LINUX SECURITY SUMMIT EUROPE

Enabling new security frontiers: Deep dive into Confidential Computing on RISC-V

Ravi Sahita
Rivos Inc.

Atish Patra
Rivos Inc.

Rivos

Outline

- RISC-V and Security Goals
- Introduction to Confidential Computing
- RISC-V **Co**nfidential **V**M **E**xtension (aka RISC-V CoVE)

RISC-V

- *RISC-V* is an open, royalty-free standard Instruction Set Architecture (ISA)
    - Base Privileged, Un-privileged ISA and extensions

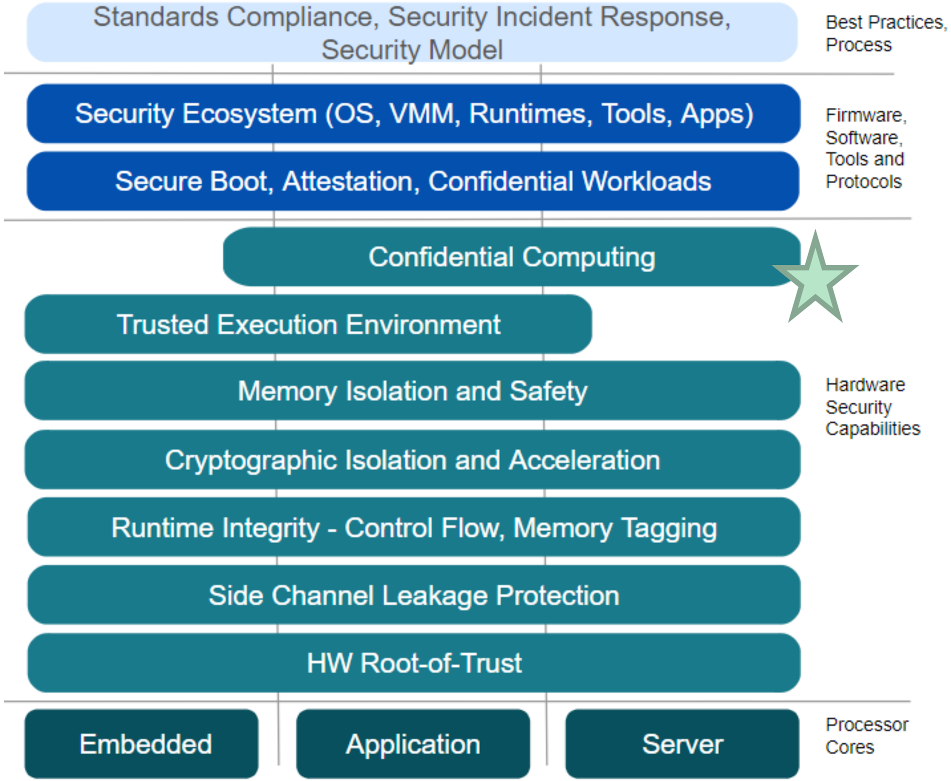
- *RISC-V International* is the global non-profit home of the ISA, non-ISA, related specifications, and stakeholder community
    - Members contribute and collaborate to define RISC-V open specifications via area-focused horizontal committees, special interest groups, and task groups.

https://riscv.org/

RISC-V and Security

RISC-V's open and clean-slate design presents a unique opportunity to ingrain security for the next generation of compute infrastructure.

- Foundational Security & Cryptography
- Software Hardening
- Trusted & Confidential Computing
- Security Model & Lifecycle



Standards Compliance, Security Incident Response, Security Model

Best Practices, Process

Security Ecosystem (OS, VMM, Runtimes, Tools, Apps)

Secure Boot, Attestation, Confidential Workloads

Firmware, Software, Tools and Protocols

Confidential Computing

Trusted Execution Environment

Memory Isolation and Safety

Cryptographic Isolation and Acceleration

Runtime Integrity - Control Flow, Memory Tagging

Side Channel Leakage Protection

HW Root-of-Trust

Hardware Security Capabilities

Embedded

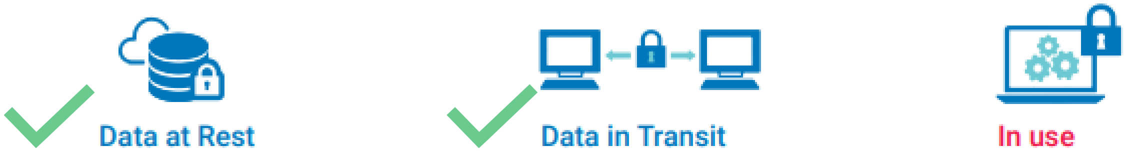
Application

Server

Processor Cores

Introduction to Confidential Computing

Confidential Computing protects *data-in-use* by performing computation in a *hardware-based*, *attested* Trusted Execution Environment (TEE).



A critical aspect of TEEs are the requirements of hardware-based isolation, and attestation (cryptographic-verification) of the Trusted Computing Base (TCB).

CONFIDENTIAL COMPUTING CONSORTIUM

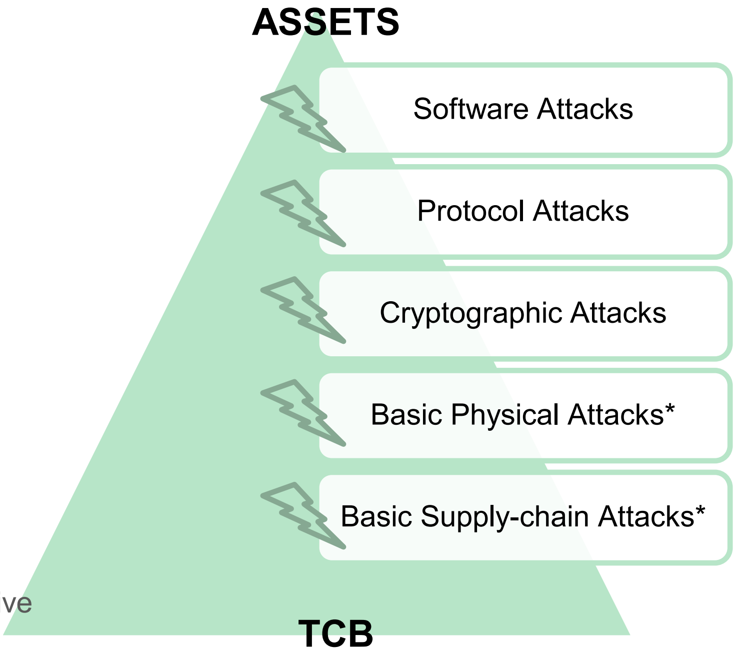
https://confidentialcomputing.io/

An example use case – Confidential AI

- Cryptographically-verifiable protection of data and models throughout the AI lifecycle, *especially when such data is in use.*
  - Enables confidentiality for: model weights, proprietary training data, inference queries
  - Enables data-controls during multi-party, federated training and inferencing
  - Mitigates insider threats to ensure model safety and training isolation
  - Mitigates sensitive data breaches and enables compliance with data privacy regulations

- Requires TEE on general purpose CPUs and GPUs
  - Must provide Data Confidentiality, Data Integrity, Code Integrity and Attestation.

Confidential Computing Threat Model (50K foot summary)


ASSETS

Software Attacks

Protocol Attacks

Cryptographic Attacks

Basic Physical Attacks*

Basic Supply-chain Attacks*

TCB

RISC-V Security Model (non-normative)

RISC-V Security Model Task Group

Version 0.3, 05/2024: This document is in development. Assume everything can change. See http://riscv.org/spec-state for details.

Adversaries

* Local/Remote Software
- Un-Priv/ Priv.

