Linux Plumbers Conference

Dublin, Ireland September 12-14, 2022

Confidential Compute on RISC-V platforms

Ravi Sahita
Rivos Inc.

Linux Plumbers Conference | Dublin, Ireland Sept. 12-14, 2022

Rivos

Background

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.

TEE properties verified via HW-rooted **attestation** of the Trusted Computing base

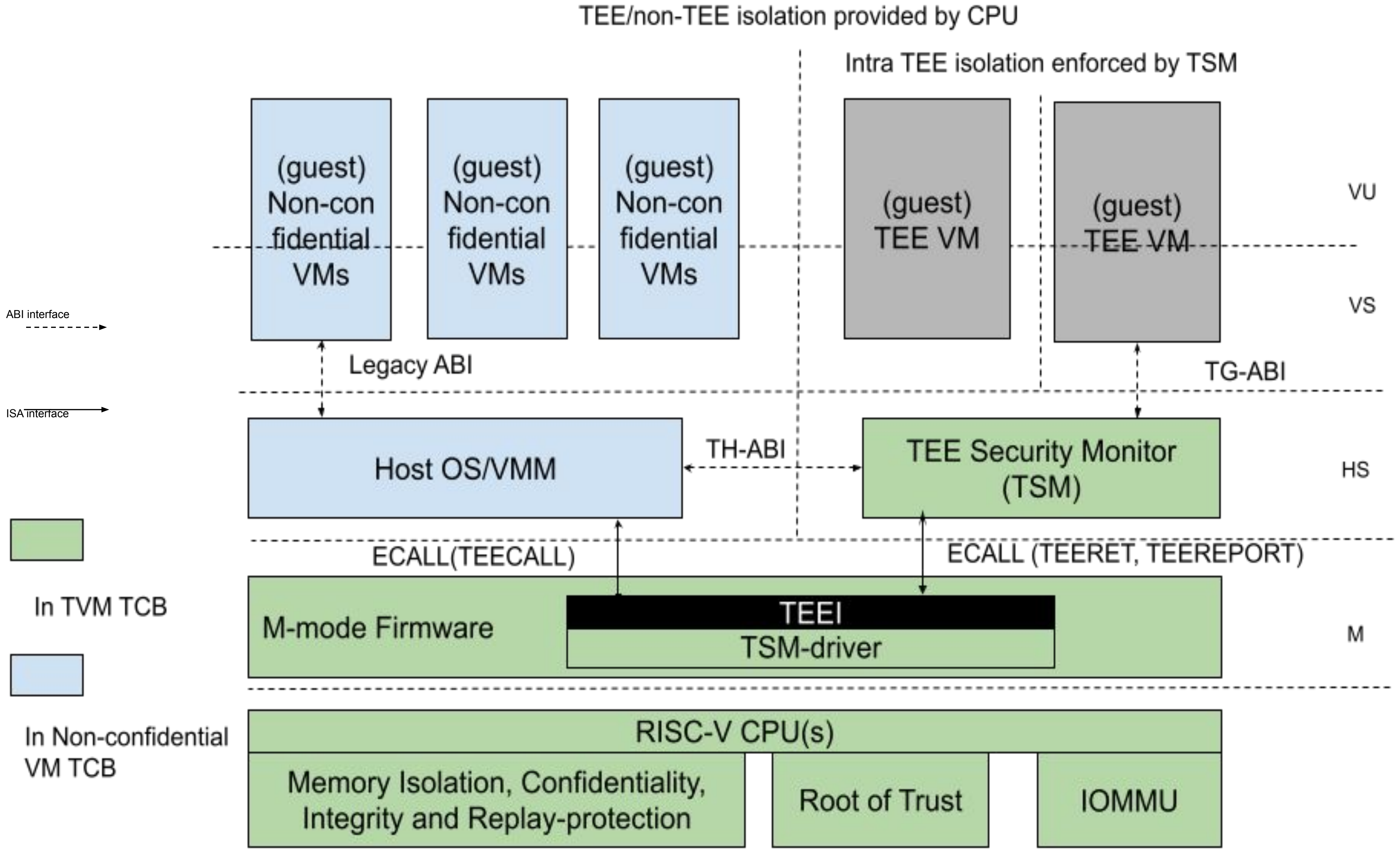
Threat Model

- Read of confidential data in memory/CPU register state
- Tamper of confidential data in memory/CPU register state
- Tamper of MMU in-memory structures (address mappings)
- Invalid execution of code handling confidential data
- Shared memory
- Read/Write of confidential data in memory via I/O devices
- Spurious generation, tamper of interrupt delivery
- Timestamp, Debug & Performance monitoring
- Denial of service
- Exposure of data via side channels (arch and u-arch)
- Attestation-oriented (boot, crypto and protocol)
- Operational feature attacks (migration)
- <others>

All current TEE approaches address aspects of this threat model

A Ref. Arch for RISC-V is proposed that has:
-non-ISA interfaces
-ISA extensions
-platform guidelines

Reference Arch



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

Legacy ABI

TG-ABI

Host OS/VMM

TH-ABI

TEE Security Monitor (TSM)

HS

ECALL(TEECALL)

ECALL (TEERET, TEEREPORT)

🟩 In TVM TCB

🟦 In Non-confidential VM TCB

M-mode Firmware

TEEI

TSM-driver

M

RISC-V CPU(s)

Memory Isolation, Confidentiality, Integrity and Replay-protection

Root of Trust

IOMMU

- Uses RISC-V H-extension*. Specifies ISA-extensions e.g. Physical Memory Attributes for Confidential memory

