Confidential Compute for RISC-V Platforms

AP-TEE TG proposal (presented by Ravi Sahita)

For Trusted Computing SIG

Assignee - Security HC

Confidential Computing

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

> *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

Key properties of a HW-based TEE for Conf. Comp.

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
- Attestatability -- This is a required from the Trusted Computing SIG perspective
- Recoverability

Confidential Compute Threat Model

User/System Software attacks

Protocol attacks

Cryptographic attacks

Basic hardware attacks

Basic upstream supply-chain attacks

Advanced hardware attacks

Upstream hardware supply-chain attacks

uArch and Arch Side-channel attacks*

Detailed threat model has been defined and documented here.

We note that different implementations will have varying degrees of resistance to attacks

The TC SIG (or proposed TG) does not aim to specify any threats as out of scope.

*continues to be actively researched

RVI Gaps → AP-TEE TG Charter

Why should we do this? And why now?

- Confidential Computing is at an inflection point and all compute domains (Data Center/Servers to Embedded) require support for it - alternate architectures have solutions in place

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

What are the key gap areas? [TG components proposed below described on next slide]

- **AP-TEE TG to cover Reference Architecture, Interfaces, Uncover potential ISA gaps**
  - Interfaces must be designed to be extensible to future ISA (via gap analysis) --normative.
  - ISA proposals -- request FT/TG as needed -- normative.
  - Security Arch for CC -- *Separate living doc* - also use as an Implementers Guide.

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), PCIe, CXL

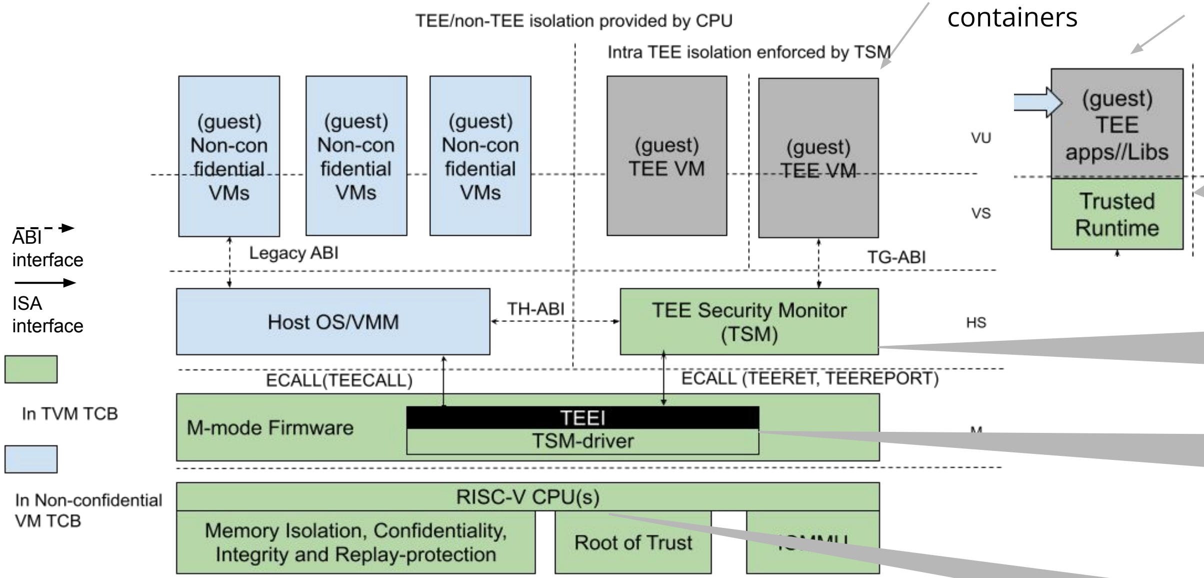
CCC
Open Enclave SDK
Keystone
Project Veraison

IETF RATS
TCG DICE
DMTF SPDM
PCIe IDE, TDISP

AP-TEE TG Charter: Reference Arch

Conf. VMs, containers

Conf. apps, libraries


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
(guest) TEE apps//Libs
Trusted Runtime
VU
VS
HS
M
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)
ECALL(TEECALL)
ECALL (TEERET, TEEREPORT)
M-mode Firmware
TEEI
TSM-driver
RISC-V CPU(s)
Memory Isolation, Confidentiality, Integrity and Replay-protection
Root of Trust
IOMMU

TVM construct allows various confidential workloads - VMs, containers, apps/libs, specialized

A new security component (expected to be HW-signed, attestatable and auditable) - in TCB -- normative AP-TEE i/fs **TH, TG-ABI**

TSM-driver functions well-defined. In M-mode and can leverage future extensions for M-mode isolation (that can proceed in parallel)

Recommended ISA and platform capabilities will extend in conjunction with this TG, e.g. IOMMU, Platform Security Model etc.