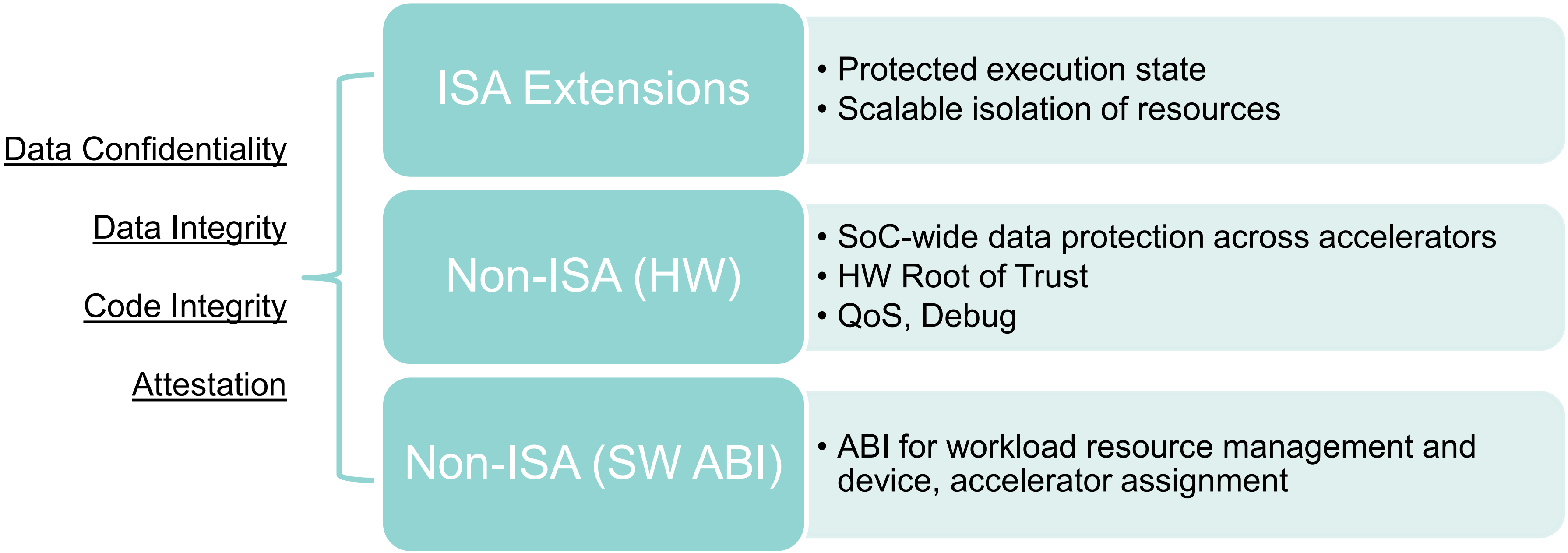
* Local/Remote Hardware
-Non-invasive/ Invasive

Out of scope:

* Advanced Physical Attacks
* Advanced Supply-chain Attacks
* Denial of Service (DoS) Attacks

https://github.com/riscv-non-isa/riscv-security-model

Confidential Computing on RISC-V



Data Confidentiality

Data Integrity

Code Integrity

Attestation

ISA Extensions

- Protected execution state
- Scalable isolation of resources

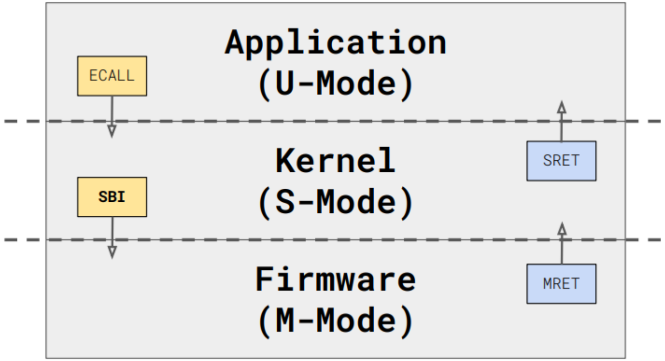
Non-ISA (HW)

- SoC-wide data protection across accelerators
- HW Root of Trust
- QoS, Debug

Non-ISA (SW ABI)

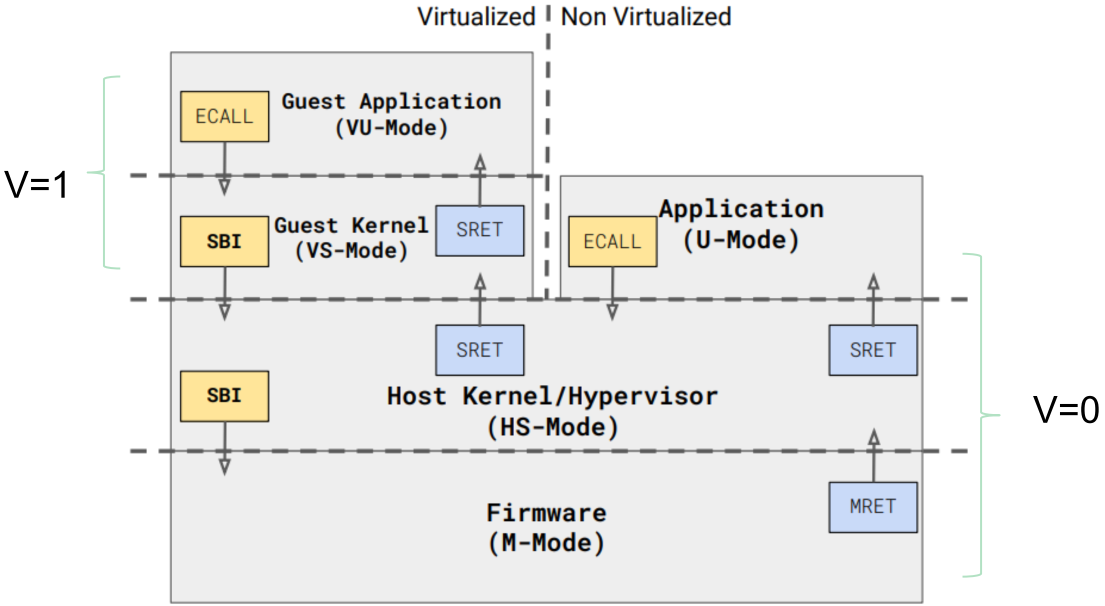
- ABI for workload resource management and device, accelerator assignment

Brief background of RISC-V Priv ISA and Hypervisor Ext.



https://github.com/riscv/riscv-isa-manual

Brief background of RISC-V Priv ISA and Hypervisor Ext.

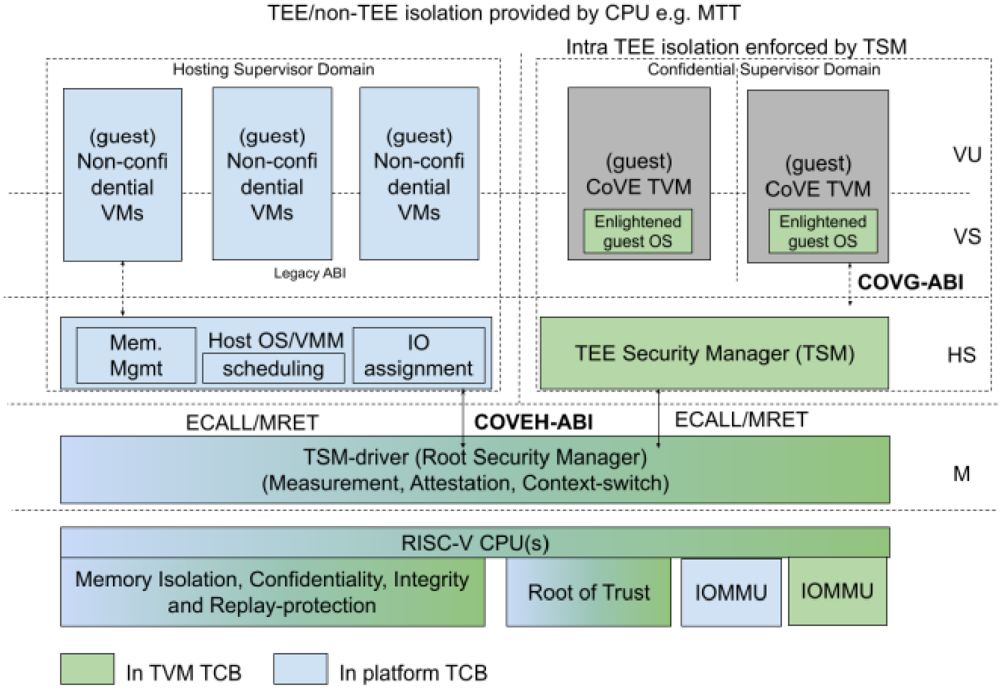


https://github.com/riscv/riscv-isa-manual

Confidential **V**M **E**xtension (CoVE) Reference Arch.

