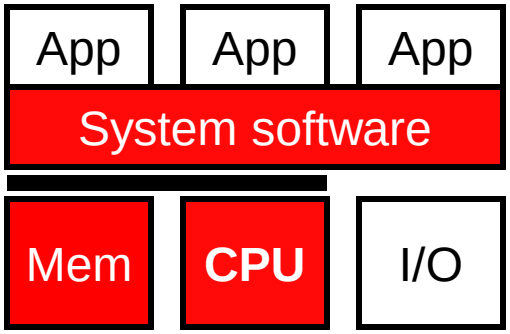
CSE 560
Computer Systems Architecture

Security

Slides originally developed by Justin Deters (Wash U)

This Unit: Security



JumpSwitches for Indirect Branches

Trusted Execution Environment

- ARM
- AMD
- Intel
- RISC-V

Hardware Design Choices

- Speculative Execution → Good
- Data Prefetching → Good
- Speculative Execution + Data Prefetching → Bad

- Design choices can interact with unintended consequences

- Need to design hardware with security in mind

JumpSwitches Talk

- Retpolines
  - Serves as the state-of-the-art defense, effectively disabling speculative execution for indirect branches
  - 20% penalty on some workloads

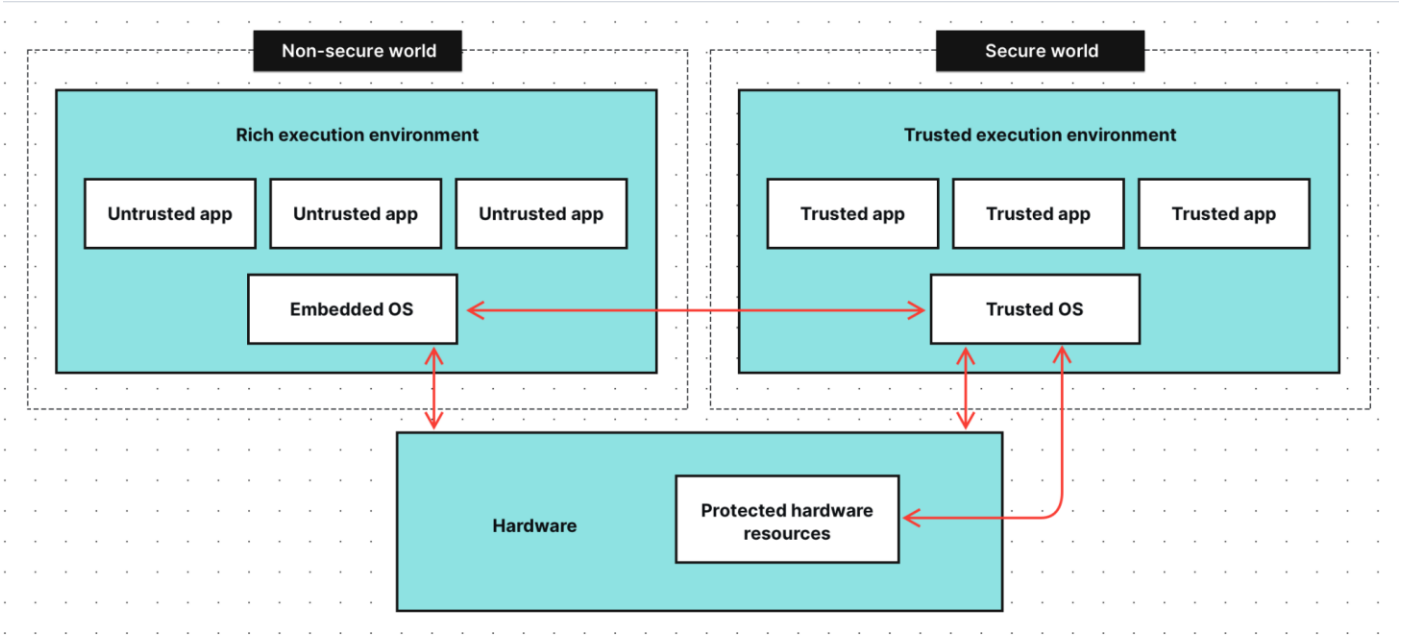
- JumpSwitches
  - Enables speculative execution of indirect branches on safe targets
  - Leverages indirect call promotion, transforming indirect calls into direct calls
  - Learn targets at runtime and perform just-in-time promotion without the overhead of binary translation