- Introduces TEE Security Manager (TSM) - Support different implementations, deployment models and process/VM workloads

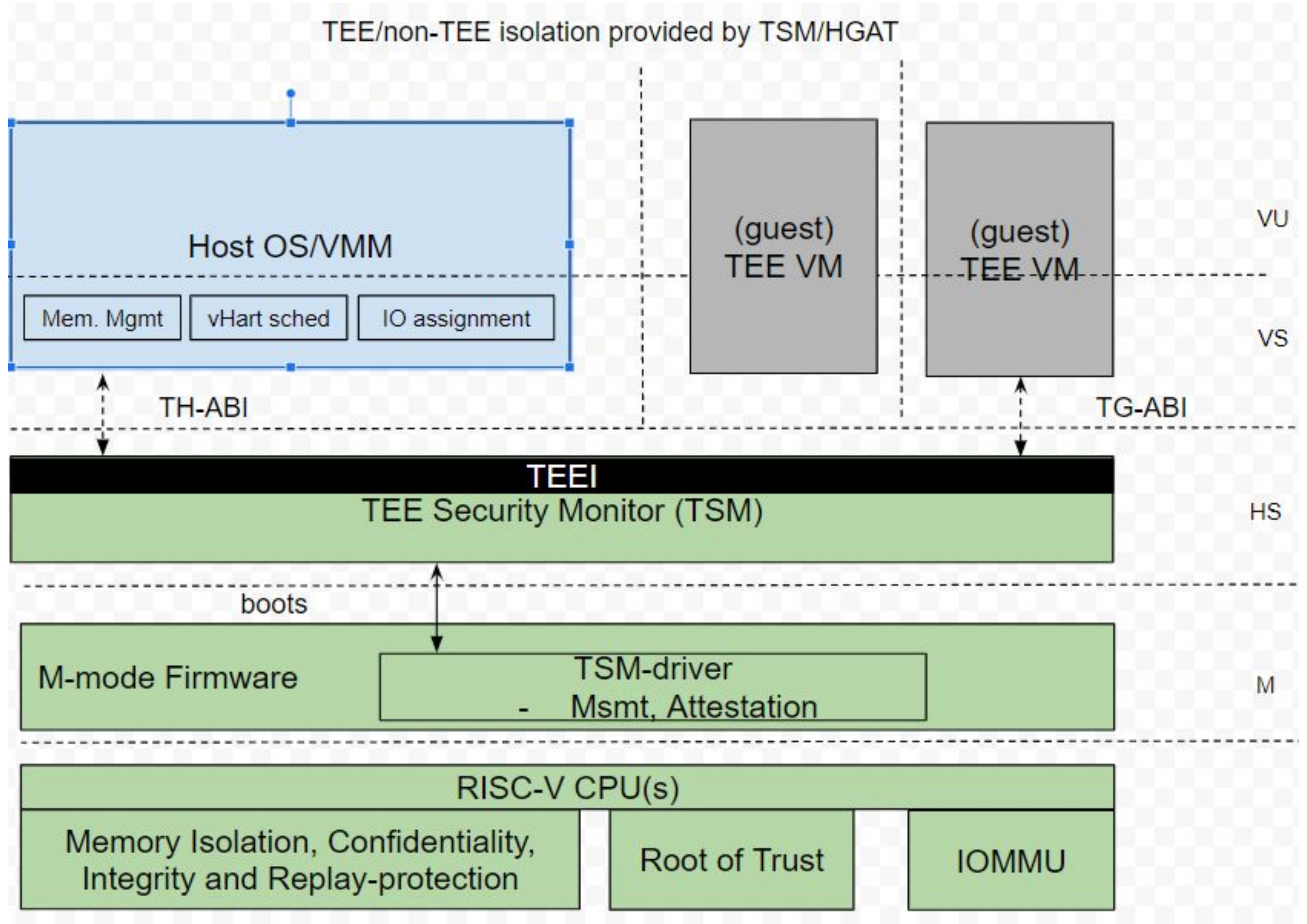
- TSM Interfaces TH-ABI and TG-ABI** abstract platform differences

