CVM Extension: Confidential Virtual Machine Extension for RISC-V

Dingji Li

IPADS, SJTU

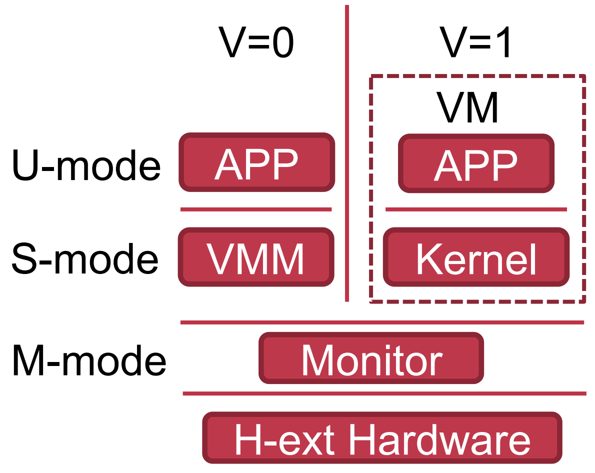
Data Security is Important

- **Security-critical applications are everywhere**
  - Scenarios: cloud, PC, end device
  - Examples: serverless, machine learning, mobile payment

- **User requirements**
  - Robust privacy protection
  - Great performance

Hardware TEE is Emerging

- **Trusted Execution Environment**
  - Hardware-enforced isolation
  - Data is only visible to authorized code
- **VM is suitable for enabling TEE**
  - Clear security boundary
  - Compatibility with existing applications