- USENIX ATC ’19
- JumpSwitches: Restoring the Performance of Indirect Branches In the Era of Spectre
  - https://www.usenix.org/conference/atc19/presentation/amit

Trusted Execution Environment

- Assume that the whole chip is compromised
- Add a Trusted Execution Environment (TEE) to the chip
  - Ensure hardware is not vulnerable to known threats
    - E.g., in-order execution
  - Take measures to diminish chances for software vulnerabilities
    - E.g., keep functionality as low as possible
- Run security sensitive tasks in the TEE
  - Encryption/decryption
  - Storing secure keys
  - Etc.

Trusted Execution Environment



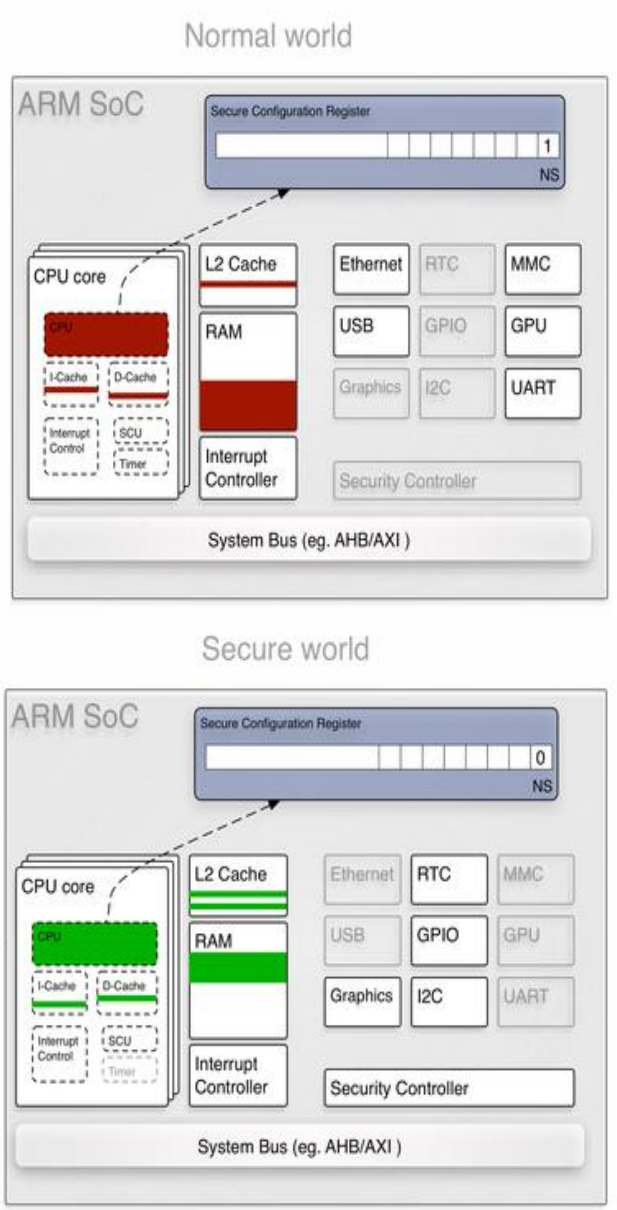
- Two OS stacks with separate memory
- Code split into trusted and non-trusted regions
- CPU core switches between secure and non-secure modes
- Strictly limited communication between the worlds

TEEs are common in industry

- ARM → TrustZone
- AMD → Platform Security Processor
- Intel → Trusted Execution Technology & SGX Software Guard Extensions
- RISC-V → Keystone
  - Open source!

