RISC-V
Confidential Computing

ARCHI’25 - sameo@rivosinc.com

Rivos

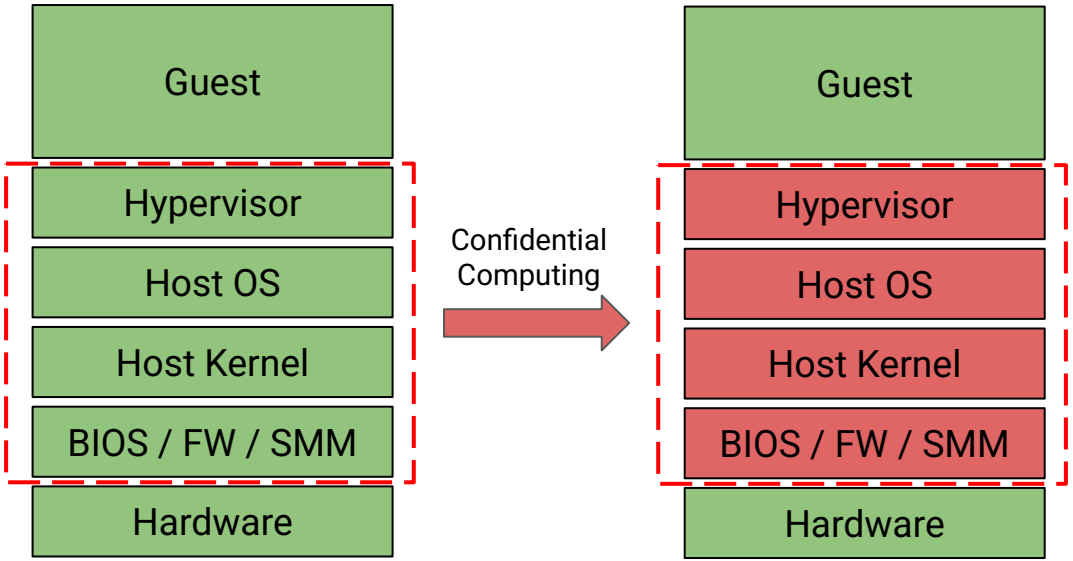
Confidential Computing

Protect data *in use*

- Data at rest or in transit are already protected
- Perform computation in a *hardware-based, attested* Trusted Execution Environment (TEE)
- Hardware-based isolation and attestation are key requirements

Protection from the infrastructure (SW and HW)

A Different Threat Model



Trusted — Guest Trusted Computing Base (TCB)

Untrusted — Untrusted Components

New adversaries

System SW (FW, Kernel, VMM, etc)

Out of Scope

DoS from non TCB components

Why? Some Use Cases…

Cloud providers no longer need to be trusted

Protection from buggy or compromised infrastructure system software

Cryptographically-verifiable data protection for e.g. regulated markets

Confidential AI

- Protection of data and models through the AI lifecycle
- Data control during federated training and inferencing

Requires confidential computing on general purpose CPUs and GPUs

- Must provide data confidentiality, data and code integrity, attestation
- Intel TDX, AMD SEV, ARM CCA

Confidential Computing on RISC-V

Defined by the CoVE and CoVE-IO RISC-V Technical Groups

Confidential VM Extensions - Confidential Computing for Application Processors

Confidential VM Extensions for I/O - Specification for CoVE trusted I/O

Both ISA and non-ISA specifications

Defines a new class of Trusted Execution Environment

Trusted Virtual Machine (TVM) - H extension (Virtualization) is required

Lift and Shift virtual machines, runtime isolated from the host OS, hypervisor and VMM

Run on top of a hardware-rooted, attestable and minimal TCB

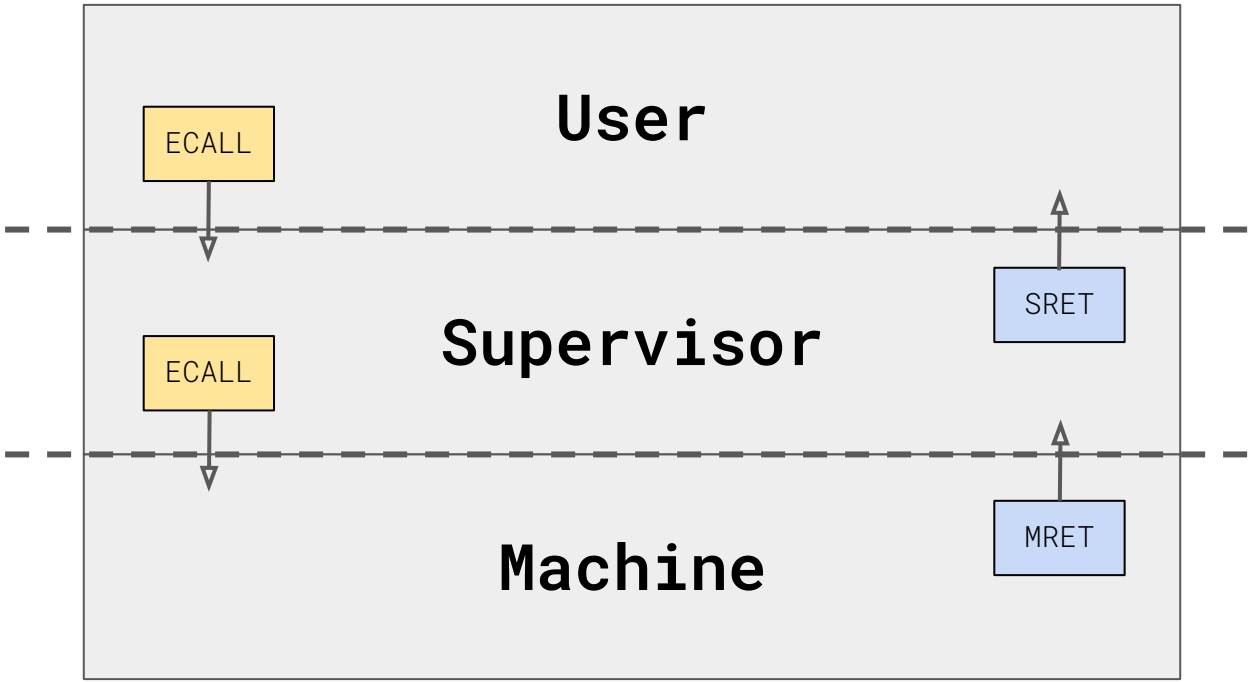
Similar goals and use cases as AMD SEV, Intel TDX or ARM CCA

Confidential Computing on RISC-V

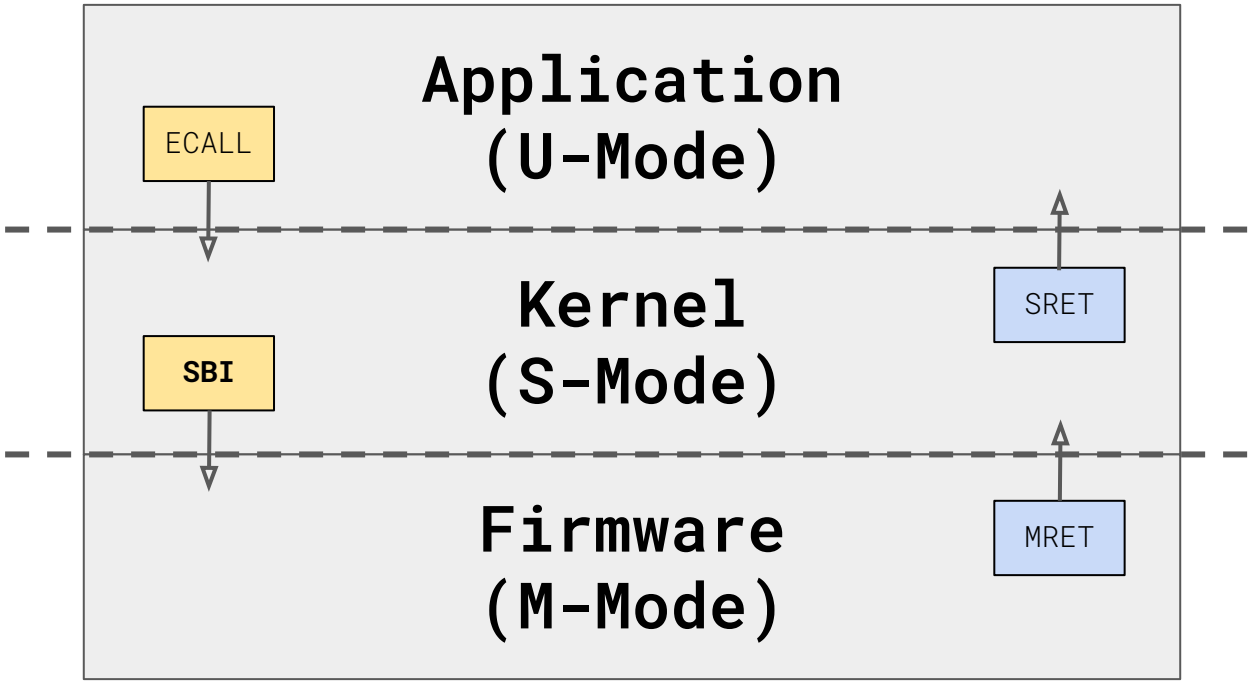
Protect the TVM confidential memory

Data confidentiality, data and code integrity, attestation

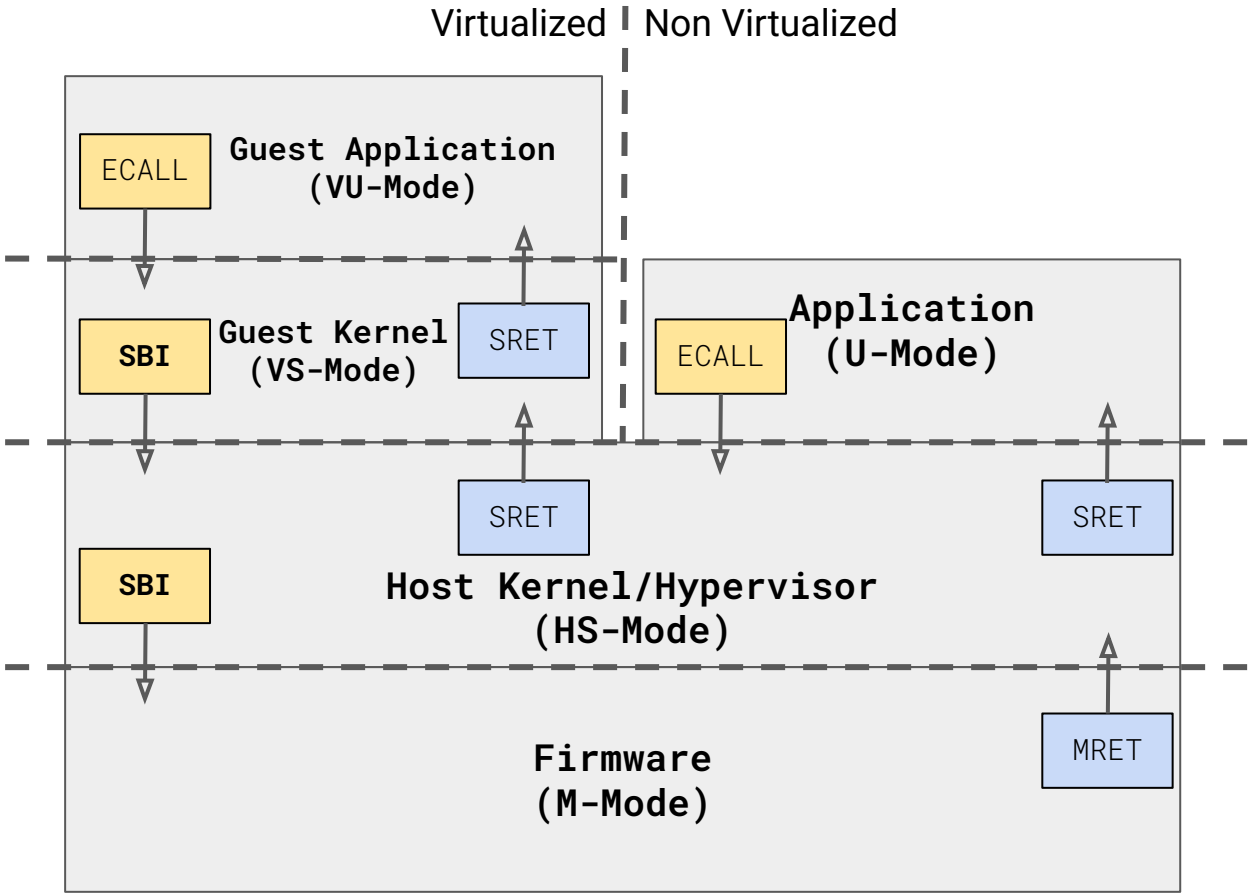
ISA Extensions		Non-ISA		
		Hardware	Software	
Specification	Role		Specification	Role
Supervisor Domain Access Protection	Memory isolation Execution state protection	HW Root-of-Trust IOMMU	***CoVE***	ABI for TVM resource management
			CoVE-IO	ABI for direct device assignment into a TVM



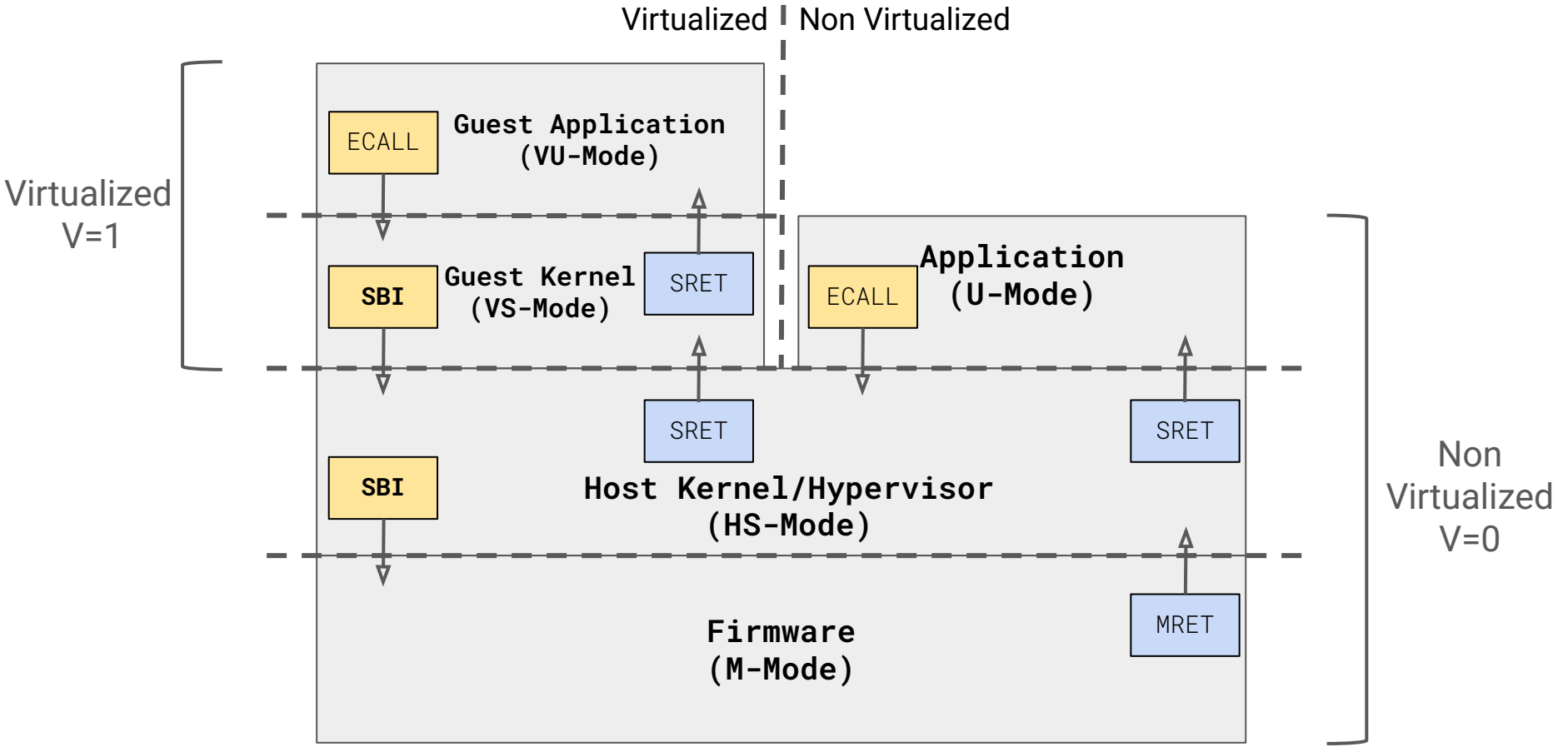
RISC-V Privilege Modes



RISC-V Privilege Modes



RISC-V Privilege Modes with **Hypervisor Extension**



RISC-V Privilege Modes with **Hypervisor Extension**

Supervisor Domains Access Protection

A set of RISC-V privileged ISA extensions

S-mode typically has full access to all physical memory

A single S-mode domain of execution controls all physical memory

Extends to more than one supervisor domain

Managed by the RooT Domain Security Manager (RDSM)

