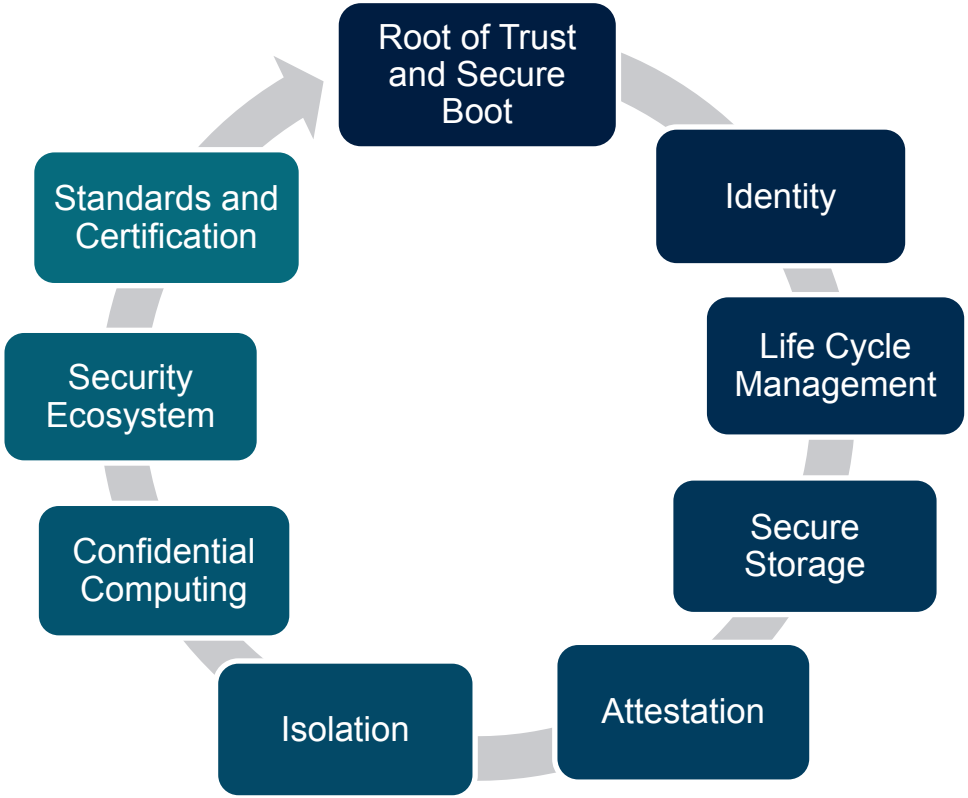
RISC-V Security Overview
CCC TAC Meeting