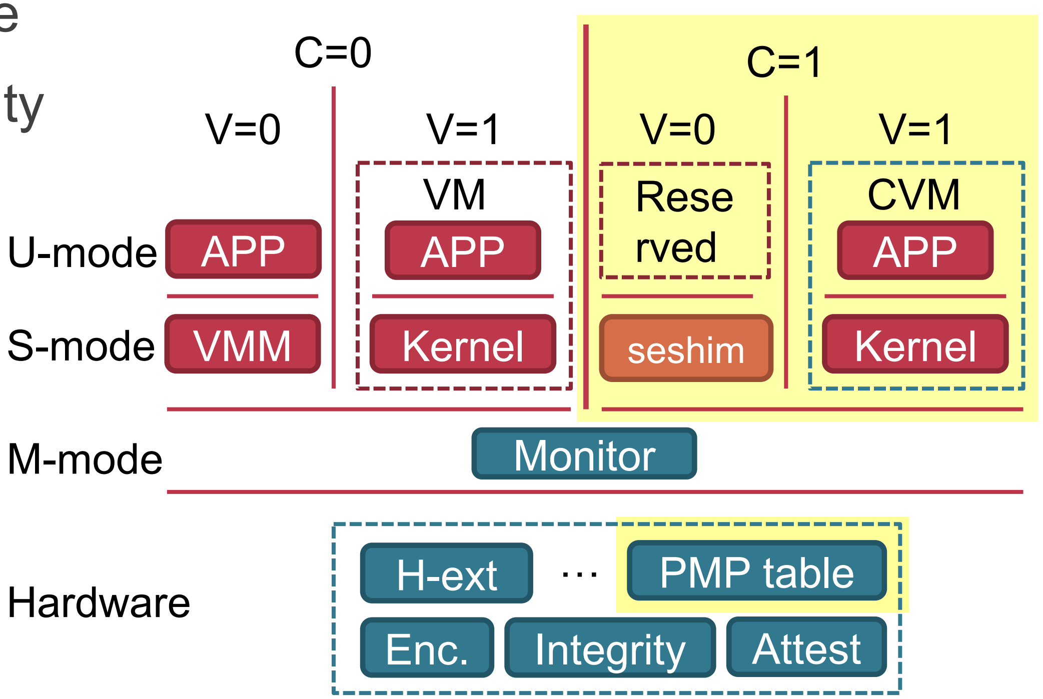
Current Hypervisor Extension

Confidential Virtual Machine Extension

CVM EXTENSION

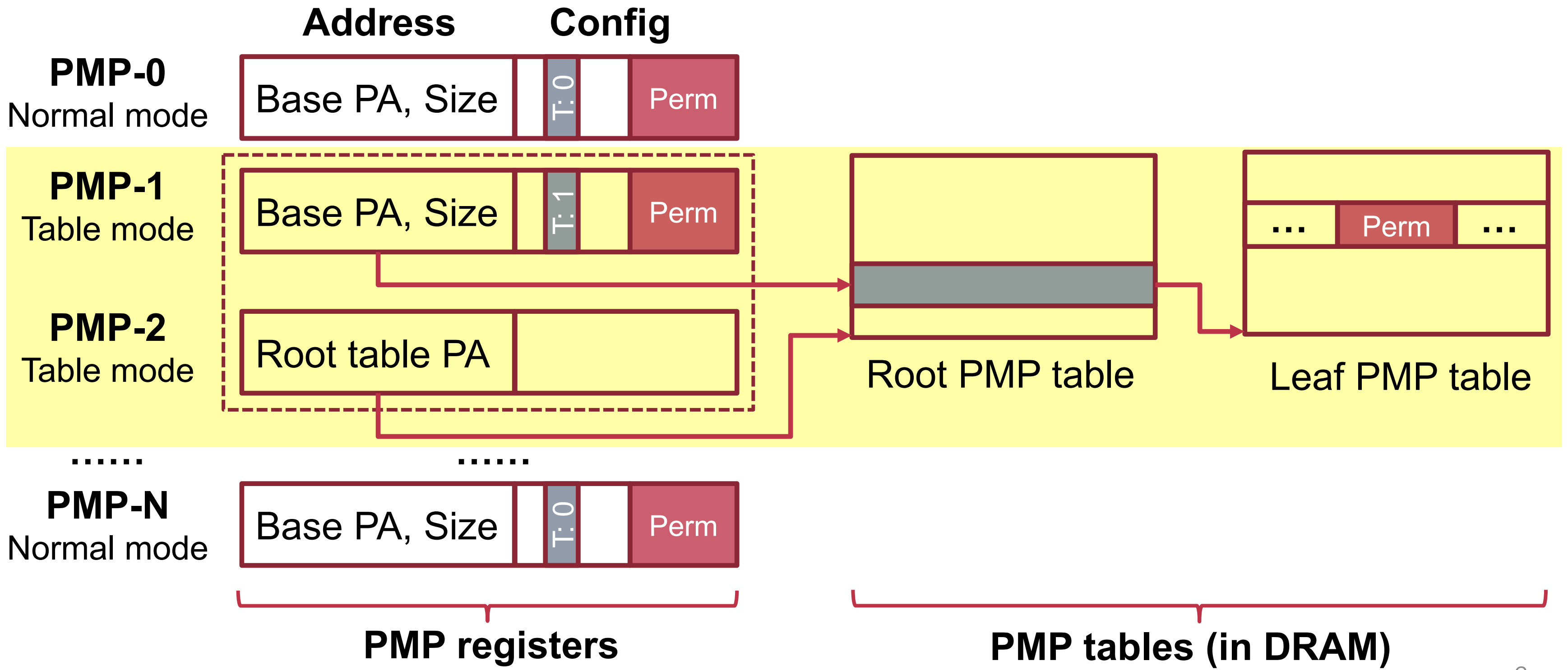
Overall Architecture

- **Confidentiality mode**
  - Isolated from non-C mode
  - Confidentiality and integrity
  - CHS-mode: secure shim
- **PMP table extension**
  - Fine-grained physical memory isolation



PMP Table Extension Overview

- **Extends a PMP mode to support protection table**



Threat Model

- **Four classes of attacks**
  - Privileged software attacks
    - E.g., compromised host OS, VMM
  - Physical attacks
    - E.g., cold boot attack (dump DRAM content)
  - Controlled channel attacks
    - Other side-channel attacks (e.g., cache, TLB) are out of scope
  - DMA attacks
    - E.g., devices overwrite memory

PMP Table Extension

MEMORY PROTECTION

PMP Configuration Register

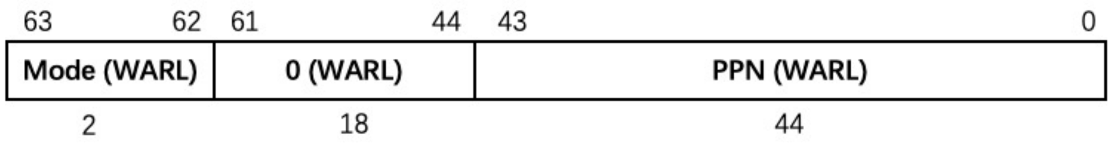
- **T bit**
  - T=0, the same as existing PMP entry (normal mode)
  - T=1, this PMP entry is in table mode
- **C bit**
  - C=0, both C and non-C mode are allowed to access the memory
  - C=1, the non-C mode software cannot access the memory