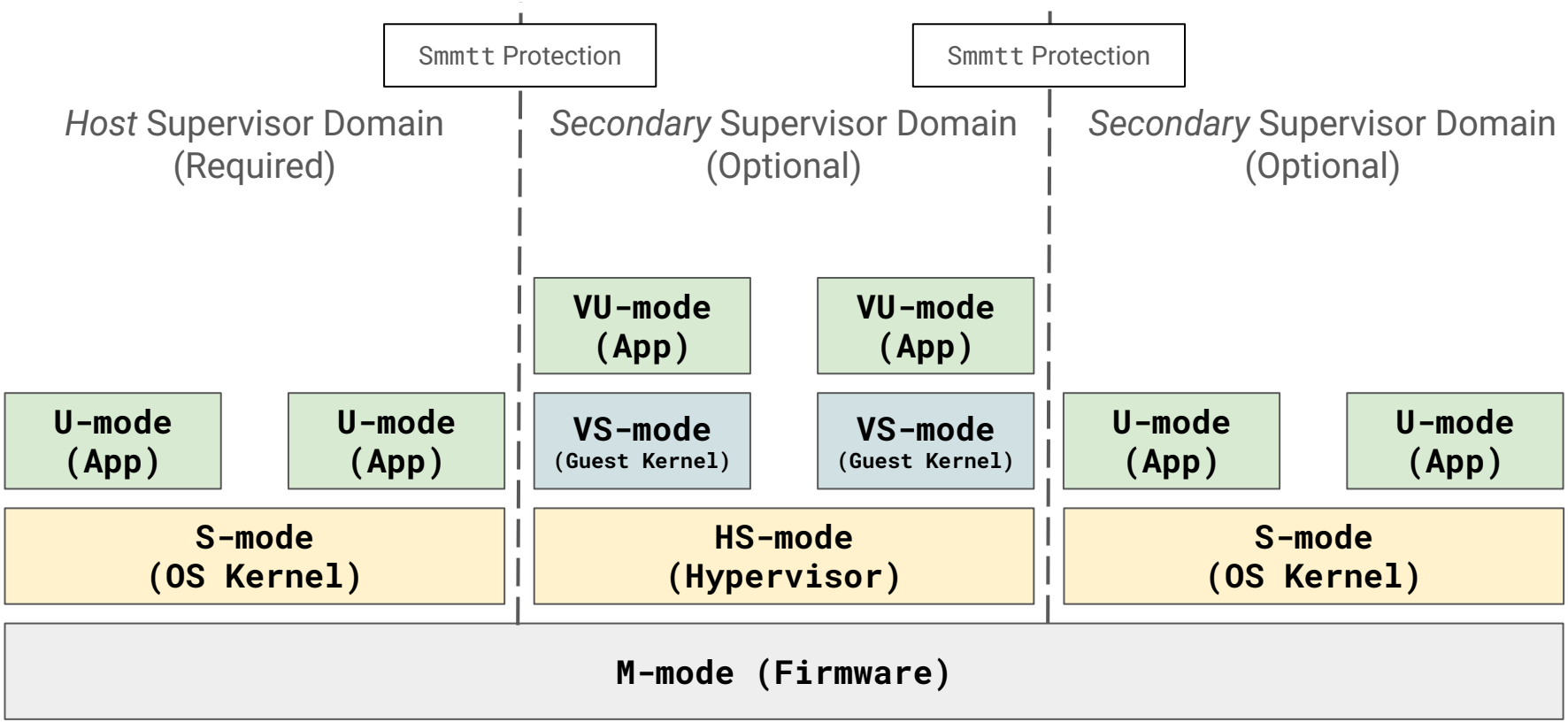
Provides physical address space isolation between all supervisor domains

For both harts and devices

Confidential computing supervisor domain

Memory integrity and confidentiality protection from the host supervisor domain

Supervisor Domains Access Protection



Supervisor Domains - “Smsdid” Extension

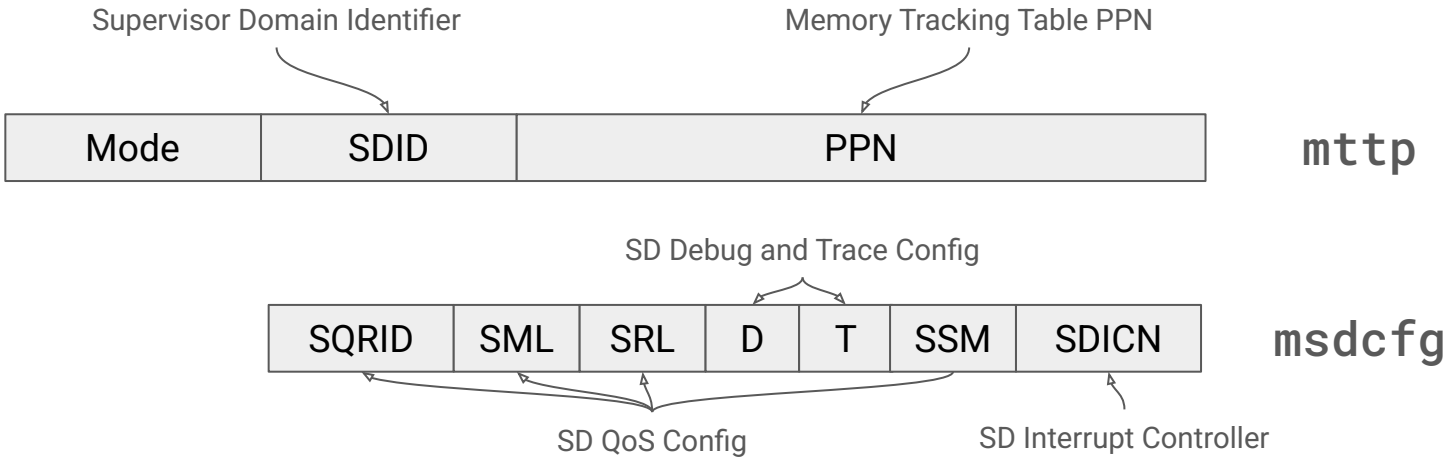
Two M-mode CSRs (`mttp` and `msdcfg`)

Assign Supervisor Domain IDs (SDIDs) to harts

Identifier to manage SD access control

Index into a Memory Tracking Table (MTT, defined in the Smmtt extension)

Two M-mode fence instructions (`MFENCE.SPA` and `MINVAL.SPA`)



Supervisor Domains - “Smmtt” Extension

One Memory Tracking Table (MTT) per supervisor domain

Defines domain memory access control, per physical page

MTT(PA) → Access Control

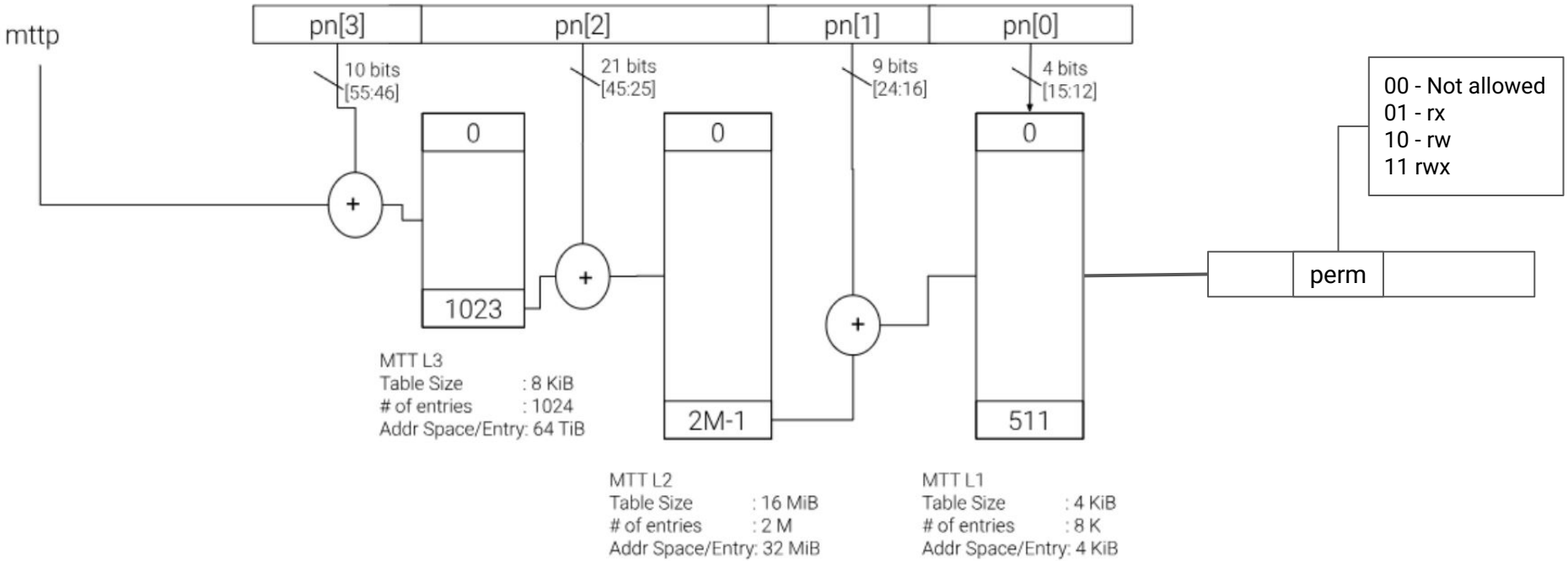
Memory Integrity

Accessing a Domain□ page from Domain□ generates a fault (unless shared)

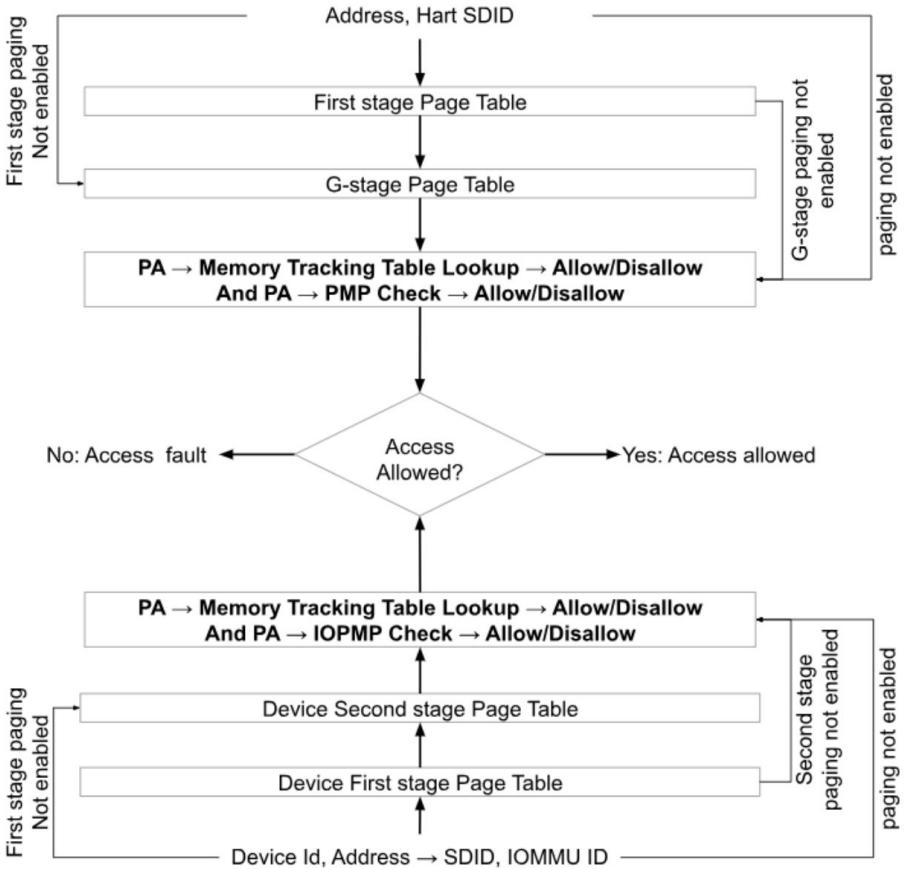
Memory Protection

Memory encryption key selection can be built from the physical address and the domain ID