Dec 2, 2021

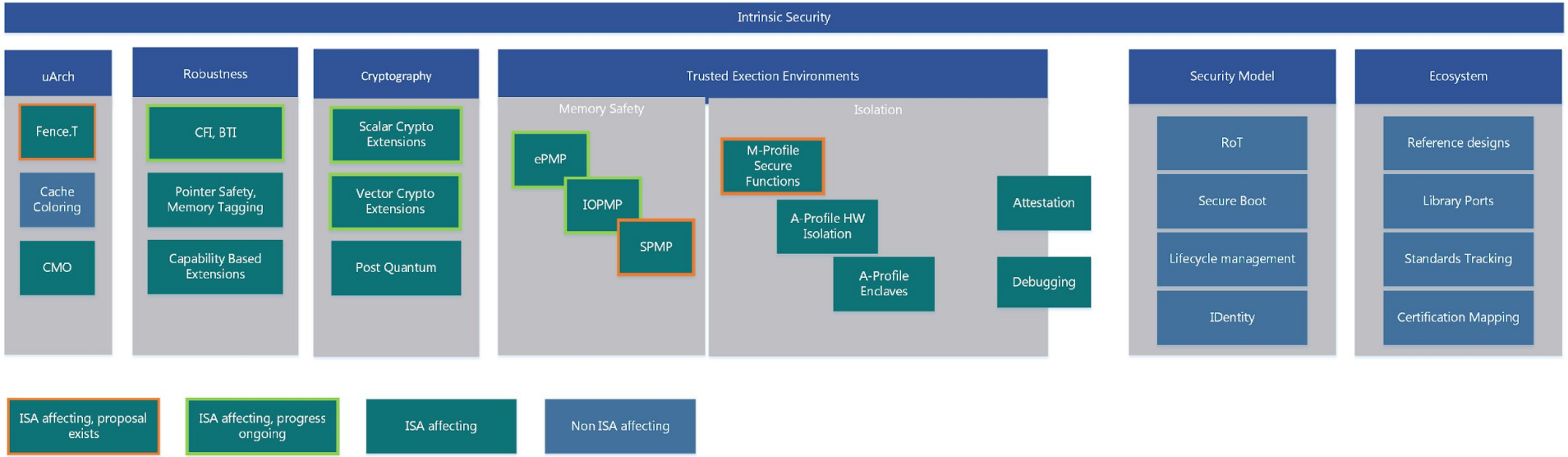
Intrinsic Security

Zero Trust Model

- Security as a basic feature of HW, SW Firmware
- Support security through entire lifecycle
- Guidelines matched to profiles



Scope


Intrinsic Security
uArch
Fence.T
Cache Coloring
CMO
Robustness
CFI, BTI
Pointer Safety, Memory Tagging
Capability Based Extensions
Cryptography
Scalar Crypto Extensions
Vector Crypto Extensions
Post Quantum
Trusted Exection Environments
Memory Safety
ePMP
IOPMP
SPMP
Isolation
M-Profile Secure Functions
A-Profile HW Isolation
A-Profile Enclaves
Attestation
Debugging
Security Model
RoT
Secure Boot
Lifecycle management
IDentity
Ecosystem
Reference designs
Library Ports
Standards Tracking
Certification Mapping
ISA affecting, proposal exists
ISA affecting, progress ongoing
ISA affecting
Non ISA affecting

Security Model

- State Goals & Rationale for RISC-V Security
- Defines scope
- Derives security requirements
- Abstracted from implementation specifics
- References existing standards when appropriate
- Referenced by appropriate sections of Platform specifications
    - By Profile, By Vertical

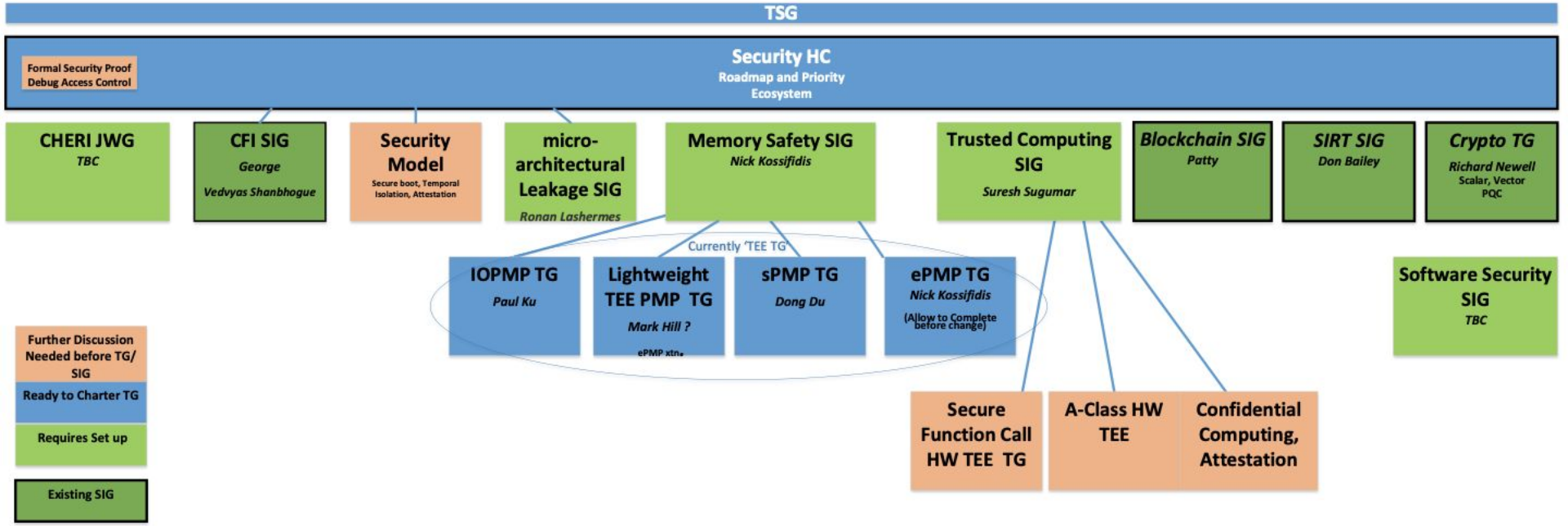
Topics:

- Lifecycle
- Platform Integrity
- Data Integrity
- Isolation
- Secure Boot
- Assurance & Attestation
- Secure storage
- Cryptography
- Applicable Standards

Security Ecosystem

- Enablement of RISC-V security services & software
- Identify and list key open-source security software and libraries
- Develop RISC-V security reference implementation(s)
- Identify, monitor, and influence applicable standards
- Identify and liaison with applicable Security Certification entities

Security Organization



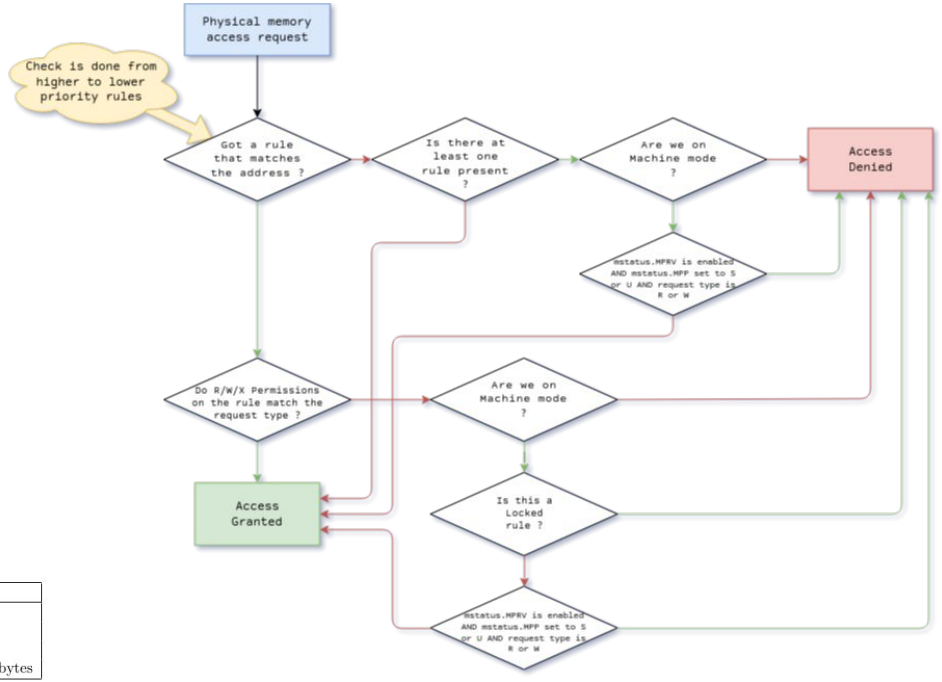
RISC-V®

Memory Safety

RISC-V®

Physical Memory Protection (PMP)

- Basic isolation between M-mode and S/U-modes.
- Normal rules apply to S/U, Locked rules (impossible to edit after adding them) apply to all modes. PMP gives access to S/U (locked down by default) and restricts M (full access by default).
- Up to 64 entries for defining physical memory regions and their permissions.
- Support for three different addressing modes (TOR, NA4, NAPOT).
- Priority matching from lower to higher indexed entries.



pmpaddr	pmpcfg.A	Match type and size
yyyy...yyyy	NA4	4-byte NAPOT range
yyyy...yyy0	NAPOT	8-byte NAPOT range
yyyy...yy01	NAPOT	16-byte NAPOT range
yyyy...y011	NAPOT	32-byte NAPOT range
...	...	...
yy01...1111	NAPOT	2^{XLEN}-byte NAPOT range
y011...1111	NAPOT	2^{XLEN+1}-byte NAPOT range
0111...1111	NAPOT	2^{XLEN+2}-byte NAPOT range
1111...1111	NAPOT	2^{XLEN+3}-byte NAPOT range

