not Availability]
  - More on adversary and threat model coming up

- Accommodate wide SW deployment models within TEE (VM, App, Container, other…)
- Natively support multi-tenancy
- Avoid software refactoring (No application changes)
- Avoid unnecessary ISA complexity; Be able to accommodate future ISA extensions
- Leverage and contribute to developing attestation standards
- Design to support key TCO features - rebootless TCB updates, confidential IO, migration, …
- Ensure collateral requirements (Debug, QoS, RAS) are preserved/met for Data Center, Edge, …

Threat Model (Attack Categories)

User/System Software attacks

Protocol attacks

Cryptographic attacks

Basic hardware attacks

uArch and Arch Side-channel attacks

Basic upstream supply-chain attacks

Upstream hardware supply-chain attacks

Advanced hardware attacks

Detailed RVI Confidential Computing threat model has been defined and documented here.

The RISC-V Trusted Computing SIG does not aim to specify any threats from this set as out of scope - noting that different implementations will have varying degrees of resistance to these attacks.

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.

Conf. Comp. on RISC-V → AP-TEE TG (Application Processor TEE)

Why should we do this? And why now?

- Confidential Computing is at an inflection point and all compute domains/market segments (Data Center/Servers to Embedded) require support for it

What is our proposed action plan?

- **AP-TEE TG to specify**
  - Reference Architecture and Interfaces (Non-ISA, normative)
  - AP-TEE Security Arch / Implementers Guide (informative living doc).
  - Identify ISA proposal(s) to propose via Priv TG (ISA, normative)

Who else do we work with?

- **Within RVI -** Other HC/SIG/TGs
- **Outside RVI -** Confidential Computing Consortium (CCC), Trusted Computing Group (TCG), Internet Engineering Task Force (IETF), Distributed Management Task Force (DMTF), GlobalPlatform, PCIe, CXL

Intel SGX, TDX
AMD SEV-ES-SNP
ARM Trustzone, CCA
IBM PEF
RISC-V AP-TEE

CCC
Open Enclave SDK
Keystone
Project Veraison

IETF RATS
TCG DICE
DMTF SPDM
PCIe IDE, TDISP

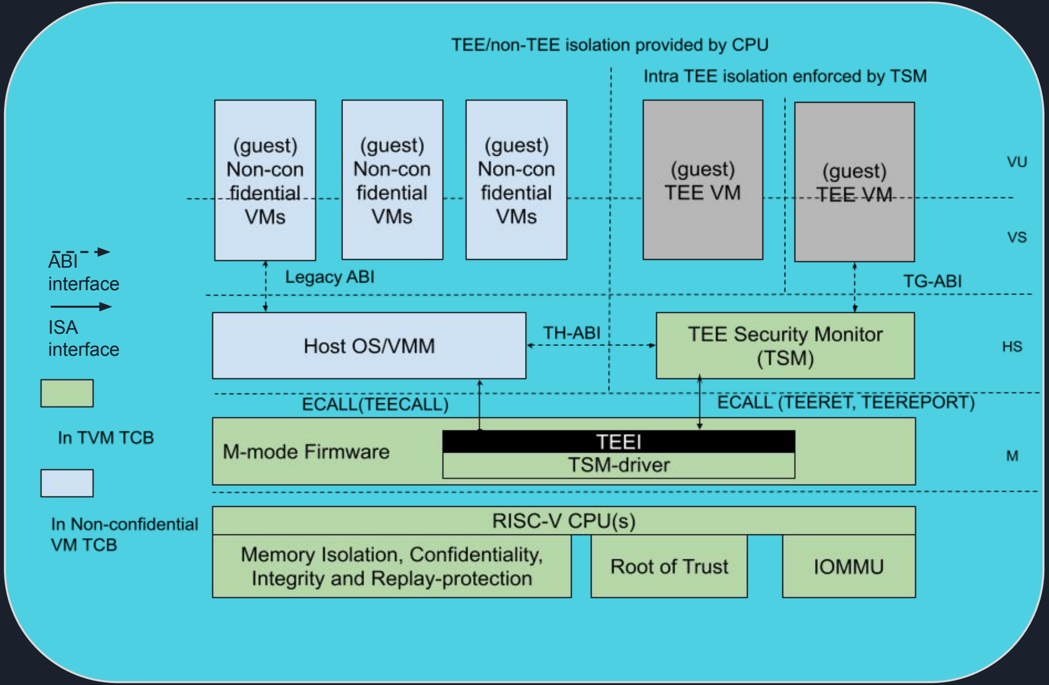
AP-TEE RVI Reference Architecture

- Reference architecture for confidential computing on RISC-V platforms

- AP-TEE TH/TG-ABIs -- normative non-ISA spec.