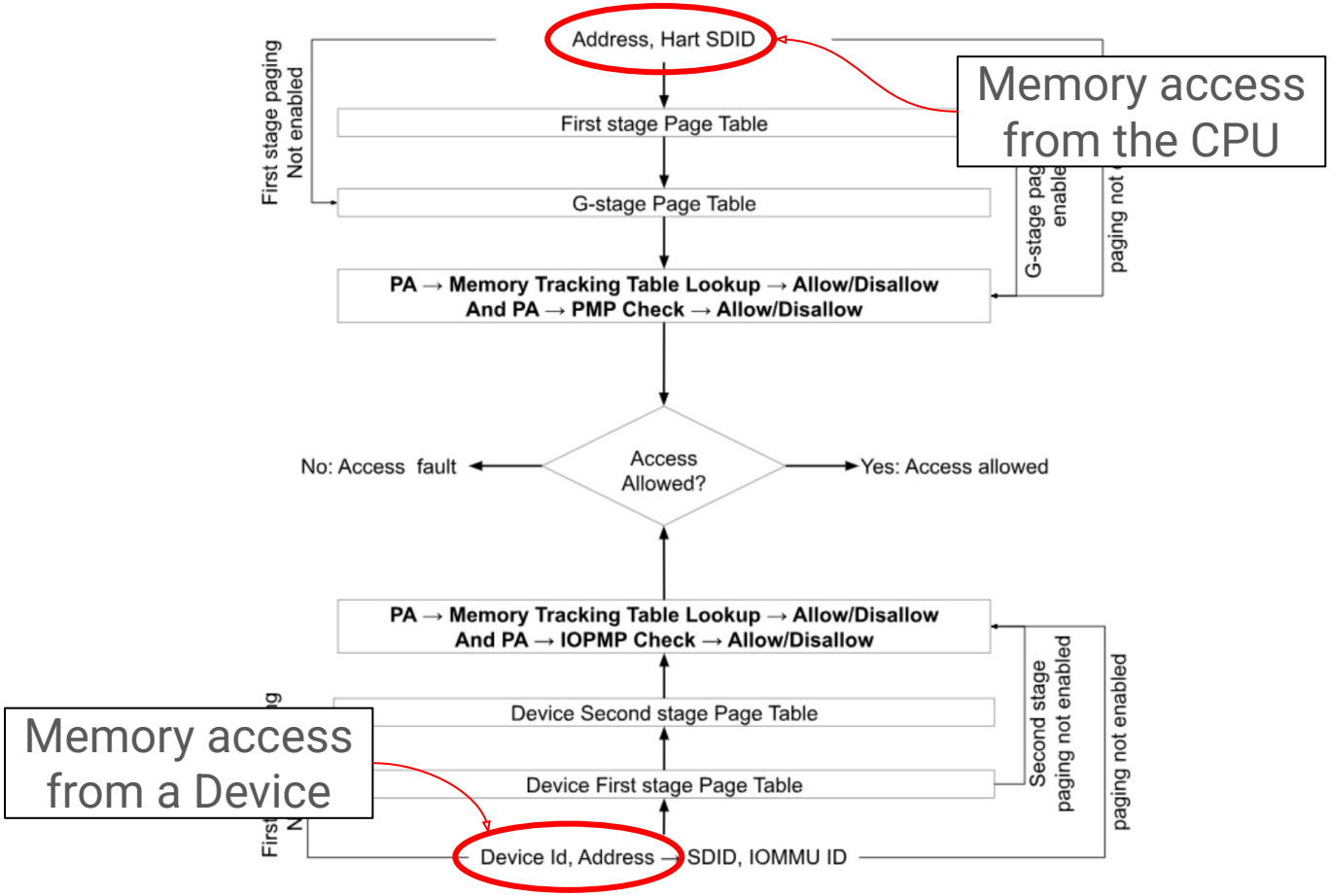
Supervisor Domains - “Smmtt” Extension



Supervisor Domains - “Smmtt” Extension



Supervisor Domains - “Smmtt” Extension



TCB

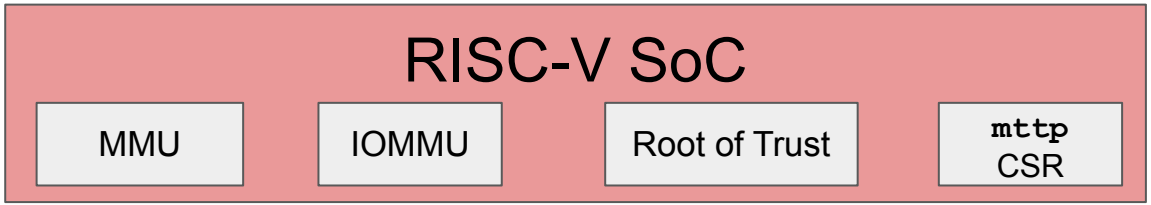
!TCB

VU-mode

VS-mode

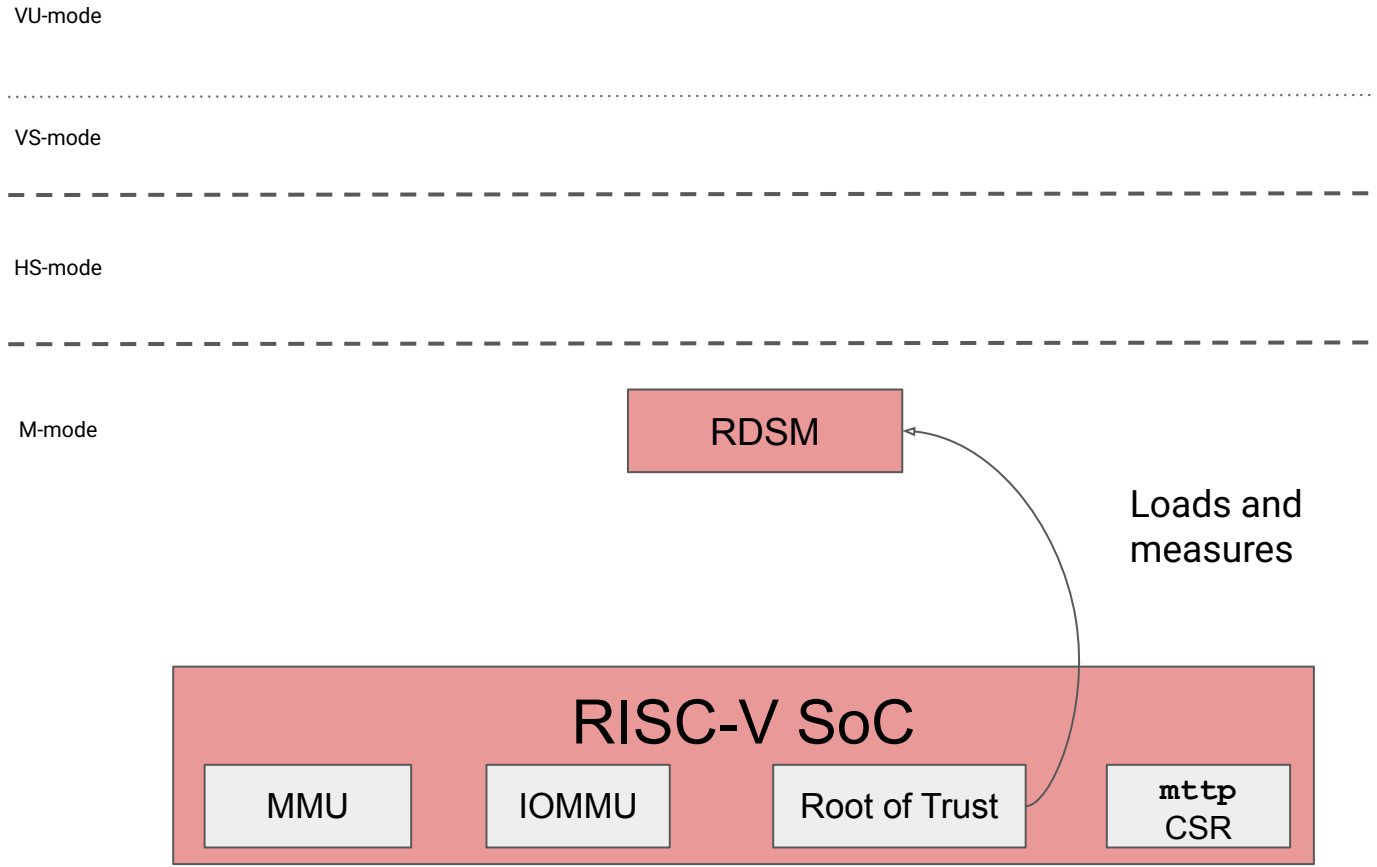
HS-mode

M-mode



TCB

!TCB


VU-mode
VS-mode
HS-mode
M-mode
RDSM
Loads and measures
RISC-V SoC
MMU
IOMMU
Root of Trust
mttp CSR

Root Domain Security Manager (RDSM)

M-mode firmware component, part of the TCB

Supervisor Domain Switcher via the `mttp` CSR

Root Domain Security Manager (RDSM)

M-mode firmware component, part of the TCB

Supervisor Domain Switcher via the `mttp` CSR

Non Confidential (Host Supervisor Domain) → Confidential (Supervisor Domain□)

1. Host hypervisor does a TEECALL SBI call
2. RDSM traps
3. RDSM updates the hart mttp CSR
4. RDSM MRET into the TSM

Root Domain Security Manager (RDSM)

M-mode firmware component, part of the TCB

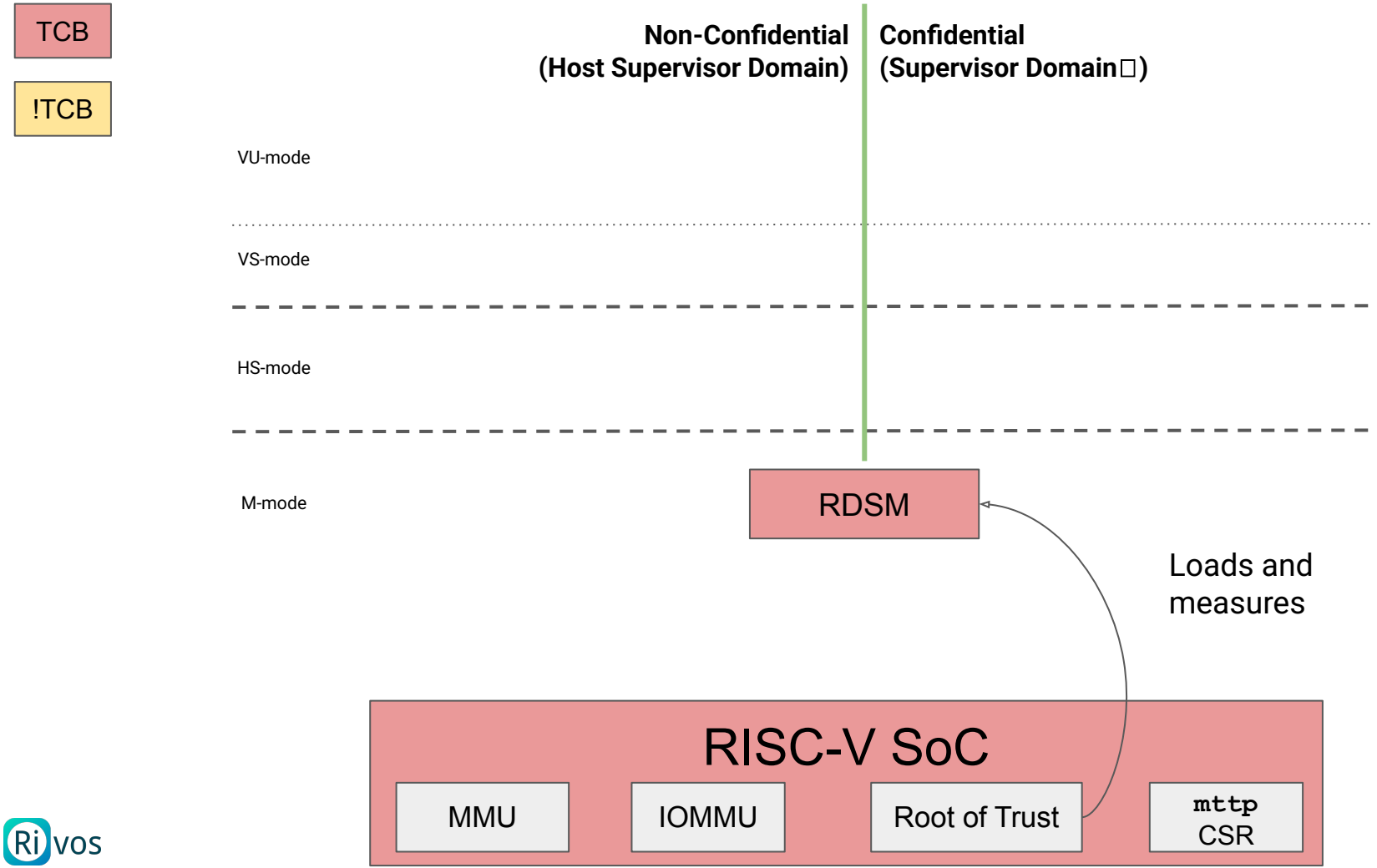
Supervisor Domain Switcher via the `mttp` CSR

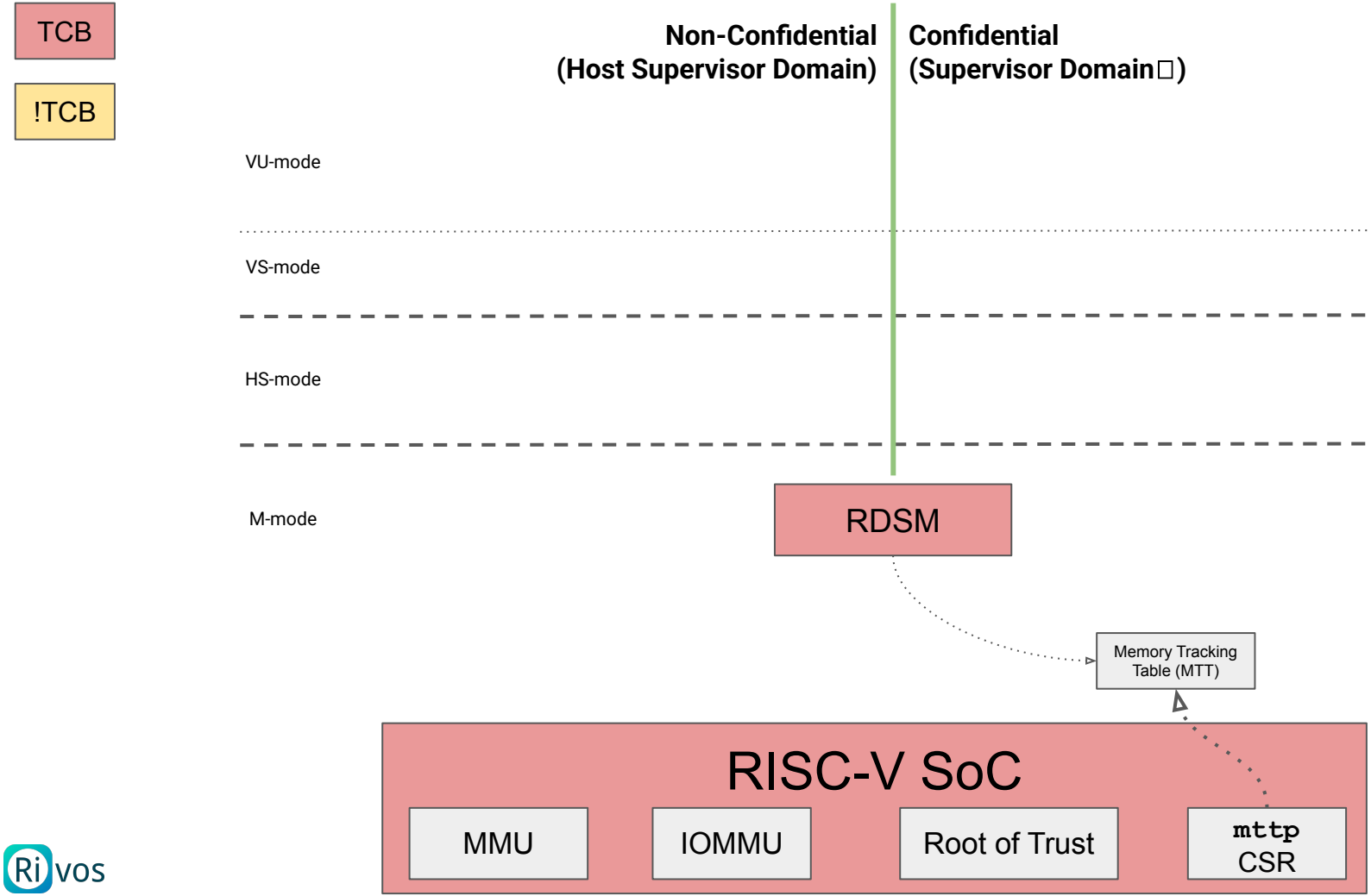
Non Confidential (Host Supervisor Domain) → Confidential (Supervisor Domain□)

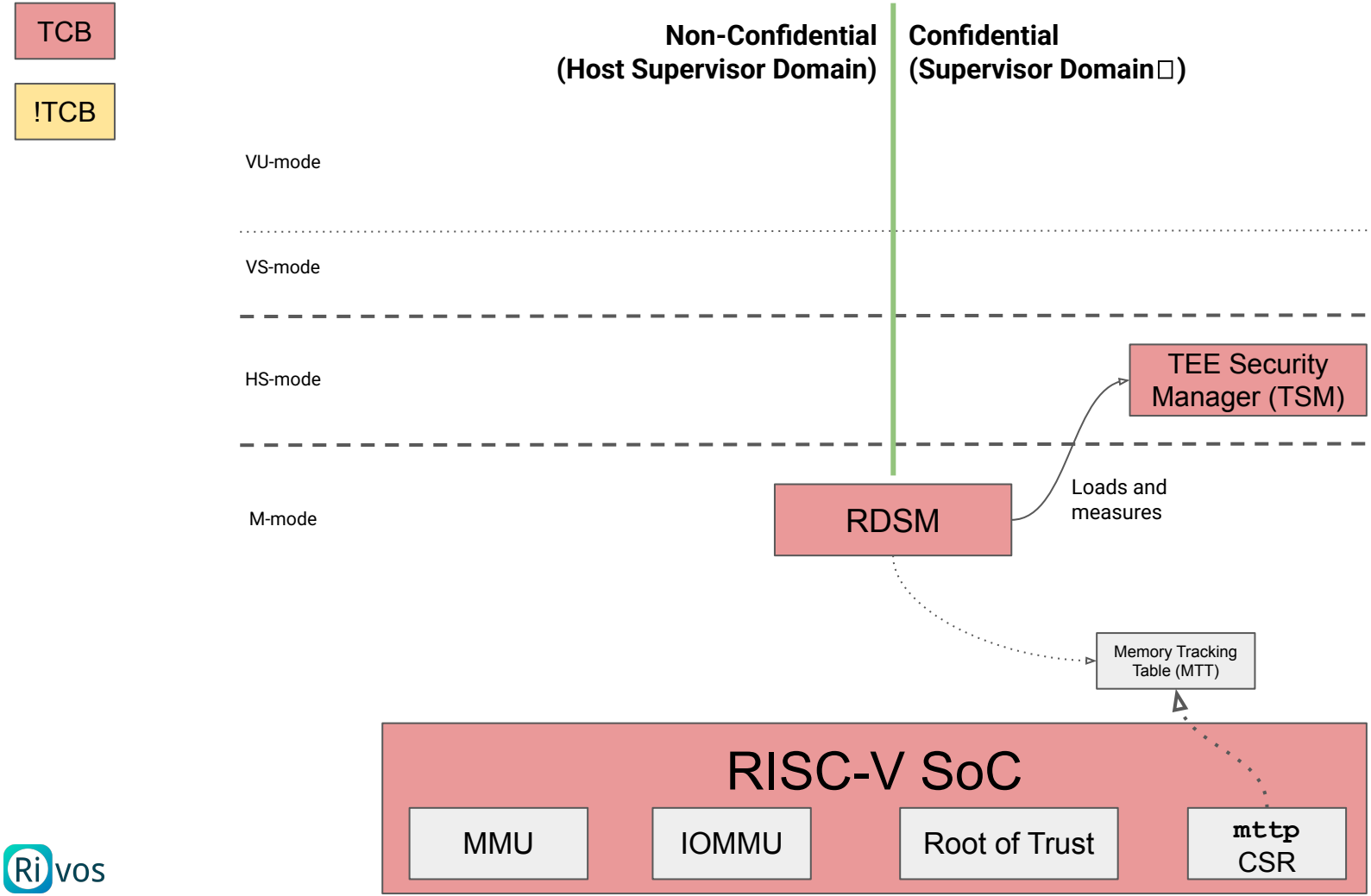
1. Host hypervisor does a TEECALL SBI call
2. RDSM traps
3. RDSM updates the hart mttp CSR
4. RDSM MRET into the TSM

