Massachusetts Institute of Technology

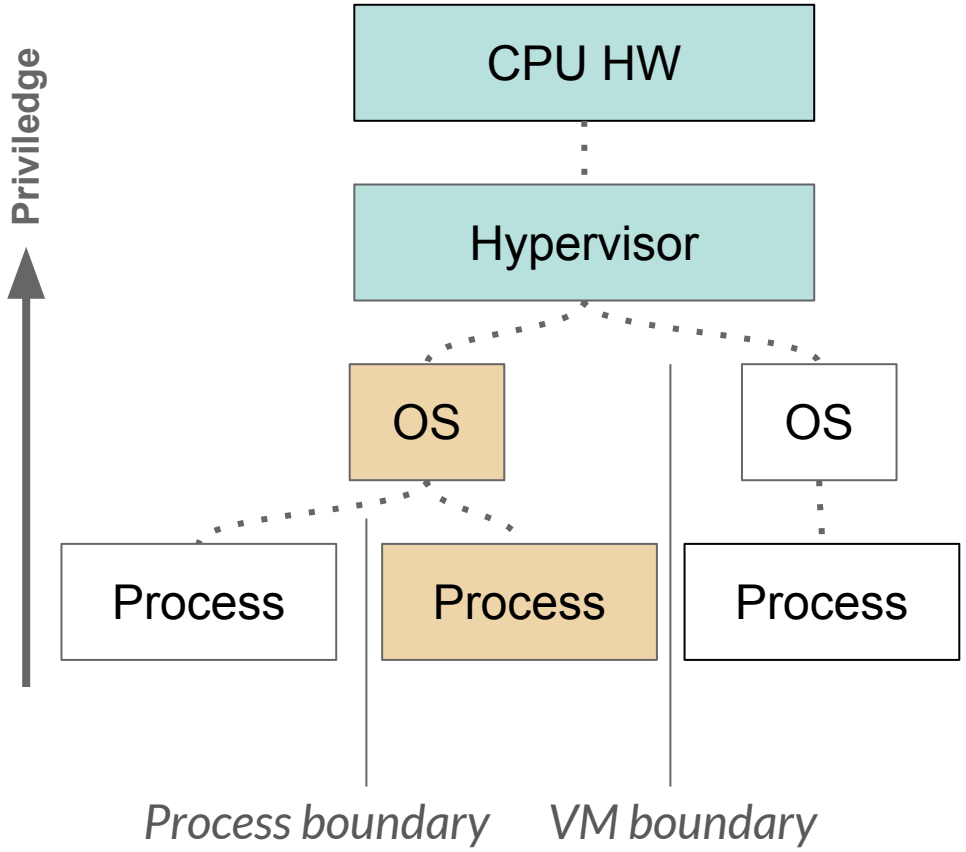
RISCV with Sanctum Enclaves

Victor Costan, Ilia Lebedev, Srini Devadas

Today, privilege implies trust (1/3)

If computing remotely, what is the TCB?

trusted computing base

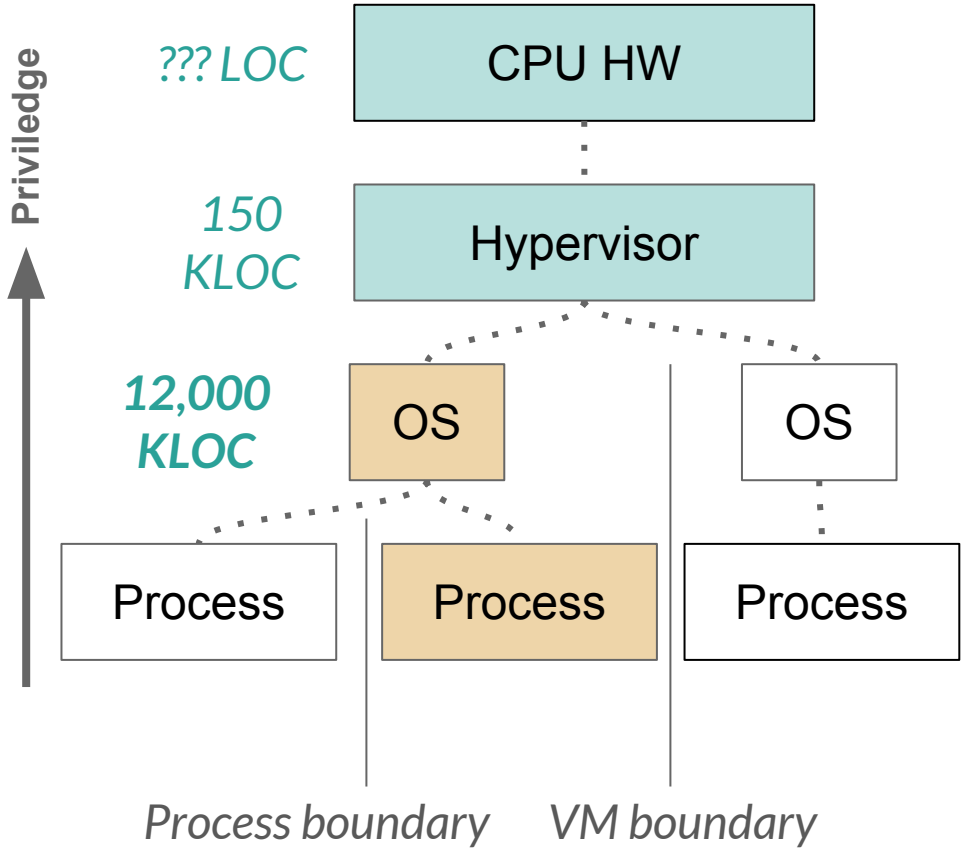


Today, privilege implies trust (2/3)

If computing remotely, what is the TCB?

trusted computing base

Sanctum **decouples** HW protection from trust!