- AP-TEE Security Arch/Implementers Guide for RISC-V confidential computing - covers platform recommendations -- informative (living spec).

- AP-TEE ISA extension(s) -- *start with current ISA*; identify ISA gaps in TG -- request FT/TG as needed -- expected to be normative spec.



TEE/non-TEE isolation provided by CPU
Intra TEE isolation enforced by TSM
(guest) Non-confidential VMs
(guest) Non-confidential VMs
(guest) Non-confidential VMs
(guest) TEE VM
(guest) TEE VM
VU
VS
ABI interface
ISA interface
🟩 In TVM TCB
▭ In Non-confidential VM TCB
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
RISC-V CPU(s)
Memory Isolation, Confidentiality, Integrity and Replay-protection
Root of Trust
IOMMU

https://github.com/riscv-non-isa/riscv-ap-tee/blob/main/specification/riscv-aptee-spec.pdf
[Document in development stage in AP-TEE TG]

AP-TEE TG Charter: Interface specs

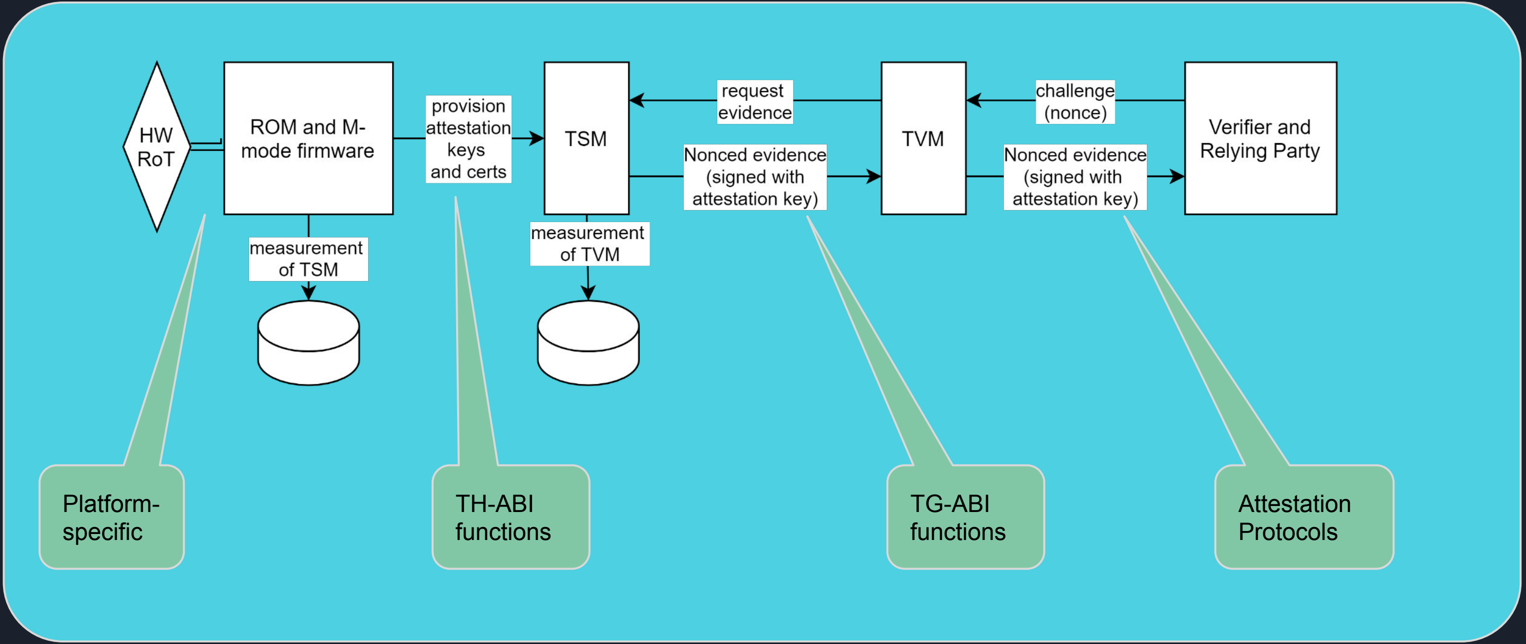
	Area	Function	Resources
Specs	AP-TEE TH-ABI	SBI Extension Interface implemented by the TSM via ECALL for use by OS/VMM to manage TVMs	TG WG members Github
	AP-TEE TG-ABI	SBI Extension Interface implemented by the TSM via ECALL for use by TVM guest workloads	
POCs	TEE Security Manager (TSM)	TSM is a RISC-V 64 bit SW module that uses RISC-V H-extension and implements TH and TG-ABI. It is in the TCB for all TVM workloads (Expected to be HW-vendor signed and may be HW-operator signed)	Rivos contributes to start collab. Github
	M-mode FW	Minimal SBI extensions (TCB component) to support TSM initialization, TEECALL, TEERET implementation. It is in the TCB for all TVM workloads (Expected to be HW-vendor signed and may be HW-operator signed) - Collab with OpenSBI	Expecting collaborators on these existing projects from Software HC discussion
	Linux, KVM (Host OS/VMM)	*Untrusted* (enlightened) host OS/VMM that manage resources for TVM-based confidential workloads [TSM enforces security properties] - Collab with Hypervisor SIG	
	Linux (TVM Guest OS), Guest Firmware	Enlightened guest OS/runtime (in TCB of TVM workload) - Collab with SW HC	