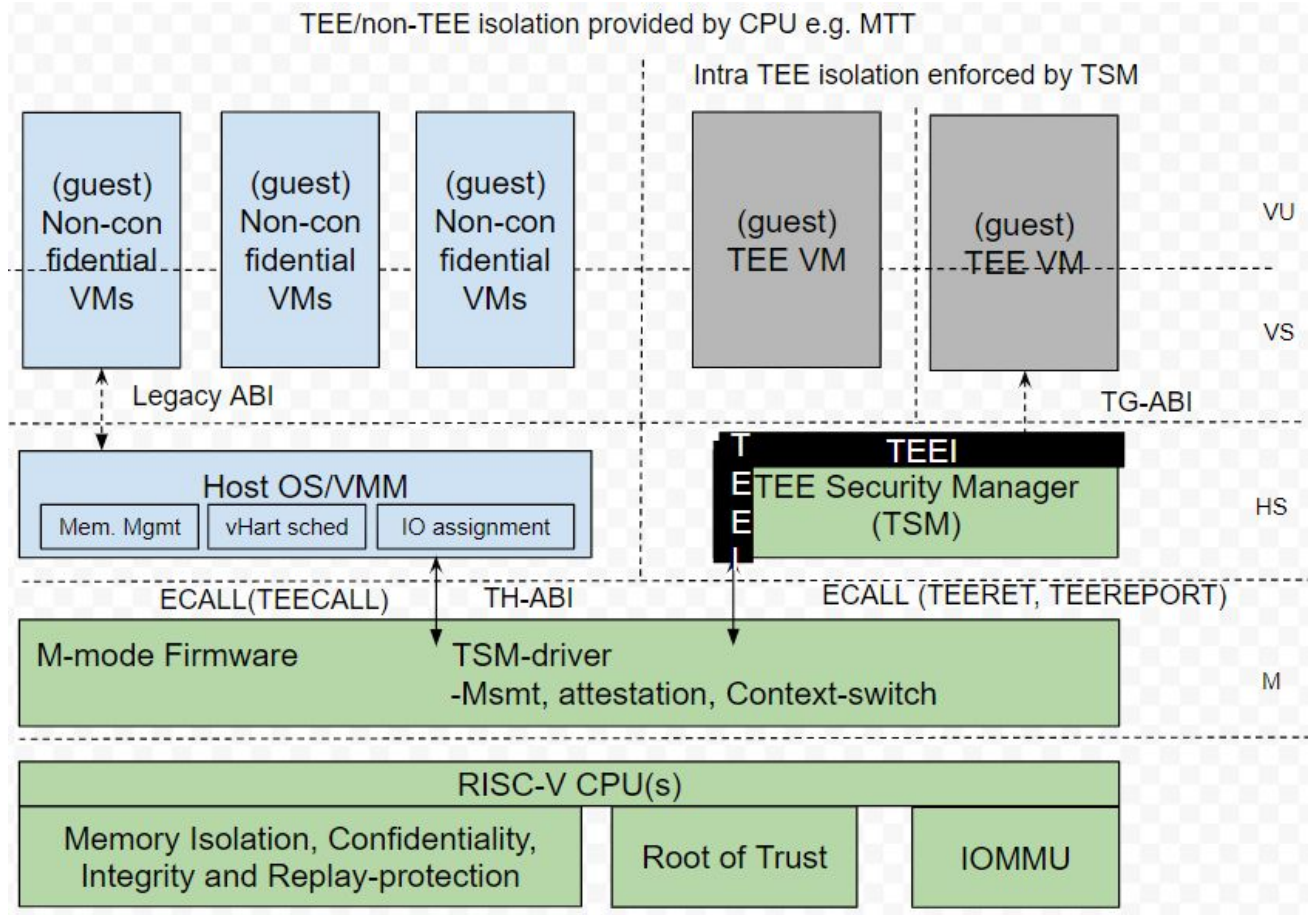
- Provide guidelines for non-normative [implementation specific] stuff

**https://github.com/riscv-non-isa/riscv-ap-tee/blob/main/specification/riscv-aptee-spec.pdf

*https://github.com/kvm-riscv/howto/wiki/KVM-RISCV64-on-QEMU

Deployment Models




TEE/non-TEE isolation provided by CPU e.g. MTT
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
Mem. Mgmt
vHart sched
IO assignment
TEEI
T E E I
TEE Security Manager (TSM)
HS
ECALL(TEECALL)
TH-ABI
ECALL (TEERET, TEEREPORT)
M-mode Firmware
TSM-driver
-Msmt, attestation, Context-switch
M
RISC-V CPU(s)
Memory Isolation, Confidentiality, Integrity and Replay-protection
Root of Trust
IOMMU

- Both deployment models require H extension, and *TSM interfaces for:*
  - *Confidential memory mgmt.*
  - *vCPU mgmt, context isolation*
  - *Measurement and Attestation*
  - *IO assignment*
- Platform-specific Implementation aspects remain the same ***e.g. memory encryption***
- Deployment model 2 requires additional HW functions ***e.g. Confidential Memory PMA***
- Additionally - implementations can further reduce ***M-mode TCB*** with future HW support

Discussion

TH-ABI Plumbing (Interfaces:1)

Get TSM Info e.g. memory pages to set aside per TVM	sbi_tee_tsm_get_info
Create/Destroy TVM	sbi_tee_create/destroy_tvm
Convert a memory region from Non-Confidential to Confidential	sbi_tee_mem_convert_pages
Issue fence for global/TVM TLB invalidations	sbi_tee_mem_initiate/local_fence
Assign TVM memory region to confidential, shared or MMIO	sbi_tee_tvm_add_shared/conf/mmio_region
Add confidential page mappings	sbi_tee_tvm_add_page_table_pages
Add a measured TVM page	sbi_tee_tvm_page_add_measured_pages
Add a zero TVM page (lazy add)	sbi_tee_tvm_page_add_zero_pages
Blocks page mappings for TVM page(s)	sbi_tee_tvm_page_range_block
Unblocks page mappings for TVM page(s)	sbi_tee_tvm_page_range_unblock
Relocate a page for an existing mapping for a TVM page.	sbi_tee_tvm_page_relocate
Promote/Demote a (set of) page mappings for a TVM	sbi_tee_tvm_page_promote/demote
Reclaim confidential pages	sbi_tee_tvm_page_reclaim

TH-ABI Plumbing (Interfaces:2)

Create a TVM vcpu context	sbi_tee_tvm_vcpu_create
Execute a TVM vcpu context	sbi_tee_tvm_vcpu_run
Others aspects that will require interfaces: Secure interrupts, TEE IO, TEE workload migration etc.	WIP

Linux/KVM Mapping

TVM creation requires some additional operations in addition to the ordinary VM creation.