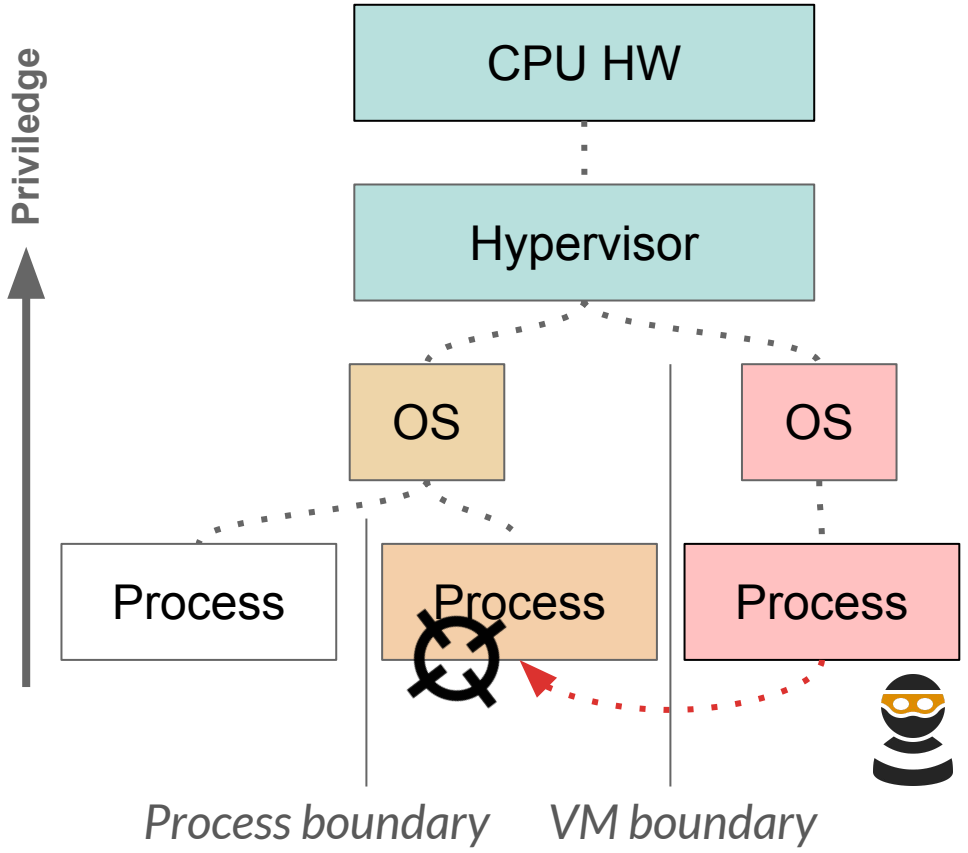


Today, privilege implies trust (3/3)

