




# Confidential Compute for RISC-V Platforms

5/3/2022 AP-TEE TG

*Assignee - Security HC*





# Disclosures



# Agenda

Disclosures

Charter Discussion

Key Deliverables and Timeline

AP-TEE Overview

Threat Model

SW Interfaces Discussion

ISA Discussion (Mem. Mgmt)

# RISC-V Gaps → AP-TEE TG Charter

*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 - alternate architectures have solutions in place => Risk to lose market share and RISC-V adopters in these domains.

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

*What are the gap areas? And what is our proposed action plan?*

- Security Platform Model TG -- *ongoing*
- **AP-TEE TG to cover Reference Arch, Interfaces, Uncover potential ISA gaps**
  - AP-TEE Interfaces - uses current ISA; extensible to future ISA (via gap analysis) -- normative spec.
  - AP-TEE Security Arch for CC -- also use as an Implementers Guide -- informative living doc.
  - ISA proposal(s) -- request FT/TG only if needed -- normative spec.

RISC-V AP-TEE TG  
addresses this gap

*Who else do /should we work with?*

- **Within RVI** - Security HC/Trusted Computing SIG, TEE TG, CFI SIG, Software HC (Hypervisor SIG), SOC infrastructure SIG (IOMMU, QoS, RAS ...), DataCenter SIG
- **Outside RVI** - Confidential Computing Consortium (CCC), Trusted Computing Group (TCG), Internet Engineering Task Force (IETF), Distributed Management Task Force (DMTF), GlobalPlatform, PCIe, CXL

**CCC**  
Open Enclave SDK  
Keystone  
Project Veraison

**IETF** RATS  
**TCG** DICE  
**DMTF** SPD  
**PCIe** IDE, TDISP

# AP-TEE TG - Principles

- 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 unnecessary 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 adversary**

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)

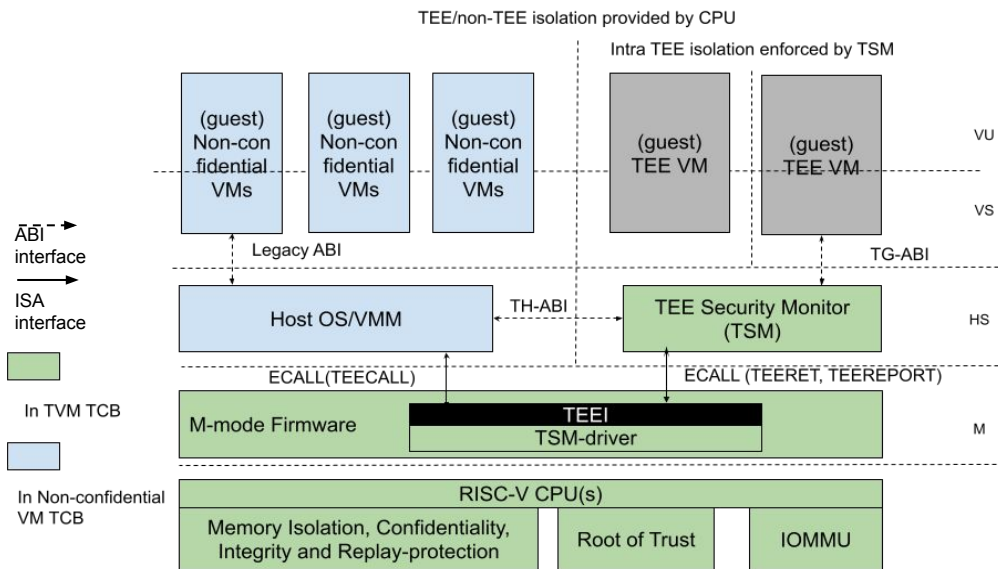
T20: **Unexpected enabling of performance monitoring and/or debug** on a TVM leading to information loss via performance monitoring events/counters and debug mode accessible information.