- Get KVM system capability to check if AP-TEE is supported on the RISC-V platform
- **New: Get the TSM info**

- VM creation (KVM_CREATE_VM)
- **New: Set TVM parameters when creating it via modified KVM_INIT_VM.**

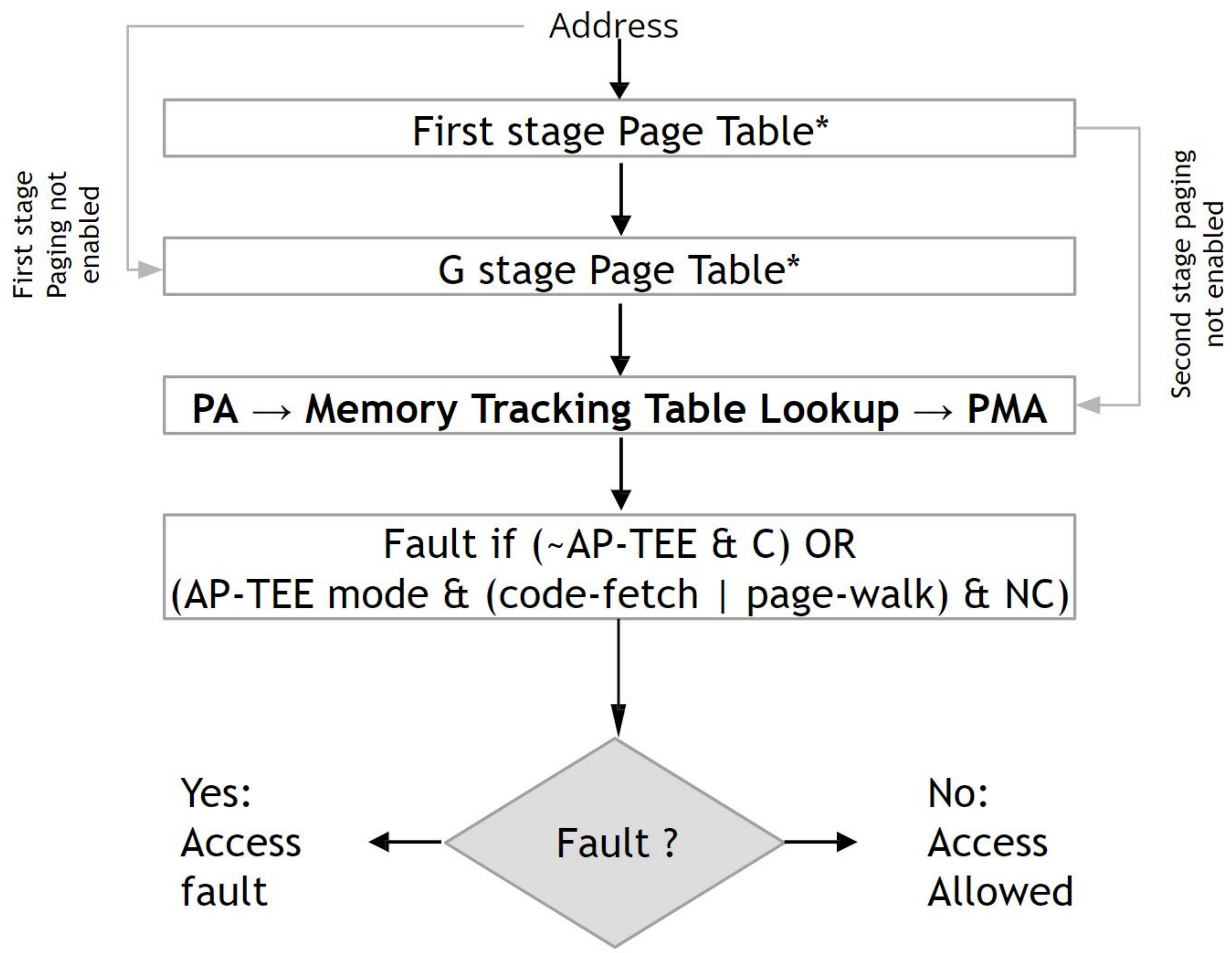
- VCPU creation (KVM_CREATE_VCPU)
- **New: Set TVM-specific VCPU parameters. KVM_TVM_INIT_VCPU.**

- Assign (shared) memory to the TVM (e.g. for virtIO)
- **New: Assign confidential guest memory and extend the TVM measurement as memory contents are loaded**

- **New: Finalize TVM so it can be executed**

- VCPU RUN (KVM_VCPU_RUN)
- **New: Schedules TVM vcpu via TSM TH-ABI**

Linux/KVM Mapping


Address
First stage Page Table*
G stage Page Table*
PA → Memory Tracking Table Lookup → PMA
Fault if (~AP-TEE & C) OR (AP-TEE mode & (code-fetch | page-walk) & NC)
First stage Paging not enabled
Second stage paging not enabled
Fault ?
Yes: Access fault
No: Access Allowed

TVM memory mgmt requires some additional operations in addition to the ordinary VM memory mgmt.

Proposal for RISC-V in discussion - **new confidential physical memory attribute enforced by the CPU as part of the MMU (and IOMMU)**

Post-TVM creation:

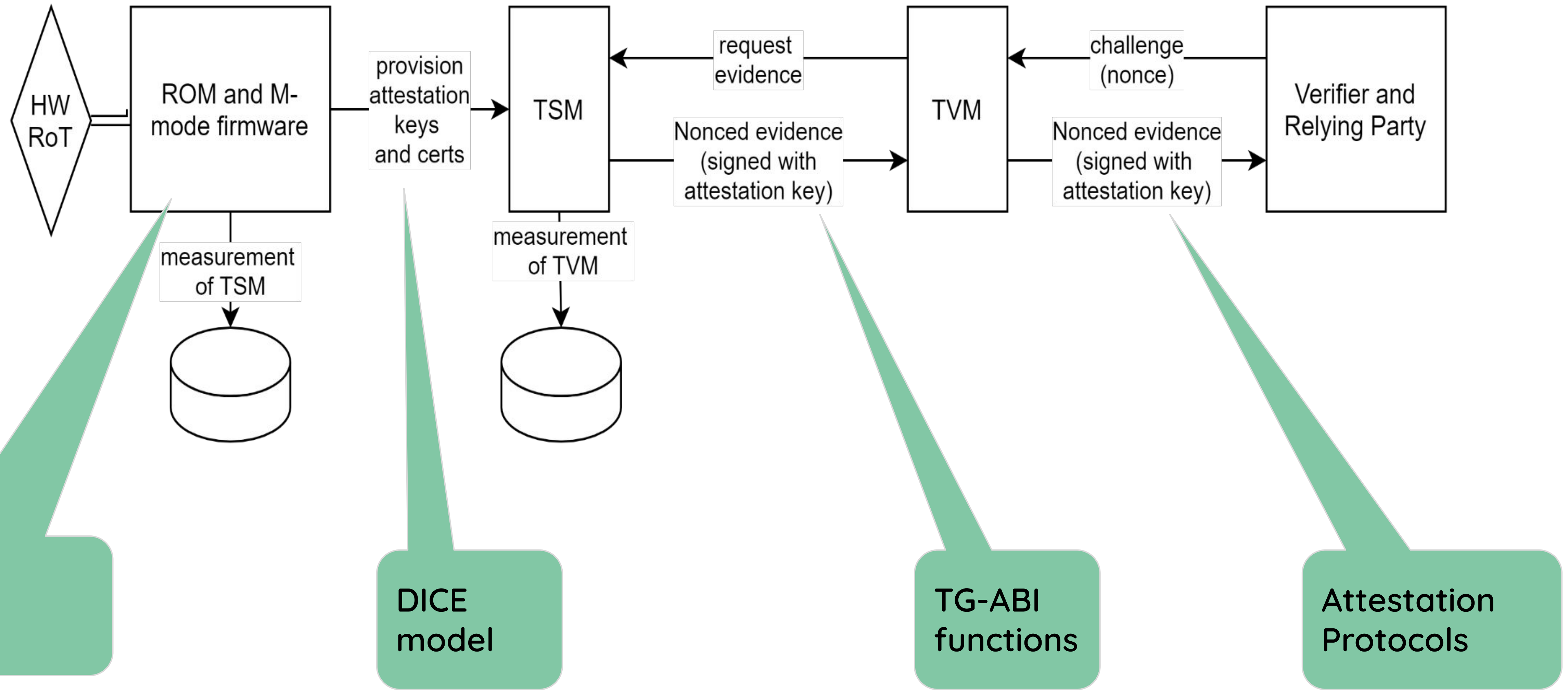
- On-demand assignment of memory
  - New: Only zero memory may be added for not-present page mappings

- Modification of mappings
  - New: Block new TLB mappings via making mapping not-present
  - Modify mapping (relocate, promote, demote etc.)
  - New: Flush old cached mappings (via TSM)

- **New: handle new fault conditions during TVM execution**
  - Access violation from untrusted host (from MTT)
  - Violation due to code fetch or page walk in non-confidential memory

TG-ABI Plumbing (Interfaces)

Get the platform attestation capabilities e.g. crypto hash supported	sbi_attestation_get_capabilities
Get the TCB attestation evidence e.g. X.509 certificate	sbi_attestation_get_evidence
Extend a measurement register	sbi_attestation_extend_measurement
Read a measurement register	sbi_attestation_read_measurement
Invoke untrusted host service e.g. virtIO	sbi_tee_req_vmm_svc
Others to be added for scenarios like IO, Migration etc.	



- Work in progress proposal at https://github.com/sameo/riscv-sbi-doc/pull/1
- PoC at https://github.com/rivosinc/salus

Call to Action

- Review and provide feedback on APTEE TH/TG-ABI (discussions on the RVI AP-TEE TG list)
  - See discussion of the interfaces here - https://docs.google.com/presentation/d/14-rFP23zEdP2t6A9D40J6e3iPYqF7sk0AuqV47OLVEw/edit#slide=id.p

- Join POC efforts for TEE Security Manager (TSM) for RISC-V implementing TH/TG-ABI https://github.com/rivosinc/salus