7	6	5	4 3	2	1	0
L (WARL)	C (WARL)	T (WARL)	A (WARL)	X (WARL)	W (WARL)	R (WARL)
1	1	1	2	1	1	1

Format of PMP configuration register

PMP Address Register

- **If i-th PMP entry is in table-mode**
  - pmpaddr[i+1] holds the base PPN of the root PMP table



Format of pmpaddr[i+1] when T=1 in pmpaddr[i] (RV64)

Value	**Name**	**Description**
0	Two-level protection table	The PMP table has two levels. The mode assumes Sv32 physical address for RV32 and Sv39 physical address for RV64.
1-3	–	Reserved for standard use

Encoding of the mode field when PMP table is enabled

PMP Table Structure

- **Radix-tree table**
  - V=0, invalid; V=1, valid
  - C=0, C & non-C can access; C=1, only C can access
  - R=W=X=0, next level; otherwise, final permission

63 49	48 23	22 14	13 5	4	3	2	1	0
Reserved	PPN[2]	PPN[1]	PPN[0]	C	X	W	R	V
15	26	9	9	1	1	1	1	1

Format of root PMP table entry (RV64)

63 60	59 56	55 52	51 48	47 44	43 40	39 36	35 32	31 28	27 24	23 20	19 16	15 12	11 8	7 4	3 0
perm15	perm14	perm13	perm12	perm11	perm10	perm9	perm8	perm7	perm6	perm5	perm4	perm3	perm2	perm1	perm0
4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4

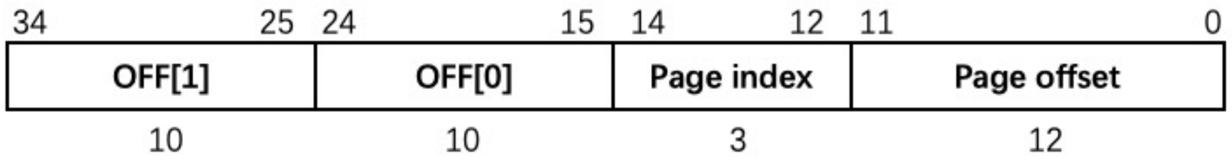
Format of leaf PMP table entry (RV64)

3	2	1	0
C (WARL)	X (WARL)	W (WARL)	R (WARL)
1	1	1	1

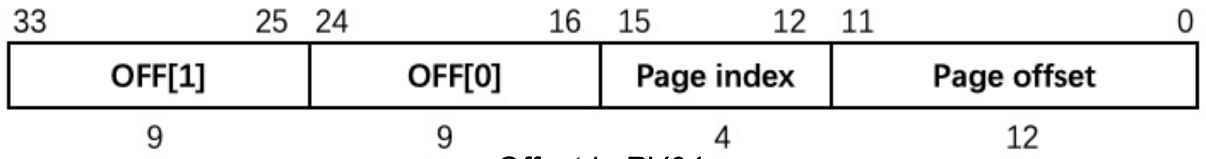
Permissions for a 4KiB physical page in leaf pmpte

Permission Indexing Process

- **Relative offset indexing**
  - offset = phy_addr - range_base
  - OFF[1]: root PMP table index
  - OFF[0]: leaf PMP table index
  - Page index: permission index in the leaf PMP entry



Offset in RV32



Offset in RV64

Rational

- **On-demand metadata costs**
  - No extra cost if the system does not enable PMP table
- **Flexibility**
  - The M-mode software can allocate the page dynamically
- **Compatible with existing PMP/ePMP design**
  - Without introducing a new layer of protection mechanism
- **Better indexing performance**
  - Only two-level tables for 16GiB physical memory

CHS-MODE CSR

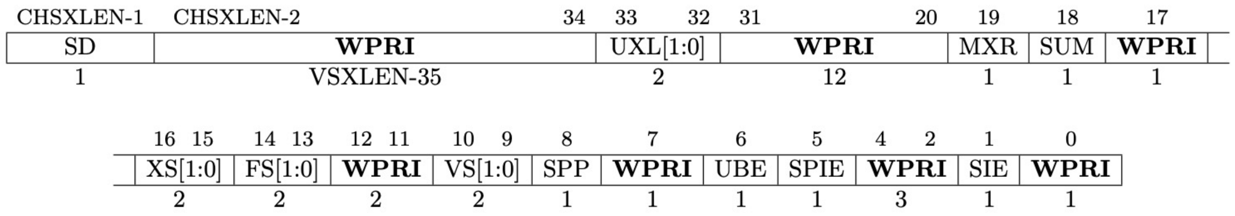
Duplicate HS-mode CSRs for CHS-mode

- **Direct switch between HS-mode and CHS-mode**
  - Software alone is unable to swap a CSR
  - Same reason as duplicating S-mode CSRs for VS-mode