New Security Objectives - Bootstrap one or more **TEE Security Manager(s) (TSM)** domain to host new type of VMs - **TEE VMs (TVMs)**

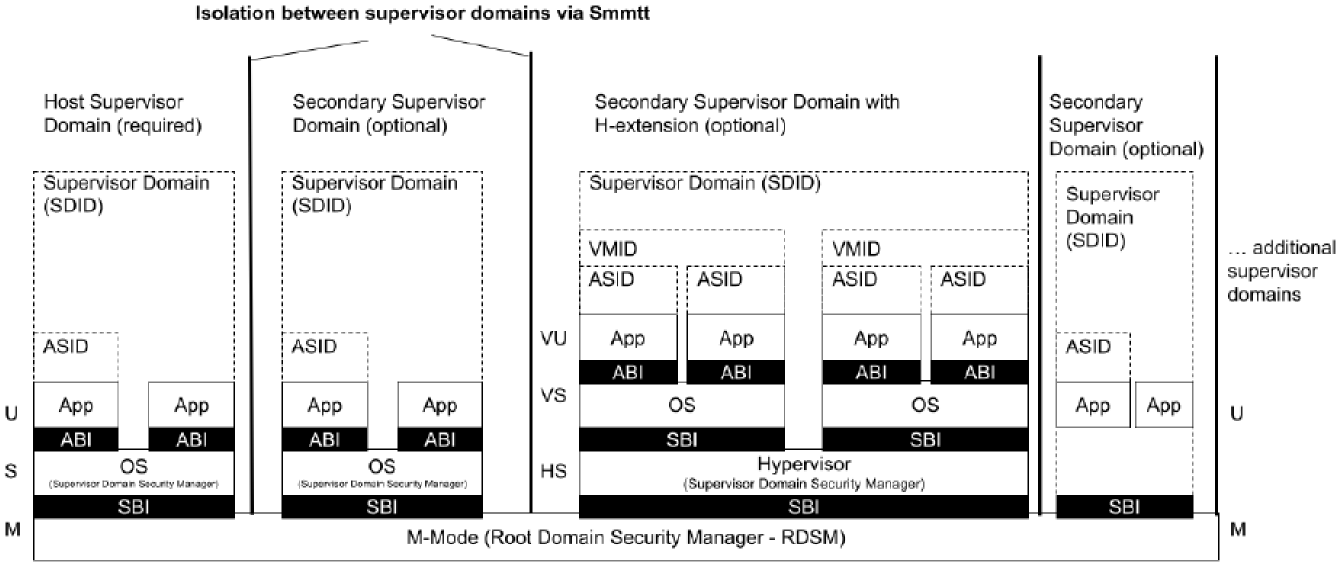
- Isolation of OS/VMM-domain-accessible memory from the TSM-domain-accessible memory, while allowing host OS/VMM to manage resource assignment
- Enable a **Root Domain Security Manager** (RDSM) to mutually isolate TSMs
- Enable the **TEE Security Manager** (TSM) to mutually isolate TVMs
- Provide **HW-rooted attestation** of the TCB (including HW and SW such as RDSM, TSM).
- Leverage platform HW to protect data-in-use across the SoC:
  - Protect data that leaves the SoC/package boundaries e.g. via memory encryption of DRAM, PCIe/CXL
  - Restrict debug, Perf., QoS monitoring
  - Enable device function binding to TVM



TEE/non-TEE isolation provided by CPU e.g. MTT
Intra TEE isolation enforced by TSM
Hosting Supervisor Domain
Confidential Supervisor Domain
(guest) Non-confi dential VMs
(guest) Non-confi dential VMs
(guest) Non-confi dential VMs
(guest) CoVE TVM
Enlightened guest OS
(guest) CoVE TVM
Enlightened guest OS
VU
VS
Legacy ABI
COVG-ABI
Mem. Mgmt
Host OS/VMM scheduling
IO assignment
TEE Security Manager (TSM)
HS
ECALL/MRET
COVEH-ABI
ECALL/MRET
TSM-driver (Root Security Manager)
(Measurement, Attestation, Context-switch)
M
RISC-V CPU(s)
Memory Isolation, Confidentiality, Integrity and Replay-protection
Root of Trust
IOMMU
IOMMU
In TVM TCB
In platform TCB

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

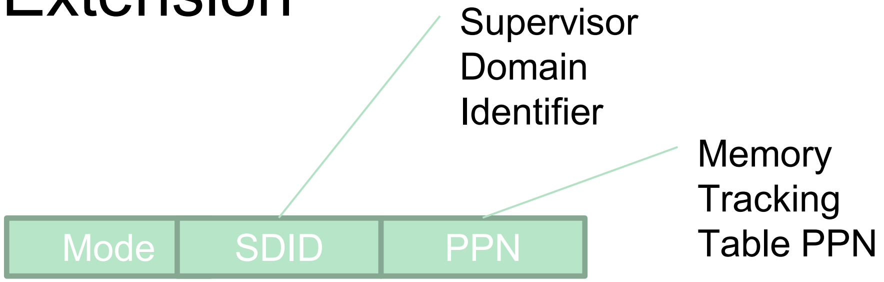
Supervisor Domains – priv. ISA extension


Isolation between supervisor domains via Smmtt
Host Supervisor Domain (required)
Secondary Supervisor Domain (optional)
Secondary Supervisor Domain with H-extension (optional)
Secondary Supervisor Domain (optional)
Supervisor Domain (SDID)
Supervisor Domain (SDID)
Supervisor Domain (SDID)
Supervisor Domain (SDID)
VMID
VMID
ASID
ASID
ASID
ASID
ASID
ASID
ASID
App
App
App
App
App
App
App
App
App
App
ABI
ABI
ABI
ABI
ABI
ABI
ABI
ABI
OS (Supervisor Domain Security Manager)
OS (Supervisor Domain Security Manager)
OS
OS
Hypervisor (Supervisor Domain Security Manager)
SBI
SBI
SBI
SBI
SBI
SBI
M-Mode (Root Domain Security Manager - RDSM)
... additional supervisor domains
U
S
M
VU
VS
HS
U
M

https://github.com/riscv/riscv-smmtt

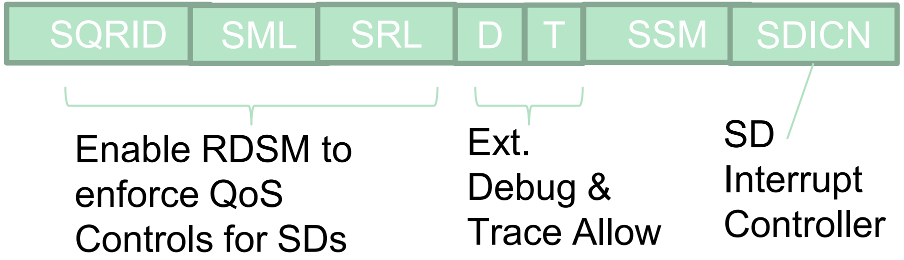
Supervisor Domain “Smsdid” Extension

- CSRs to manage “Supervisor Domain Identifiers” aka SDID assignment to harts
  - Local identifiers to manage access-control properties on harts (extends VMID, ASID)
  - Physical memory permissions programmed via a **Memory Tracking Table**
- M-mode SD fence instructions
  - MFENCE.SPA
  - MINVAL.SPA