Confidential (Supervisor Domain□) → Non Confidential (Host Supervisor Domain)

1. TSM does a TEERET SBI call
2. RDSM traps
3. RDSM updates the hart mttp CSR
4. RDSM MRET into the hypervisor


TCB
!TCB
Non-Confidential
(Host Supervisor Domain)
Confidential
(Supervisor Domain□)
VU-mode
VS-mode
HS-mode
M-mode
RDSM
Loads and
measures
RISC-V SoC
MMU
IOMMU
Root of Trust
mttp
CSR


TCB
!TCB
Non-Confidential
(Host Supervisor Domain)
Confidential
(Supervisor Domain□)
VU-mode
VS-mode
HS-mode
M-mode
RDSM
Memory Tracking
Table (MTT)
RISC-V SoC
MMU
IOMMU
Root of Trust
mttp
CSR


TCB
!TCB
Non-Confidential
(Host Supervisor Domain)
Confidential
(Supervisor Domain□)
VU-mode
VS-mode
HS-mode
M-mode
TEE Security
Manager (TSM)
Loads and
measures
RDSM
Memory Tracking
Table (MTT)
RISC-V SoC
MMU
IOMMU
Root of Trust
mttp
CSR

TEE Security Manager (TSM)

A trusted intermediary between the host VMM and the TVMs

The Supervisor Domain Security Manager (SDSM) for the Confidential domain

- Open source reference implementation at https://github.com/rivosinc/salus

Manages all TVM second-stage (G-stage) page tables

- TVM G-stage page tables must be in confidential memory

Passive component

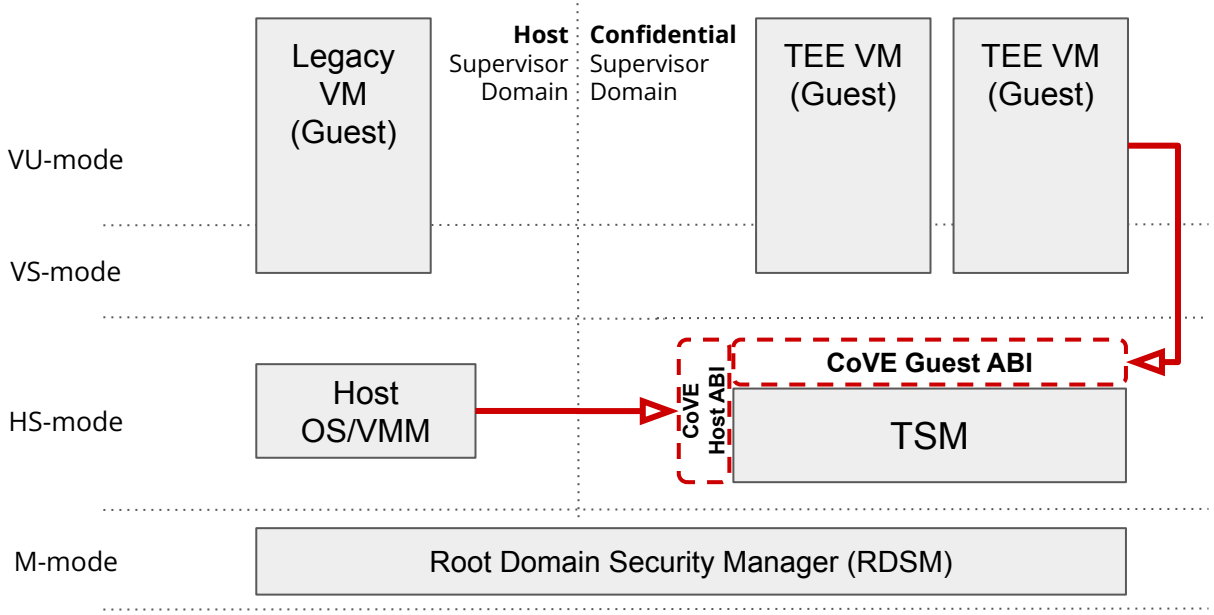
- Enforces CC security attributes for the TVMs (through G-stage and MTT)
- Does not schedule TVMs. Does not handle interrupts.
- Implements the CoVE and CoVE-IO specifications

RISC-V CoVE

A non-ISA specification for managing TVMs and their resources

Describes the RISC-V confidential computing ABI

Split between host and guest parts



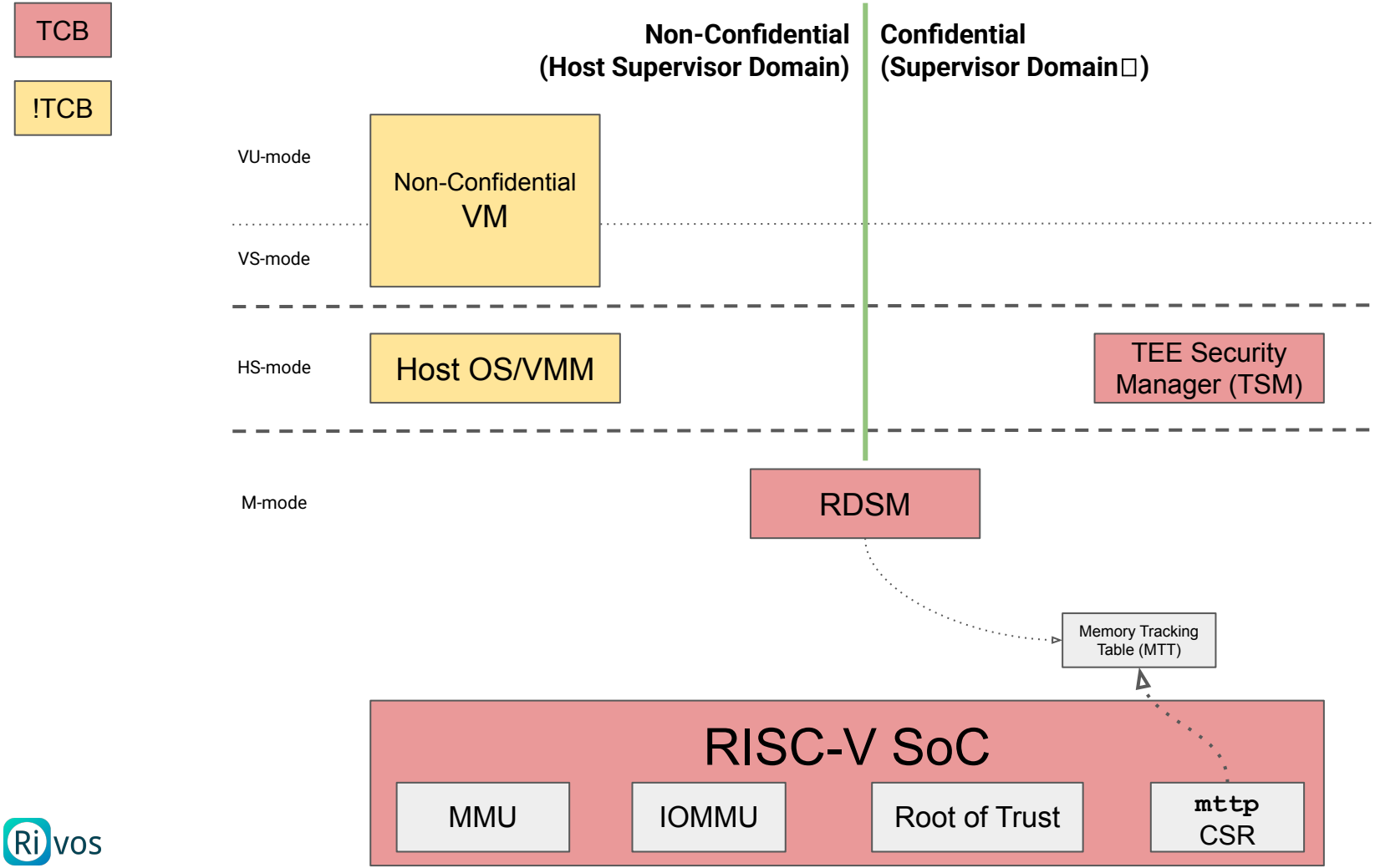
RISC-V CoVE

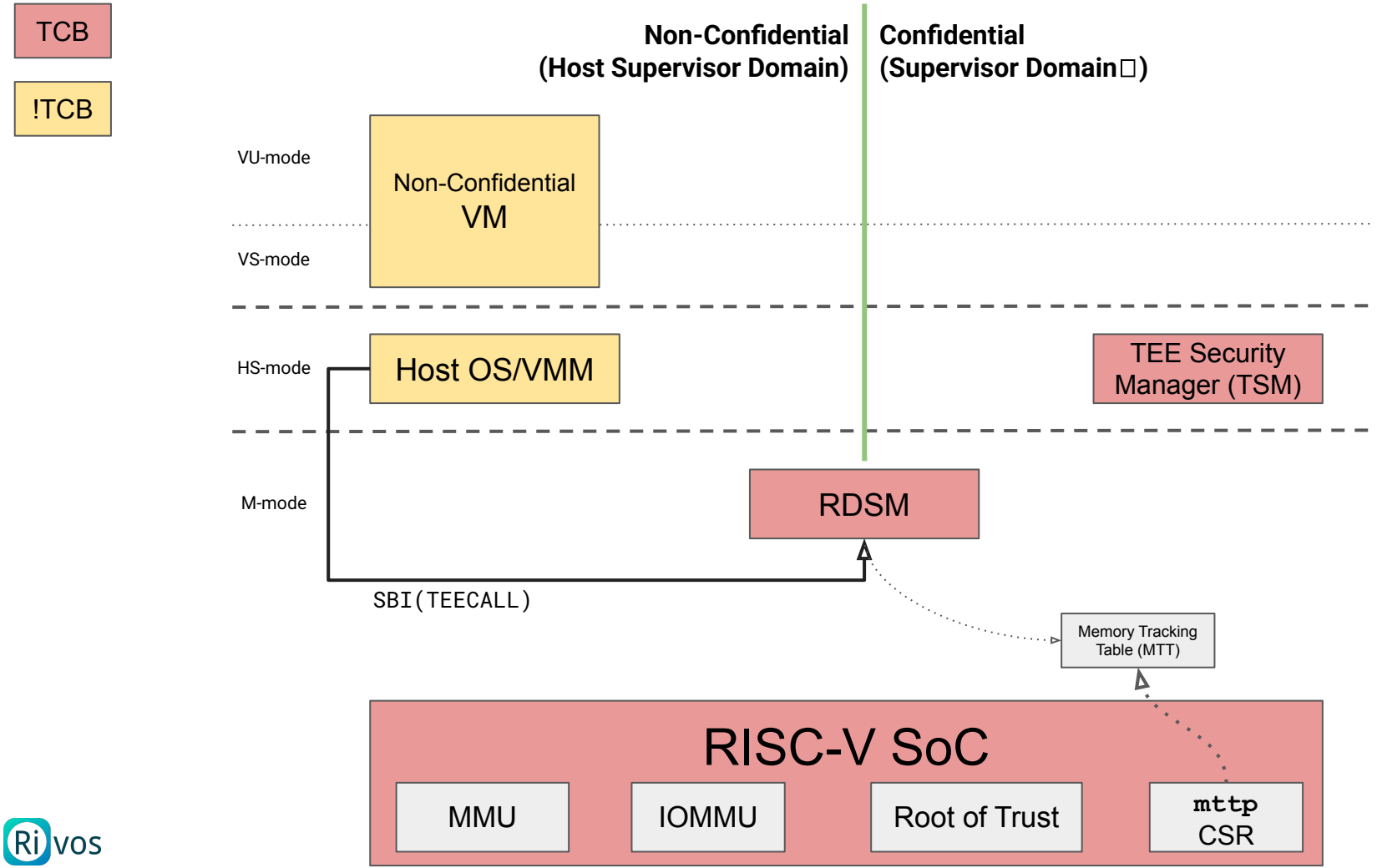
ABI is proxied through the RDSM

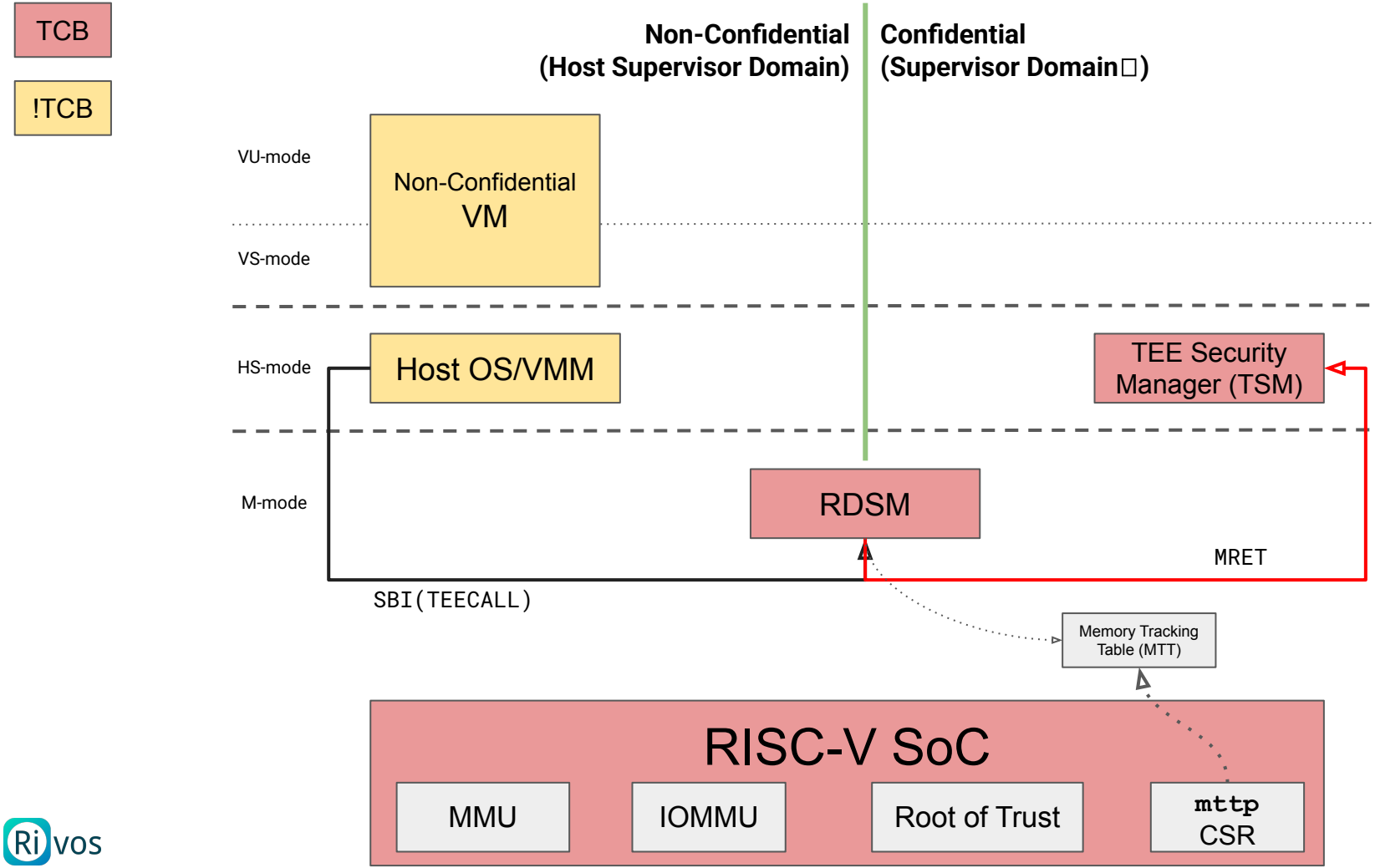
1. Host VMM call the CoVE Host ABI
2. RDSM traps the host SBI call and MRET into the TSM
3. TSM services the host VMM request
4. TSM makes a TEERET upon request completion
5. RDSM traps the TSM TEERET and MRET back to host VMM