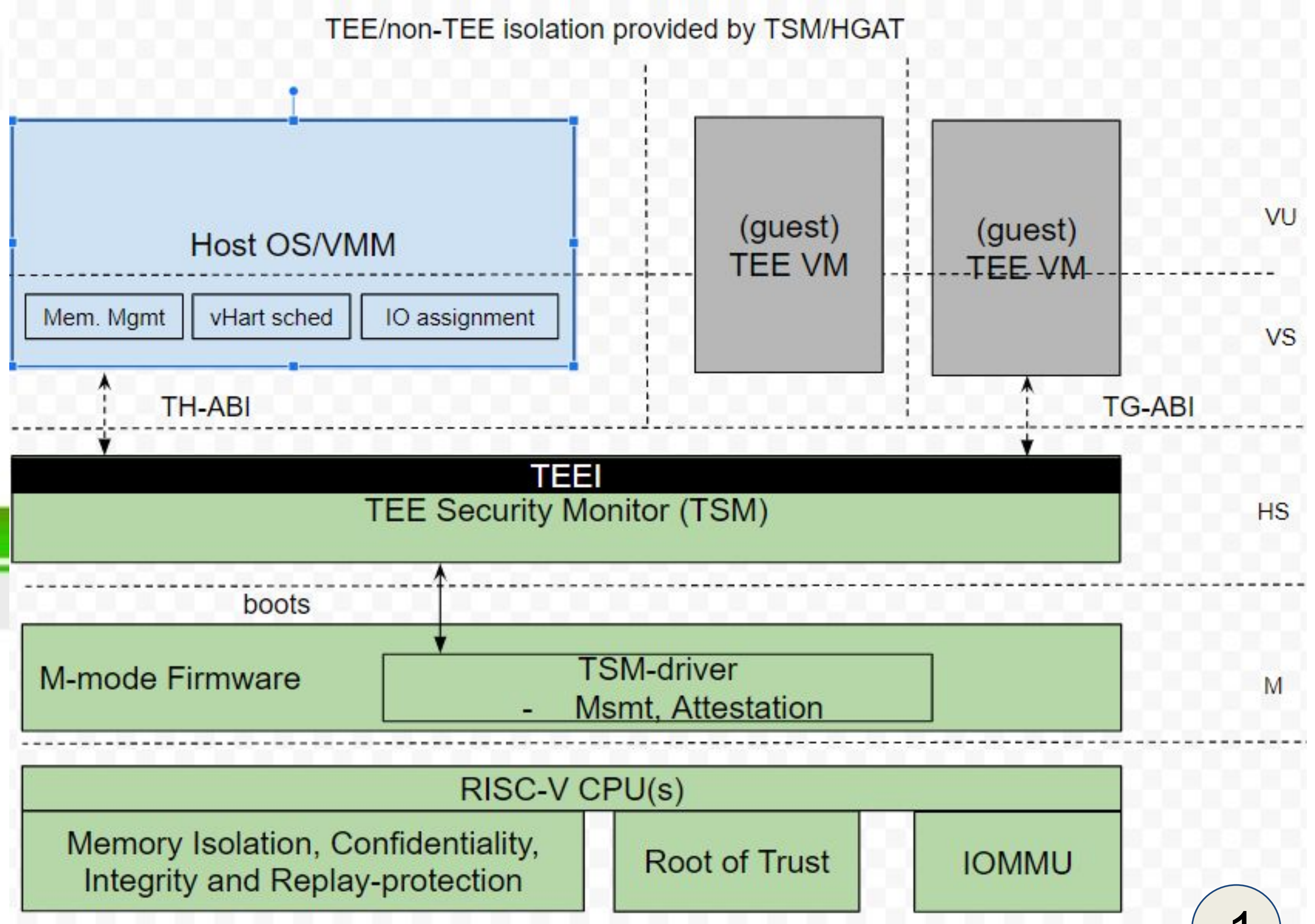
- Extend RISC-V-KVM to interface with proposed TH/TG-ABI -- active task in AP-TEE TG and RVI Hypervisor SIG

- Develop common test cases to evaluate compatibility for Linux/KVM TVM guests across different architectures and scenarios