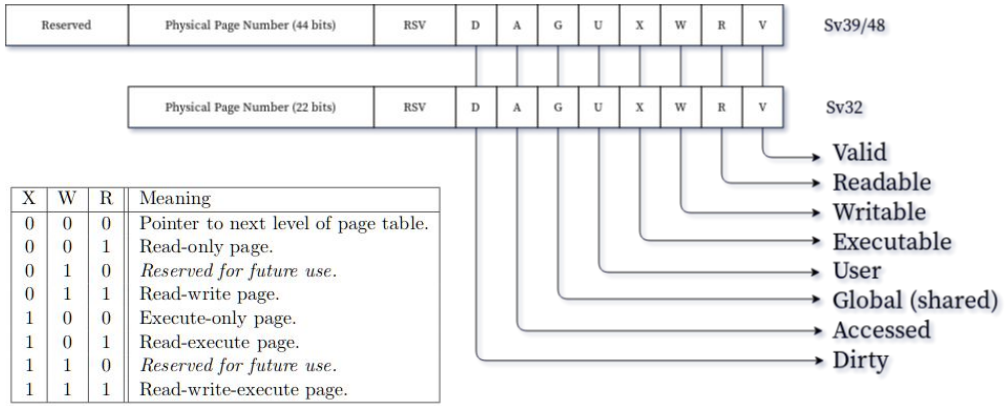
A	Name	Description
0	OFF	Null region (disabled)
1	TOR	Top of range
2	NA4	Naturally aligned four-byte region
3	NAPOT	Naturally aligned power-of-two region, ≥8 bytes

Table 3.8: Encoding of A field in PMP configuration registers.

RISC-V®

Virtual memory

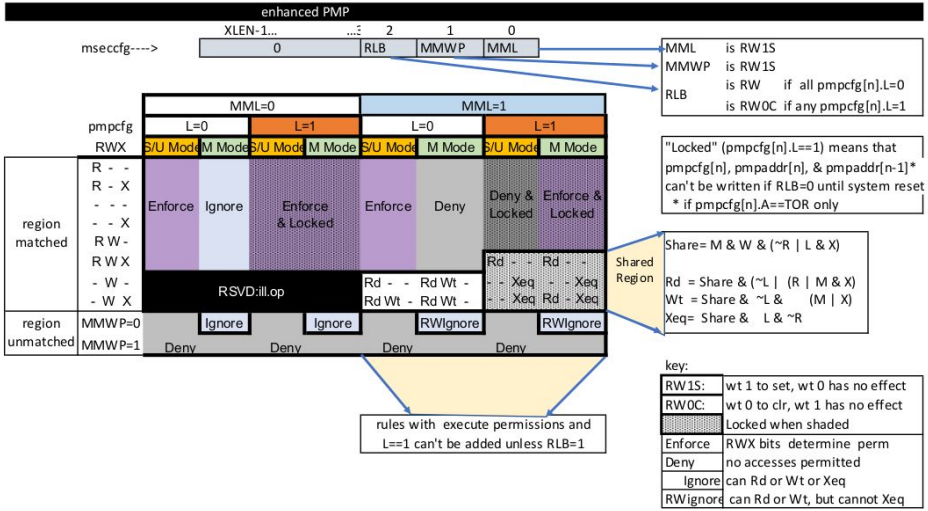
- Isolation between S and U mode and between tasks on U mode.
- Also used for isolation between guests (VS - VS), and between the guests and the host (HS - VS), using a 2nd translation stage.
- SMEP is always in place, there is no way for S mode to execute pages marked with the U bit.
- SMAP is on by default but can be disabled temporarily (through sstatus.SUM) so that S-mode can read/write data from U-mode on specific code paths (e.g. copy_to/from_user() on Linux).



X	W	R	Meaning
0	0	0	Pointer to next level of page table.
0	0	1	Read-only page.
0	1	0	*Reserved for future use.*
0	1	1	Read-write page.
1	0	0	Execute-only page.
1	0	1	Read-execute page.
1	1	0	*Reserved for future use.*
1	1	1	Read-write-execute page.

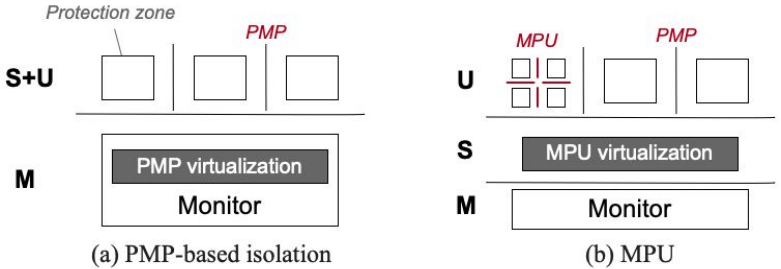
Enhanced Physical Memory Protection (ePMP)