ARM TrustZone

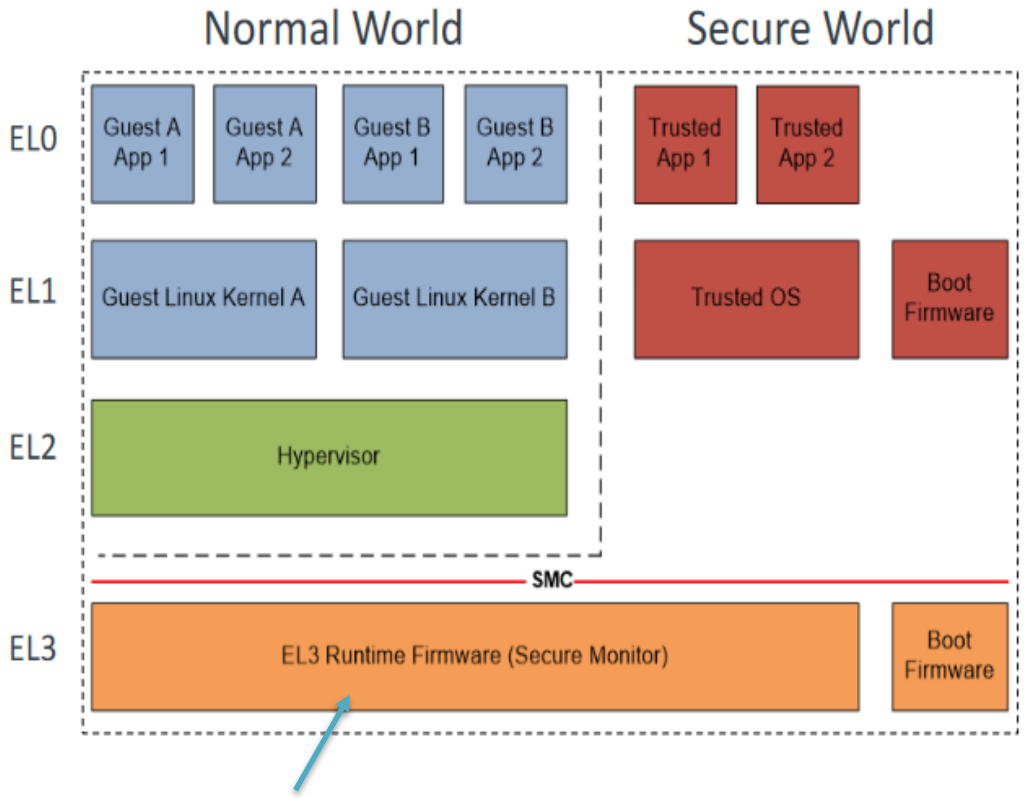
- CPU has a secure mode
  - OS cannot see this mode
  - Firmware responsible for changing modes
  - Normal World vs. Secure World
- Software has different view of hardware in secure world
- Change in security state propagates over system bus



https://genode.org/documentation/articles/trustzone

Example System

- Software must request to be moved to secure world
  - Used hardware exceptions to initiate this process
  - Firmware is in charge of preserving software state between transitions

