Open Source CoCo with RISC-V

sameo@rivosinc.com

FOSDEM 2023

A Free and Open ISA

RISC ISA

Free and Open

Specifications are released under the CC BY 4.0

Volume 1 - Unprivileged Specs (2019)

Volume 2 - Privileged Specs (2021)

Controlled by a non-profit organization - RISC-V International

RISC-V ISA

Simple

No μ-arch dependencies

300 pages specs

Modular

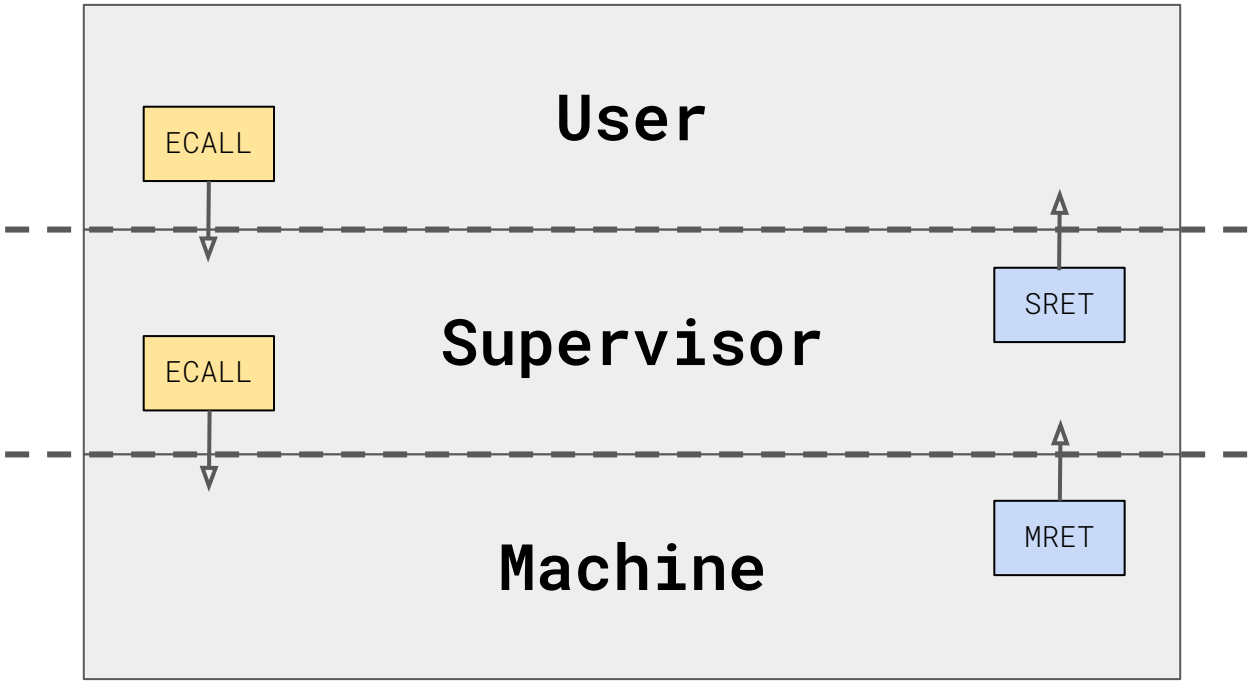
Same specs for everyone

Stable

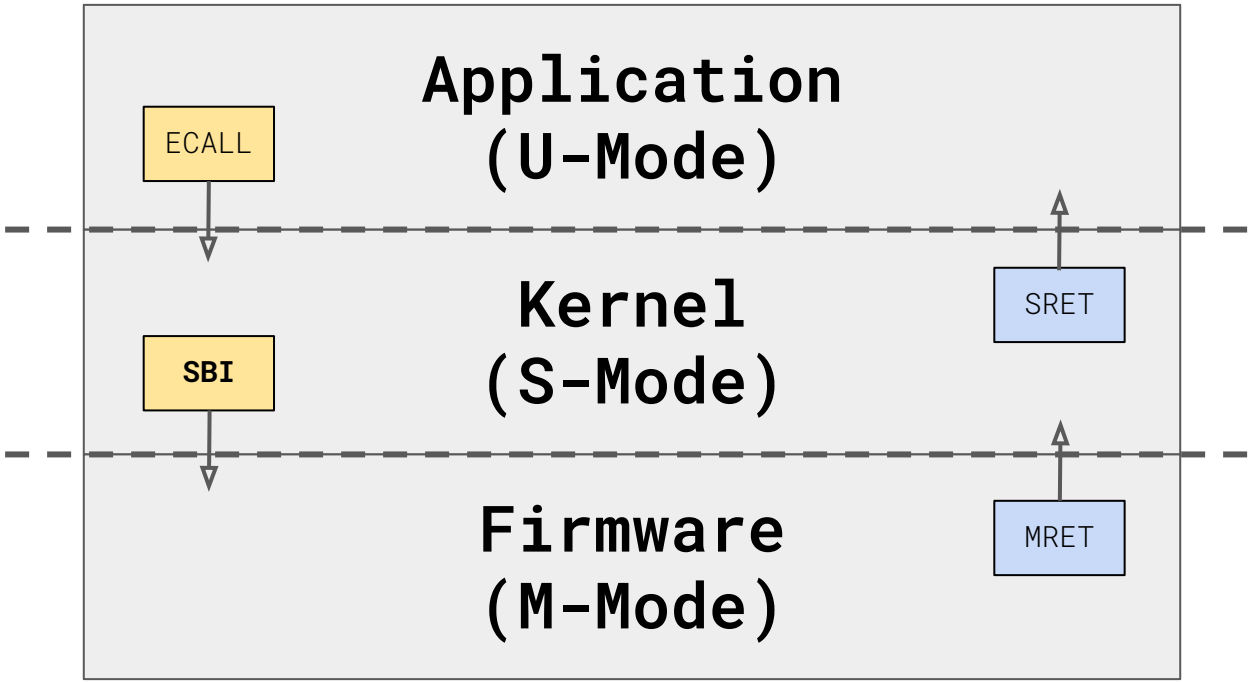
Base ISA and standard extensions are frozen