- **New CSRs**
  - *chsstatus, chstvec, chsscratch, chsepc, chscause, chstval, chsatp*

Confidential Hypervisor-extended Supervisor Status Register (chsstatus)

- **UXL and UBE fields**
  - Hardwired to 0 and reserved for future use



CHSXLEN-1	CHSXLEN-2 … 34	33 … 32	31 … 20	19	18	17
SD	**WPRI**	UXL[1:0]	**WPRI**	MXR	SUM	**WPRI**
1	VSXLEN-35	2	12	1	1	1

16 … 15	14 … 13	12 … 11	10 … 9	8	7	6	5	4 … 2	1	0
XS[1:0]	FS[1:0]	**WPRI**	VS[1:0]	SPP	**WPRI**	UBE	SPIE	**WPRI**	SIE	**WPRI**
2	2	2	2	1	1	1	1	3	1	1

Confidential hypervisor-extended supervisor status register (chsstatus) when CHSXLEN=64

Other Confidential Hypervisor-extended CSRs

- **When C=1 and V=0**
  - CHS-mode CSRs substitute usual HS-mode CSRs
  - Same format as HS-mode CSRs

Substitution	**Usual**
chstvec	stvec
chsscratch	sscratch
chsepc	sepc
chscause	scause
chstval	stval
chsatp	satp

Restrictions on Interrupt CSRs in CHS-mode

- **The shim layer may be untrusted**
  - Can interfere or forge interrupts to the hypervisor in HS-mode

- **CHS-mode software cannot modify *sie* and *sip***
  - The secure shim cannot modify the interrupt states before switching to the hypervisor in the HS-mode

VM load/store/fence, exception delegation

H-EXT SECURITY ENHANCEMENT

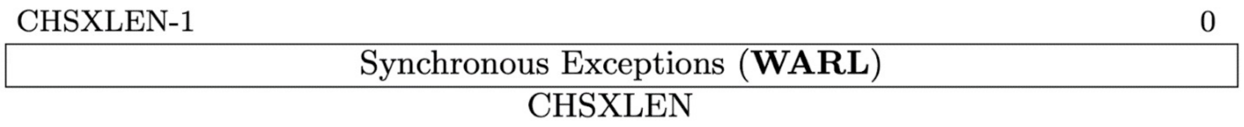
VM Load/Store/Fence

- **HLV, HLVX, HSV**
  - When C=0, no software can access CVM's private memory
  - When C=1, CHS-mode software can access

- **HFENCE.VVMA, HFENCE.GVMA**
  - Only CHS-mode software can apply to CVMs
  - CHS-mode software provides interfaces for HS-mode

Confidential Hypervisor-extended Hypervisor Exception Delegation Register (chhedeleg)

- **Allow guest exception delegation**
  - Inform CVMs of operations that require hypervisor emulations
  - C=1 and V=0, *chhedeleg* substitutes for the *hedeleg*



Confidential hypervisor-extended hypervisor exception delegation register (chhedeleg)

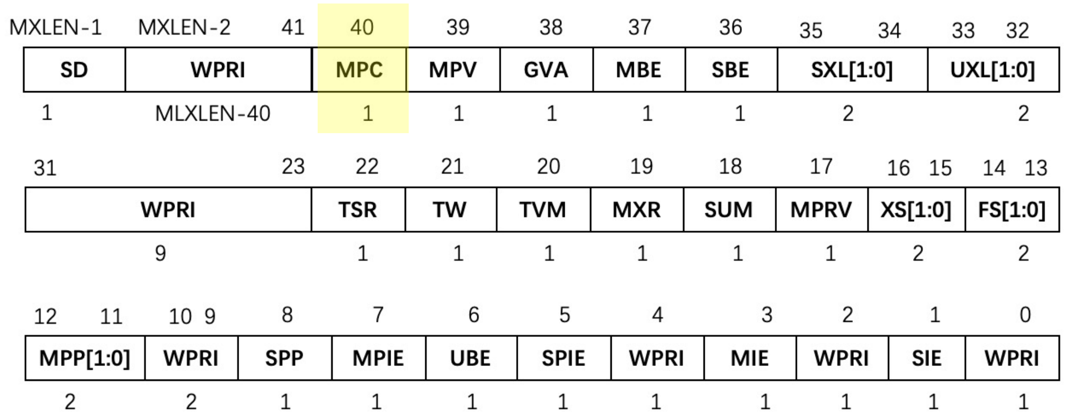
Bit	Attribute	Corresponding Exception
20	Writable	Instruction guest-page fault
21	Writable	Load guest-page fault
22	Writable	Virtual instruction
23	Writable	Store/AMO guest-page fault