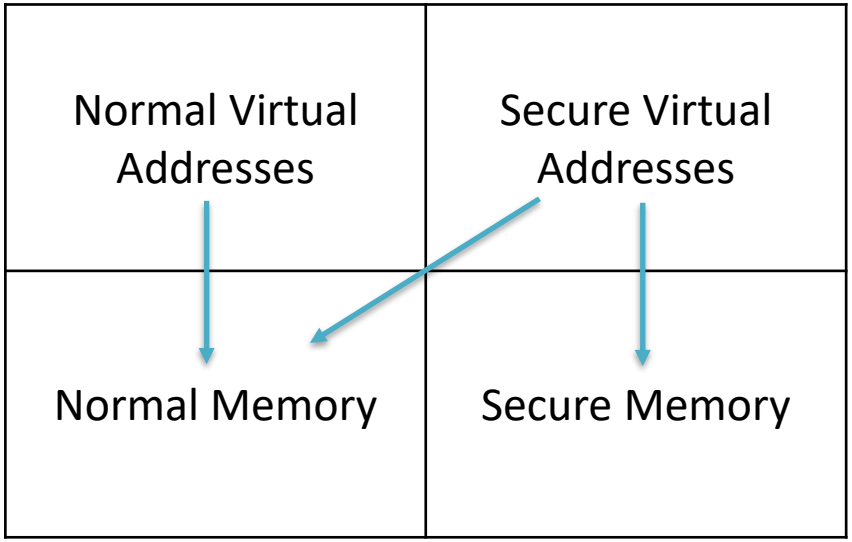


Handles moving software from Normal to Secure World

https://elinux.org/images/0/05/Elc-tfa.pdf

Virtual Memory

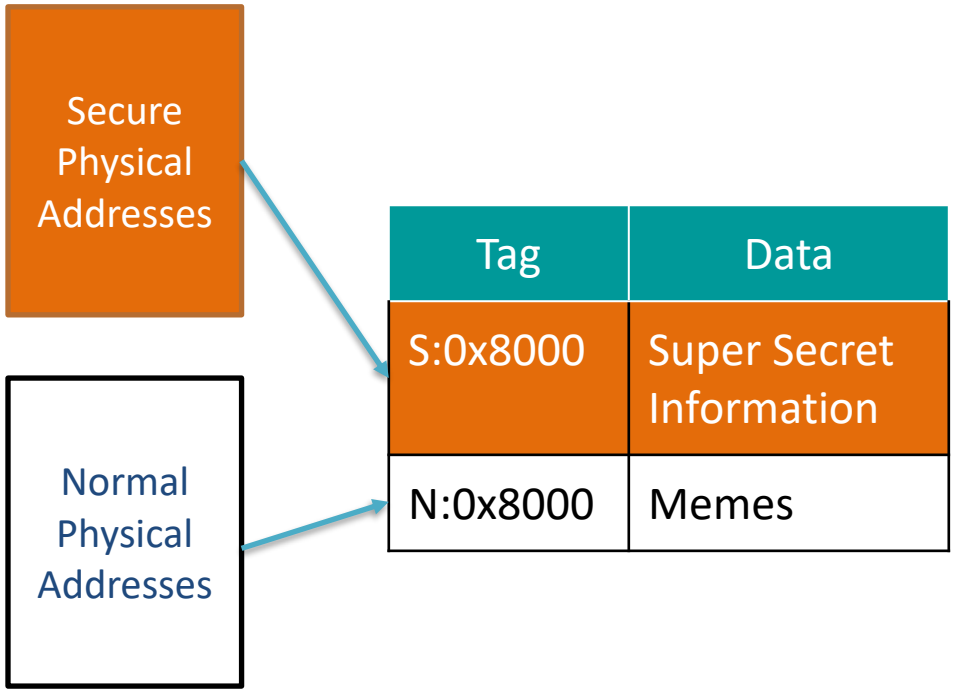
- Separate *virtual addresses* for Normal and Secure addresses
- Separate *physical addresses* for Normal and Secure memory
  - Normal Mode can only see Normal memory
  - Secure Mode can see both Normal and Secure memory



Caches

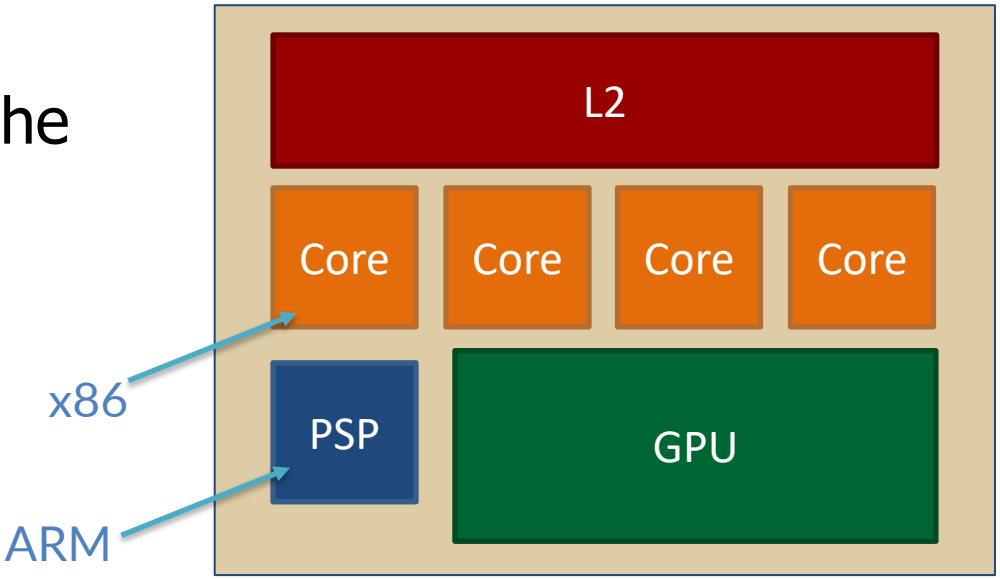
- Caches are tagged with *physical addresses*