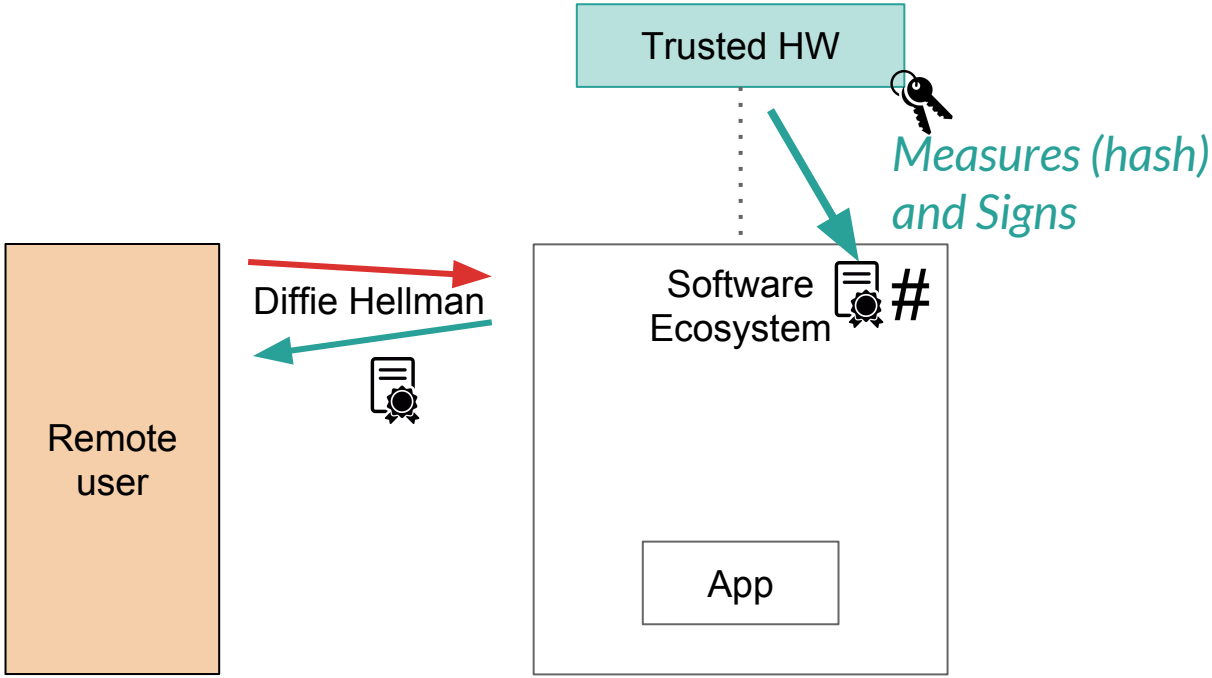
Side channels leak privacy

Sanctum uses
hardware-assisted isolation

Strong privacy & integrity
with low overhead

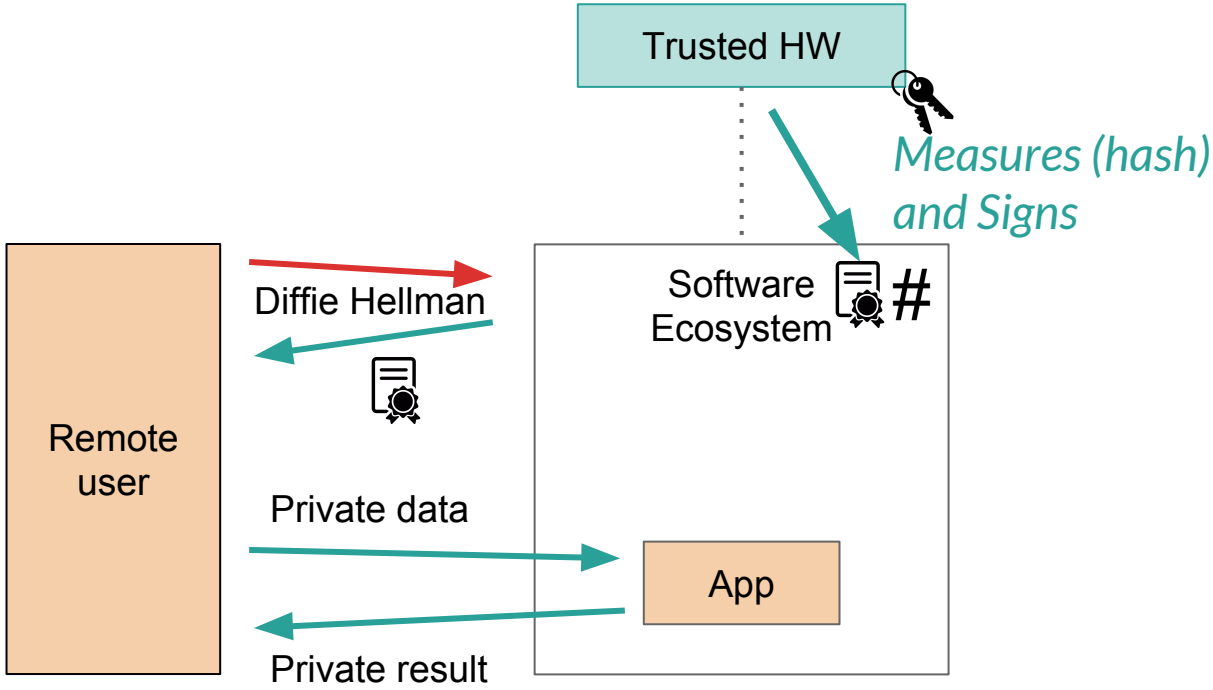


Remote Software Attestation (1 /2)



Trusted HW creates proof for remote user

Remote Software Attestation (2 /2)

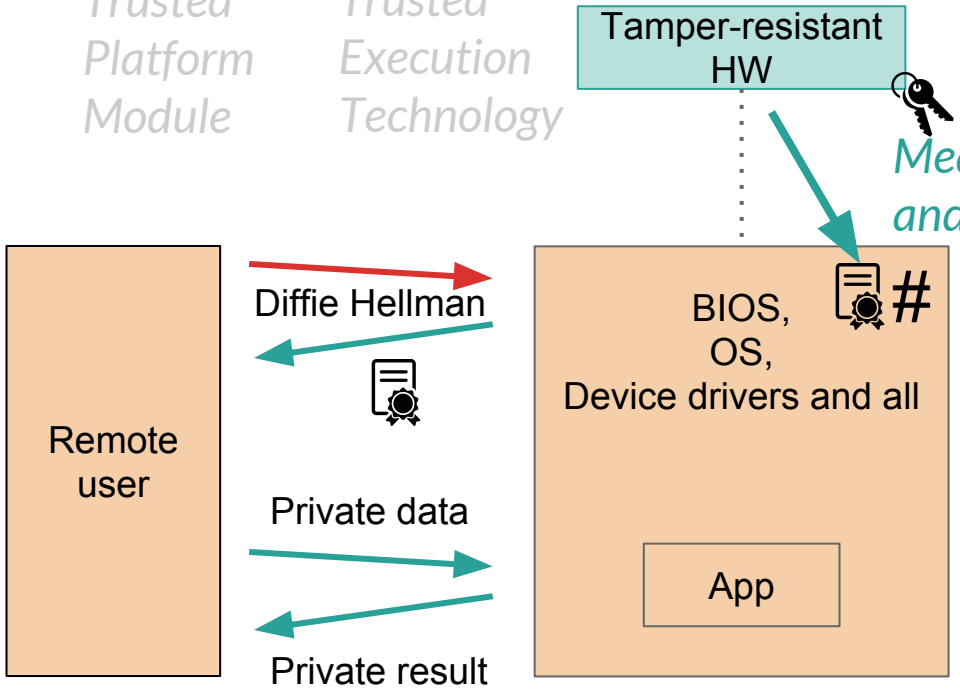


Remote user decides whether or not to trust certificate

Prior work: TPM, TXT

Trusted Platform Module

Trusted Execution Technology



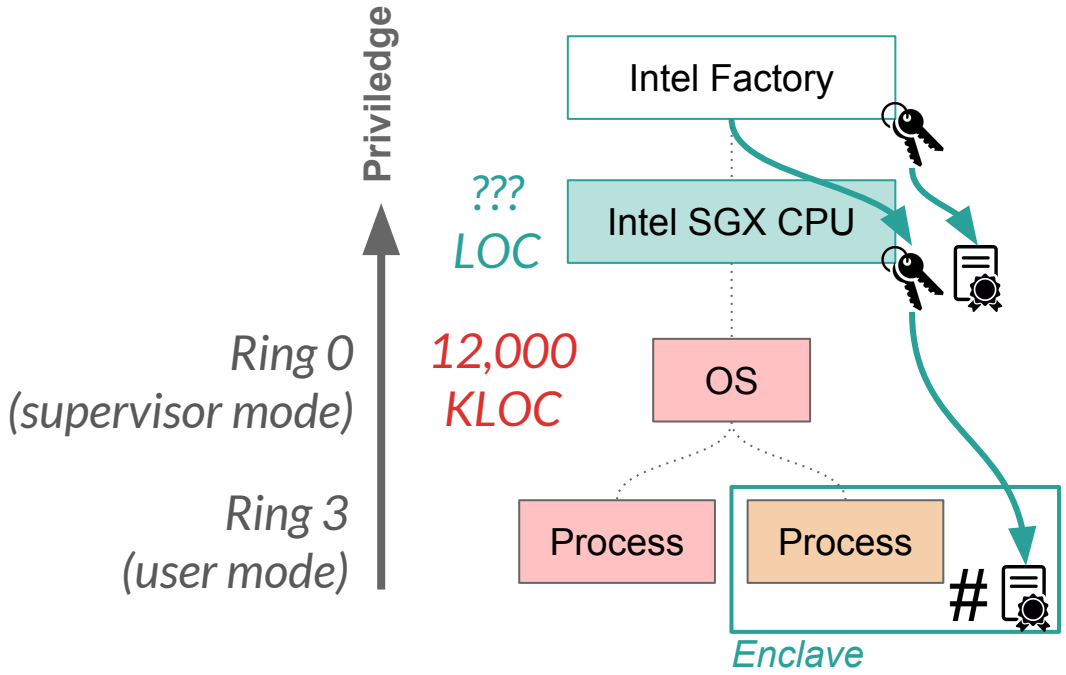
Trusted HW must keep its keys private!

Software ecosystem must not be vulnerable

Prior work included too much SW in their attestation

12,000+ KLOC

Recent “Success”: Intel SGX



Enclave ≔ immutable commitment of resources to one process.

Threat Model (Intel SGX)

Protect privacy* and integrity** of an **enclave** against

a privileged SW adversary (OS/Hypervisor)

SGX prevents:

Directly reading/ tampering with:

- Config. registers
- Enclave memory
- Enclave process structures
- DMA-capable devices

Some specific physical attacks: ← Physical access!