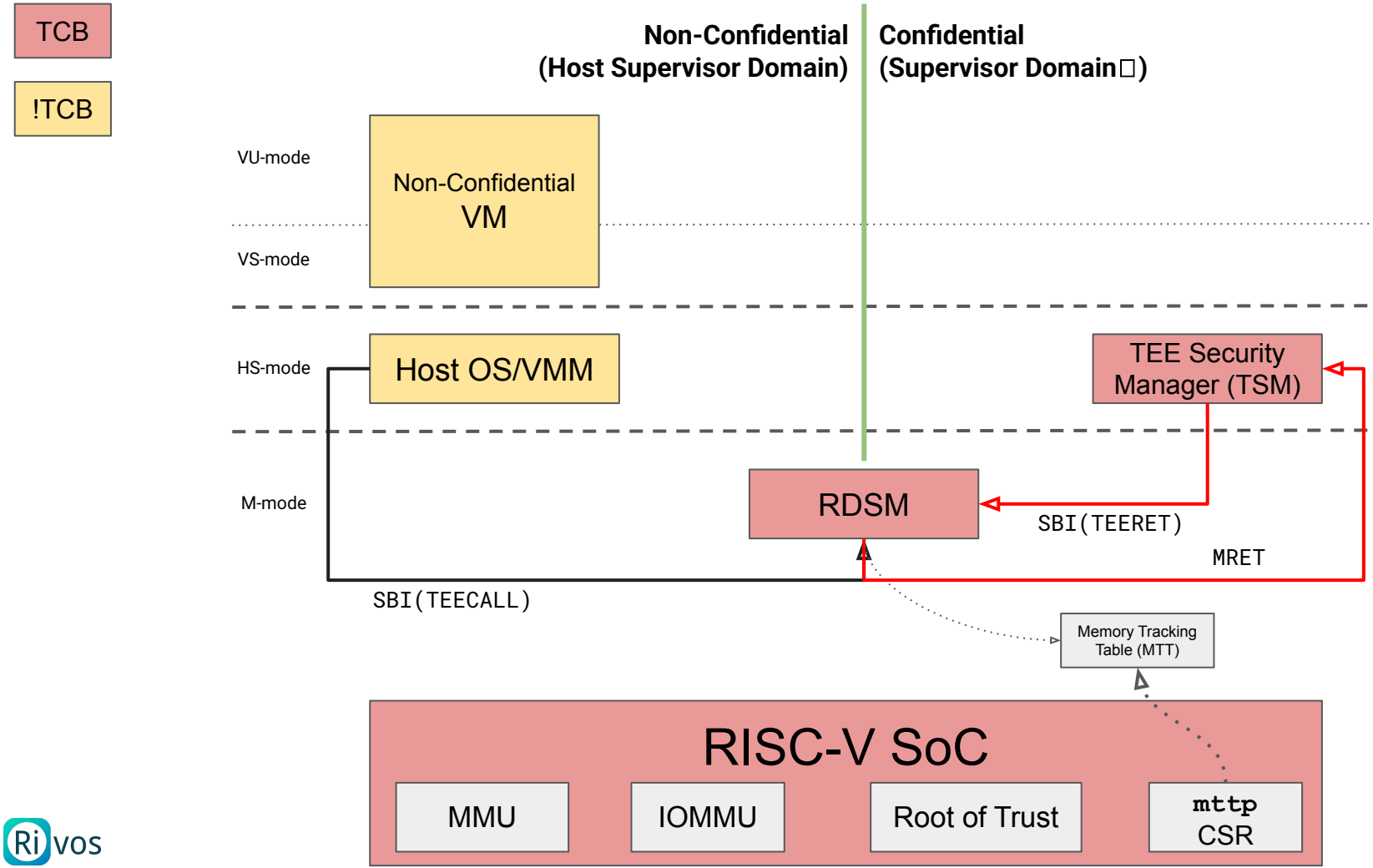
Examples

- Creating and destroying a TVM
- Converting !confidential memory to confidential, reclaiming confidential memory
- Mapping measured and zero pages into a TVM address space
- Donating confidential memory to the TSM
- Creating and running a TVM vCPU


TCB
!TCB
Non-Confidential
(Host Supervisor Domain)
Confidential
(Supervisor Domain□)
VU-mode
VS-mode
Non-Confidential
VM
HS-mode
Host OS/VMM
TEE Security
Manager (TSM)
M-mode
RDSM
Memory Tracking
Table (MTT)
RISC-V SoC
MMU
IOMMU
Root of Trust
mttp
CSR

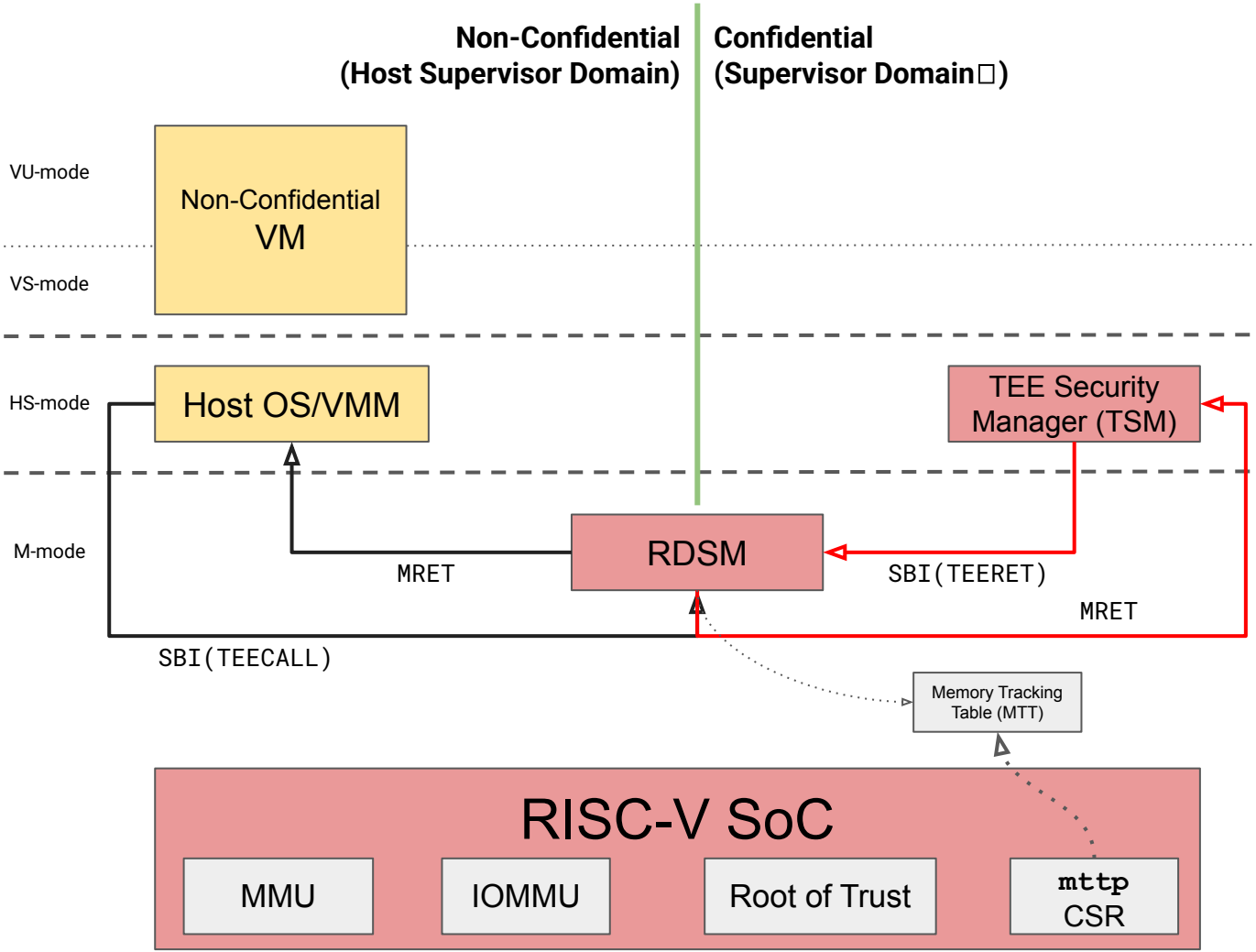

TCB
!TCB
Non-Confidential
(Host Supervisor Domain)
Confidential
(Supervisor Domain□)
VU-mode
VS-mode
Non-Confidential
VM
HS-mode
Host OS/VMM
TEE Security
Manager (TSM)
M-mode
RDSM
SBI(TEECALL)
Memory Tracking
Table (MTT)
RISC-V SoC
MMU
IOMMU
Root of Trust
mttp
CSR


TCB
!TCB
Non-Confidential
(Host Supervisor Domain)
Confidential
(Supervisor Domain□)
VU-mode
VS-mode
Non-Confidential
VM
HS-mode
Host OS/VMM
TEE Security
Manager (TSM)
M-mode
RDSM
MRET
SBI(TEECALL)
Memory Tracking
Table (MTT)
RISC-V SoC
MMU
IOMMU
Root of Trust
mttp
CSR


TCB
!TCB
Non-Confidential
(Host Supervisor Domain)
Confidential
(Supervisor Domain□)
VU-mode
VS-mode
Non-Confidential
VM
HS-mode
Host OS/VMM
TEE Security
Manager (TSM)
M-mode
RDSM
SBI(TEERET)
MRET
SBI(TEECALL)
Memory Tracking
Table (MTT)
RISC-V SoC
MMU
IOMMU
Root of Trust
mttp
CSR

TCB

!TCB

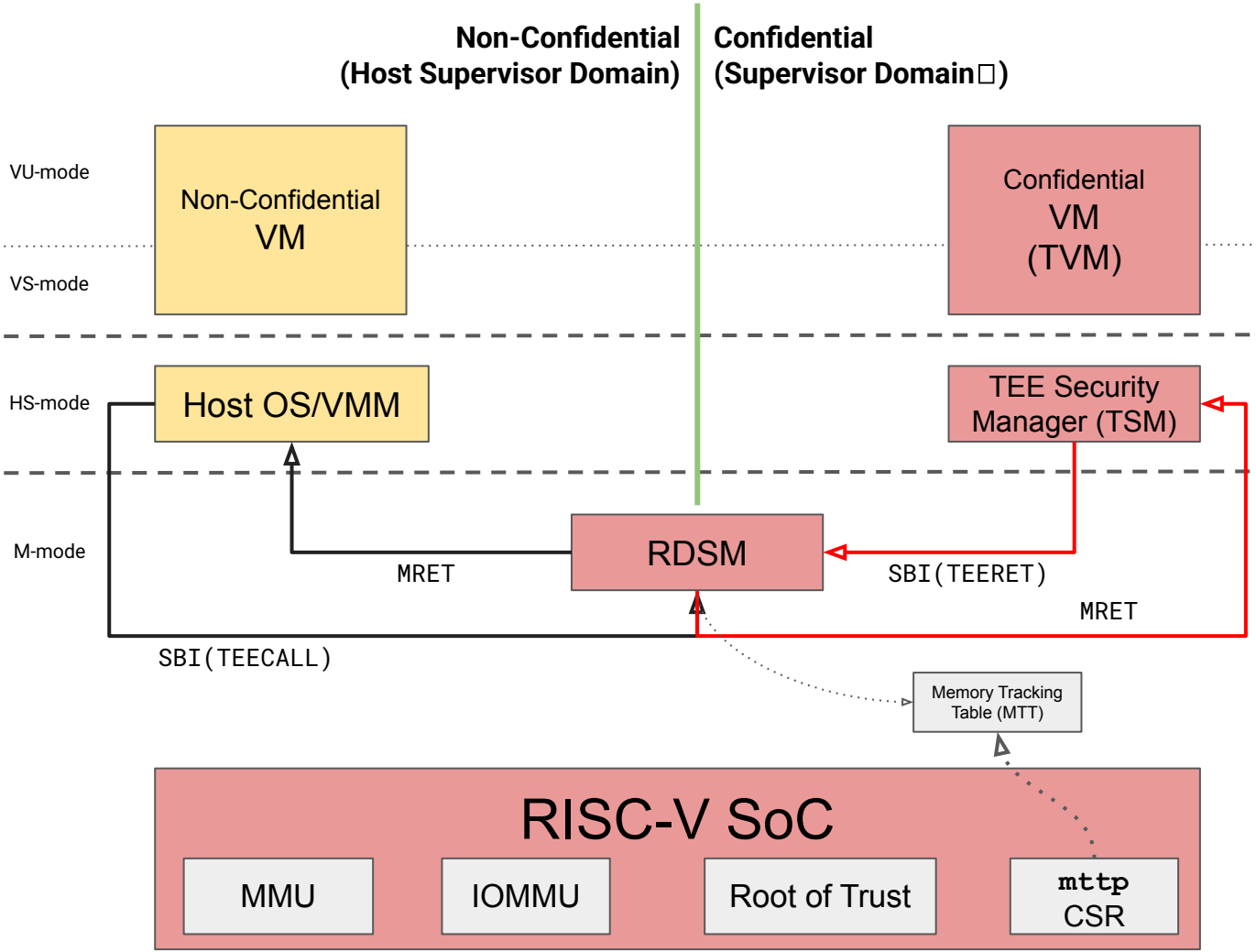

Non-Confidential
(Host Supervisor Domain)
Confidential
(Supervisor Domain□)
VU-mode
VS-mode
Non-Confidential
VM
HS-mode
Host OS/VMM
TEE Security
Manager (TSM)
M-mode
RDSM
MRET
SBI(TEERET)
MRET
SBI(TEECALL)
Memory Tracking
Table (MTT)
RISC-V SoC
MMU
IOMMU
Root of Trust
mttp
CSR

CoVE ABI (Host, COVH-ABI) - TVM Creation

1. Create a TVM context
   a. sbi_covh_create_tvm( )
2. Donate confidential memory for the TVM 2nd stage page tables
   a. sbi_covh_add_tvm_page_table_pages()
3. Reserve TVM confidential memory regions
   a. sbi_covh_add_tvm_memory_region()
4. Add measured and zero pages to the TVM
   a. sbi_covh_add_tvm_measured_pages(), sbi_covh_add_tvm_zero_pages()
5. Create the TVM vCPUs
   a. sbi_covh_create_tvm_vcpu()
6. Finalize the TVM
   a. sbi_covh_finalize_tvm()
7. Run a TVM
   a. sbi_covh_run_tvm_vcpu()

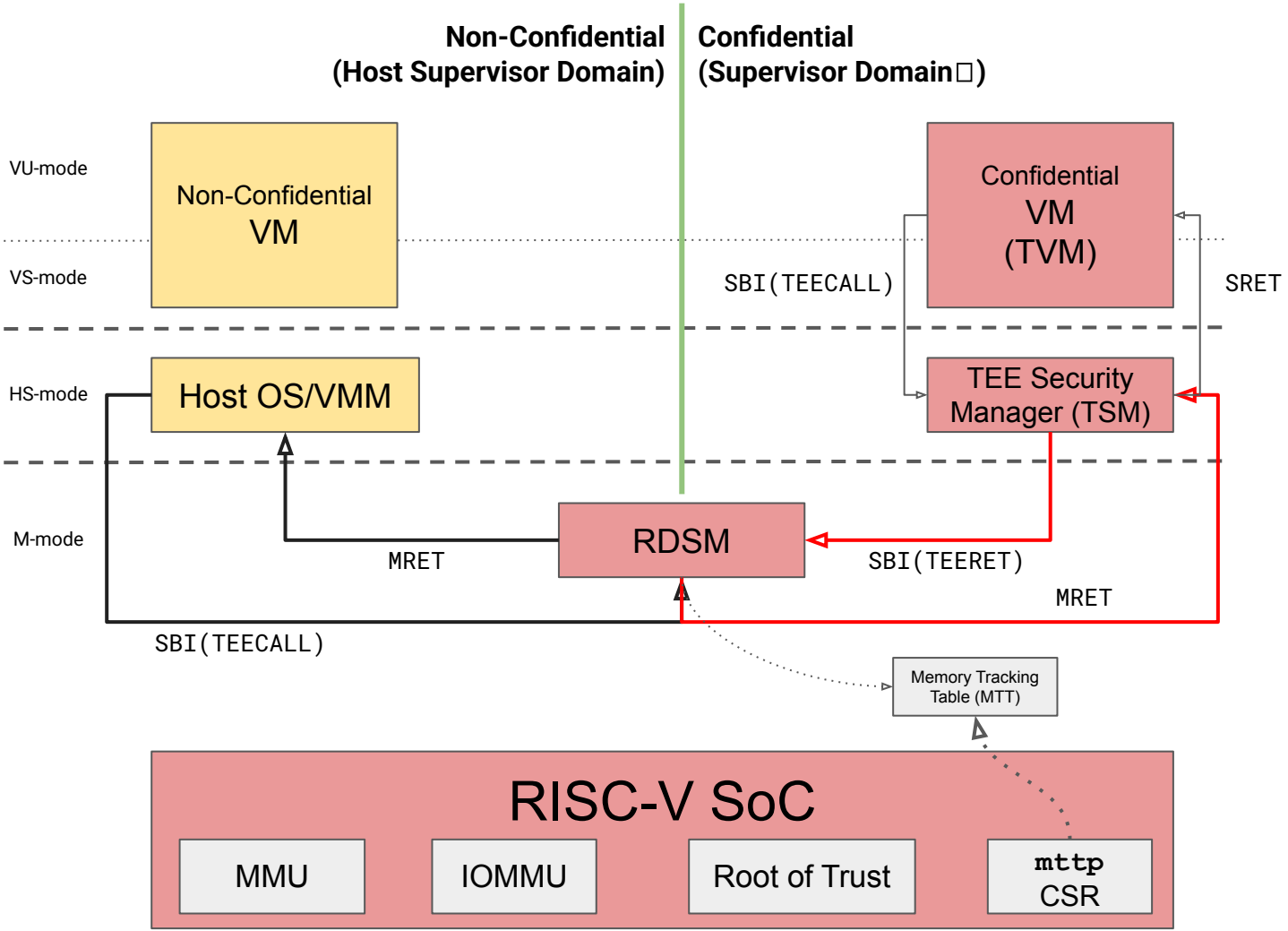
TCB

!TCB


Non-Confidential
(Host Supervisor Domain)
Confidential
(Supervisor Domain□)
VU-mode
VS-mode
Non-Confidential
VM
Confidential
VM
(TVM)
HS-mode
Host OS/VMM
TEE Security
Manager (TSM)
M-mode
RDSM
MRET
SBI(TEERET)
MRET
SBI(TEECALL)
Memory Tracking
Table (MTT)
RISC-V SoC
MMU
IOMMU
Root of Trust
mttp
CSR