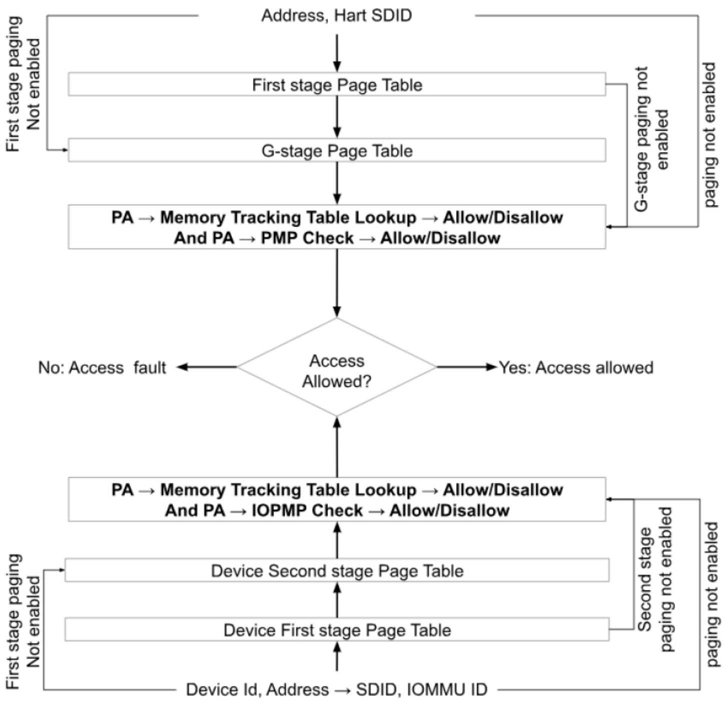


Memory Tracking Table Pointer register

M-mode SD config register

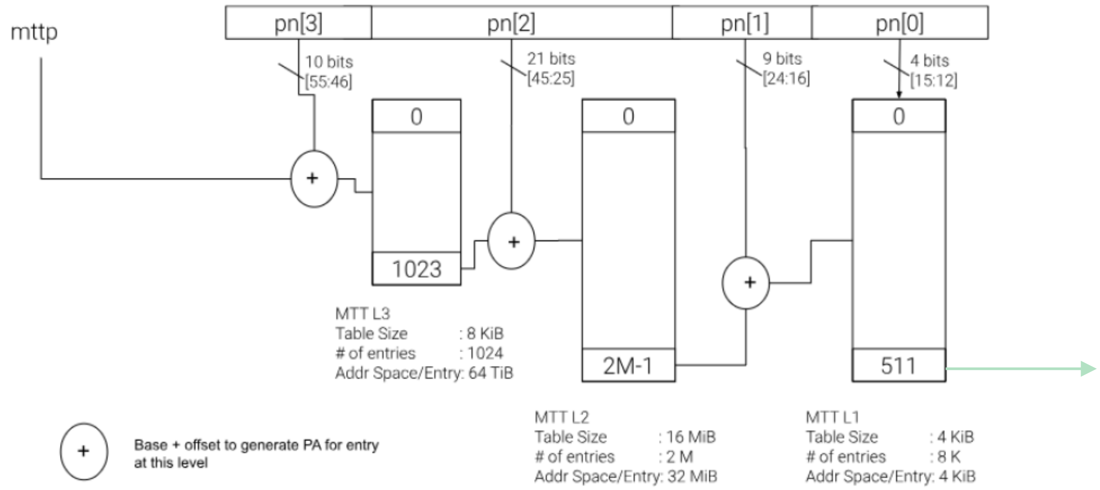


Supervisor Domains “Smmtt” Extension


Address, Hart SDID
First stage Page Table
G-stage Page Table
PA → Memory Tracking Table Lookup → Allow/Disallow
And PA → PMP Check → Allow/Disallow
First stage paging
Not enabled
G-stage paging not
enabled
paging not enabled
Access
Allowed?
No: Access fault
Yes: Access allowed
PA → Memory Tracking Table Lookup → Allow/Disallow
And PA → IOPMP Check → Allow/Disallow
Device Second stage Page Table
Device First stage Page Table
Device Id, Address → SDID, IOMMU ID
First stage paging
Not enabled
Second stage
paging not enabled
paging not enabled

Supervisor Domains “Smmtt” Extension

55 – 46	45 – 25	24 – 16	15 – 12	11 – 0
pn[3]	pn[2]	pn[1]	pn[0]	page offset
10	21	9	4	12



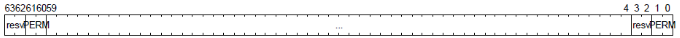
MTT L3
Table Size : 8 KiB
# of entries : 1024
Addr Space/Entry: 64 TiB

MTT L2
Table Size : 16 MiB
# of entries : 2 M
Addr Space/Entry: 32 MiB

MTT L1
Table Size : 4 KiB
# of entries : 8 K
Addr Space/Entry: 4 KiB

+ Base + offset to generate PA for entry at this level

MTTL1 Access-permission encoding	Description
00b	The entry specifies access to the 4 KiB address space is **not allowed** for the domain.
01b	The entry specifies **read** and **execute** (but **no write**) access is allowed to the 4 KiB address space for the domain.
10b	The entry specifies **read** and **write** (but **no execute**) access is allowed to the 4 KiB address space for the domain.
11b	The entry specifies **read**, **write** and **execute** access is allowed to the 4 KiB address space for the domain.