- DRAM contents
- Device secret

**Indirect observation?*

***Can SGX keep its keys private?*

Threat Model (Sanctum)

Protect privacy and integrity of an **enclave** against

a privileged SW adversary (OS/Hypervisor)

Sanctum prevents:

Directly read/ tamper with:

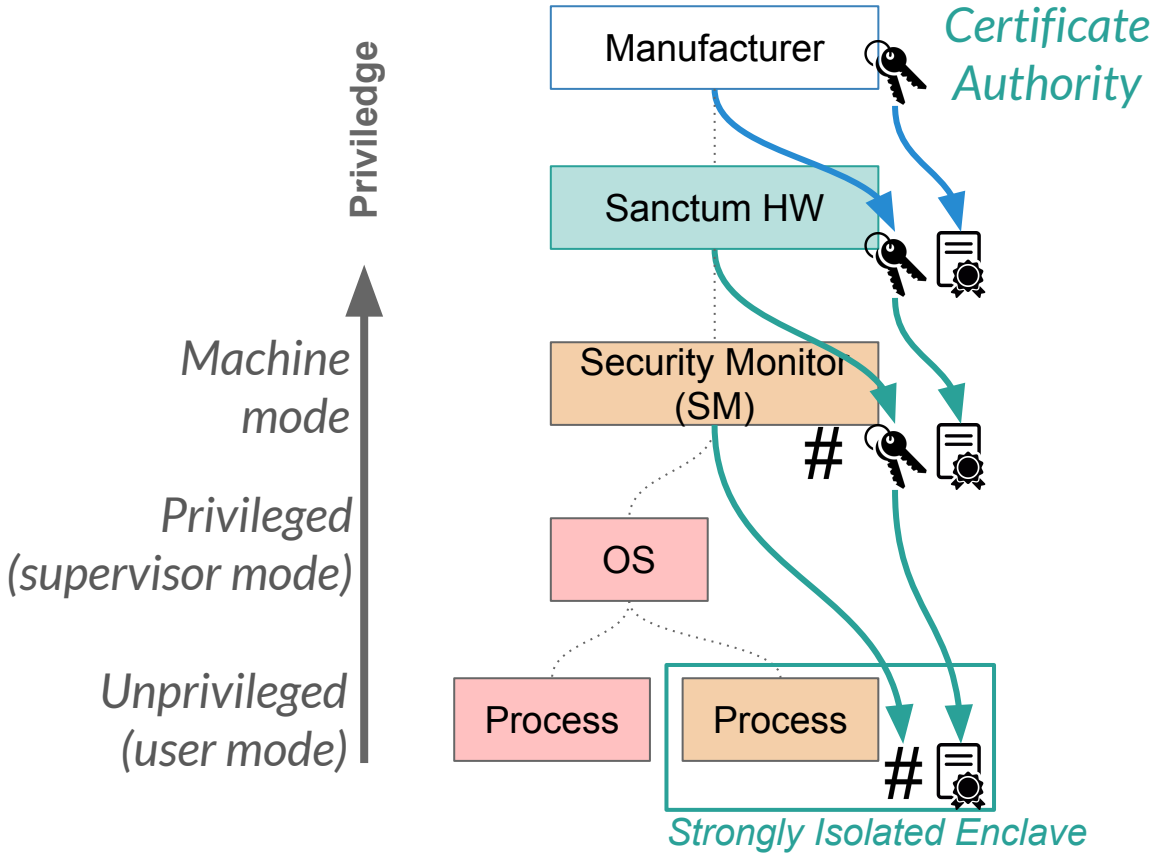
- Config. registers
- Enclave memory
- Enclave process structures
- DMA-capable devices

Indirectly* observe private state via shared:

- Caches
- Microarchitectural state
- Data structures managed by the OS
- Interrupts / Faults