Extensions are optional

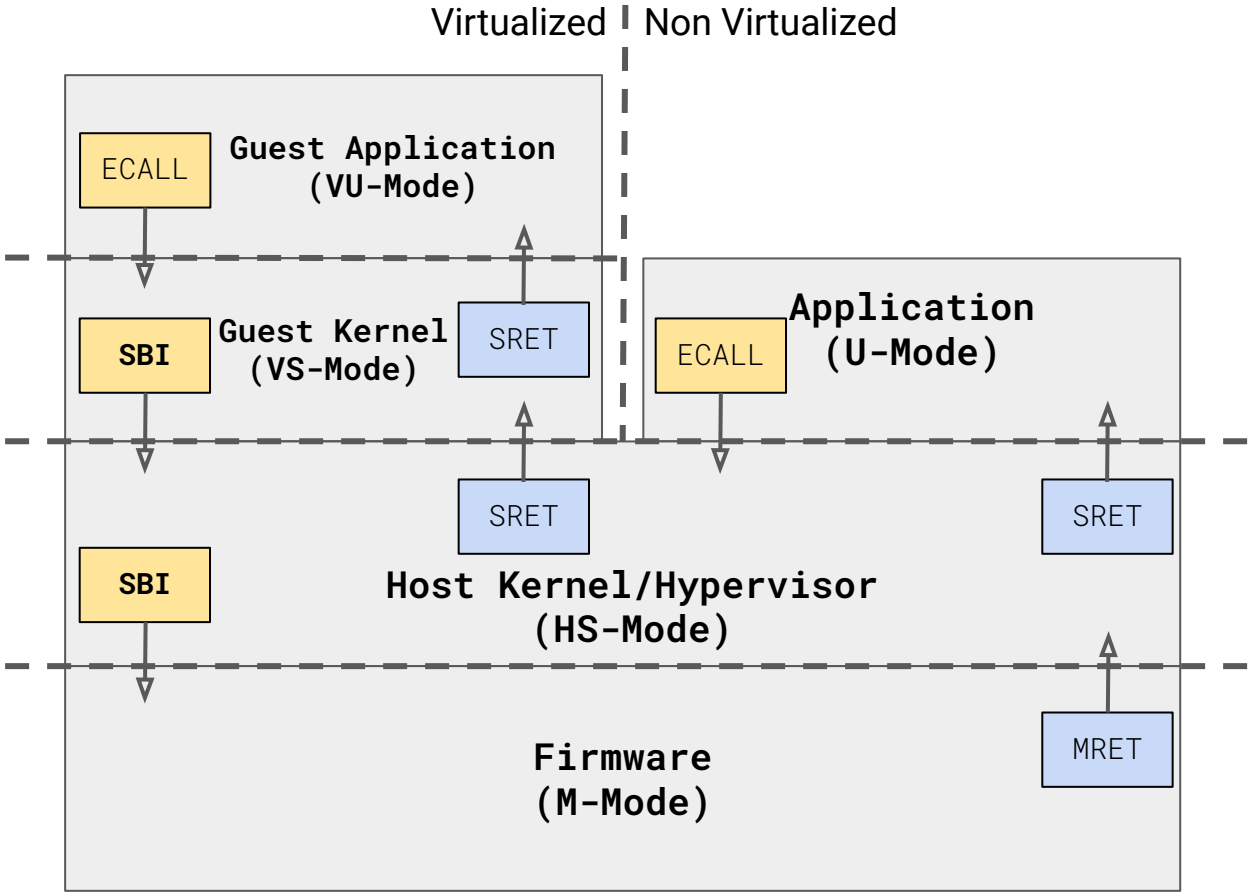
imm[31:12]			rd	0110111	LUI
imm[31:12]			rd	0010111	AUIPC
imm[20|10:1|11|19:12]			rd	1101111	JAL
imm[11:0]	rs1	000	rd	1100111	JALR
imm[12|10:5]	rs2	rs1	000	imm[4:1|11]	1100011	BEQ
imm[12|10:5]	rs2	rs1	001	imm[4:1|11]	1100011	BNE
imm[12|10:5]	rs2	rs1	100	imm[4:1|11]	1100011	BLT
imm[12|10:5]	rs2	rs1	101	imm[4:1|11]	1100011	BGE
imm[12|10:5]	rs2	rs1	110	imm[4:1|11]	1100011	BLTU
imm[12|10:5]	rs2	rs1	111	imm[4:1|11]	1100011	BGEU
imm[11:0]		rs1	000	rd	0000011	LB
imm[11:0]		rs1	001	rd	0000011	LH
imm[11:0]		rs1	010	rd	0000011	LW
imm[11:0]		rs1	100	rd	0000011	LBU
imm[11:0]		rs1	101	rd	0000011	LHU
imm[11:5]	rs2	rs1	000	imm[4:0]	0100011	SB
imm[11:5]	rs2	rs1	001	imm[4:0]	0100011	SH
imm[11:5]	rs2	rs1	010	imm[4:0]	0100011	SW
imm[11:0]		rs1	000	rd	0010011	ADDI
imm[11:0]		rs1	010	rd	0010011	SLTI
imm[11:0]		rs1	011	rd	0010011	SLTIU
imm[11:0]		rs1	100	rd	0010011	XORI
imm[11:0]		rs1	110	rd	0010011	ORI
imm[11:0]		rs1	111	rd	0010011	ANDI
0000000	shamt	rs1	001	rd	0010011	SLLI
0000000	shamt	rs1	101	rd	0010011	SRLI
0100000	shamt	rs1	101	rd	0010011	SRAI
0000000	rs2	rs1	000	rd	0110011	ADD
0100000	rs2	rs1	000	rd	0110011	SUB
0000000	rs2	rs1	001	rd	0110011	SLL
0000000	rs2	rs1	010	rd	0110011	SLT
0000000	rs2	rs1	011	rd	0110011	SLTU
0000000	rs2	rs1	100	rd	0110011	XOR
0000000	rs2	rs1	101	rd	0110011	SRL
0100000	rs2	rs1	101	rd	0110011	SRA
0000000	rs2	rs1	110	rd	0110011	OR
0000000	rs2	rs1	111	rd	0110011	AND
fm | pred | succ		rs1	000	rd	0001111	FENCE
000000000000		00000	000	00000	1110011	ECALL
000000000001		00000	000	00000	1110011	EBREAK



RISC-V Privilege Modes



RISC-V Privilege Modes



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

RISC-V Confidential Computing

AP-TEE RISC-V Technical Group

AP-TEE: Application Processor Trusted Execution Environment

Reference confidential computing architecture for RISC-V

Non-ISA specification - identifies ISA gaps (e.g. confidentiality PMA)

Defines a new class of Trusted Execution Environment

Trusted Virtual Machine (TVM) - H extension 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 or Intel TDX

Architecture Components

Per Hart AP-TEE bit

TEE Security Manager (TSM)