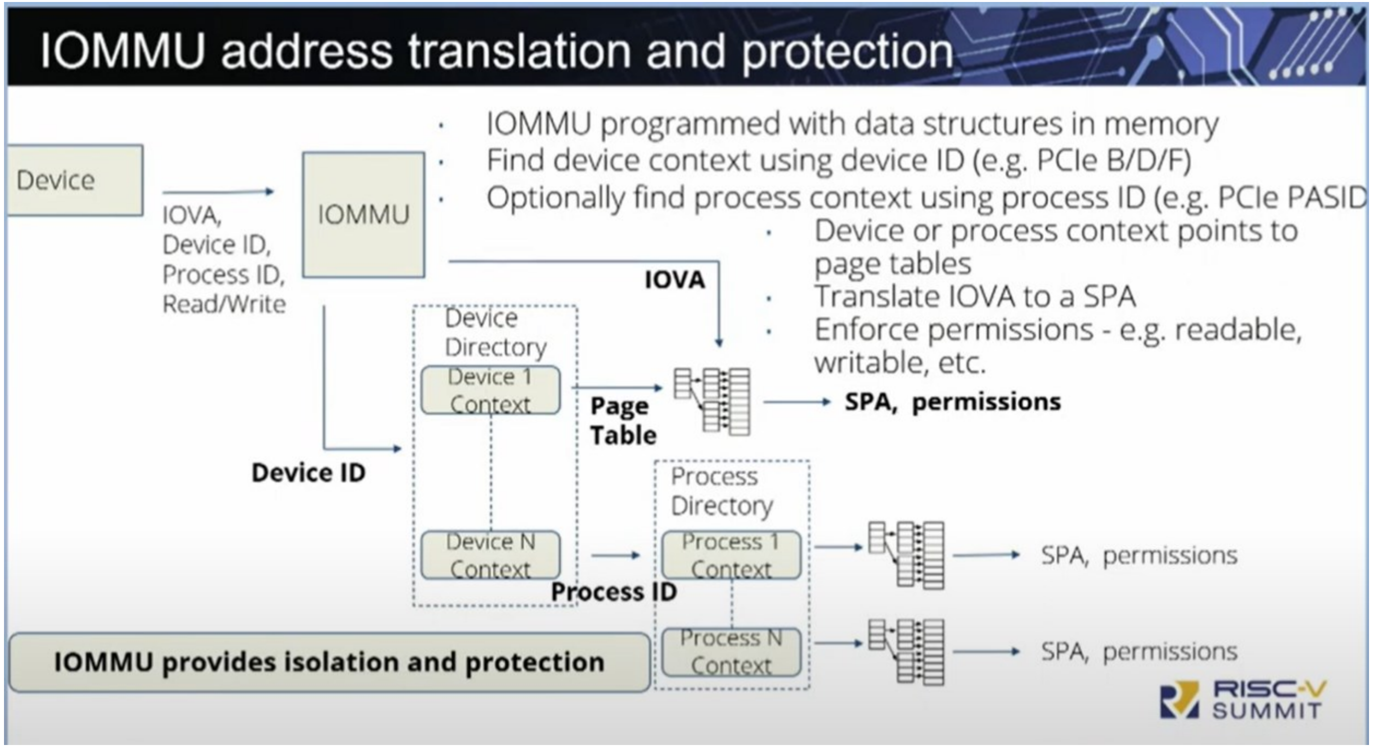


RISC-V AIA & IOMMU (Background)

RISC-V Advanced Interrupt Arch. (AIA) and IOMMU specifications ratified!

https://github.com/riscv/riscv-aia

https://github.com/riscv-non-isa/riscv-iommu



IOMMU address translation and protection

- IOMMU programmed with data structures in memory
- Find device context using device ID (e.g. PCIe B/D/F)
- Optionally find process context using process ID (e.g. PCIe PASID
  - Device or process context points to page tables
  - Translate IOVA to a SPA
  - Enforce permissions - e.g. readable, writable, etc.

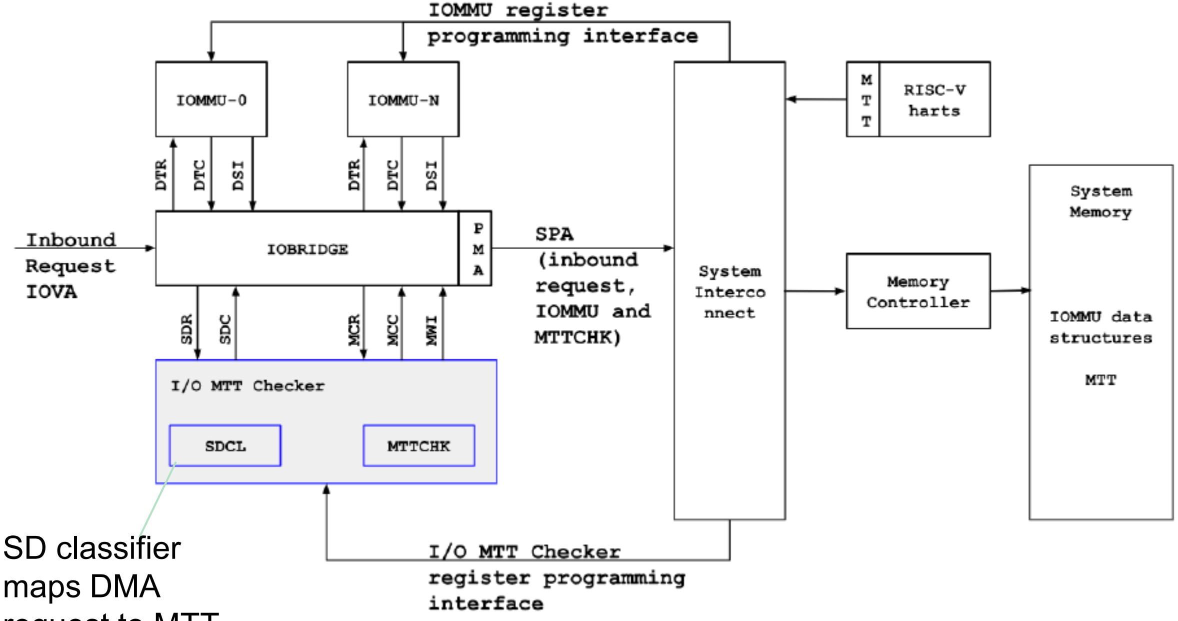
IOMMU provides isolation and protection

Supervisor Domains “IO-MTT” extension

- Supervisor domains may be granted control over DMA-capable devices by assigning IOMMU instances to the SD.

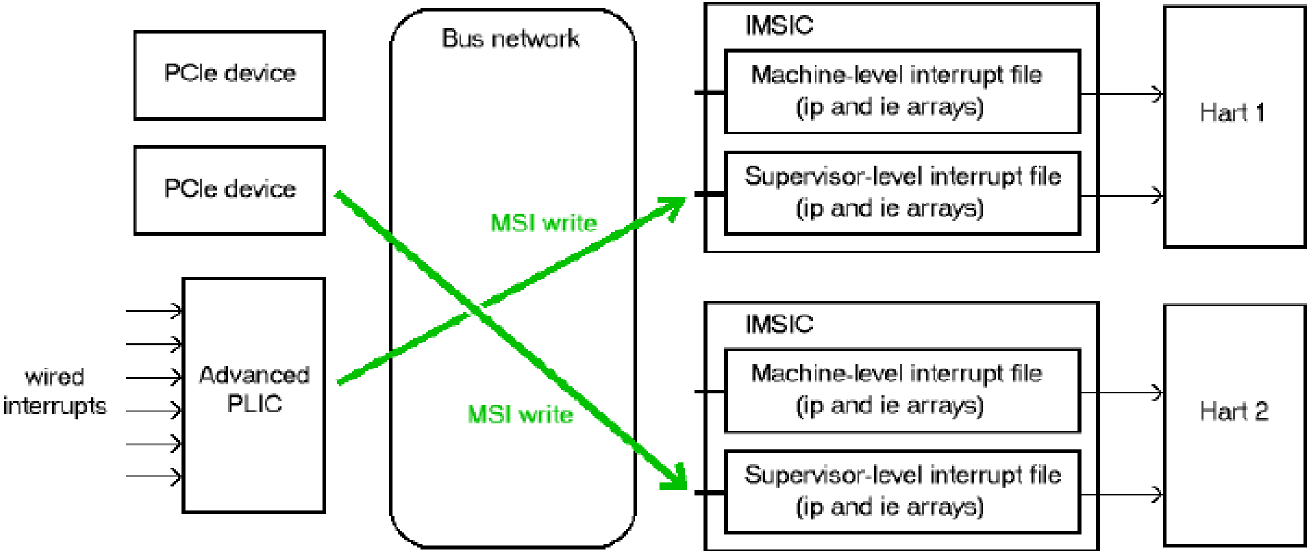
- **Security Objective** - DMA from the devices and the IOMMU linked with a SD must adhere strictly to the access protections encoded in the MTT of the respective SD.

- Also, using the MTT, the RDSM enforces that the IOMMU memory-mapped programming regions are access-restricted to the SD the IOMMU is assigned to.