AP-TEE TG Charter: Platform & ISA (Scope)

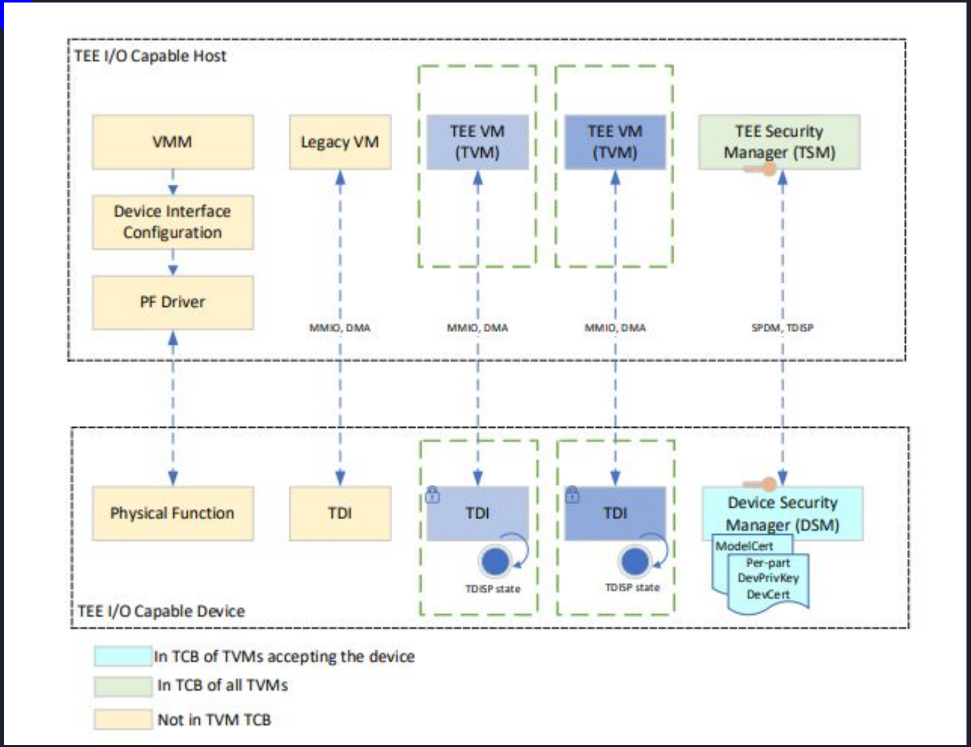
Area	Function	Resources
CPU	Evaluate AP-TEE mode qualifier, Sparse (page-based) confidential memory access-control	TG members
IOMMU	AP-TEE mode qualifier; Sparse (page-based) confidential memory access-control and fabric i/f	w/ IOMMU TG
TLB, Caches	AP-TEE mode qualifier and other micro-architectural structures	TG members
Interconnect, Fabric	Platform-specific cryptographic memory isolation and mode qualifier	TG members to document + Implementation feedback *Currently in AP-TEE spec Github
Memory	Platform-specific cryptographic memory isolation and mode qualifier	
HW Root-of-trust	Platform-specific subsystem to support HW Attestation, Sealing interfaces	
Devices	Device-specific subsystem to support Device attestation, link security	
QoS, RAS, DC	Platform-specific, Domain-specific	w/ SOC Infra