TSM-driver

Hardware Root of Trust

Memory Tracking Table

TCB

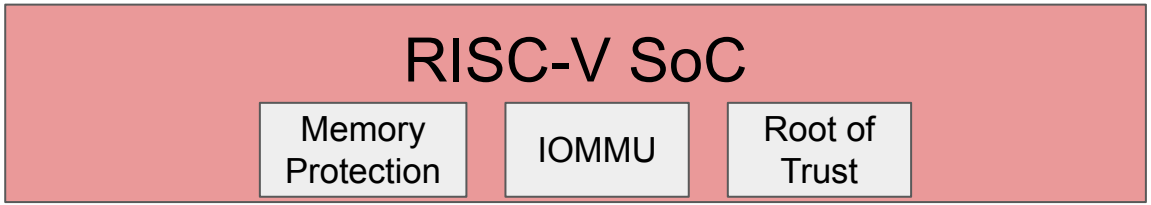
!TCB

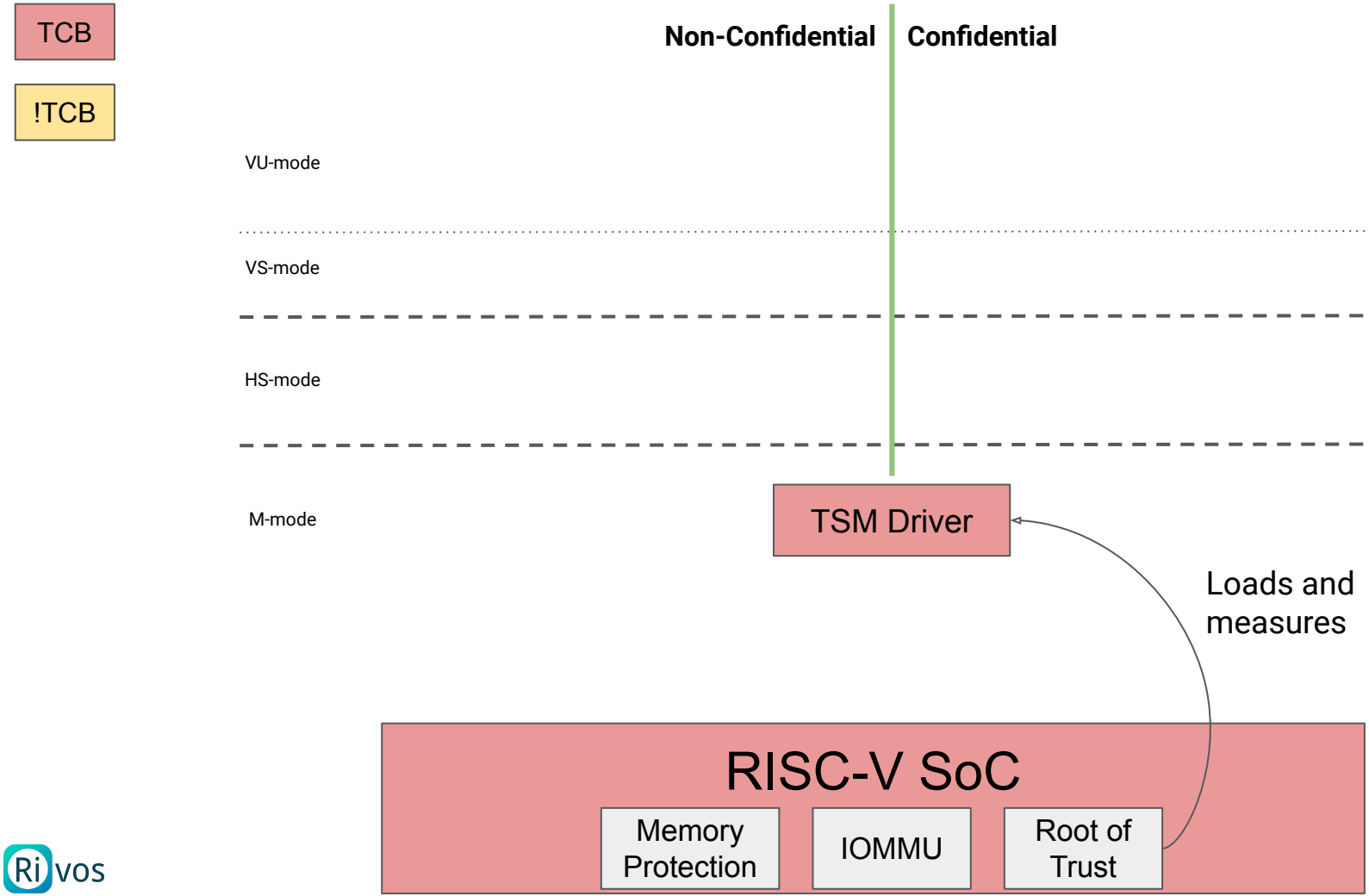
VU-mode

VS-mode

HS-mode

M-mode




TCB
!TCB
Non-Confidential
Confidential
VU-mode
VS-mode
HS-mode
M-mode
TSM Driver
Loads and measures
RISC-V SoC
Memory Protection
IOMMU
Root of Trust