- Only secure applications can hit a cache line tagged with a secure physical address



AMD: Platform Security Processor

- Embeds a 32-Bit ARM microcontroller
  - Isolated ROM and SRAM
  - Cryptographic Processor
  - Two different ISAs on the same chip!
- Microcontroller manages the security of the processor.
- Communicates with processor via interrupts

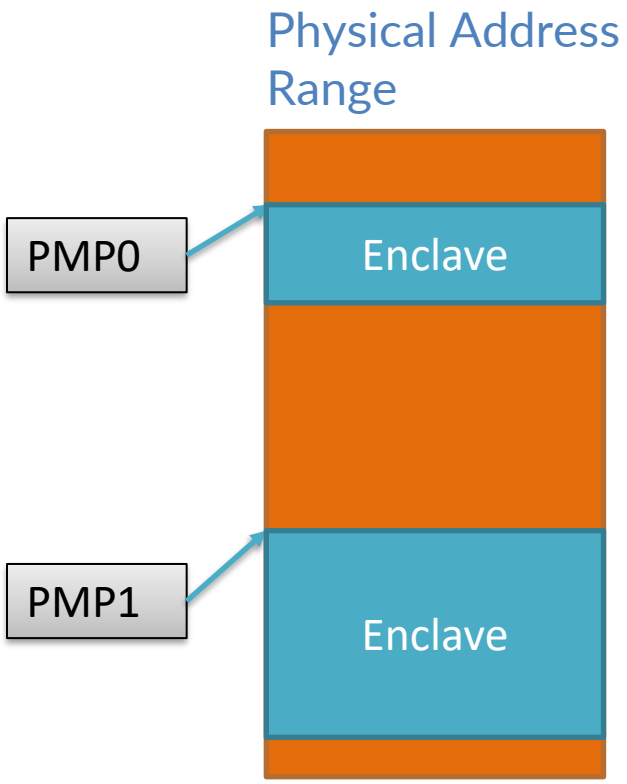


RISC-V: Keystone

- TEE for RISC-V
- RISC-V
  - Open source ISA
  - Community gets to decide what goes in
  - Many chip implementations are also open source
  - Some industry adoption
    - Western Digital
    - SiFive
    - Google → Titan M2 in Pixel 6
- Keystone uses ISA features of RISC-V!