Attributes in chhedeleg that are different from hedeleg

TRAPS & MODE SWITCHING

Machine Status Registers (mstatus)

- **MPC bit (Machine Previous Confidentiality Mode)**
  - Similar to MPP bit
  - MPRV bit is affected by current confidentiality mode



MXLEN-1	MXLEN-2 41	40	39	38	37	36	35 34	33 32
SD	WPRI	MPC	MPV	GVA	MBE	SBE	SXL[1:0]	UXL[1:0]
1	MLXLEN-40	1	1	1	1	1	2	2

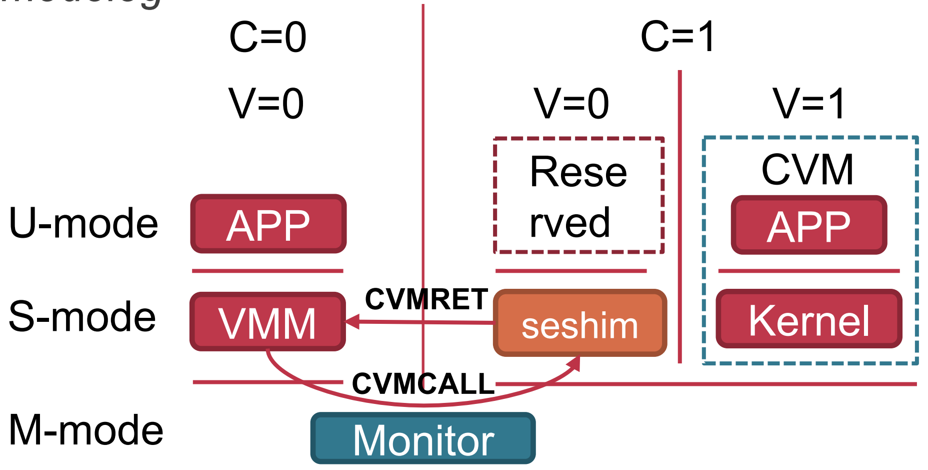
31 23	22	21	20	19	18	17	16 15	14 13
WPRI	TSR	TW	TVM	MXR	SUM	MPRV	XS[1:0]	FS[1:0]
9	1	1	1	1	1	1	2	2

12 11	10 9	8	7	6	5	4	3	2	1	0
MPP[1:0]	WPRI	SPP	MPIE	UBE	SPIE	WPRI	MIE	WPRI	SIE	WPRI
2	2	1	1	1	1	1	1	1	1	1

Status register format (RV64) when the CVM extension is implemented

Mode Switching

- **CVMCALL and CVMRET**
  - Switch between HS-mode and CHS-mode directly
  - Assert CVM-enter and CVM-leave exceptions
    - Add two new bits in *medeleg*