TSM-Driver

M-mode firmware component, part of the TCB

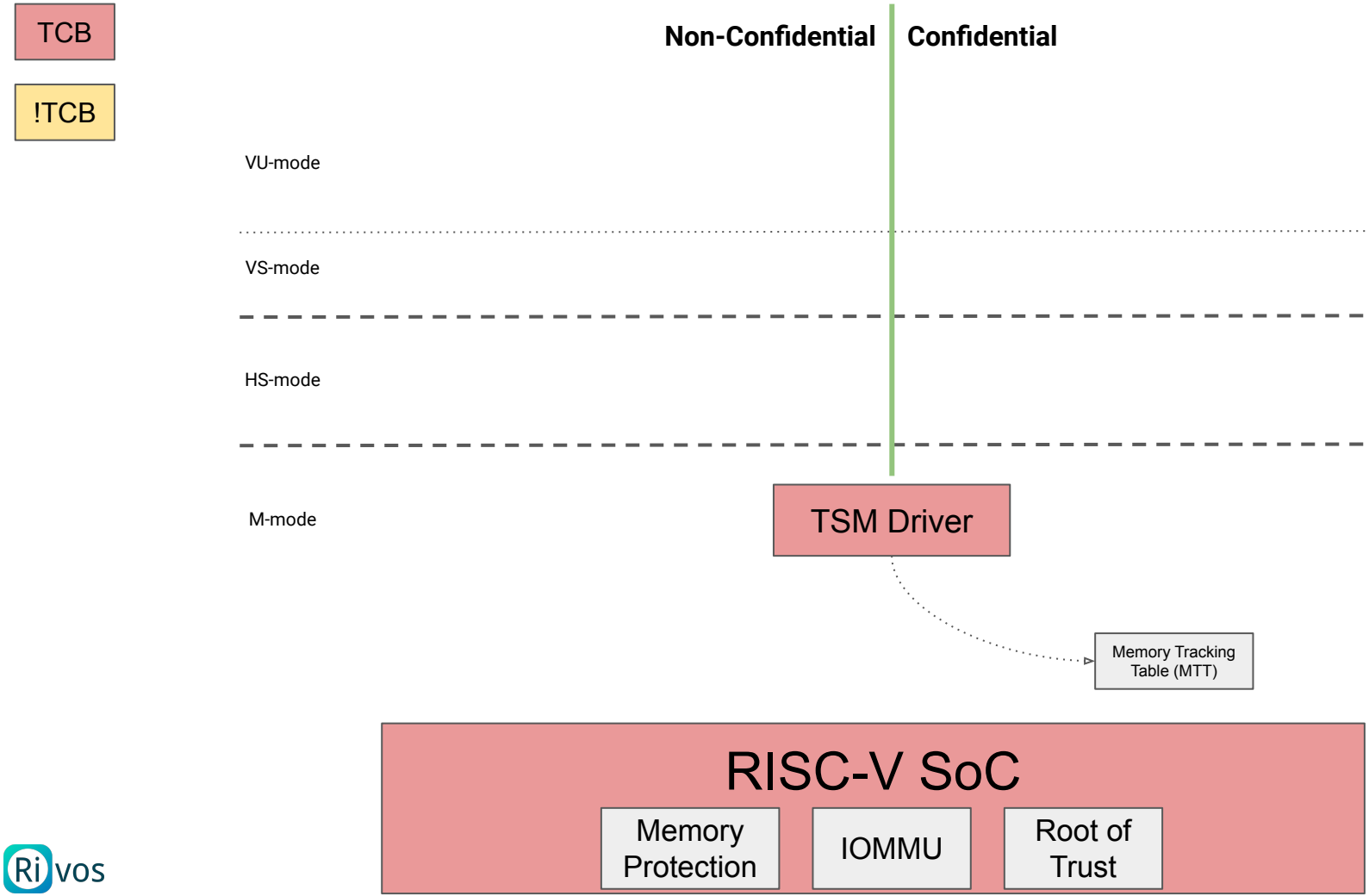
Confidential world switcher

Non Confidential → Confidential

1. Hypervisor does a TEECALL SBI call
2. TSM-driver traps
3. TSM-driver toggles the hart AP-TEE bit
4. TSM-driver MRET into the TSM

Confidential → Non Confidential

1. TSM does a TEERET SBI call
2. TSM-driver traps
3. TSM-driver toggles the hart AP-TEE bit
4. TSM-driver MRET into the hypervisor


TCB
!TCB
Non-Confidential
Confidential
VU-mode
VS-mode
HS-mode
M-mode
TSM Driver
Memory Tracking Table (MTT)
RISC-V SoC
Memory Protection
IOMMU
Root of Trust

Memory Tracking Table (MTT)

Confidential memory attribute (RISC-V PMA) page tracker

Defines if a page is in confidential memory or not

MTT(Physical address) → Confidential or !Confidential address

Memory Integrity

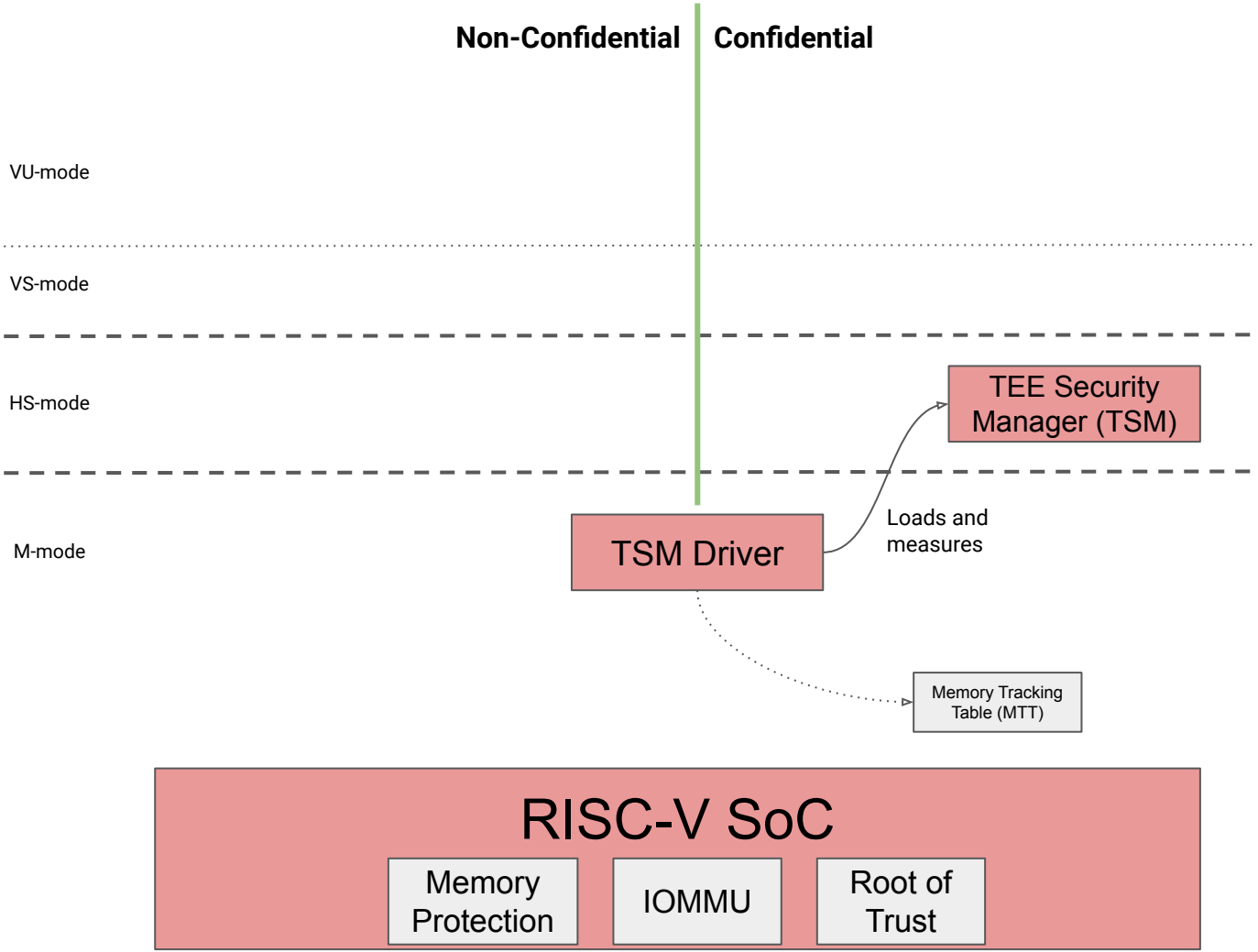
Accessing a !Confidential page from !AP-TEE generates a fault

Memory Protection

Memory encryption key selection can be built from (PA, PMA)

TCB

!TCB


Non-Confidential
Confidential
VU-mode
VS-mode
HS-mode
M-mode
TEE Security Manager (TSM)
Loads and measures
TSM Driver
Memory Tracking Table (MTT)
RISC-V SoC
Memory Protection
IOMMU
Root of Trust

TEE Security Manager (TSM)