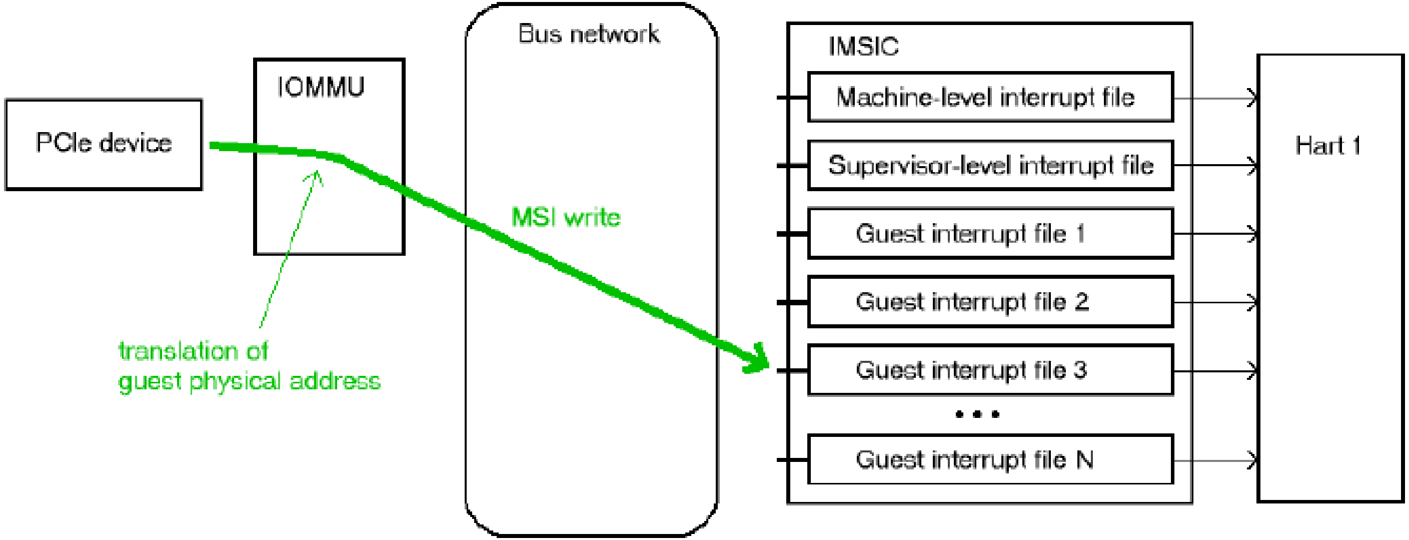
SD classifier maps DMA request to MTT

RISC-V Advanced Interrupt Arch. (Background)


PCIe device
PCIe device
wired interrupts
Advanced PLIC
Bus network
MSI write
MSI write
IMSIC
Machine-level interrupt file (ip and ie arrays)
Supervisor-level interrupt file (ip and ie arrays)
IMSIC
Machine-level interrupt file (ip and ie arrays)
Supervisor-level interrupt file (ip and ie arrays)
Hart 1
Hart 2

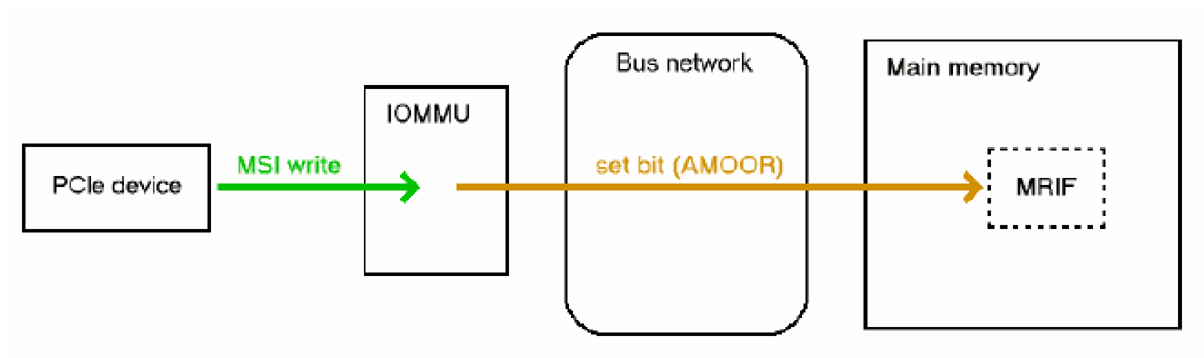
MSI – Message signaled interrupt

RISC-V AIA



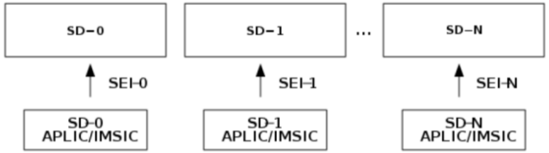
PCIe device
IOMMU
Bus network
MSI write
translation of
guest physical address
IMSIC
Machine-level interrupt file
Supervisor-level interrupt file
Guest interrupt file 1
Guest interrupt file 2
Guest interrupt file 3
• • •
Guest interrupt file N
Hart 1