No protection against physical access or fault injection

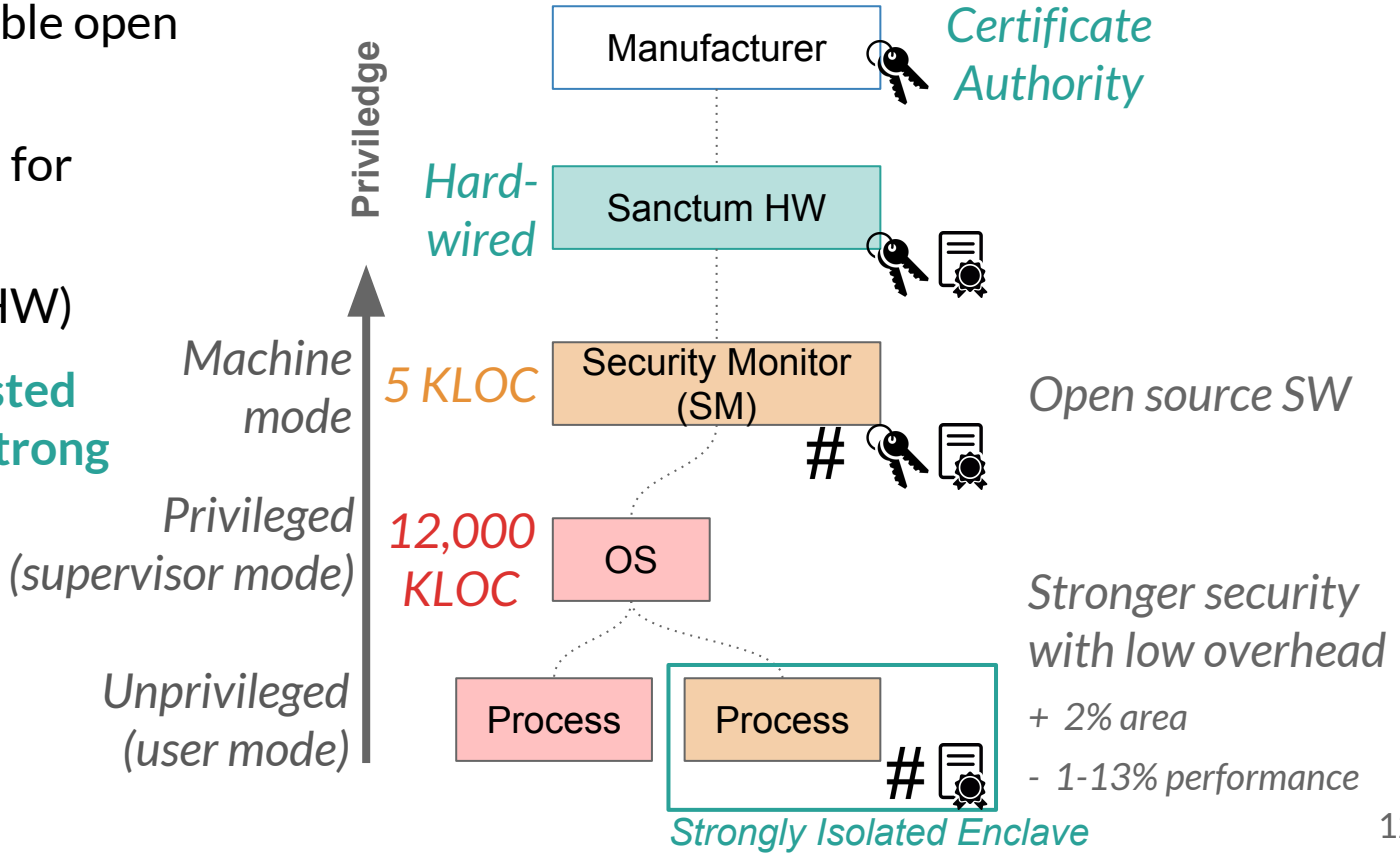
Sanctum's Chain of Trust



Sanctum's Contributions

- Formally verifiable open source SM
- Truly small TCB for attestation

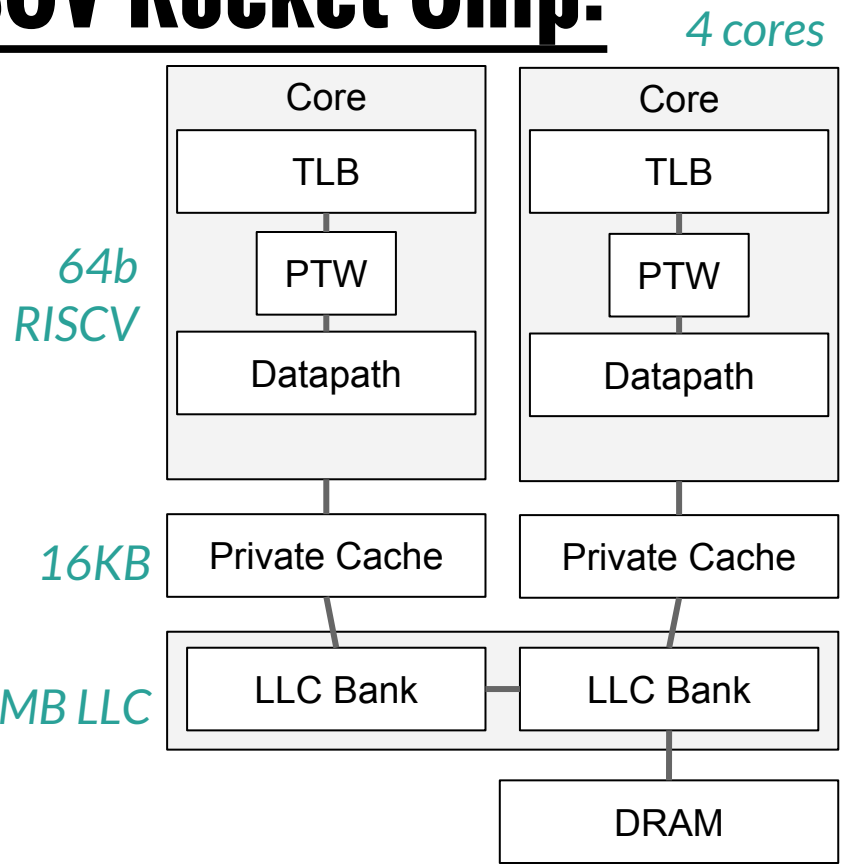
  (small SW and HW)
- **Hardware-assisted isolation with strong guarantees**



Sanctum modifies a RISCV Rocket Chip:

Build hardware support for small, trusted software to enforce isolation guarantees for unprivileged software.

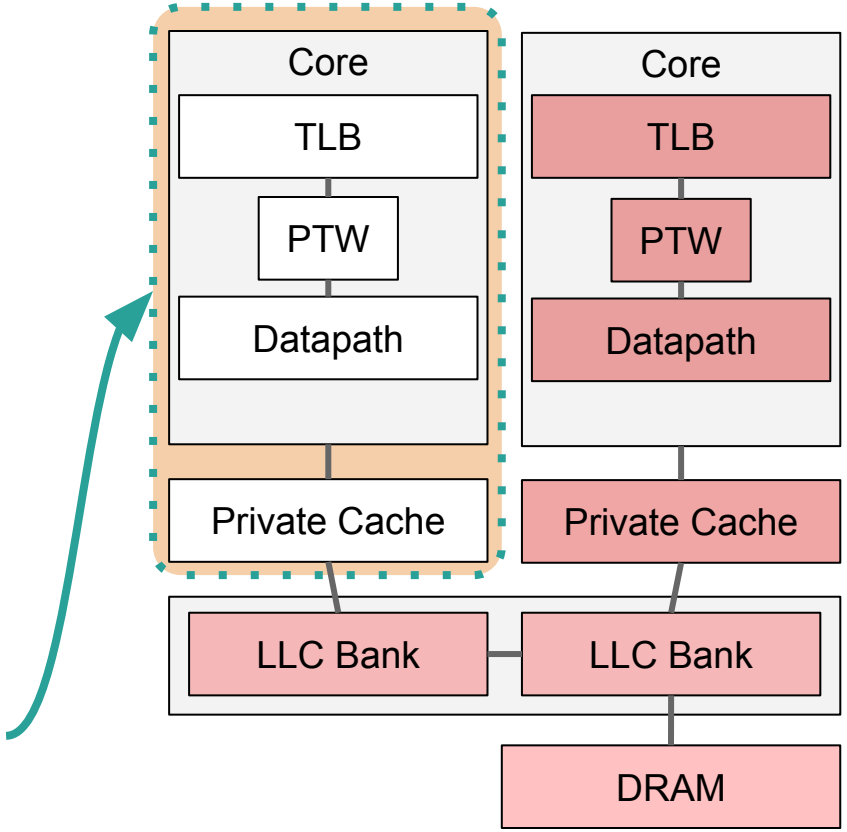
Explicitly multiplex between private and OS-controlled state