A trusted intermediary between the host VMM and the TVMs

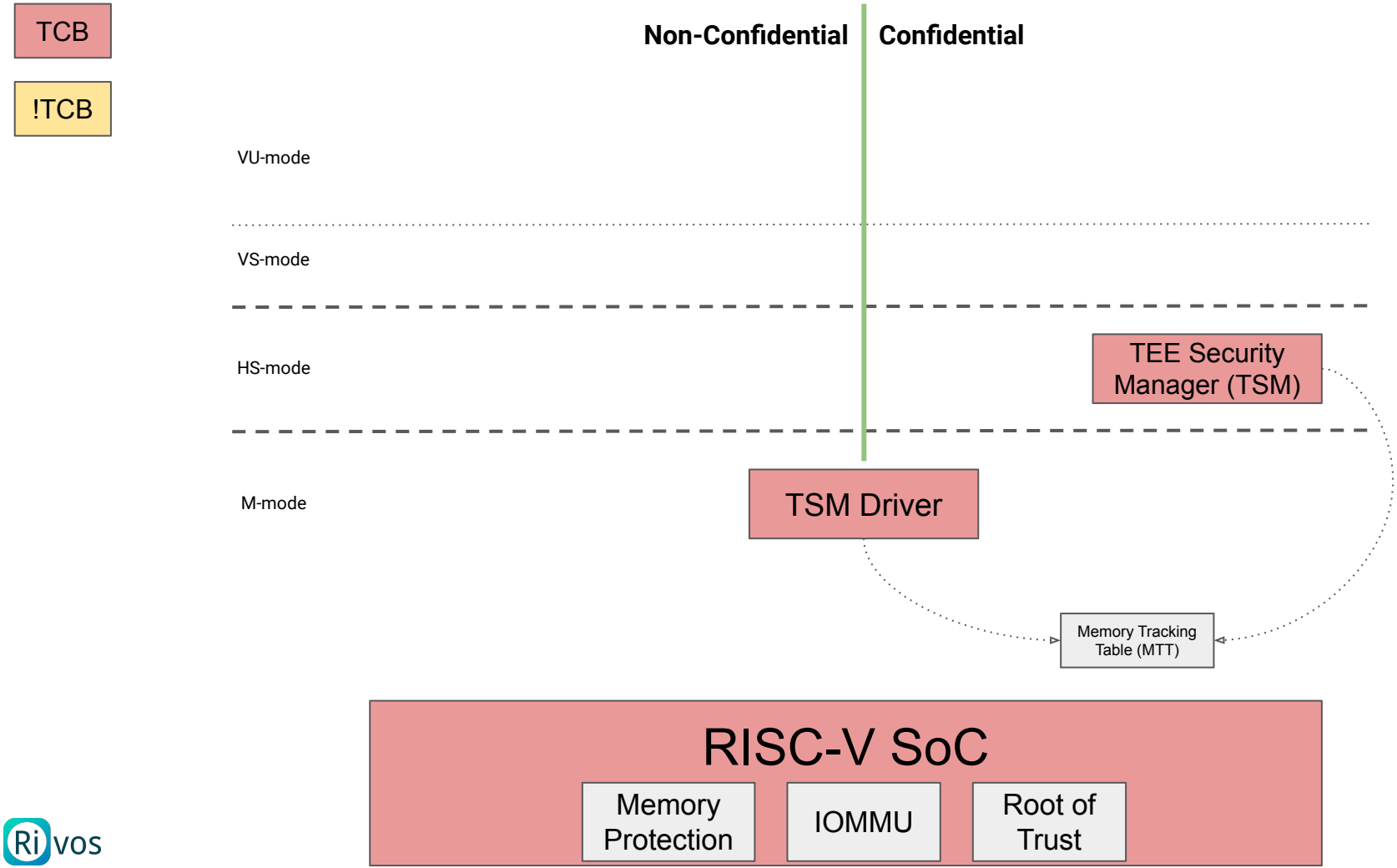
Manages all TVM second-stage (G-stage) page tables

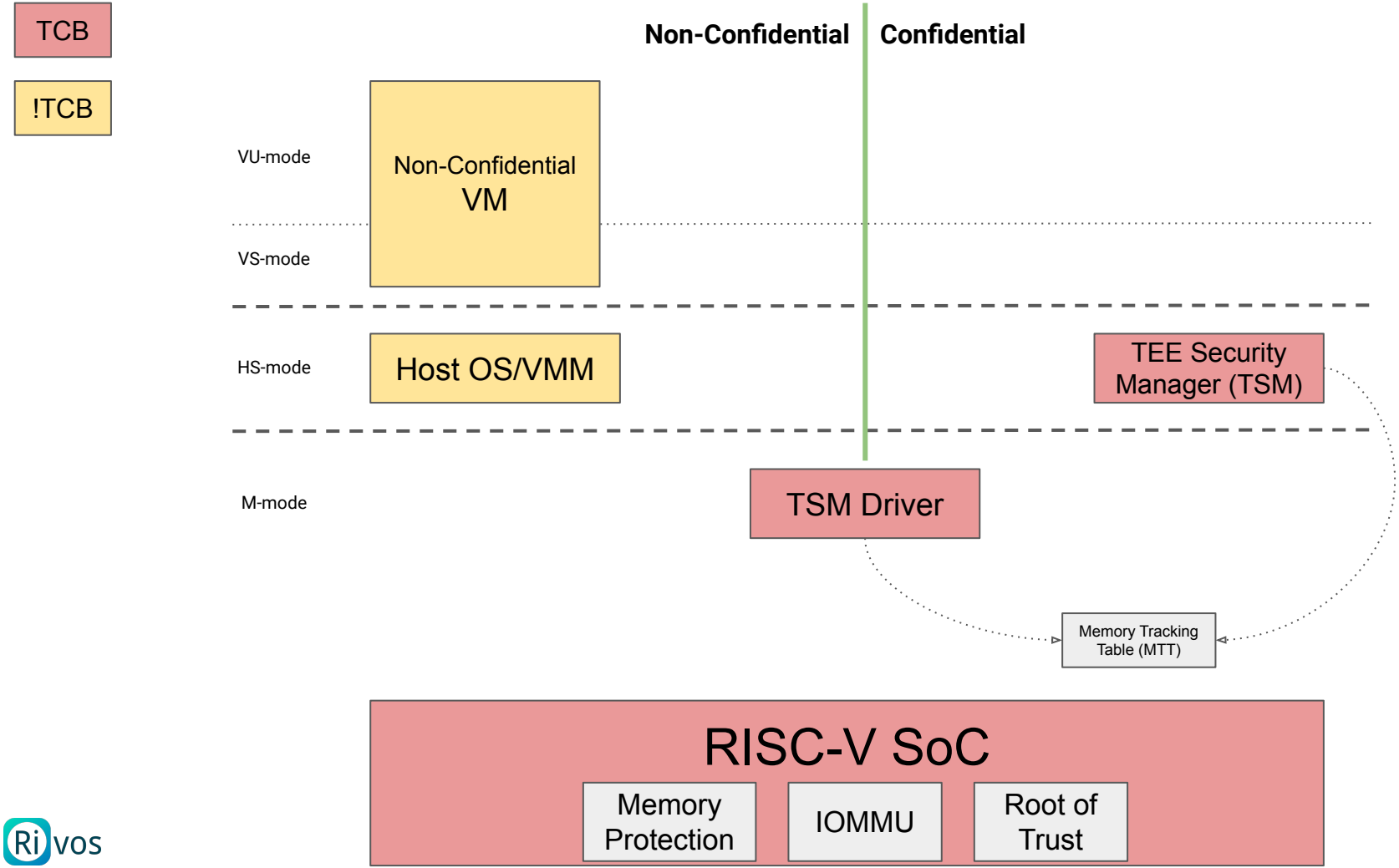
- TVM G-stage page tables must be in confidential memory