TCB

!TCB


Non-Confidential
(Host Supervisor Domain)
Confidential
(Supervisor Domain□)
VU-mode
VS-mode
Non-Confidential
VM
Confidential
VM
(TVM)
SBI(TEECALL)
SRET
HS-mode
Host OS/VMM
TEE Security
Manager (TSM)
M-mode
RDSM
MRET
SBI(TEERET)
MRET
SBI(TEECALL)
Memory Tracking
Table (MTT)
RISC-V SoC
MMU
IOMMU
Root of Trust
mttp
CSR

CoVE ABI (Guest, COVG-ABI)

TVMs can also call into the TSM, through the CoVE ABI

Memory management

Open memory region for emulated MMIO

Interrupt delivery

Allow for external interrupts to be safely delivered

Attestation

Retrieve attestation capabilities and evidence

RISC-V Confidential Computing Attestation

Confidential computing without attestation is pointless

What is the point of protecting data without knowing how it will be used?

A tenant must be able to cryptographically verify:

- The software stack that is running the TEE
- The platform state and origin on top of which the TEE is running

a.k.a Attestation Evidence

CoVE Attestation

Attestation model

Attestation evidence format

Attestation flows and ABIs

Root of Trust for attestation

Layered Attestation

Layered architecture based on TCG DICE

Each TCB layer requests the root-of-trust to load the next one

The root-of-trust verifies, loads and endorses each layer

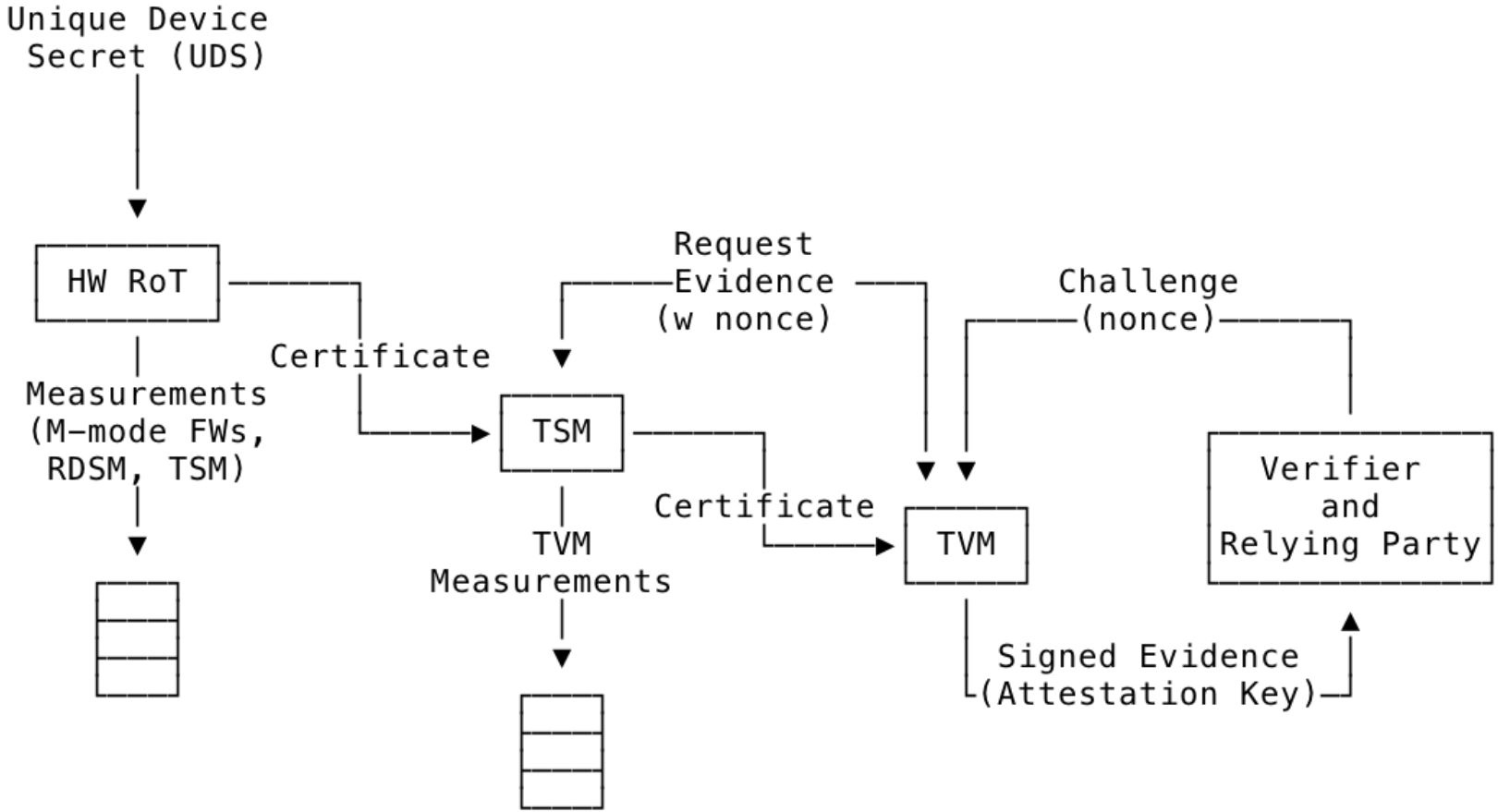
- And builds a TCB certificate chain, rooted into its silicon creator endorsed certificate

The root-of-trust loads the TSM in confidential memory

The TSM delivers attestation evidences to the TVMs

- Signed by the root-of-trust

Layered Attestation



Layered Attestation Evidence

Platform, TSM and TVM tokens

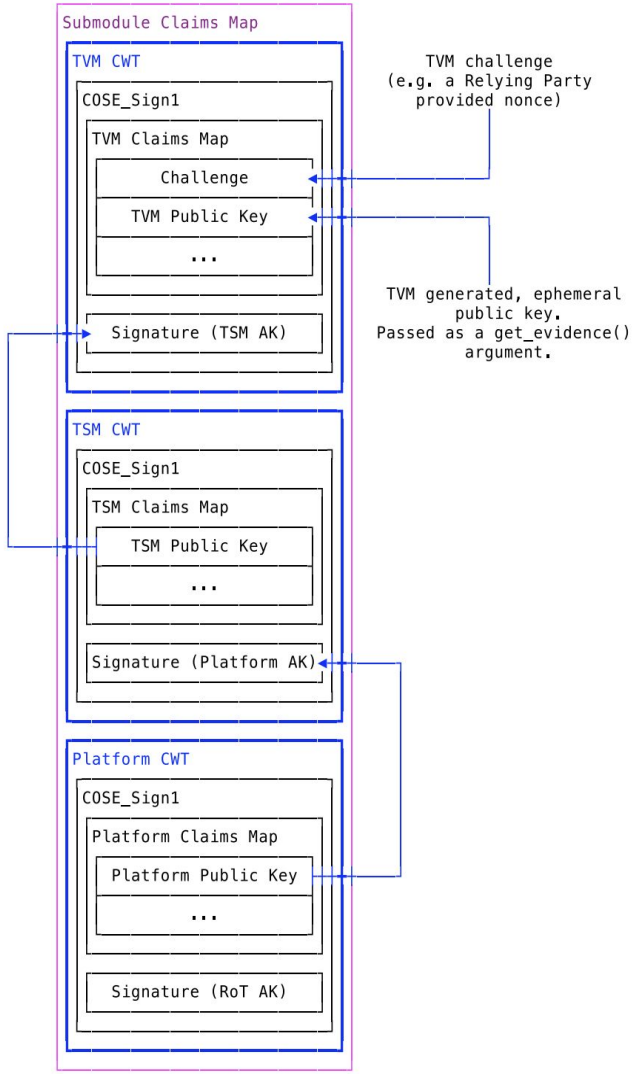
Cryptographically linked together

An attestation certificate chain

Cryptographic proof of the TVM TCB

Verifiable, standardized view of the TVM TCB

For the relying party to verify an appraise



Rivos

Attestation ABI

Defined by the CoVE specification

A COVG (Guest <-> TSM) ABI

- Get TSM attestation capabilities
- Extend runtime measurements
- Get attestation evidence

The TVM asks for an attestation evidence (triggered by the relying party)

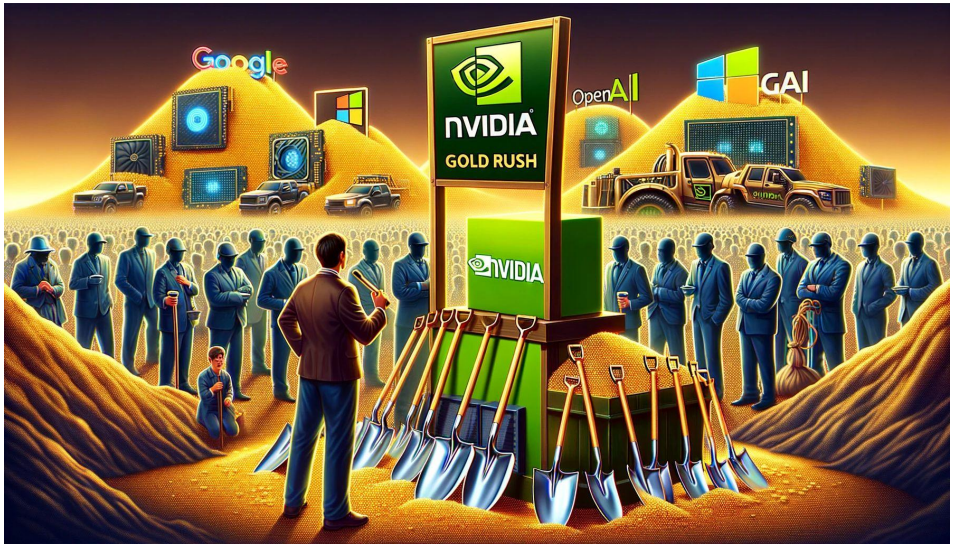
The TSM generates and signs the attestation certificate

Confidential Computing and I/O

Remember the Confidential AI use case?

Most TVM require fast and performant I/O

Accelerators, NICs, etc



Challenges

A devices attached to a TVM is a part of its TCB

Devices can DMA in and out of a TVM confidential memory

- Asynchronously, not under the TSM or RDSM control
- Even when the supervisor domain is not active

Devices can inject interrupts into the TVM

Devices are not trustworthy

- Identity? State? Firmware?

Confidential computing security threat

How do you extend a TVM TCB with devices?

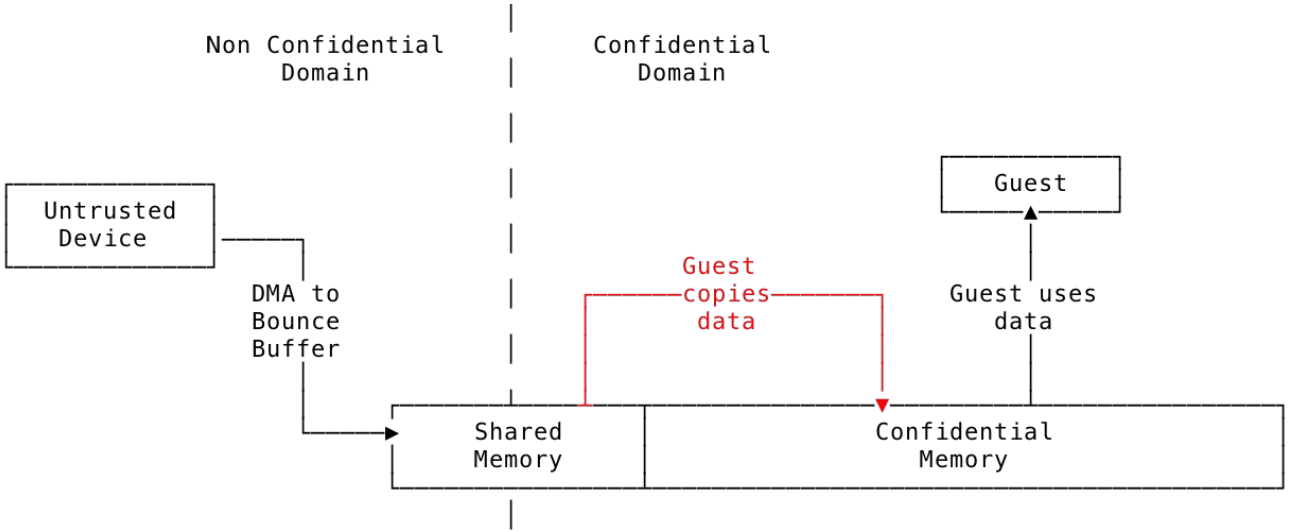
Paravirtualized Model

Devices DMA into a bounce buffer - Guest copies into confidential memory

e.g. Linux swiotlb

High latency, poor throughput

Only used for paravirtualized I/O (virtio devices)



Direct Device Assignment

Extend a TVM TCB with physical devices

Protect the TVM confidential memory from:

- The assigned devices
- All other devices on the platform
- The untrusted host SW
- Physical attacks

Direct Device Assignment

Extend a TVM TCB with physical devices

Protect the TVM confidential memory from:

The assigned devices - Device attestation, IOMMU, IO-MTT, Smsdia (Secure interrupts)

All other devices on the platform - IOMMU, IO-MTT

The untrusted host SW - PCIe TDISP, IOMMU, IO-MTT

Physical attacks - PCIe Integrity and Data Encryption

Direct Device Assignment

Extend a TVM TCB with physical devices

Protect the TVM confidential memory from:

- The assigned devices - Device attestation, IOMMU, IO-MTT, Smsdia (Secure interrupts)
- All other devices on the platform - IOMMU, IO-MTT
- The untrusted host SW - PCIe TDISP, IOMMU, IO-MTT
- Physical attacks - PCIe Integrity and Data Encryption