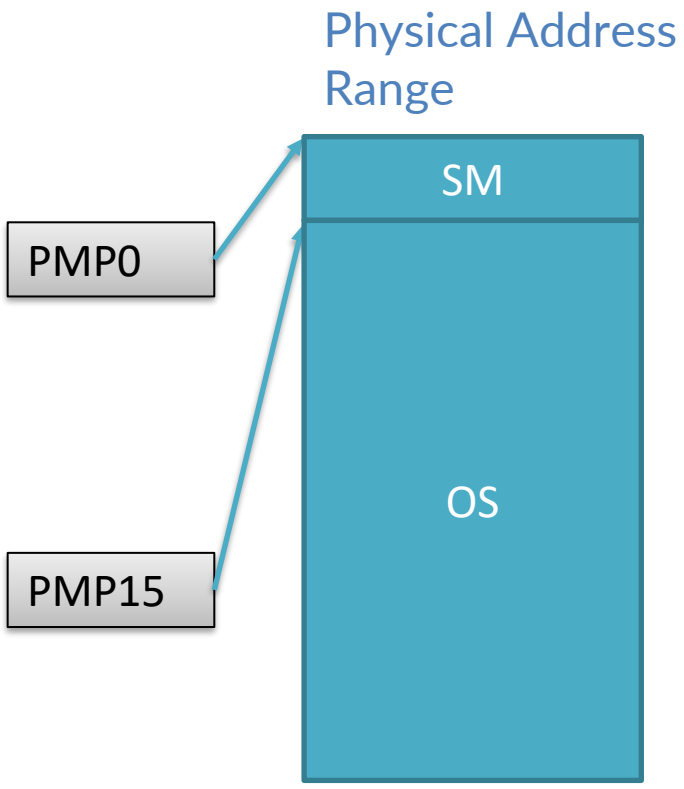
RISC-V Physical Memory Protection

- New ISA feature
  - Each core has its own set of Physical Memory Protection (PMP) registers
  - Configuration of registers done through the ISA
- PMP controls User Mode and Supervisor Mode *physical memory* access permissions
- PMP is used to create enclaves, which are secure environments with access to their own protected memory region, isolated from the rest of the system
- While in an enclave, if the *requested physical memory* address is outside the locked range → Denied
  - Can also allow for read, write, and execute granularity of the regions
- Give permission for a *contiguous* address range of a certain size



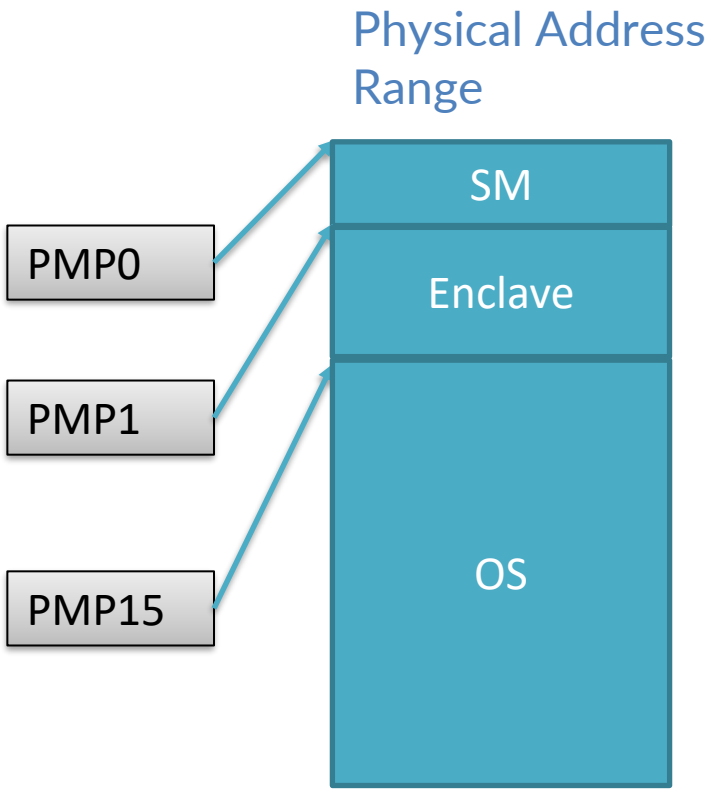
Setting up Keystone

- PMP registers are statically prioritized
  - 0 → highest priority
  - 15 → lowest priority
- Keystone creates a PMP entry for the Security Manager (SM) at the highest priority
- Keystone creates a PMP entry for the whole address range at the lowest priority
- SM launched for each core in the system