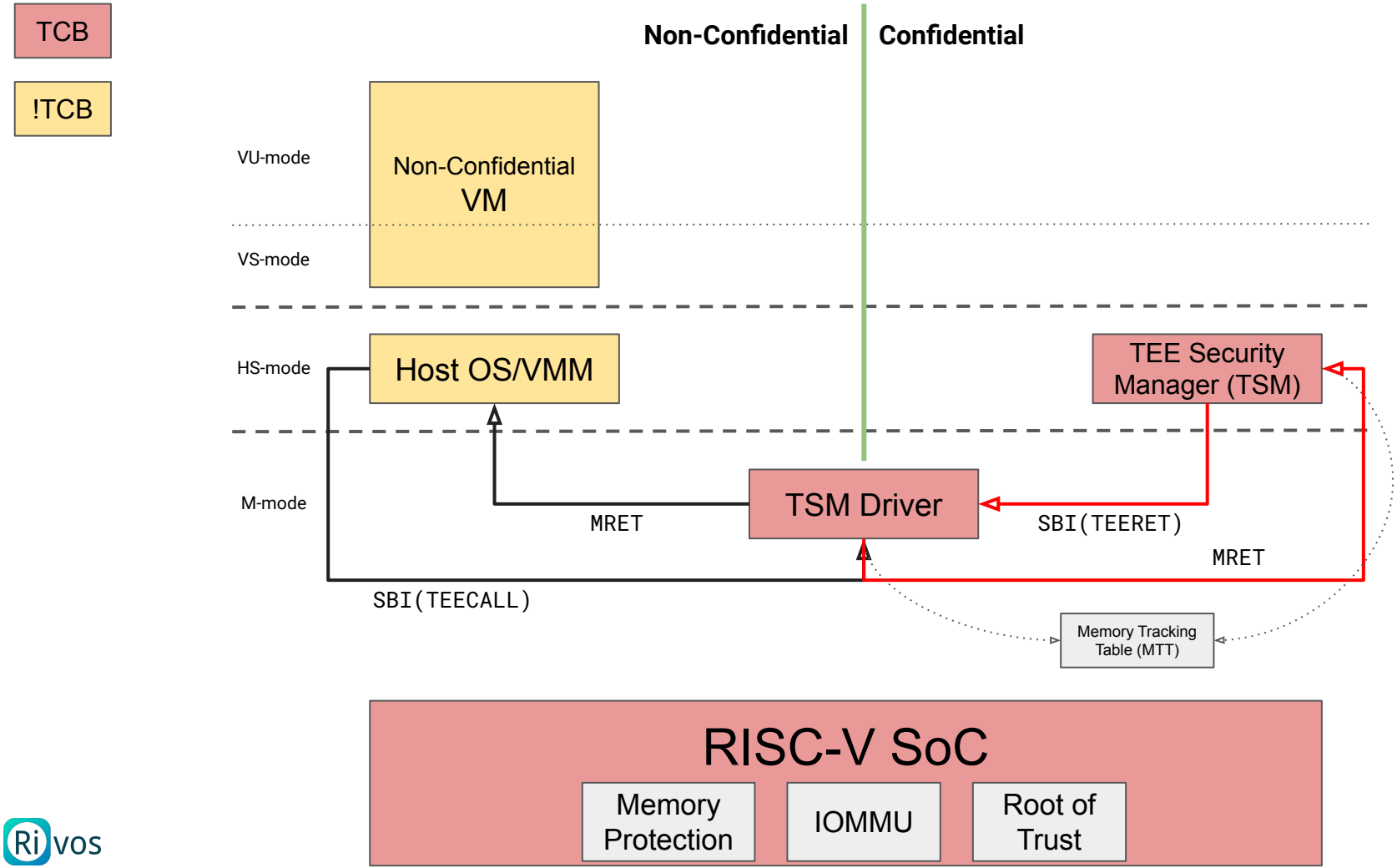
Passive component

- Implements CC security services called by the host VMM
- Enforces CC security attributes for the TVMs
- Does not schedule TVMs. Does not handle interrupts.

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


TCB
!TCB
Non-Confidential
Confidential
VU-mode
VS-mode
HS-mode
M-mode
TEE Security Manager (TSM)
TSM Driver
Memory Tracking Table (MTT)
RISC-V SoC
Memory Protection
IOMMU
Root of Trust


TCB
!TCB
Non-Confidential
Confidential
VU-mode
VS-mode
Non-Confidential VM
HS-mode
Host OS/VMM
TEE Security Manager (TSM)
M-mode
TSM Driver
Memory Tracking Table (MTT)
RISC-V SoC
Memory Protection
IOMMU
Root of Trust


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

TEE Host ABI (TH-ABI)