Defined by the CoVE-IO specification

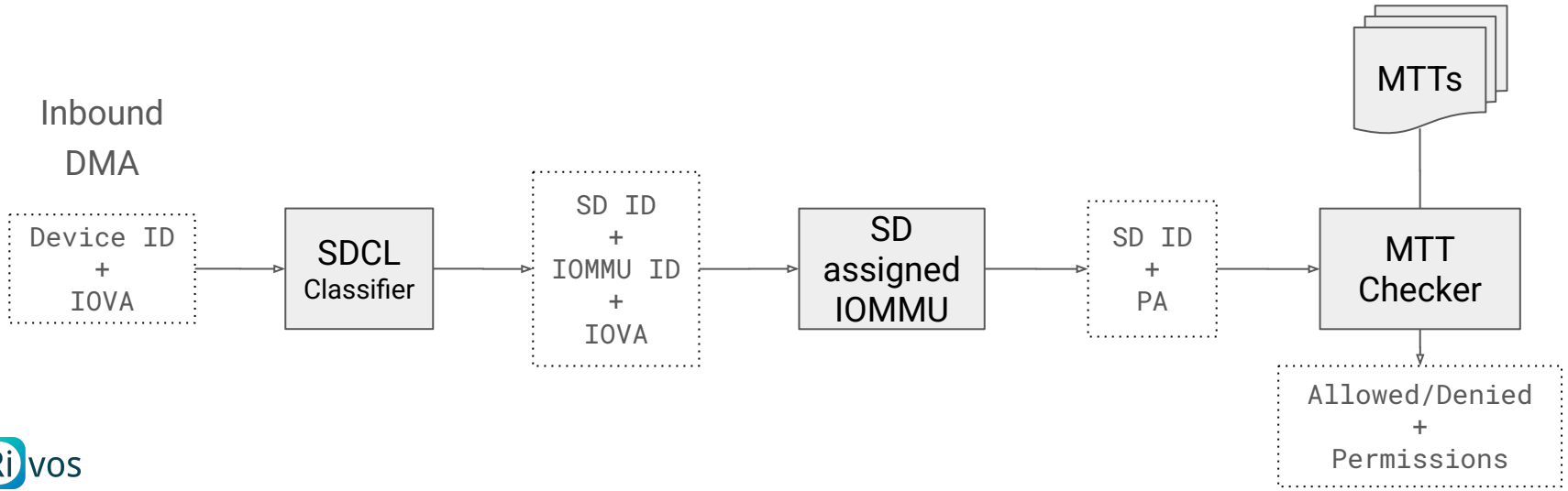
Supervisor Domains - “IO-MTT” Extension

Assigns IOMMU instances to security domains

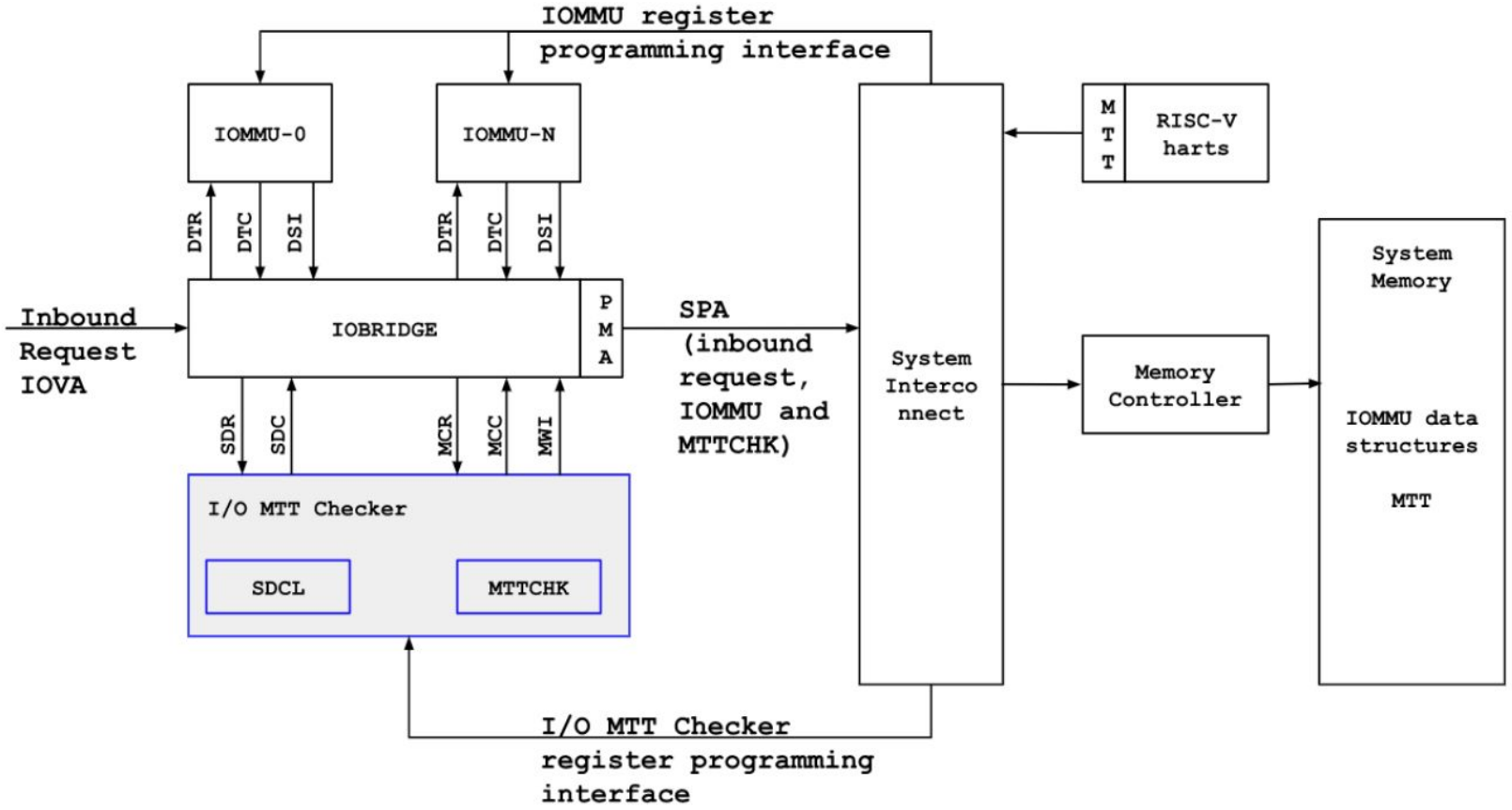
IOMMU access controlled via MTT as well

SD manager configures DMA mappings through its assigned IOMMU

Inbound DMA must adhere strictly to the device assigned MTT permissions



Supervisor Domains - “IO-MTT” Extension



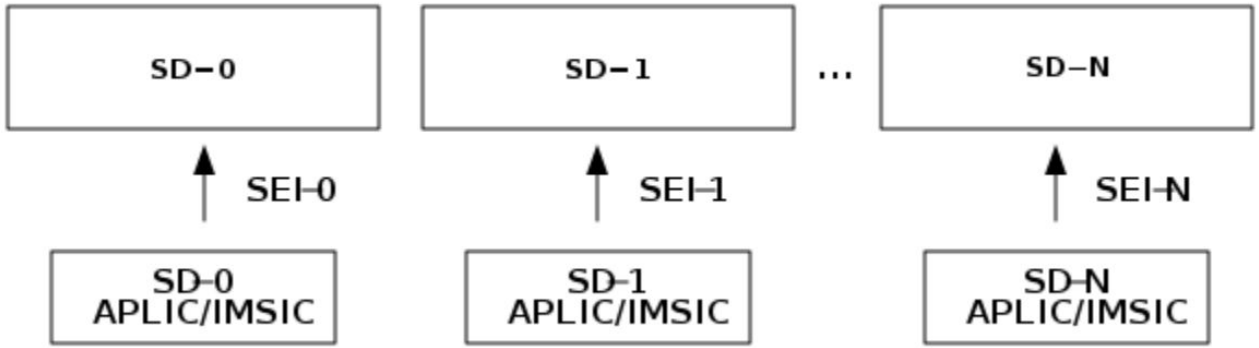
Supervisor Domains - “Smsdia” Extension

Assigns interrupt controllers to supervisor domains

- Per hart msdcfg CSR - msdeie CSR for notifying the RDSM of pending interrupts
- MTT to limit and enforce exclusive access from the supervisor domain

Extends the RISC-V interrupt architecture (AIA)

External interrupts are directly delivered to supervisor domains



Supervisor Domains - Confidential Computing I/O

IO-MTT enforces DMA to be MTT controlled

Smsdia allows for secured interrupt delivery on a supervisor domain basis

Is that enough?

Does the TVM know anything about the device?

Does the host have access to the TVM assigned devices?

Is the physical link protected?

PCIe TEE-I/O

Restricting confidential computing I/O to PCIe devices

TEE Device Interface Secure Protocol (TDISP) - PCIe 6 specified

Three major components

- The TDISP state machine
- SPDM
- PCIe Integrity and Data Encryption (IDE)

TDISP is a CoVE-IO requirement

TDISP

TEE Device Interface Security Protocol (PCI SIG)

Implements the TEE-IO Security Model

TEE-IO Device Interface

a.k.a. TDI = Device or a slice of a device (cf PCIe SR-IOV)

Host Side - TSM

Drives TDISP enabled devices lifecycle (Upon VMM requests)

Binds TDIs to TVMs

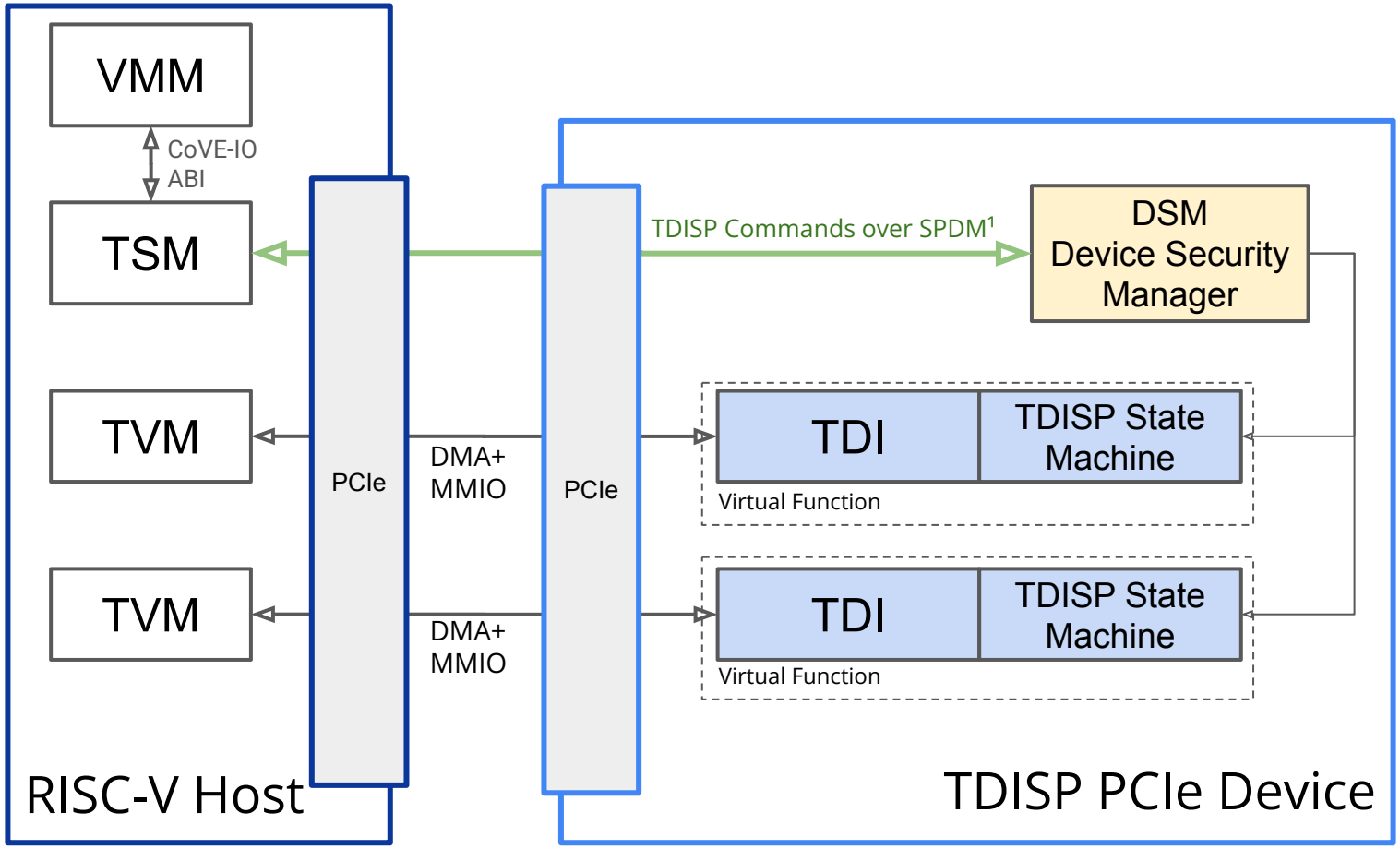
Enforces memory isolation between TVM bound TDIs and the untrusted host

Device Side - Device Security Manager (DSM)

Communicates with the TSM over a Secure Protocol and Data Model (SPDM) channel

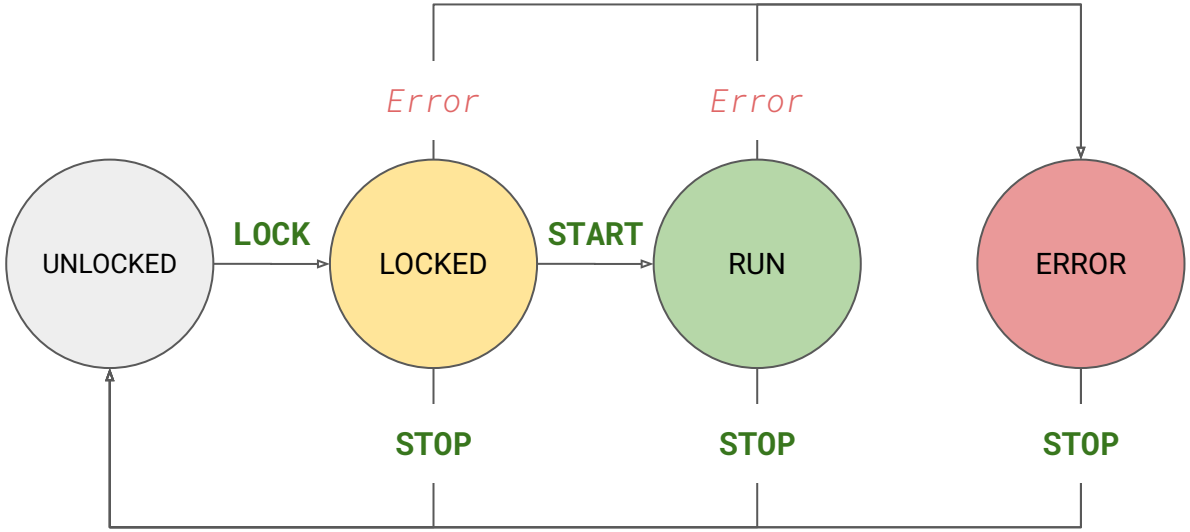
Provides TDIs memory isolation and protection