RISC-V AIA

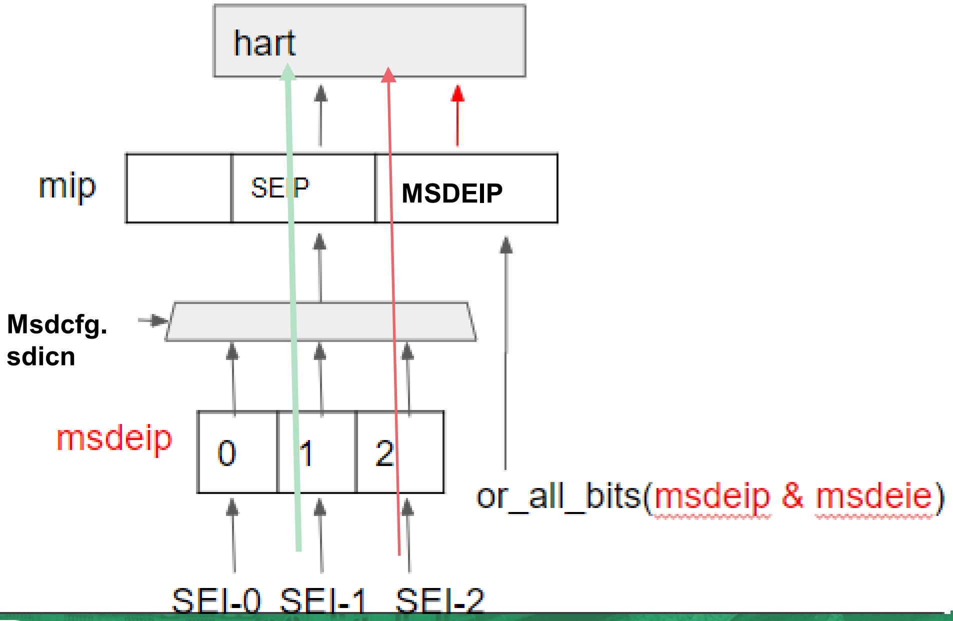


AMO – Atomic Memory operation
MRIF – Memory resident interrupt file

Supervisor Domains “Smsdia” Extension

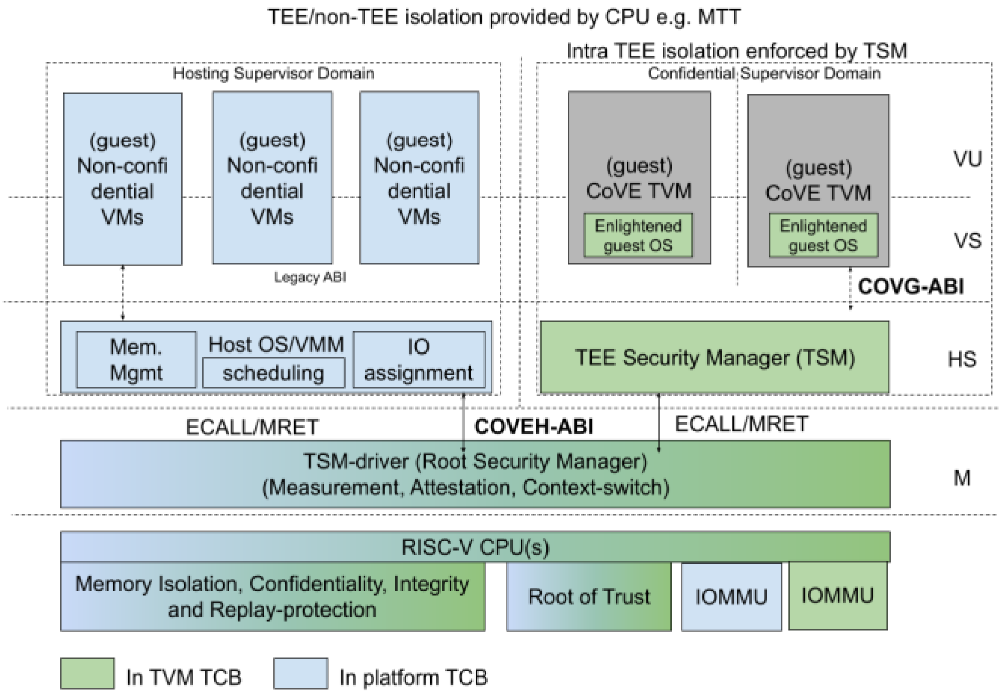


- **Security Objective** – RDSM must enforce integrity of interrupt delivery to the Supervisor Domain
- Ideally for devices assigned to a SD, external interrupts can be directly assigned to the SD. Smsdia enables this functionality.
- RDSM uses MTT to limit access and enforce exclusive SD access to assigned interrupt controllers, and uses the **msdcfg** CSR to select the interrupt controller to associate to the SD. The RDSM uses CSRs **msdeip** and **msdeie** to get notifications when SD is not active.
- Once an implemented interrupt controller is selected for SD, the H/S mode CSR interaction remains the same as defined in AIA.



Non-ISA (CoVE ABI)

- CoVE specifies two primary interfaces:
  - COVH – ABI between OS/VMM and the TSM
  - COVG – ABI between the TVM and the TSM
- COVH provides interfaces for:
  - TSM and TVM Measurement and Attestation
  - Memory Conversion between Domains
  - TVM HW state isolation & execution
  - Secure Interrupt Mgmt
  - Debug & Performance monitoring
- COVG provides interfaces for:
  - Extending dynamic measurements
  - Getting attestation credentials
- Salus is the Rivos open source (Rust-based) TSM. https://github.com/rivosinc/salus
- CoVE-IO extends the ABI to enable device function binding to the TVM (see next slides)



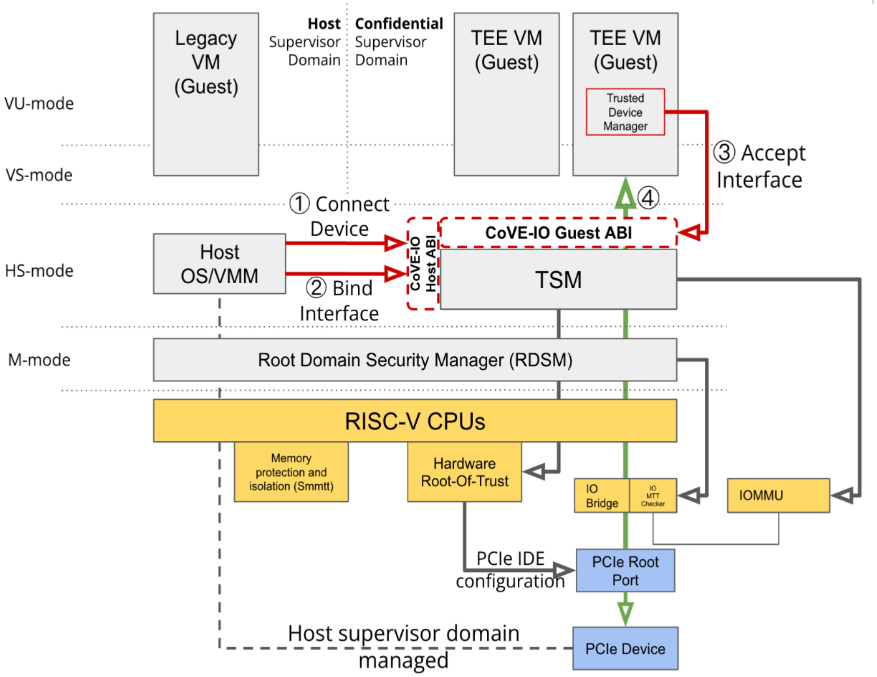
TEE/non-TEE isolation provided by CPU e.g. MTT
Intra TEE isolation enforced by TSM
Hosting Supervisor Domain
Confidential Supervisor Domain
(guest) Non-confi dential VMs
(guest) Non-confi dential VMs
(guest) Non-confi dential VMs
(guest) CoVE TVM
Enlightened guest OS
(guest) CoVE TVM
Enlightened guest OS
VU
VS
Legacy ABI
COVG-ABI
Mem. Mgmt
Host OS/VMM
scheduling
IO assignment
TEE Security Manager (TSM)
HS
ECALL/MRET
COVEH-ABI
ECALL/MRET
TSM-driver (Root Security Manager)
(Measurement, Attestation, Context-switch)
M
RISC-V CPU(s)
Memory Isolation, Confidentiality, Integrity and Replay-protection
Root of Trust
IOMMU
IOMMU
In TVM TCB
In platform TCB

Smmtt QEMU at https://github.com/grg-haas/smmtt

SW Security WG forming in: RISE RISC-V Software Ecosystem

Extending CoVE for TEE-IO

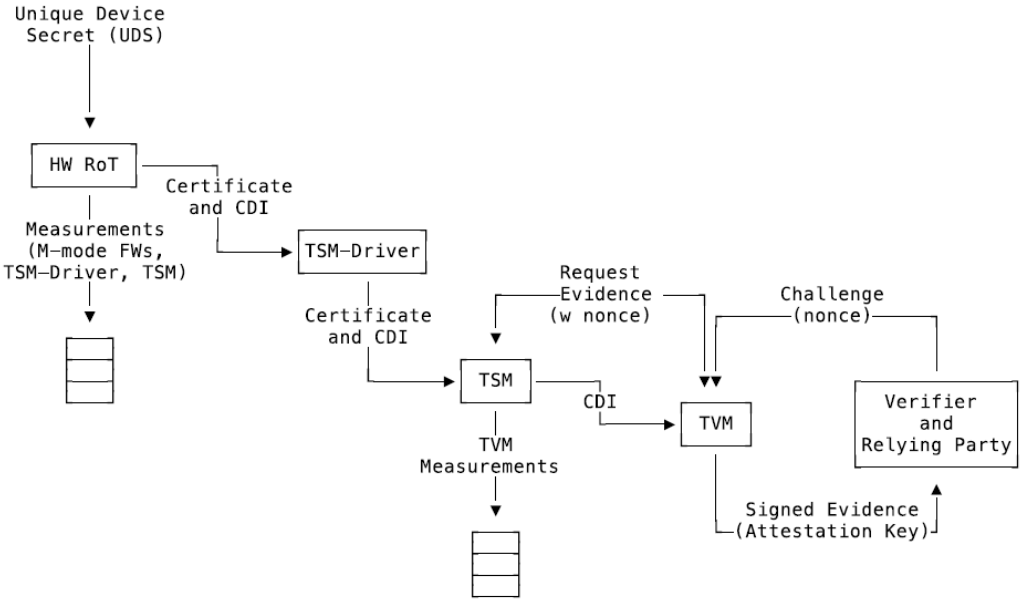
- Uses IO-MTT, and Smsdia to enable RDSM to manage IOMMU and device assignment isolation
- CoVE-IO extends ABI to assign devices to TVMs
  - Common API for connect, bind, accept being discussed in the Linux CoCo community
- CoVE-IO also uses industry standards such as SPDM, and PCIe IDE and TDISP for device authentication and state management.