Enclaves execute on private cores

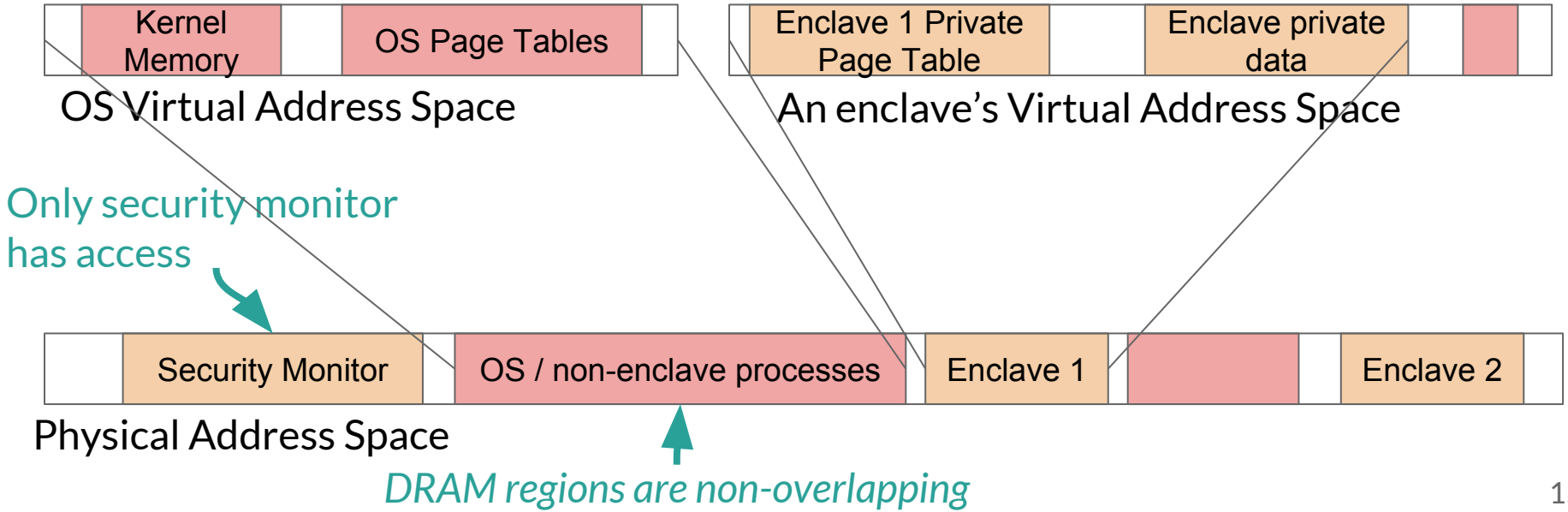
- Private L1 caches
- Private registers,
- Private branch target buffer
- Private TLB