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

For comments/feedback spec is at: https://docs.google.com/document/d/1TXiuy4ac3hQmEKvtTtM5aFVHLnNKCrYxeRZFYPRq2Xw/edit#

AP-TEE TG Charter: Interfaces

	Area	Function	Resources
Specs	AP-TEE TH-ABI	SBI Extension Interface implemented by the TSM via TEECALL 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	
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
	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 SW HC
	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)

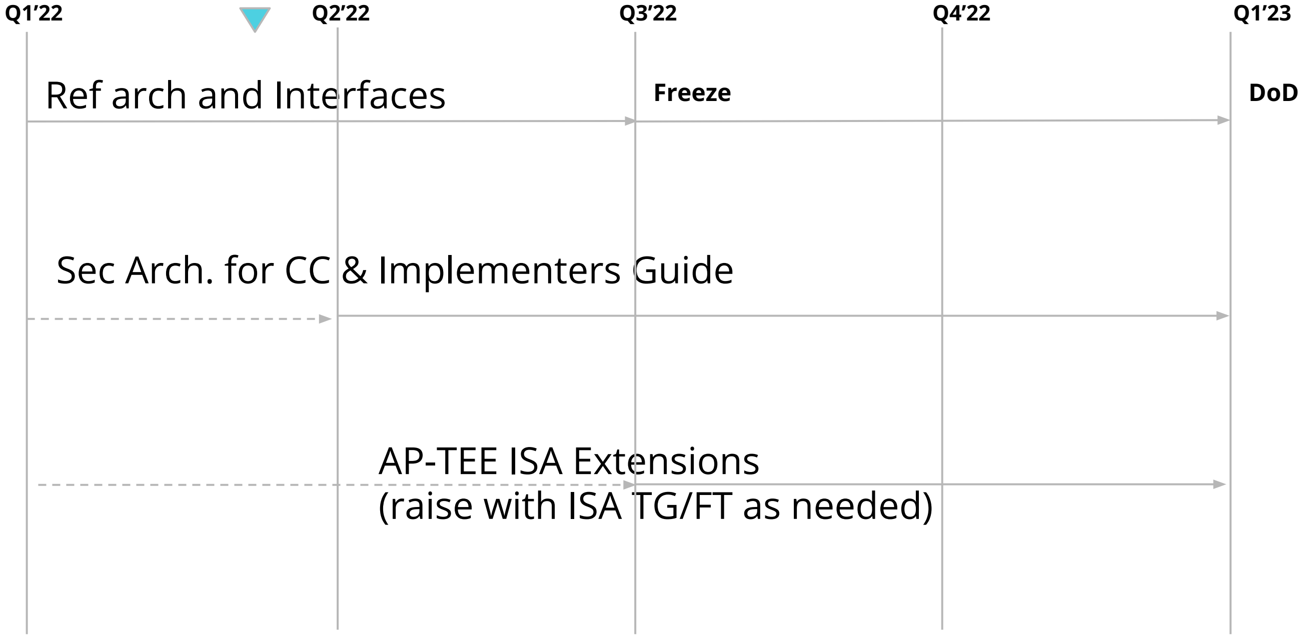
Area	Function	Resources
CPU	AP-TEE mode qualifier ; Memory page access-control isolation properties	TG members
IOMMU	AP-TEE mode qualifier; Memory page access-control isolation properties	+ IOMMU TG
TLB, Caches	AP-TEE mode qualifier	TG members
Interconnect, Fabrics	**Platform-specific** cryptographic memory isolation and mode qualifier	AP-TEE TG members to document + Implementation feedback
Memory (volatile and persistent)	**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	

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

Proposed AP-TEE TG workstreams

- Proposed ratification plan
- DoD checklist
- Infra requirements

- Initial AP-TEE spec being discussed at the TC SIG.
- CVM discussion
- Confidential Computing Survey



Q1’22 Q2’22 Q3’22 Q4’22 Q1’23

Ref arch and Interfaces

Freeze

DoD

Sec Arch. for CC & Implementers Guide

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

Seeking TSC Approval to form AP-TEE TG with this charter

Extra Slides

(AP-TEE charter)

The RISC-V 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 a security architecture analysis per the 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 their RISC-V platforms.
The proposed RISC-V AP-TEE task group will collaborate to define:

a) ***AP-TEE reference architecture and SBI extension interface (non-ISA)*** *which specifies the TH-ABI and TG-ABI interfaces (normative) 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). ***Architectural ISA extensions (normative) as needed for supporting confidential workloads****. The interfaces in item a will be defined to be extensible to these ISA extensions. The TG will start with the programming interfaces and discuss if any architecture extensions are required. Any ISA extensions will be modeled as part of the ratification process via tools such as QEMU/Spike.*

c). ***A security architecture analysis of the reference architecture as a living document*** *supporting recommendations and implementation-specific guidelines (non-normative).*

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.