AP-TEE Security Arch for CC and Implementers Guide covers recommendations on:
- Mapping of mitigations to threat model
- Recommendations for crypto modes
- Attestation protocols, formats

TVM Attestation



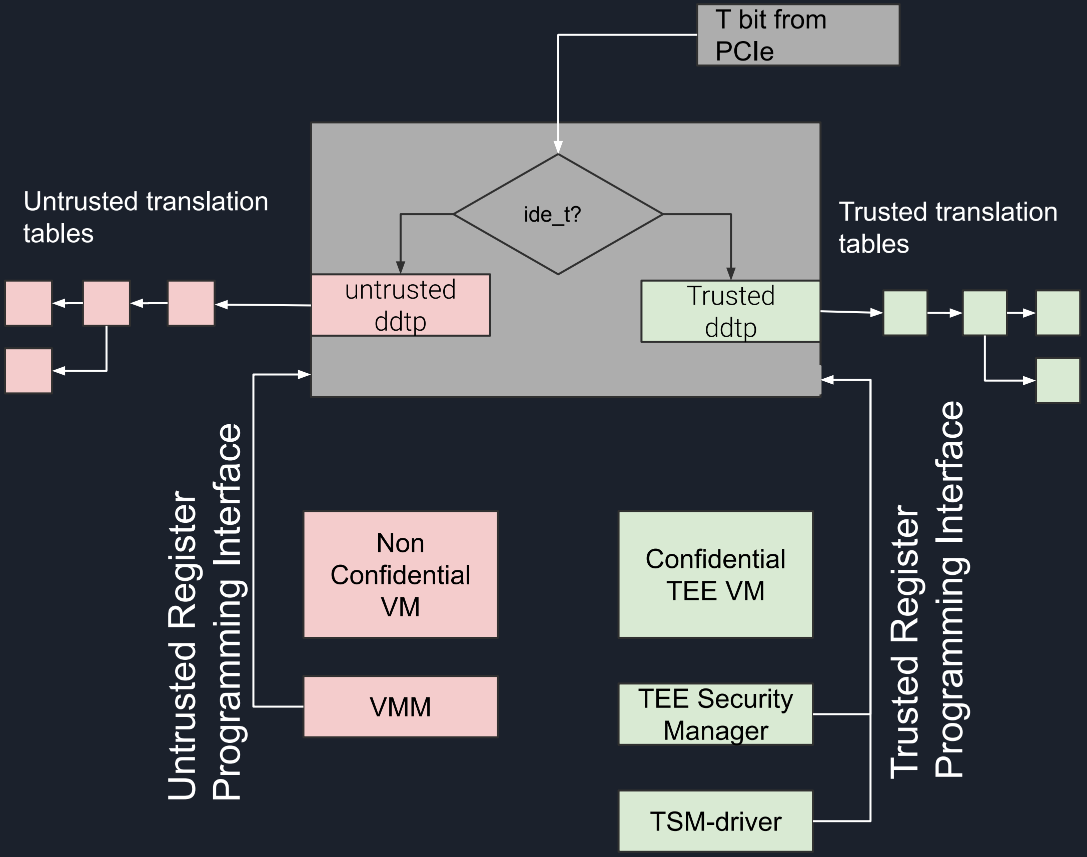
Device Implications



- Assumption - Devices require access to TVM confidential data to process it
- Device hosts a Device Security Manager (DSM) FW module
  - Implements PCIe-SIG defined TEE Device Interface Security Protocol Responder
  - TSM interacts with Device Security Manager to securely assign / reclaim DPA functions to TVMs
- Capabilities needed in HW (RISC-V IOMMU)
  - Support for trusted translation tables, queues
  - Page walker and IOTLB support for enforcing confidential memory properties