- Locked rules that apply only to M mode.
- Access/execution prevention from M-mode to S/U-mode.
- Ability to switch policy from blacklist to whitelist for M-mode.
- Ability to prevent adding new executable regions on M-mode.
- Shared regions with reduced privileges between M-mode and S/U-modes.
- Allow for greater flexibility to support more use cases.
- Specification Ratified in Nov 2021



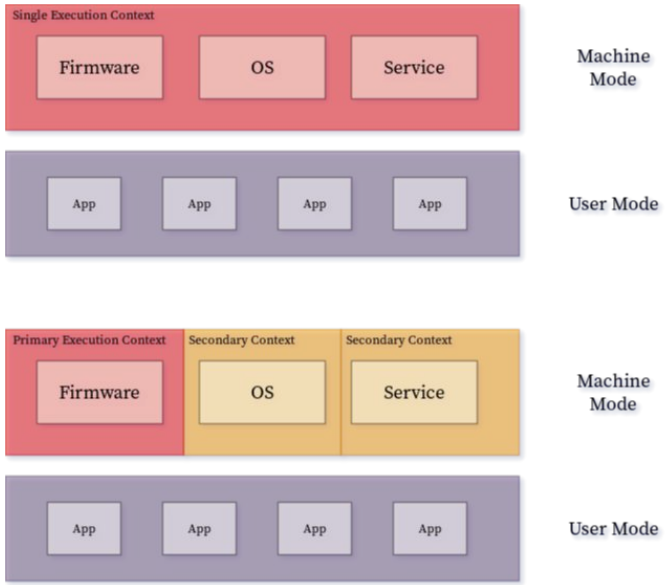
Memory Protection Unit (MPU)

- May be used instead of the MMU on S/HS/VS-modes.
- Similar encodings to ePMP.
- Fast switching between sets of rules.
- Useful for:
  - Supporting small trusted hypervisors on HS-mode (VS to VS and HS to VS isolation)
  - TEEs on S/U/VS-mode
  - Small IoT devices without MMU
- Under development, goal is to freeze by Q1 2022.



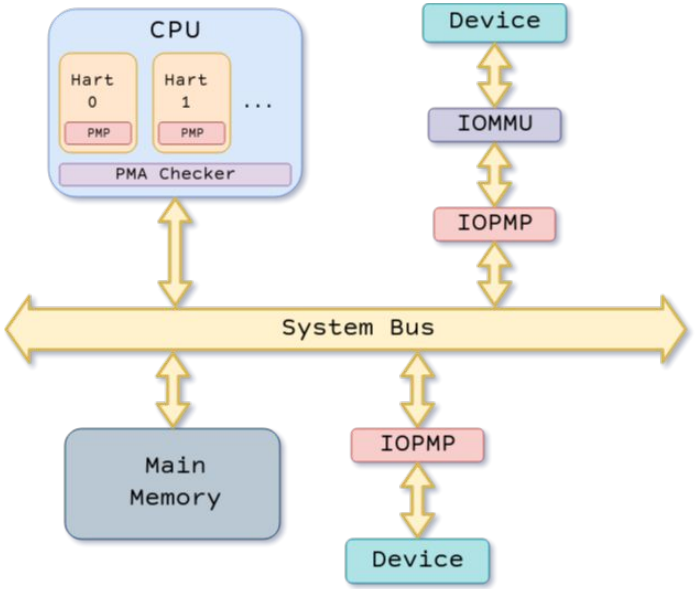
Lightweight TEE

- Memory isolation scheme for small M/U systems.
- Adds a secondary context to M mode so that the Firmware running on primary context (regular M mode, backwards compatible), can context switch between other elements running on M mode.
- Extends ePMP to allow M-mode rules for the secondary context, to be modified by the primary context.
- Most common case: Isolate Firmware from an OS running on M mode (since there is no S mode for it).
- Under development



I/O Physical Memory Protection (IOPMP)

- System level component to prevent malicious memory access from memory requesters (e.g. DMA) in the system.
- A critical mechanism to define system-wide security domains for memory isolation.
- Optionally context-aware.
- Under development, goal is to freeze by Q2 2022.