Other Non-ISA (Platform)

HW Root Of Trust

- Supports identity, attestation to provide cryptographic evidence of the TCB elements:
  - ○ HW RoT HW and FW
  - ○ RDSM (TSM-driver)
  - ○ TEE Security Manager for a SD
  - ○ TEE VM
- The HW RoT should support a layered attestation model e.g. TCG DICE (Device Identity Composition Engine), and the IETF RATS framework for attestation
- E.g. OpenTitan (Darjeeling) uses a RISC-V core and is open-source secure silicon RoT for instantiation within a larger SoC or chiplet. https://opentitan.org/


Unique Device Secret (UDS)
HW RoT
Certificate and CDI
Measurements (M-mode FWs, TSM-Driver, TSM)
TSM-Driver
Certificate and CDI
Request Evidence (w nonce)
Challenge (nonce)
TSM
CDI
TVM
Verifier and Relying Party
TVM Measurements
Signed Evidence (Attestation Key)

CDI = Compound Device Identity

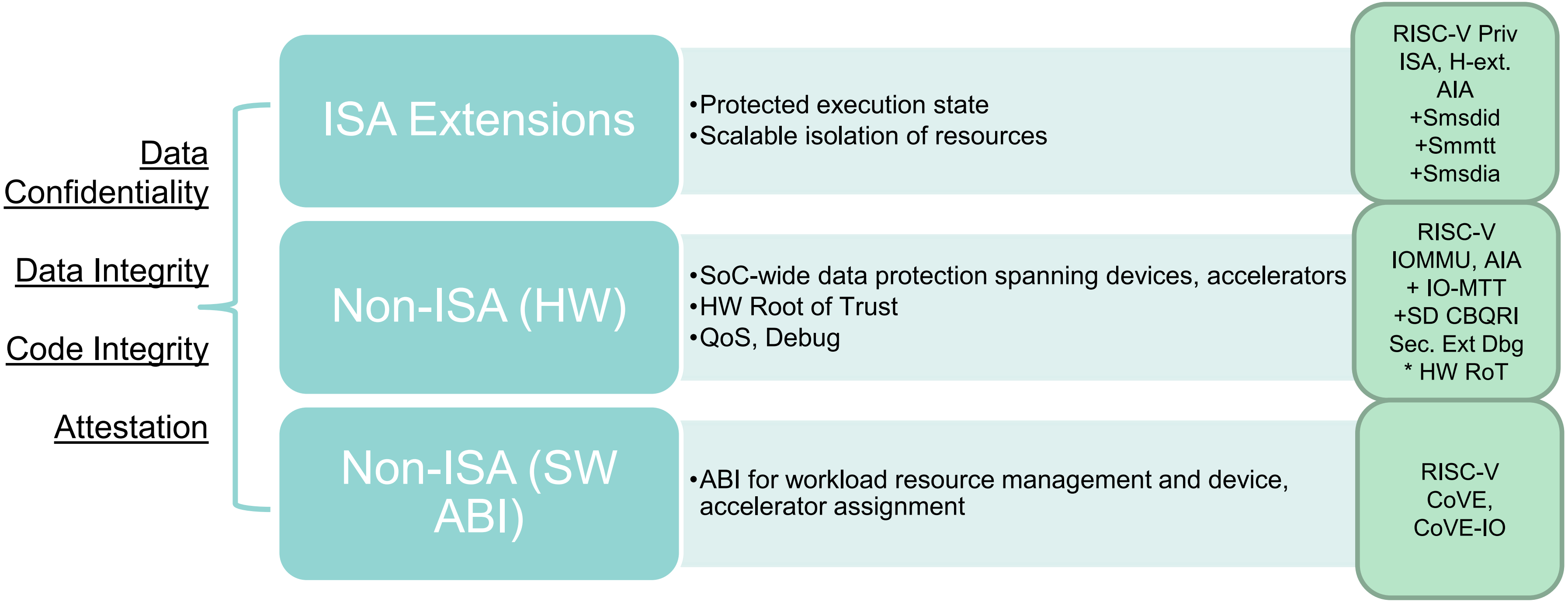
Other Non-ISA (Platform)

- Mitigate other threats to TEE data on SoC:
    - Data leaving the SoC package to DRAM, PCIe, CXL etc.
    - Invasive Debug via scan, trace etc;
    - Machine monitoring: QoS, Performance counters
    - Critical configuration of address decoders, routing tables etc.;

- Mitigations
    - Cryptographic protections (confidentiality, integrity and replay protection)
    - Filtering by Supervisor Domain (by HW or TCB SW)
    - Opt-in and Activation status reflected in attestation
    - Restricted access, configure and/or verified by TCB HW/SW (eg. RoT, RDSM)

https://github.com/riscv-non-isa/riscv-external-debug-security

Confidential Computing on RISC-V



Data Confidentiality

Data Integrity

Code Integrity

Attestation

ISA Extensions	•Protected execution state •Scalable isolation of resources	RISC-V Priv ISA, H-ext. AIA +Smsdid +Smmtt +Smsdia
Non-ISA (HW)	•SoC-wide data protection spanning devices, accelerators •HW Root of Trust •QoS, Debug	RISC-V IOMMU, AIA + IO-MTT +SD CBQRI Sec. Ext Dbg * HW RoT
Non-ISA (SW ABI)	•ABI for workload resource management and device, accelerator assignment	RISC-V CoVE, CoVE-IO

Summary & Call to Action

- *RISC-V's open and clean-slate design presents a unique opportunity to ingrain security for the next generation of compute infrastructure.*
- *Confidential computing is a key security capability for RISC-V platforms for scalable multi-tenant data-in-use protection.*
- ***RVI Task groups are actively working on ratification of CoVE[-IO] ABIs, as well as RISC-V Priv. ISA extensions for Supervisor Domains.***
  - ***Review and provide feedback (via issues/PRs) on specs from TGs***
  - ***Participate in open source SW development towards ratification***

LINUX SECURITY SUMMIT EUROPE

Thanks for your attention

Acknowledgements (in alphabetical order)

Andrew Bresticker, Andy Dellow, Anup Patel, Atish Patra, Atul Khare, Beeman Strong, Christian Bolis, Dean Liberty, Deepak Gupta, Dingji Li, Dong Du, Dylan Reid, Eckhard Delfs, Fabrice Marinet, Gokhan Kaplayan, Greg Favor, Gregor Haas, Guerney Hunt, Jiewen Yao, Kailun Qin, Krste Asanovic, Mark Hill, Manuel Offenberg, Nick Kossifidis, Nick Wood, Osman Koyuncu, Paul Elliott, Qing Li, Rajnesh Kanwal, Rob Bradford, Ravi Sahita, Samuel Holland, Samuel Ortiz, Vedvyas Shanbhogue, Wojchiech Ozga, Yann Loisel and others.

Folks contributing in Security HC, Trusted Computing SIG, AP-TEE TG, AP-TEE-IO TG, Smmtt TG, Runtime Integrity SIG, SoC Infra HC, DTPM SIG, Hypervisor SIG, PRS TG, Priv IC

Rivos