Enforces TDIs security states



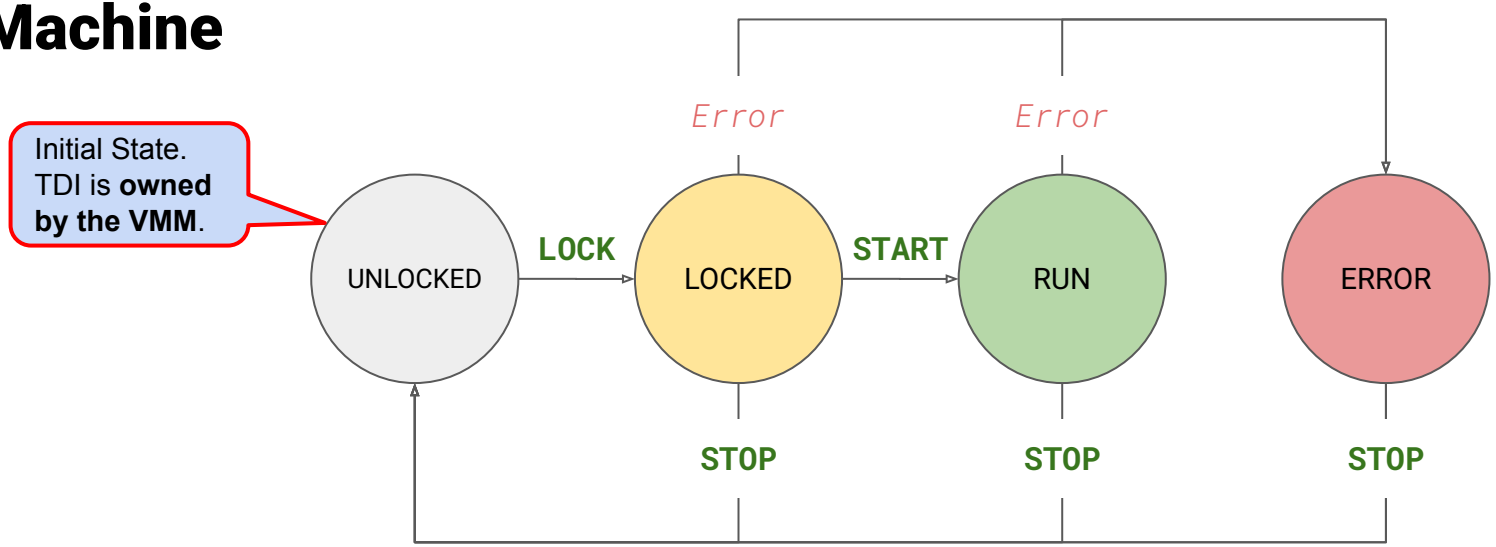
[1] Secure Protocol and Data Model

Machine



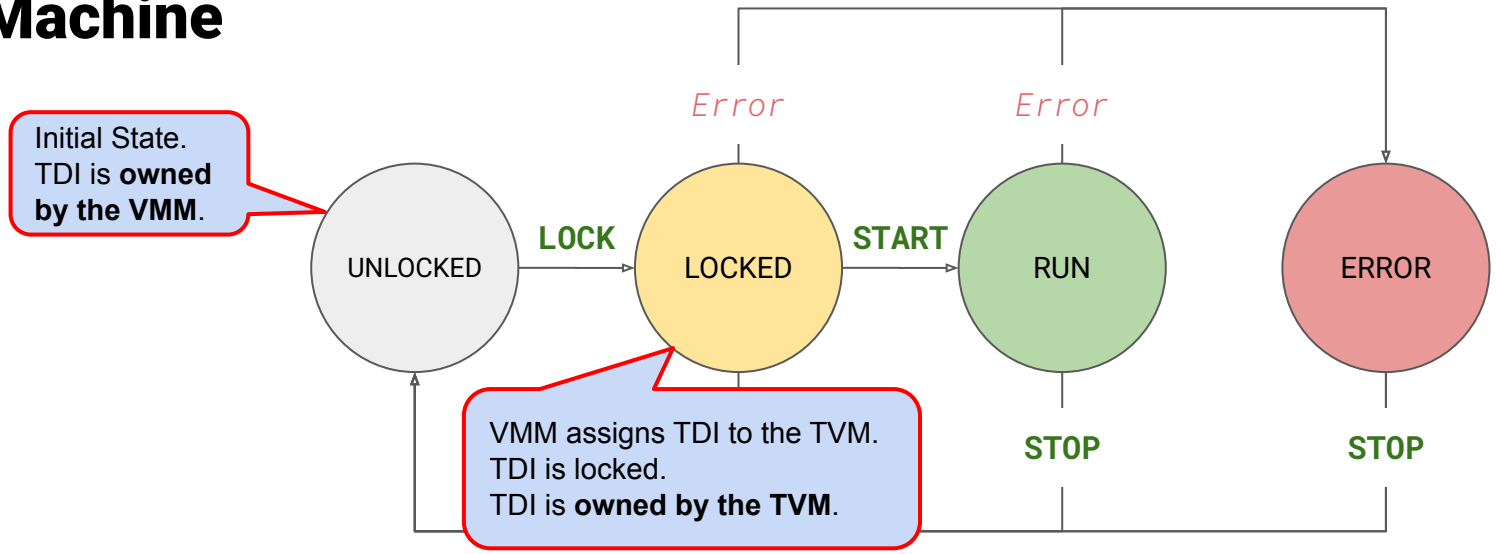
State	Device Config Changes?	DMA/MMIO	Device hold confidential data?	Usage
UNLOCKED	Yes	Yes - Not Confidential	No	Legacy
LOCKED	No	No	No	Verification by TVM
RUN	No	Yes - Confidential	**Yes**	TDI in use by guest
ERROR	No	No	**Yes**	Fatal Error Confidential data wiped

Machine



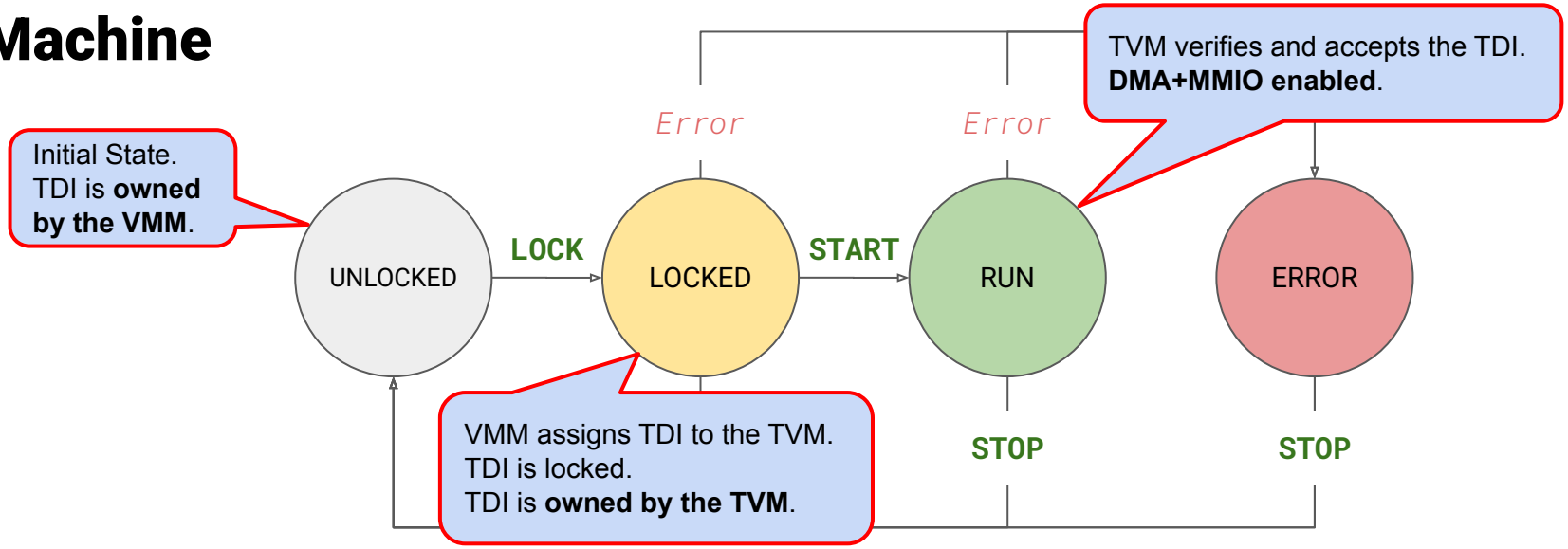
State	Device Config Changes?	DMA/MMIO	Device hold confidential data?	Usage
UNLOCKED	Yes	Yes - Not Confidential	No	Legacy
LOCKED	No	No	No	Verification by TVM
RUN	No	Yes - Confidential	**Yes**	TDI in use by guest
ERROR	No	No	**Yes**	Fatal Error Confidential data wiped

Machine



State	Device Config Changes?	DMA/MMIO	Device hold confidential data?	Usage
UNLOCKED	Yes	Yes - Not Confidential	No	Legacy
LOCKED	No	No	No	Verification by TVM
RUN	No	Yes - Confidential	**Yes**	TDI in use by guest
ERROR	No	No	**Yes**	Fatal Error Confidential data wiped

TDISP State Machine



State	Device Config Changes?	DMA/MMIO	Device hold confidential data?	Usage
UNLOCKED	Yes	Yes - Not Confidential	No	Legacy
LOCKED	No	No	No	Verification by TVM
RUN	No	Yes - Confidential	**Yes**	TDI in use by guest
ERROR	No	No	**Yes**	Fatal Error Confidential data wiped

TDISP

With TDISP, a device is exclusively bound to a security domain

TDISP removes the untrusted host security domain from the TCB

TSM and DSM (Device Security Manager) collaboration

Is the DSM trustworthy?

Device Attestation

The TVM must verify the device *before* accepting it into its TCB

- TSM must not enable DMA and MMIO before acceptation
- The LOCKED -> RUN TDISP transition

Verification can e.g. be done through attestation

SPDM provides cryptographic evidence for the device TCB

- TSM established a secured SPDM channel with the device

TVM fetches the evidence and calls into a Relying Party

- Communicates its decision to the TSM through the CoVE-IO ABI

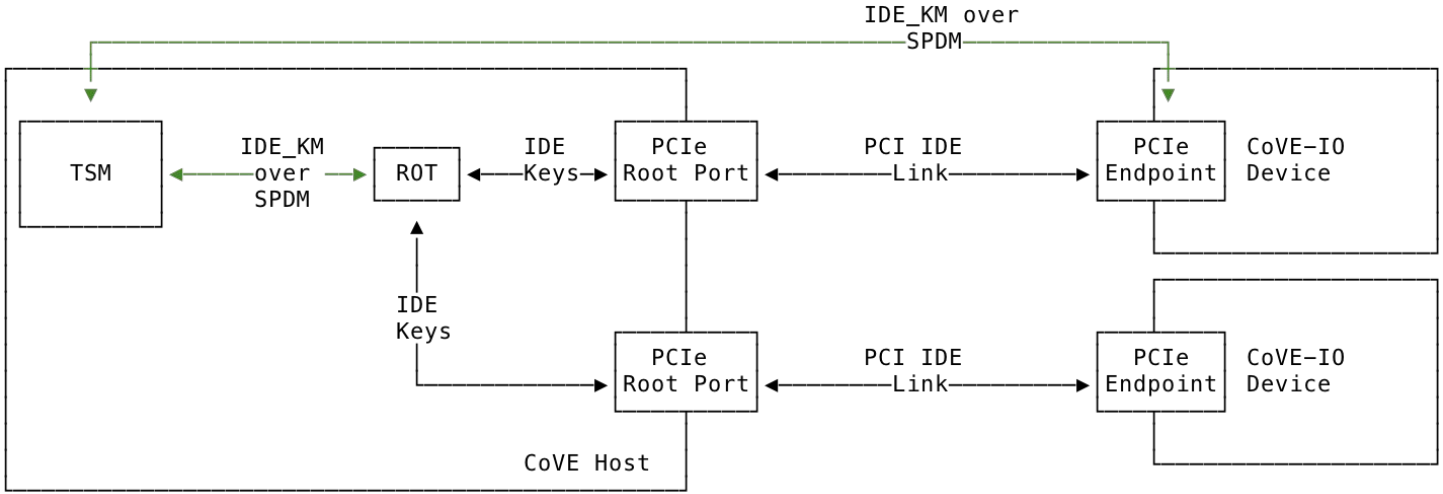
Physical Link Protection

For external PCIe devices

Protection from physical attacks on the PCIe link

PCIe TLPs encryption with PCIe IDE

The TSM generates and distributes the IDE keys



CoVE-IO

TDISP, SPDM, PCIe IDE are all architecture agnostic specifications

CoVE-IO defines the ABI to integrate TEE-IO into CoVE

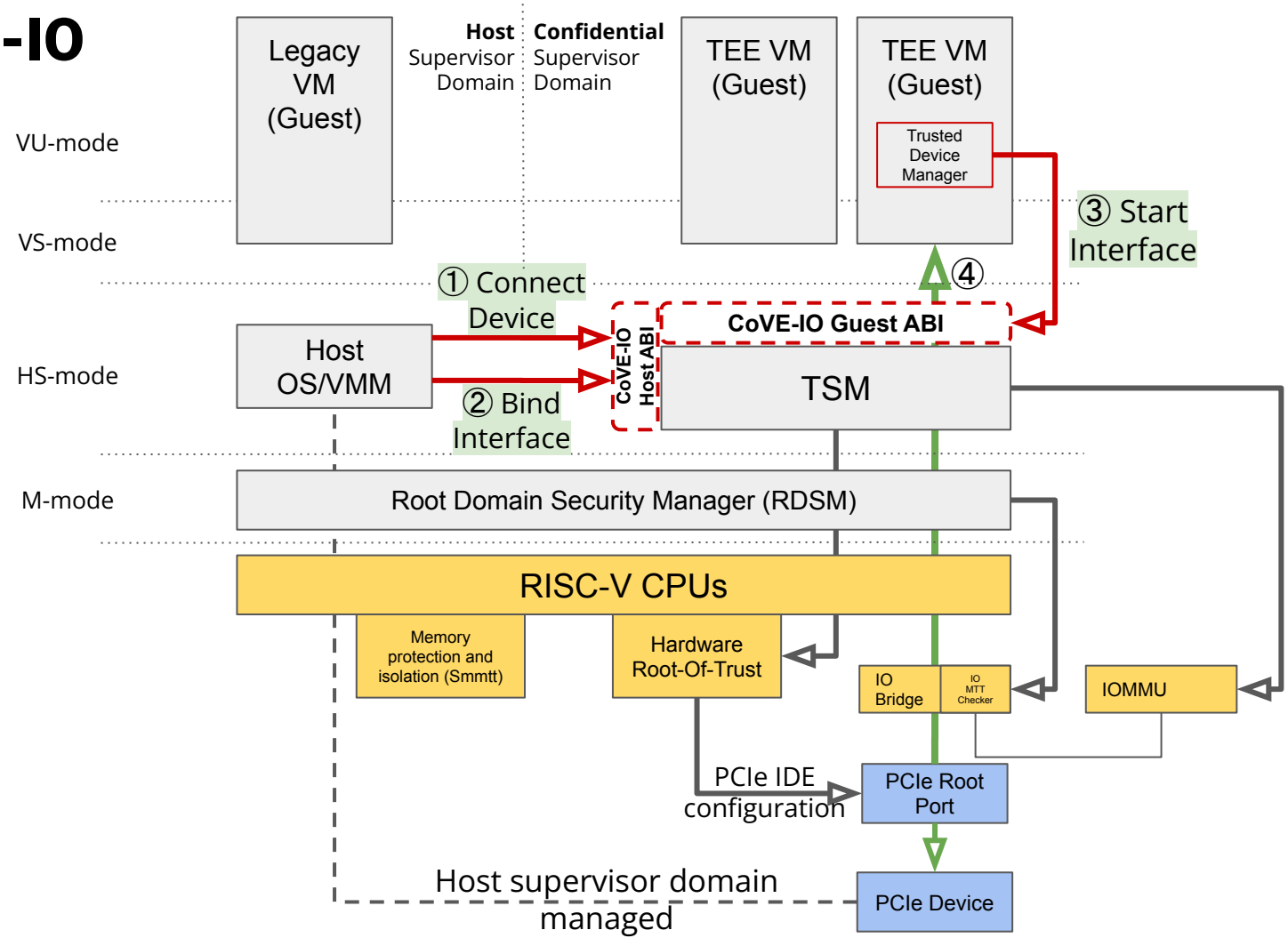
TEE-IO flows mapped to CoVE ABIs

`covh_connect_device` - Host requires TSM to establishes an SPDM connection with the device

`covh_bind_interface` - Host assigns a device to a TVM (TDISP UNLOCKED → LOCKED)

`covh_start_interface` - TVM accepts a device into its TCB (TDISP LOCKED → RUN)

CoVE-IO


Host Supervisor Domain
Confidential Supervisor Domain
VU-mode
VS-mode
HS-mode
M-mode
Legacy VM (Guest)
TEE VM (Guest)
TEE VM (Guest)
Trusted Device Manager
③ Start Interface
④
① Connect Device
② Bind Interface
Host OS/VMM
CoVE-IO Host ABI
CoVE-IO Guest ABI
TSM
Root Domain Security Manager (RDSM)
RISC-V CPUs
Memory protection and isolation (Smmtt)
Hardware Root-Of-Trust
IO Bridge
IO MTT Checker
IOMMU
PCIe IDE configuration
PCIe Root Port
Host supervisor domain managed
PCIe Device

RISC-V Confidential Computing

Lift and shift model on Trusted Virtual Machines (TVM)

Three specifications - CoVE, CoVE-IO and Smmtt

- Dependencies on HW Root-of-Trust, AIA, H-extension and IOMMU

Protection for TVM data confidentiality, data and code integrity

- The host and other devices are new adversaries

Provides TVM attestation rooted into the silicon creator

Working on ratifications on all three specifications

Working on emulated PoCs for CoVE and CoVE-IO

References

CoVE - https://github.com/riscv-non-isa/riscv-ap-tee

CoVE-IO - https://github.com/riscv-non-isa/riscv-ap-tee-io

Supervisor Domain Access Protection - https://github.com/riscv/riscv-smmtt/tree/main

TSM reference implementation - https://github.com/rivosinc/salus

RVI AP-TEE Technical Group - https://lists.riscv.org/g/tech-ap-tee

RVI AP-TEE-IO Technical Group - https://github.com/riscv-admin/ap-tee-io