Security monitor cleans up private state when allocating cores to enclaves / OS



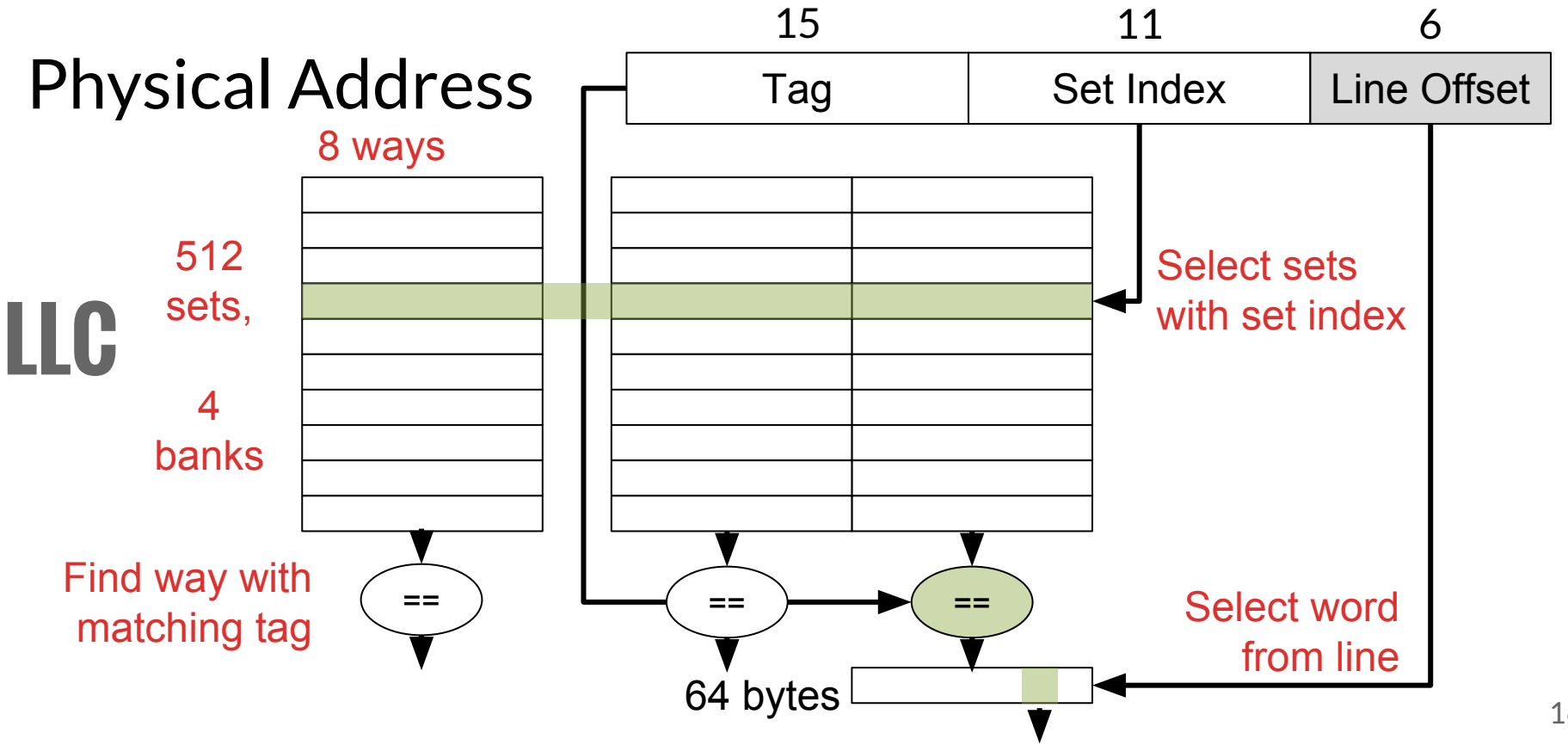
Isolated Physical Memory in Sanctum

"DRAM region" is a unit of isolated memory



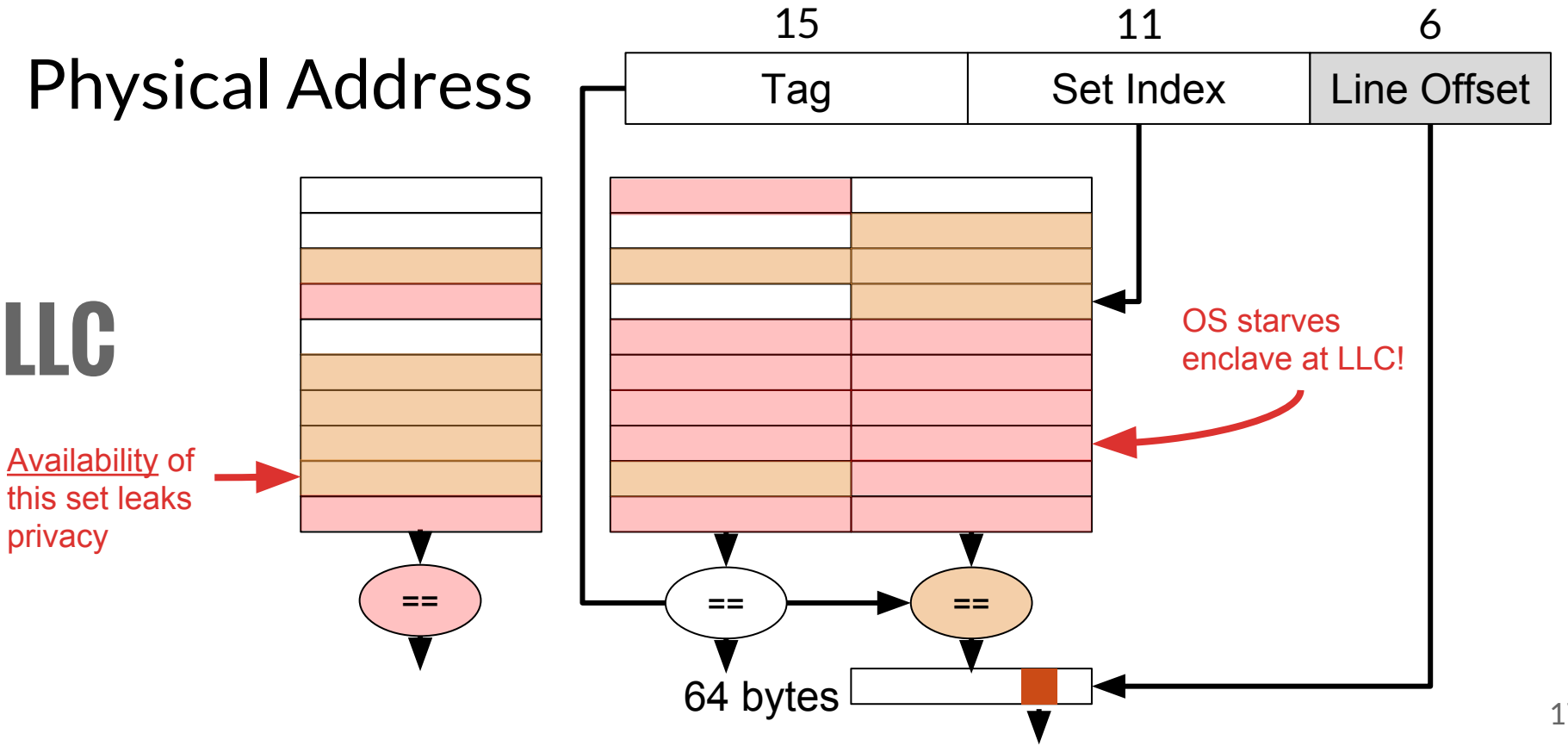
Isolating in the LLC (1/8)

Set associative (banked) cache



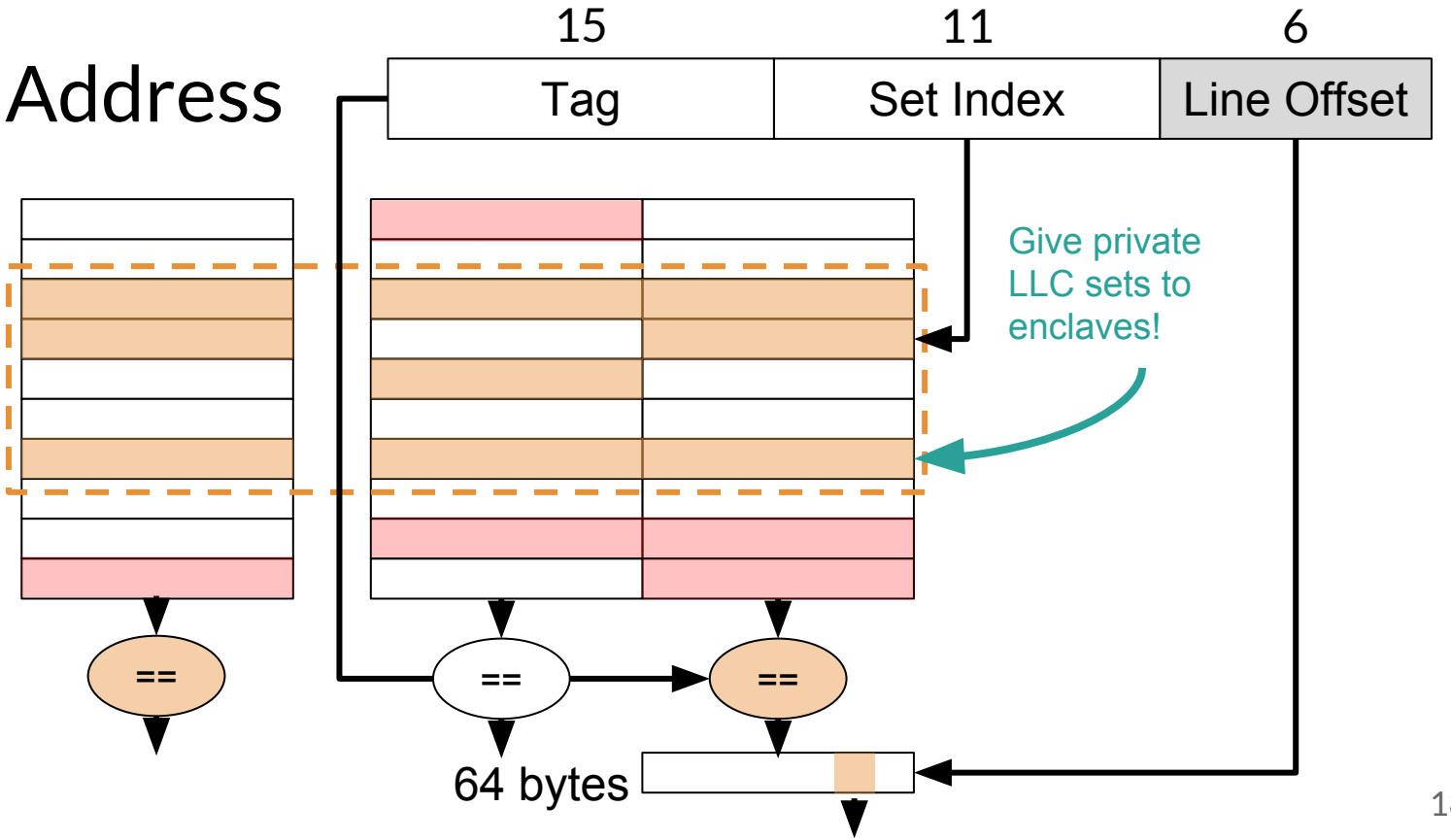
Isolating in the LLC (2/8)