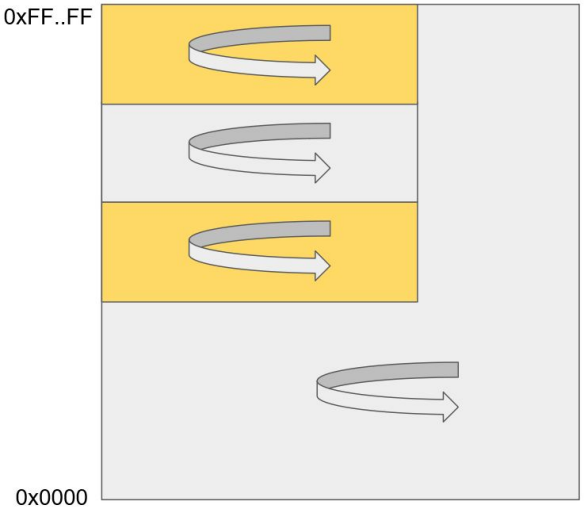
RISC-V®

Other mechanisms

Pointer Masking

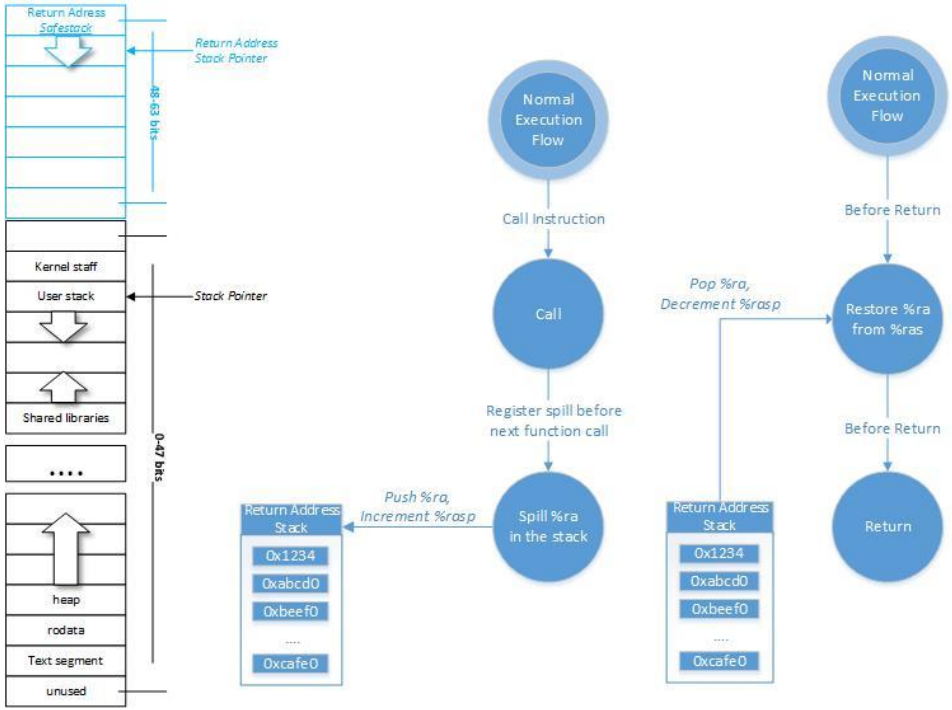
actual_address = (requested_address & ~mpmmask) | mpmbase

- Bits on the mask are ignored by the hardware and replaced by the contents of mpmbase. Works for both physical and virtual addresses.
- Can be used for software-based memory tagging, by using the masked bits of the address for tags and checking them on software.
- Can also be used as a simple memory protection mechanism, by allowing code to restrict its allowed memory range (e.g. restrict a library to only access code / data within an address range, without being able to access the rest of the task's memory).
- Each privilege mode has own copy of pointer masking CSR register. It appears as the *mpmmask*, *spmmask*, *vspmmask* and *upmmask* registers in the M-mode, HS/S-mode, VS-mode and (V)U-mode ISAs, respectively.
- Each privilege mode has its own copy of pointer base CSR register. It appears as the *mpmbase*, *spmbase*, *vspmbase* and *upmbase* registers in the M-mode, HS/S-mode, VS-mode and (V)U-mode ISAs, respectively.
- On its way to public review, likely by Q2 2022



Control Flow Integrity (CFI)

- New CFI execution mode
- Shadow stack to protect from ROP attacks, architecturally protected.
- Forward-edge protection using labeled landing points.
- In early stages of development.



Other proposals in progress...

- **FENCE.T / SEC.FLUSH**: Instructions to flush microarchitectural state, to mitigate spectre-like side channels.
- **Secure function calling**: Enter/Exit an isolated environment on S/U, without going through the secure monitor on M mode.


Thanks !
RISC-V®