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

# AP-TEE TG Charter

## Define

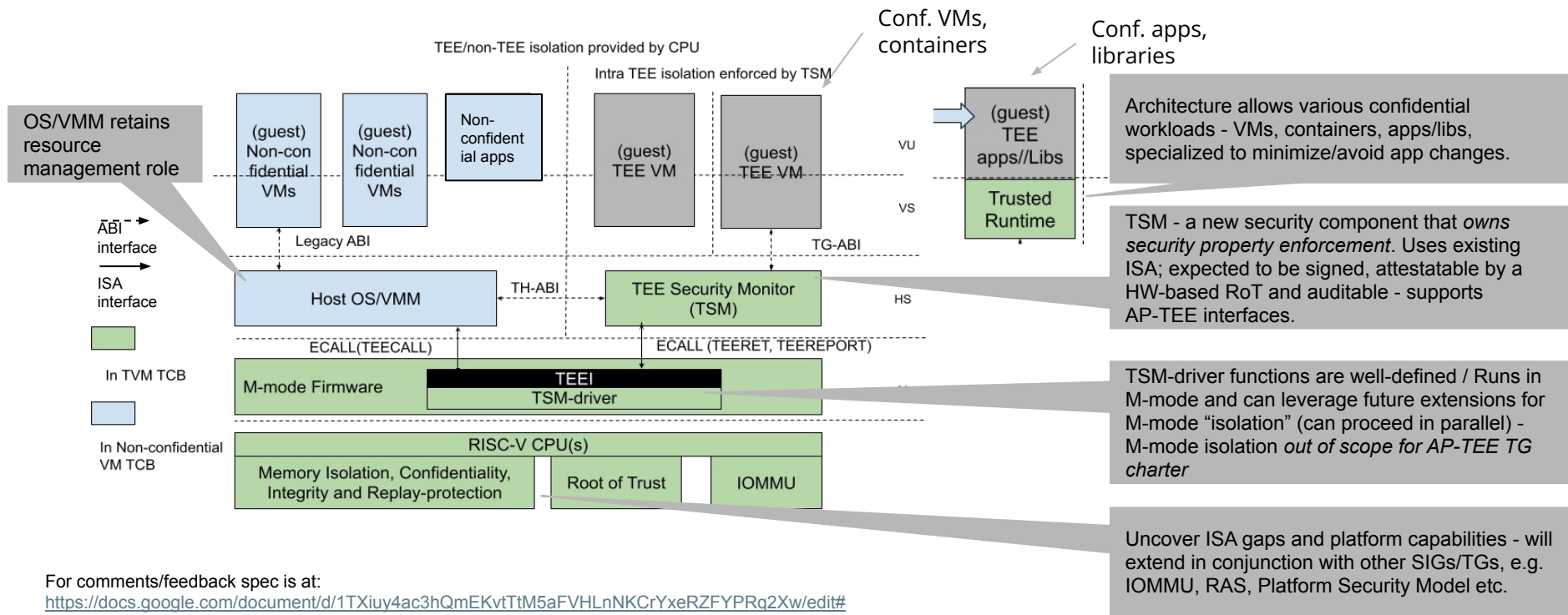
- Reference architecture for confidential compute on RISC-V platforms
- AP-TEE TH/TG-ABIs -- normative non-ISA spec.
- AP-TEE Security Arch/Implementers Guide for RISC-V confidential compute - covers platform recommendations -- informative (living spec).
- AP-TEE ISA extension(s) -- *start with current ISA*; identify ISA gaps in TG -- request FT/TG as needed -- expected to be normative spec.





# AP-TEE TG Charter: Reference Arch

**AP-TEE interfaces allow confidential compute models to be built using the ratified RISC-V ISA (with implementation specific micro-architecture and platform support)**



# AP-TEE TG Charter: Interface specs

Specs



Area	Function	Resources
AP-TEE TH-ABI	SBI Extension Interface implemented by the TSM via ECALL for use by OS/VMM to manage TVMs	TG WG members
AP-TEE TG-ABI	SBI Extension Interface implemented by the TSM via ECALL for use by TVM guest workloads	
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.
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
Linux, KVM (Host OS/VMM)	<i>Untrusted</i> (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	

POCs



# 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
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

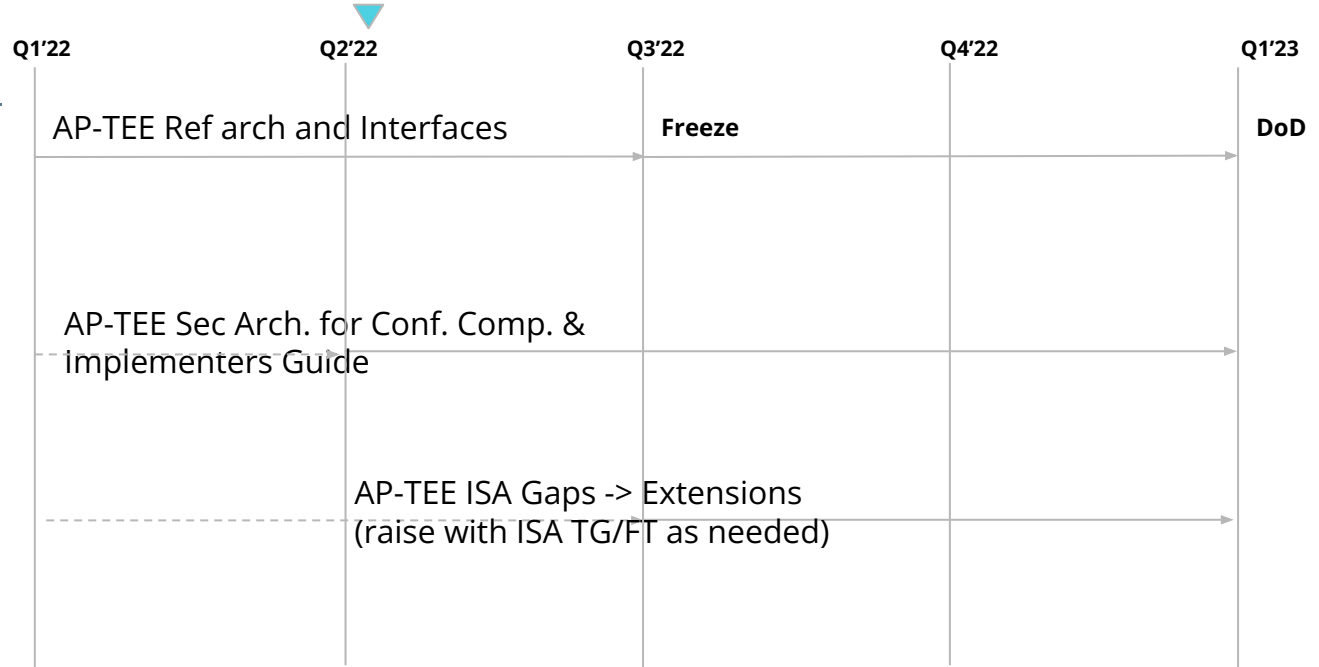
**AP-TEE Security Arch for CC and Implementers Guide** covers recommendations on:

- Mapping of mitigations to threat model
- Recommendations for crypto modes
- Attestation protocols, formats

# AP-TEE TG workstreams

- [Proposed ratification plan](#)
- [DoD checklist](#)
- [Infra requirements](#)

- 
- Initial [AP-TEE spec](#) being discussed at the TC SIG.
  - [Confidential Computing Survey](#)



# TG Discussion areas

- SW Lifecycle
- Confidential Memory Mgmt
- Interrupt Mgmt
- Attestation
- Physical Protection - Encryption
- Confidential process (SW model)
- Direct IO assignment to TVMs
  - IOMMU discussions ongoing to establish direction
- Domain partitioning
- Stolen time
- TVM Live Migration
- TVM Checkpointing
- Others?

# (AP-TEE TG charter)

The RISC-V Application Platform - Trusted Execution Environment Task Group (AP-TEE TG) will collaborate to define the reference architecture for confidential computing on RISC-V platforms - including the ABI required to enable systems software to manage confidential workloads on a multi-tenant platform, while keeping the OS/hypervisor and entities that develop the OS/VMM and/or operate/manage the platform outside the TCB. The TG will design the interfaces to comprehend existing (ratified) ISA and ensure extensibility of the interfaces to new Architectural ISA extensions as required for security or performance of confidential workloads. In addition to the normative specifications mentioned, the TG will produce an AP-TEE-specific security architecture analysis per the confidential computing threat model agreed upon as a living (non-normative) document supporting security recommendations, implementation-specific guidelines and relevant standard protocols for attestation for implementers of the AP-TEE capability on RISC-V platforms. The proposed RISC-V AP-TEE task group will collaborate to define these three specifications:

a. **AP-TEE reference architecture and SBI extension interface (non-ISA, normative)** which specifies the TH-ABI and TG-ABI interfaces to enable the OS/Hypervisor to manage confidential workloads on a multi-tenant platform, while keeping the OS/hypervisor and entities that develop the OS/VMM and/or operate/manage the platform outside the TCB. The interfaces are defined between:

1. A new platform-specific security service called the Trusted Security Manager (TSM) operating in RISC-V HS-mode and a general-purpose OS/Hypervisor executing in S/HS-mode - called the **TH-ABI**. The TH-ABI should cover aspects of: TVM creation and tear down, TVM measurement and attestation, TVM memory management and protection, TVM virtual-hart state management and protection, TVM execution and IO.

2. A Trusted Security Manager (TSM) running in HS-mode and a general-purpose OS executing in VS-mode - called the **TG-ABI**. The TG-ABI should cover aspects the TVM is involved in: TVM measurement extension and attestation, TVM memory conversion, TVM IO and other services used from host

b). **AP-TEE architecture security analysis (non-normative living document)** supporting recommendations and implementation guidelines for e.g. coverage of threat model, attestation protocols, crypto modes

c). **AP-TEE ISA extensions (normative)** to be proposed as needed for enforcing confidential workload security and performance requirements. The interfaces in item a. will be defined to be extensible to any such future ISA extensions. The TG will start with the definition of the programming interfaces and identify ISA gaps. ISA proposals made will be modeled via tools such as QEMU/Spike.

The goal of the AP-TEE interface specification is to **enable open-source reference implementations of the RISC-V AP-TEE interfaces** for platform-specific TSM implementations that enable confidential compute and trusted execution for different use case scenarios (Server, Automotive, Embedded etc.). To support this goal, a POC is defined that consists of: An SBI extension implementation for AP-TEE will be used as a reference implementation. A TSM implementation will be developed by the community as part of the ratification of the interfaces. The required changes will be made to the Linux/KVM host and guest software to validate the interface specifications.

# Discussion and follow up ARs (5/3)

- Clarify charter to focus on OS-A (Rich OS)
- Clarify dependency of AP-TEE on Priv ISA (including H-extension)
- Contrast w/ Global Platform interfaces