Binary interface for the host VMM to request CC services from the TSM

Proxied through the TSM-Driver

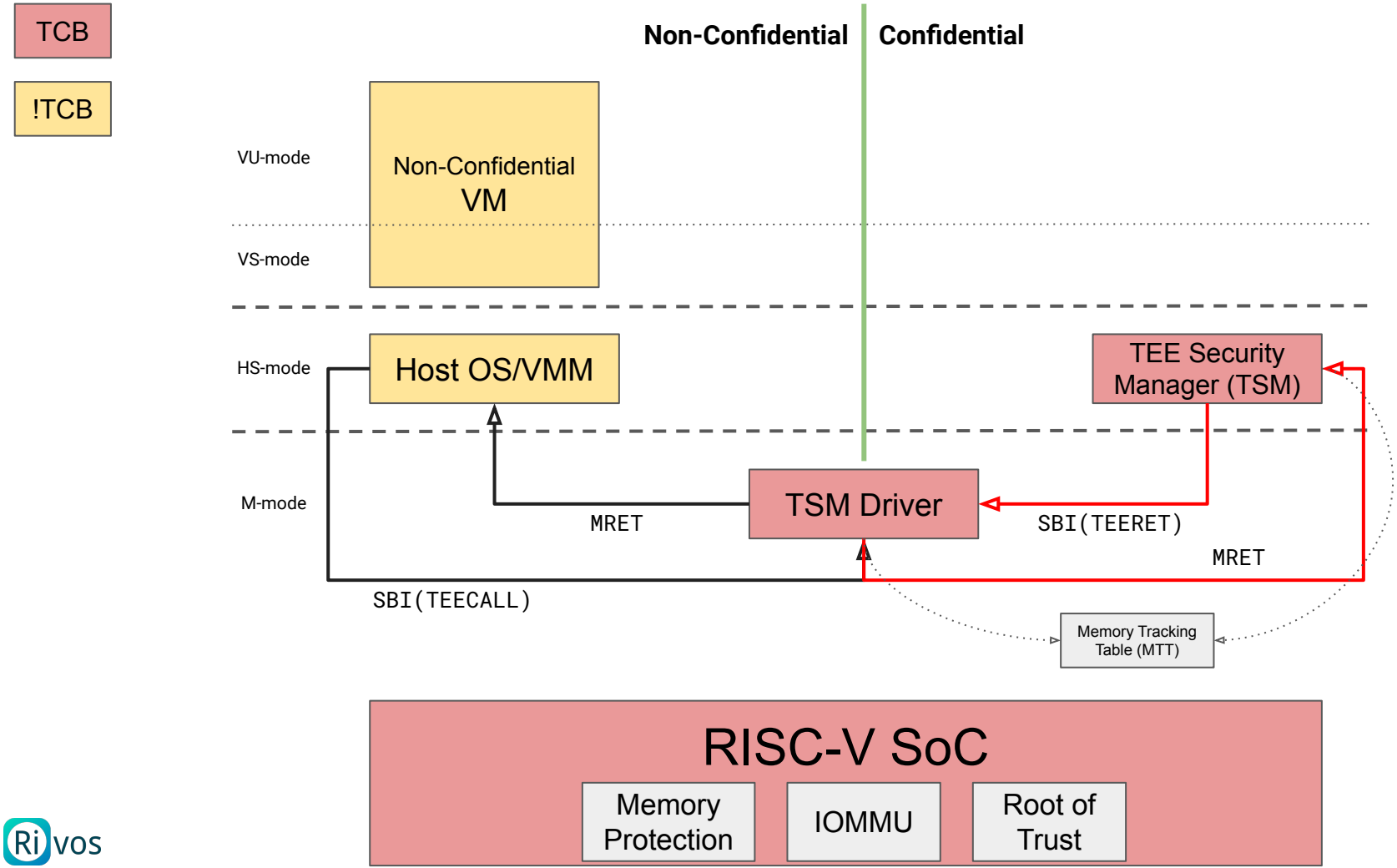
TSM-driver traps the host SBI call and MRET into the TSM