Ref: PCIe TDISP

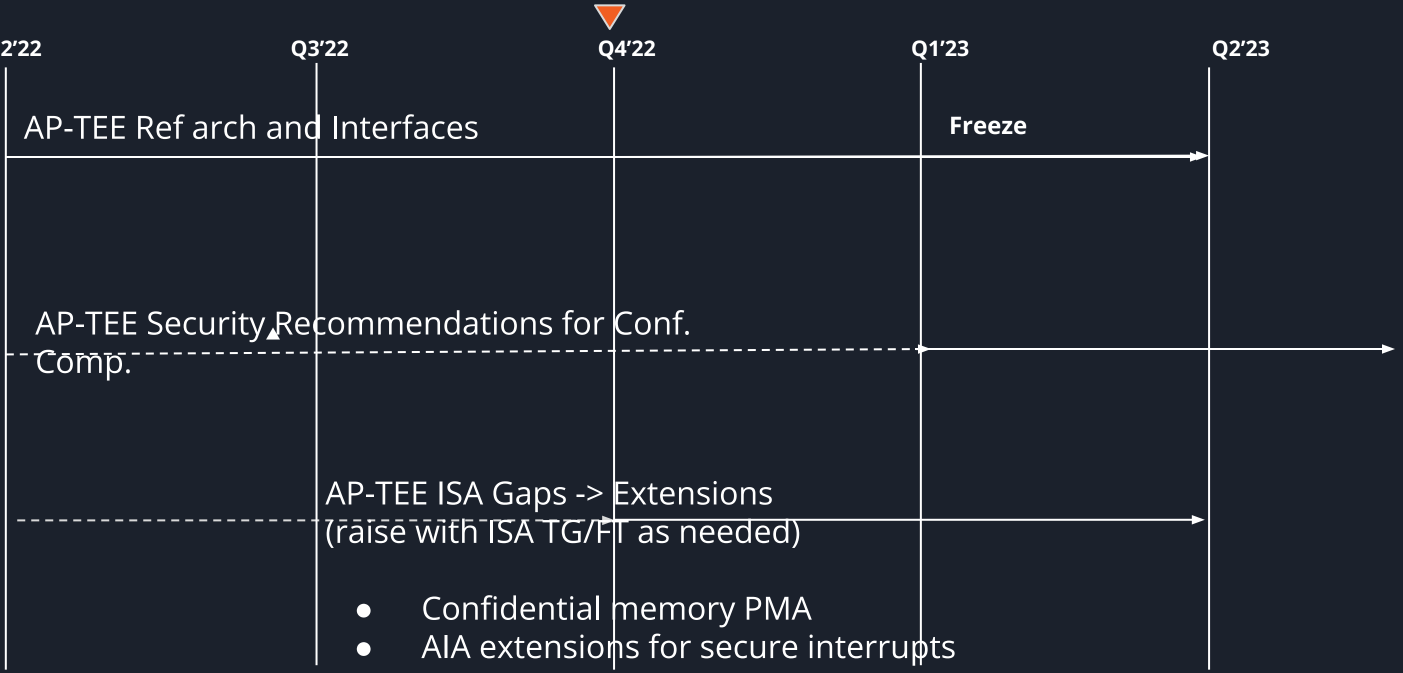
RISC-V IOMMU Conf. IO


T bit from PCIe
ide_t?
Untrusted translation tables
untrusted ddtp
Trusted ddtp
Trusted translation tables
Untrusted Register Programming Interface
Non Confidential VM
VMM
Confidential TEE VM
TEE Security Manager
TSM-driver
Trusted Register Programming Interface

AP-TEE TG workstreams

- Proposed ratification plan
- DoD checklist
- Infra requirements

- AP-TEE spec proposed to RVI
- Confidential Computing VM Design Survey



Q2'22 Q3'22 Q4'22 Q1'23 Q2'23

AP-TEE Ref arch and Interfaces

Freeze

AP-TEE Security Recommendations for Conf. Comp.