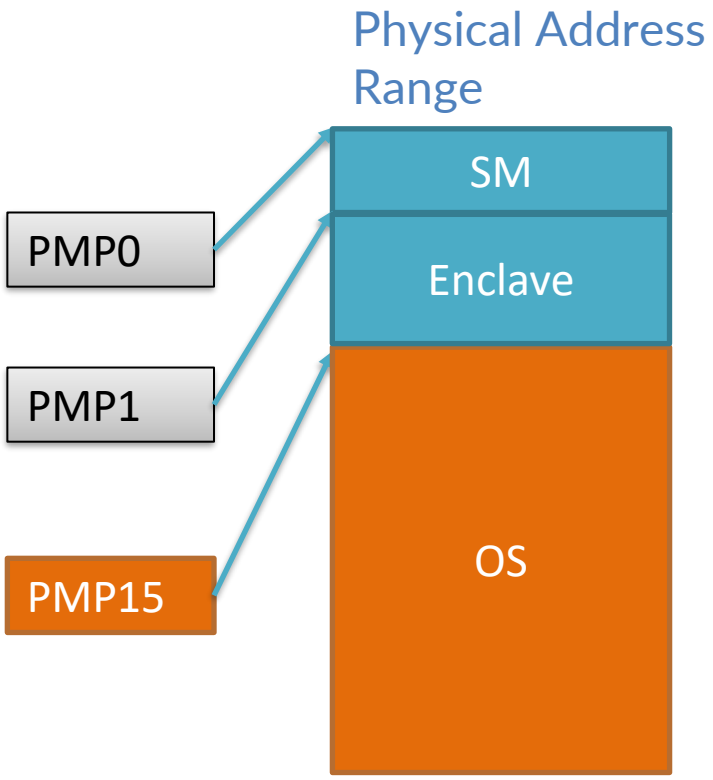
Creating an Enclave

- OS finds a free contiguous physical memory range → Calls the SM
- SM adds PMP entry
  - Higher priority than OS and user processes
- Enclave regions cannot overlap with each other or the SM



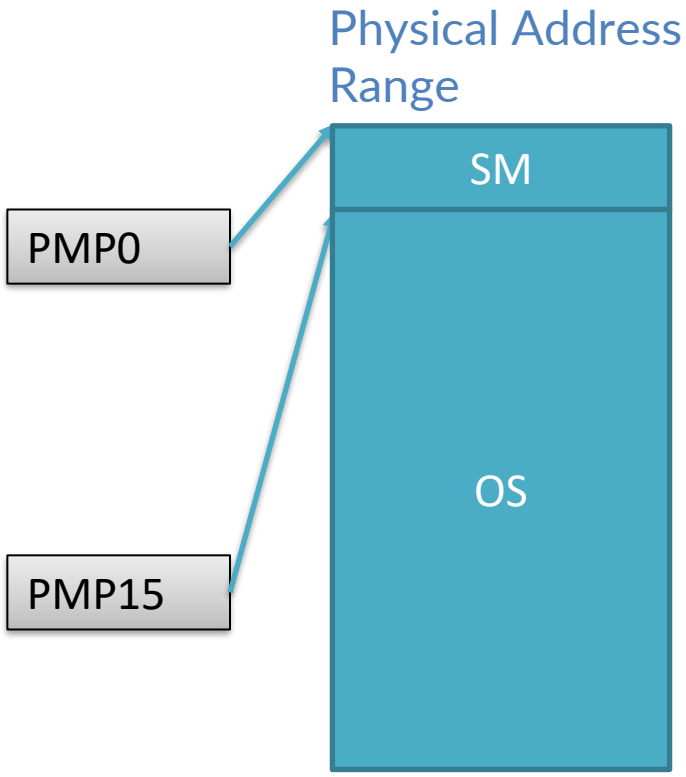
Control Transfer

- SM enables the permission bits for the enclave PMP entry
- SM removes permission for the OS PMP entry
- Enclaves can only access itself
- Nobody can access the enclave



Destruction

- SM disables all permission for the enclave
- Clears the memory of the enclave
- Gives memory back to OS
- Re-enables the OS PMP entry
- PMP for the enclave is freed
- If the OS cannot interact, how is memory managed?



Memory Management

- Keystone manages the virtual memory *in the enclave*
- Each enclave has its own page table
- Only the enclave knows its own virtual-to-physical mapping
- The OS only knows what contiguous memory range used
- Does this solve Spectre and Meltdown?

Can We Trust the Trusted Execution Environment?

- Who watches the Watchmen?
- Open source software relies on code audits and the community to find and fix security flaws
- Vast majority of hardware designs are closed source
  - Nobody can audit them except for the companies themselves
- Many security flaws make it to market
  - Many exploits aren't discovered for years
- Bad Actors
  - Zero-Day exploits are sold by private entities to governments
  - Some governments have exploits built into processors
- Perfect security is impossible
  - You can only mitigate security risks