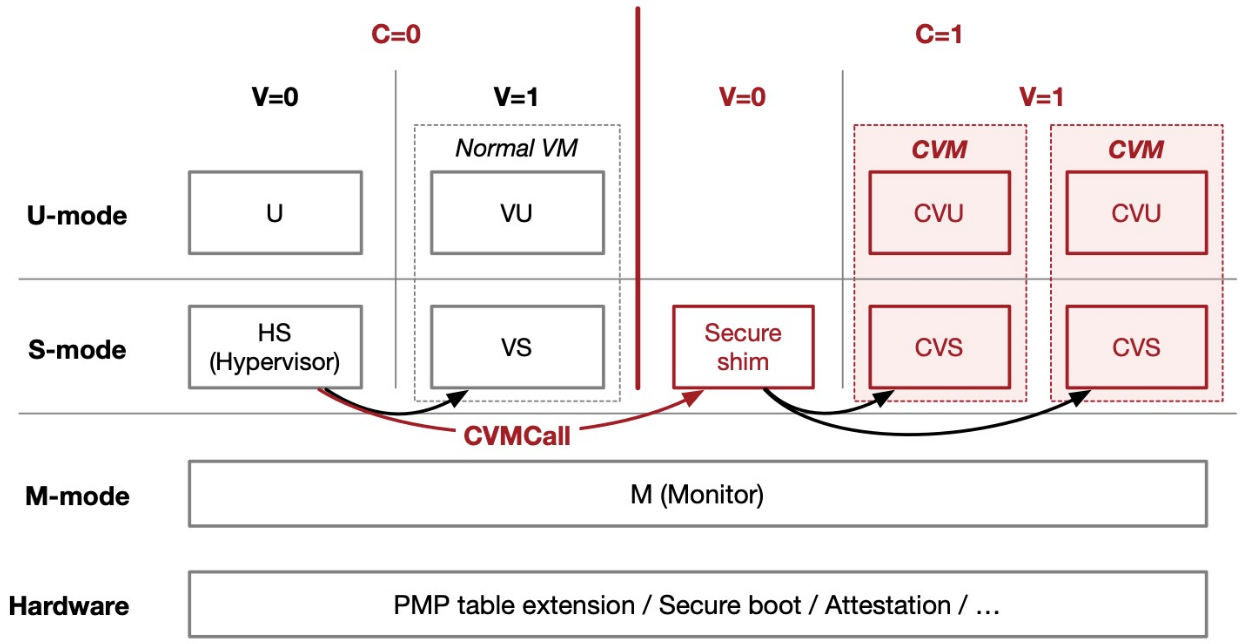
An example of CVMCALL (not delegated) and CVMRET (delegated)

Possible software design cases

SOFTWARE GUIDELINE

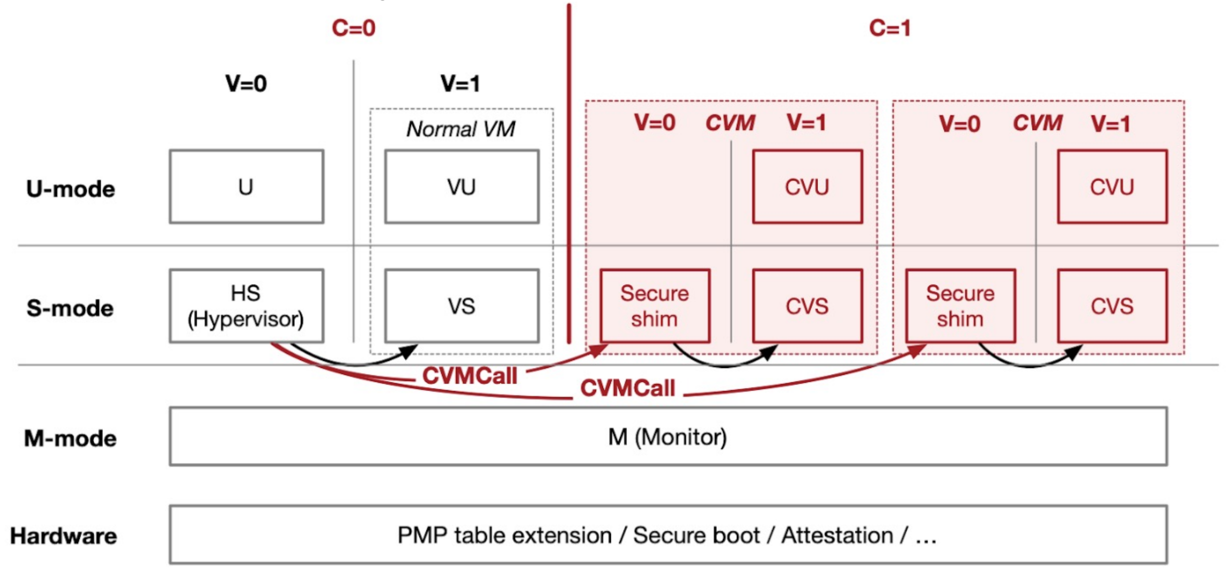
Global Trusted Secure Shim Model

- **Trusted secure shim**
  - Responsible to protect the CPU states (e.g., CSRs) for all CVMs



Per-CVM Secure Shim Model

- **Untrusted secure shim**
  - Responsible to protect the CPU states (e.g., CSRs) for a single-CVM
  - Can be customized by end-users



Conclusion

- CVM-ext aims to enable VM-based TEE on RISC-V
  - Strong threat model
    - Privileged software/physical/side-channel/DMA attacks
  - CPU states: flexible secure shim software
  - Memory: fine-grained isolation, confidentiality, integrity
- Compatible with existing H-ext and PMP

Thanks!