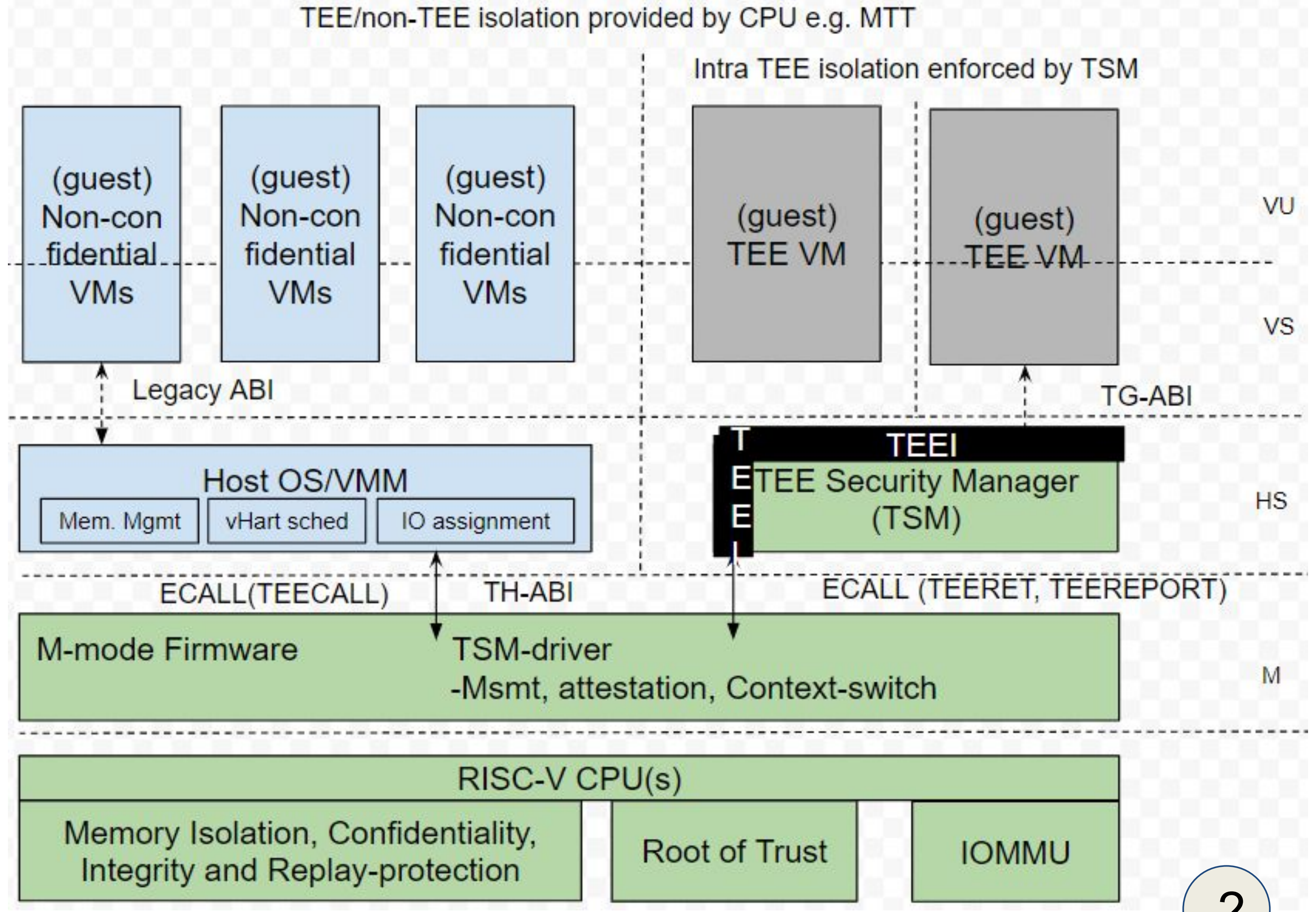
Linux
Plumbers
Conference

Dublin, Ireland September 12-14, 2022

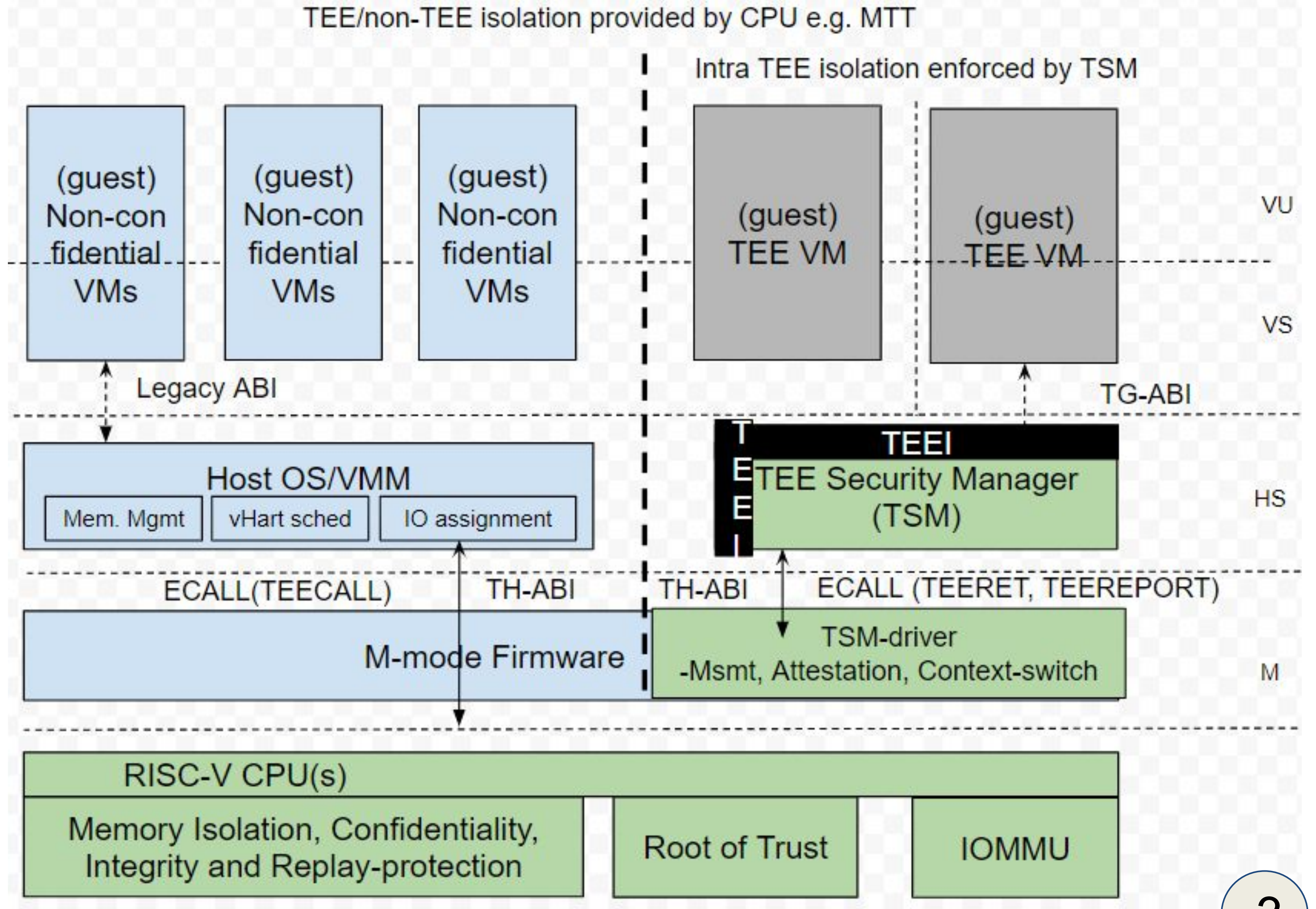
Example Deployment Cases



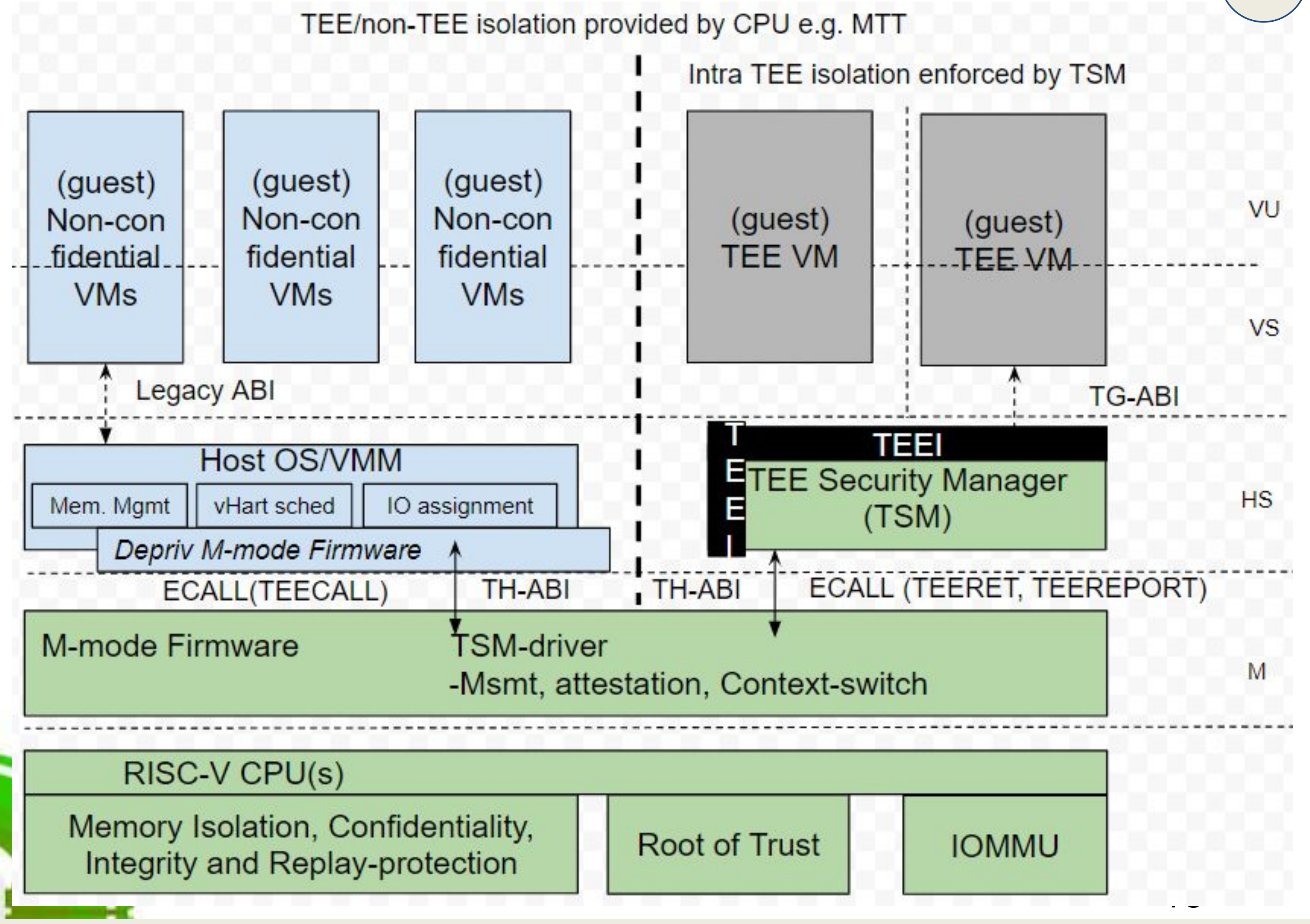
1


TEE/non-TEE isolation provided by CPU e.g. MTT
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
Legacy ABI
TG-ABI
Host OS/VMM
Mem. Mgmt
vHart sched
IO assignment
TEEI
TEE Security Manager (TSM)
HS
ECALL(TEECALL)
TH-ABI
ECALL (TEERET, TEEREPORT)
M-mode Firmware
TSM-driver
-Msmt, attestation, Context-switch
M
RISC-V CPU(s)
Memory Isolation, Confidentiality, Integrity and Replay-protection
Root of Trust
IOMMU

2


TEE/non-TEE isolation provided by CPU e.g. MTT
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
Legacy ABI
TG-ABI
Host OS/VMM
Mem. Mgmt
vHart sched
IO assignment
TEEI
TEE Security Manager (TSM)
HS
ECALL(TEECALL)
TH-ABI
TH-ABI
ECALL (TEERET, TEEREPORT)
M-mode Firmware
TSM-driver
-Msmt, Attestation, Context-switch
M
RISC-V CPU(s)
Memory Isolation, Confidentiality, Integrity and Replay-protection
Root of Trust
IOMMU

3


TEE/non-TEE isolation provided by CPU e.g. MTT
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
Legacy ABI
TG-ABI
Host OS/VMM
Mem. Mgmt
vHart sched
IO assignment
Depriv M-mode Firmware
TEEI
TEE Security Manager (TSM)
HS
ECALL(TEECALL)
TH-ABI
TH-ABI
ECALL (TEERET, TEEREPORT)
M-mode Firmware
TSM-driver
-Msmt, attestation, Context-switch
M
RISC-V CPU(s)
Memory Isolation, Confidentiality, Integrity and Replay-protection
Root of Trust
IOMMU

🟩 In TVM TCB

🟦 In Non-confidential VM TCB

- All deployment models require priv. ISA w/ H extension, and *TSM TEEI for:*
  - *Confidential memory mgmt.*
  - *vCPU mgmt, context isolation*
  - *Measurement and Attestation*
  - *IO assignment*
- Platform-specific Implementation aspects remain the same ***e.g. memory encryption***
- Deployment model 2 requires additional HW function ***e.g. Confidential Memory PMA***
- Deployment models 3 require additional HW/SW function ***e.g. M-mode TCB reduction***