AP-TEE ISA Gaps -> Extensions (raise with ISA TG/FT as needed)

- Confidential memory PMA
- AIA extensions for secure interrupts

References

AP-TEE TG admin Github

AP-TEE Specification Github

TSM POC - Salus Github

Linux Plumbers RISC-V Micro-conference discussion on AP-TEE

Extra Slides

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.

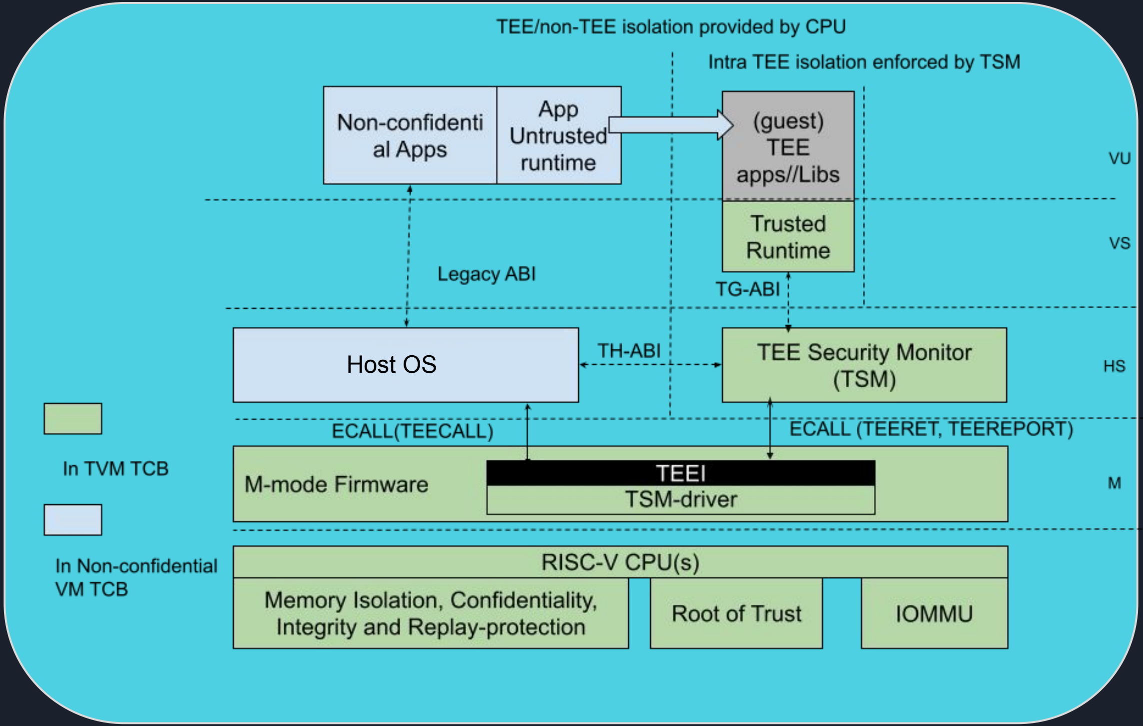
T18: **Forging attestation reports** from the RoT

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

Ref. Arch used for Confidential Applications


TEE/non-TEE isolation provided by CPU
Intra TEE isolation enforced by TSM
Non-confidential Apps
App Untrusted runtime
(guest) TEE apps//Libs
VU
Trusted Runtime
VS
Legacy ABI
TG-ABI
Host OS
TH-ABI
TEE Security Monitor (TSM)
HS
ECALL(TEECALL)
ECALL (TEERET, TEEREPORT)
M-mode Firmware
TEEI
TSM-driver
M
RISC-V CPU(s)
Memory Isolation, Confidentiality, Integrity and Replay-protection
Root of Trust
IOMMU
🟩 In TVM TCB
▢ In Non-confidential VM TCB