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

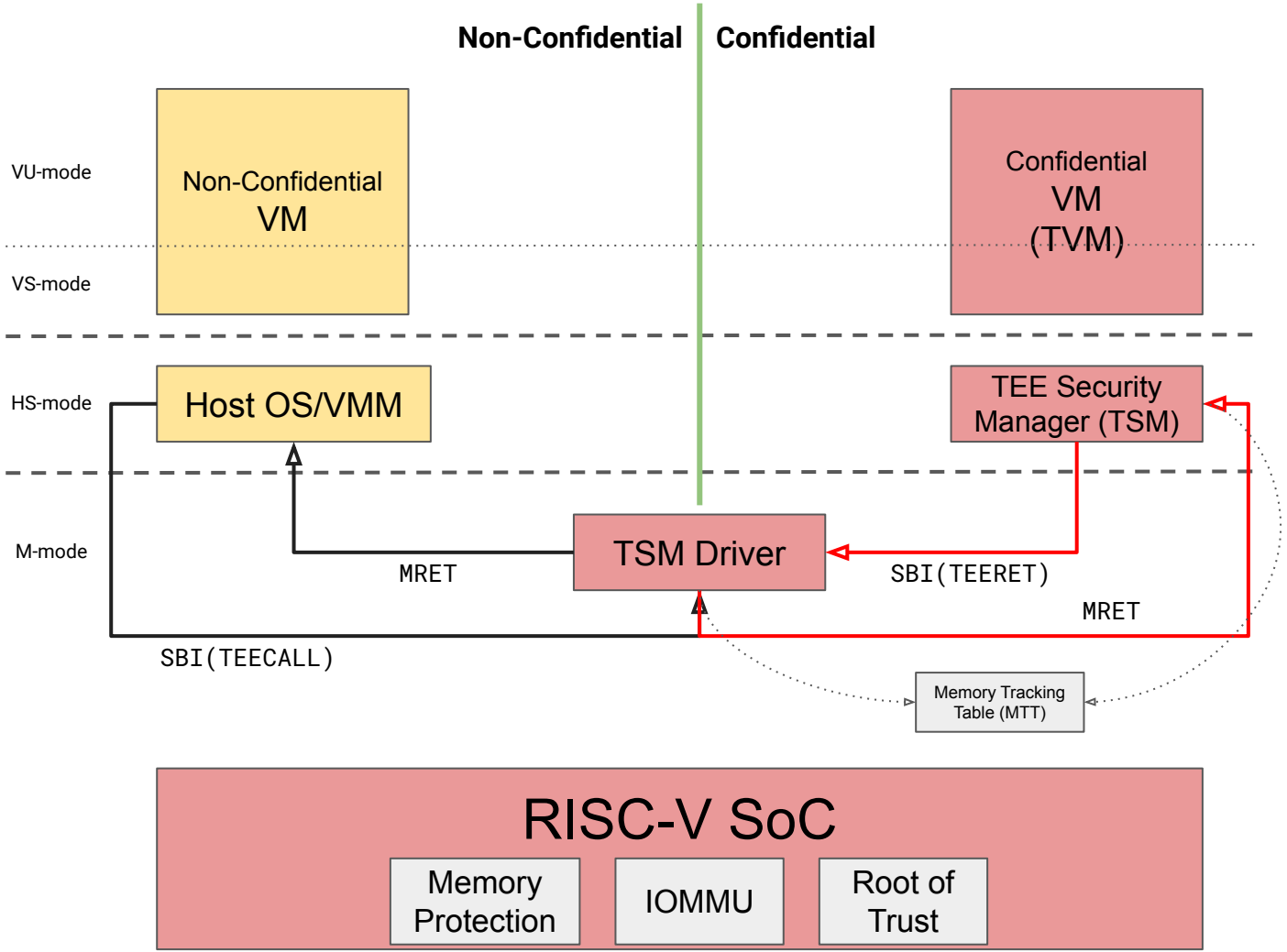
TEE Host ABI (TH-ABI) - TVM Creation

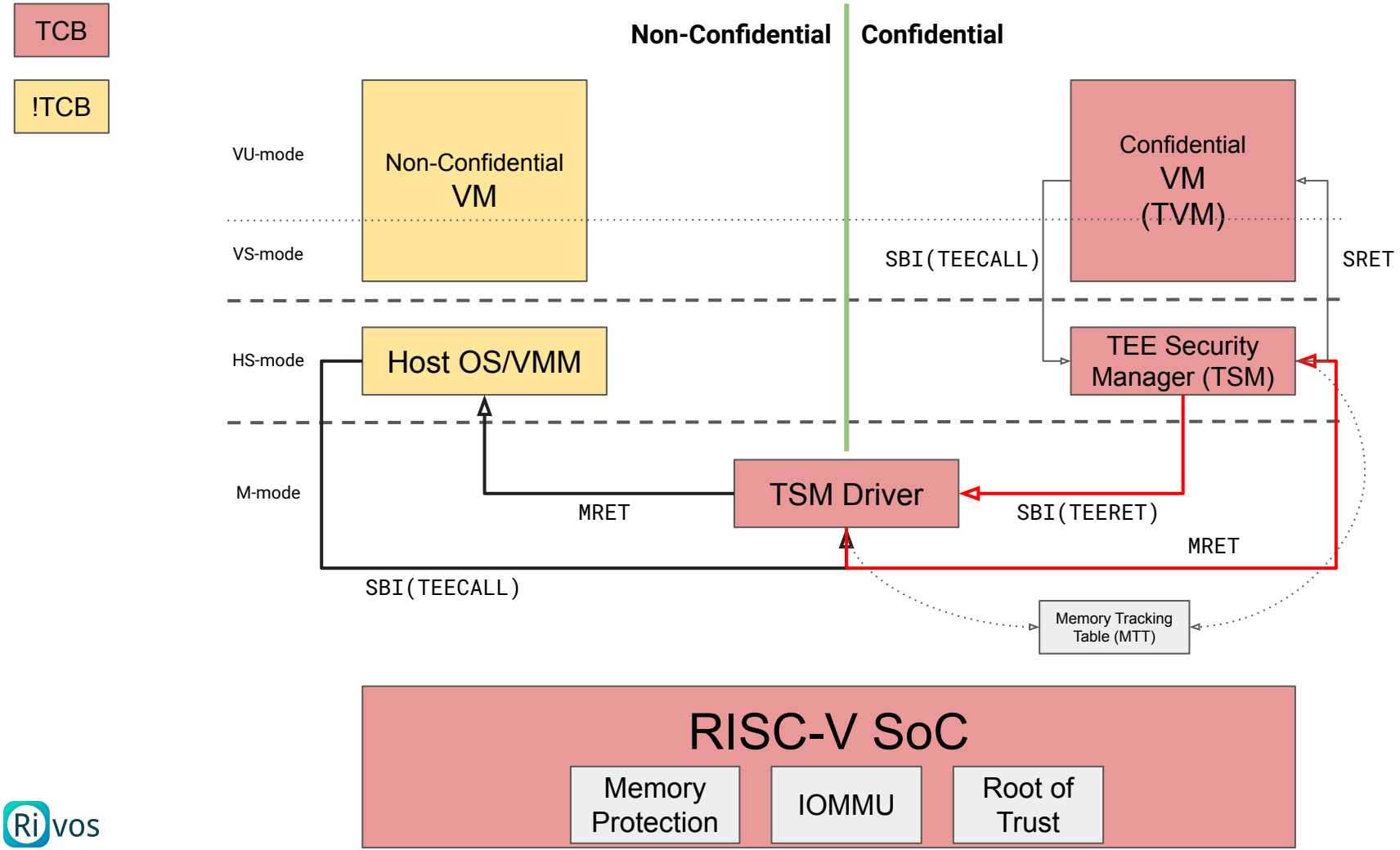
1. Create a TVM context
   a. sbi_tee_host_create_tvm( )
2. Donate confidential memory for the TVM 2nd stage page tables
   a. sbi_tee_host_add_tvm_page_table_pages()
3. Reserve TVM confidential memory regions
   a. sbi_tee_host_add_tvm_memory_region()
4. Add measured and zero pages to the TVM
   a. sbi_tee_host_add_tvm_measured_pages(), sbi_tee_host_add_tvm_zero_pages()
5. Create the TVM vCPUs
   a. sbi_tee_host_create_tvm_vcpu()
6. Finalize the TVM
   a. sbi_tee_host_finalize_tvm()
7. Run a TVM
   a. sbi_tee_host_run_tvm_vcpu()


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

TCB

!TCB


Non-Confidential
Confidential
VU-mode
VS-mode
Non-Confidential VM
Confidential VM (TVM)
HS-mode
Host OS/VMM
TEE Security Manager (TSM)
M-mode
TSM Driver
MRET
SBI(TEERET)
MRET
SBI(TEECALL)
Memory Tracking Table (MTT)
RISC-V SoC
Memory Protection
IOMMU
Root of Trust


TCB
!TCB
Non-Confidential
Confidential
VU-mode
VS-mode
HS-mode
M-mode
Non-Confidential VM
Confidential VM (TVM)
Host OS/VMM
TEE Security Manager (TSM)
TSM Driver
SBI(TEECALL)
SRET
MRET
SBI(TEERET)
MRET
SBI(TEECALL)
Memory Tracking Table (MTT)
RISC-V SoC
Memory Protection
IOMMU
Root of Trust

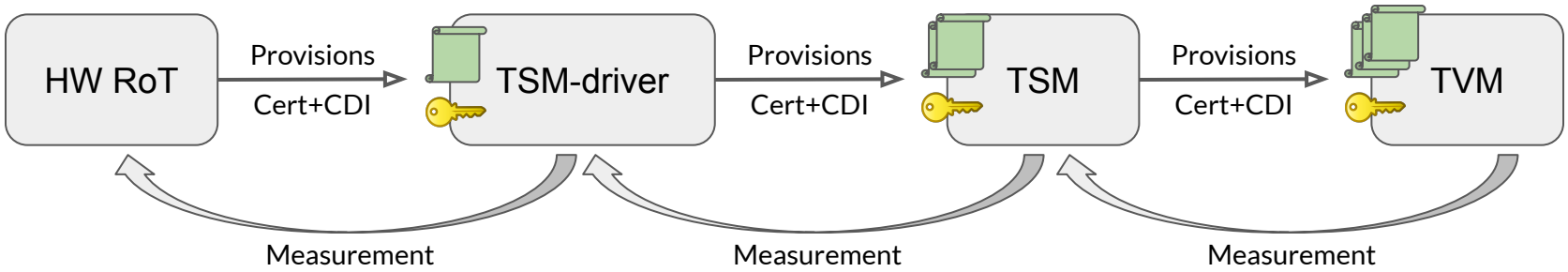
Attestation

Layered Architecture based on TCG DICE

Each layer loads, measures and certifies the next one

The TVM gets a certificate from the TSM

- Rooted back to the hardware root of trust
- Contains the TCB measurements
- Attestation Evidence



IO

Paravirtualized a.k.a. virtio

Shared memory with the host VMM

swiotlb for buffer bouncing between C and !C

Host may be trusted (virtio devices in VMM)

Guest hardening

Direct Assignment

Extend the TCB with an external, unknown, untrusted device

Device authentication and attestation (SPDM+TDISP)

PCI link protection (PCI IDE)

IOMMU collaboration

Complex…

Thanks!

AP-TEE Spec: https://github.com/riscv-non-isa/riscv-ap-tee
TSM Reference Implementation:
https://github.com/rivosinc/salus