LLC sharing leaks privacy!

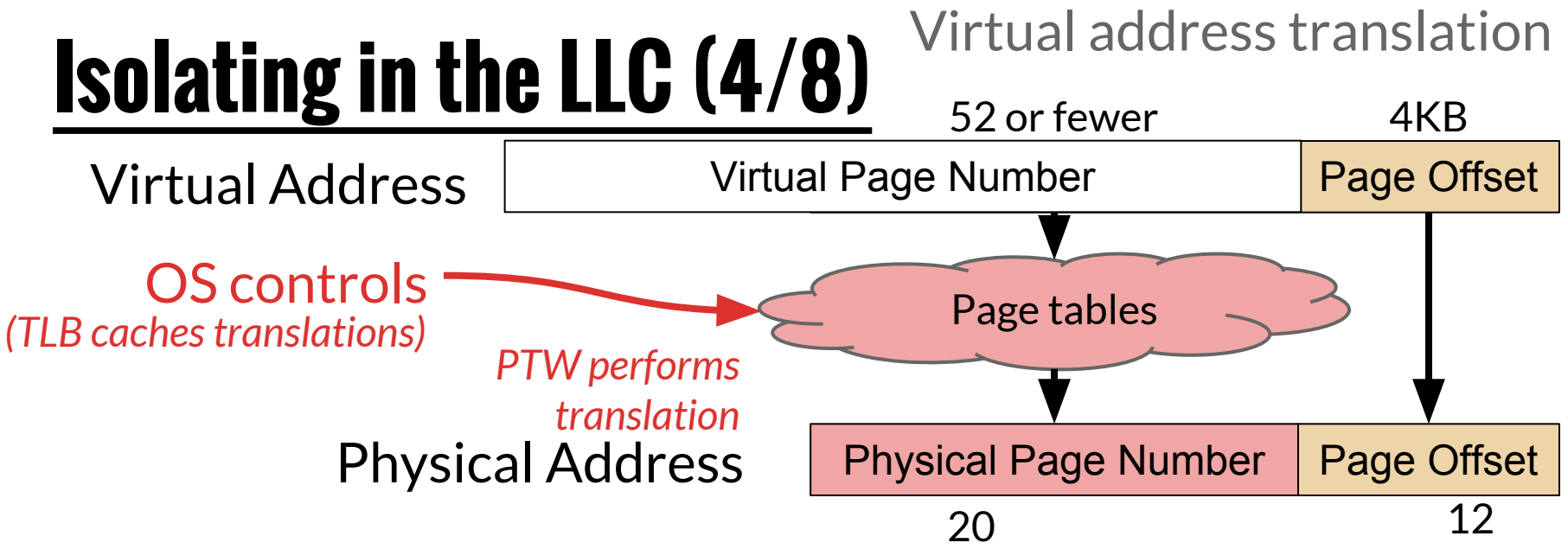


Isolating in the LLC (2/8)

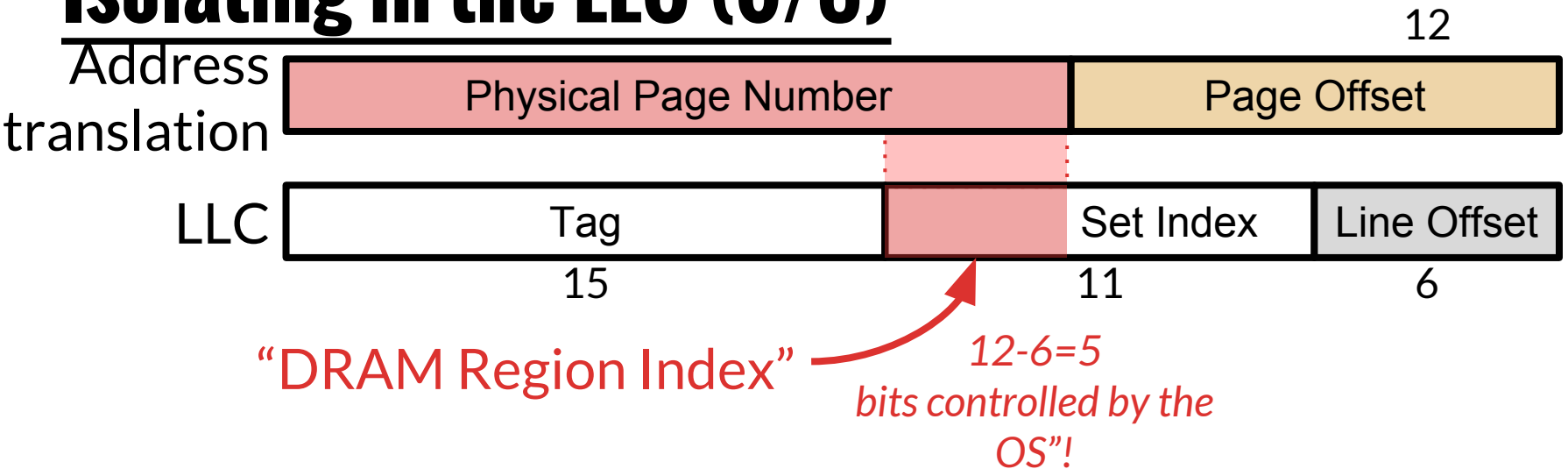
LLC sharing leaks privacy!



Isolating in the LLC (4/8)

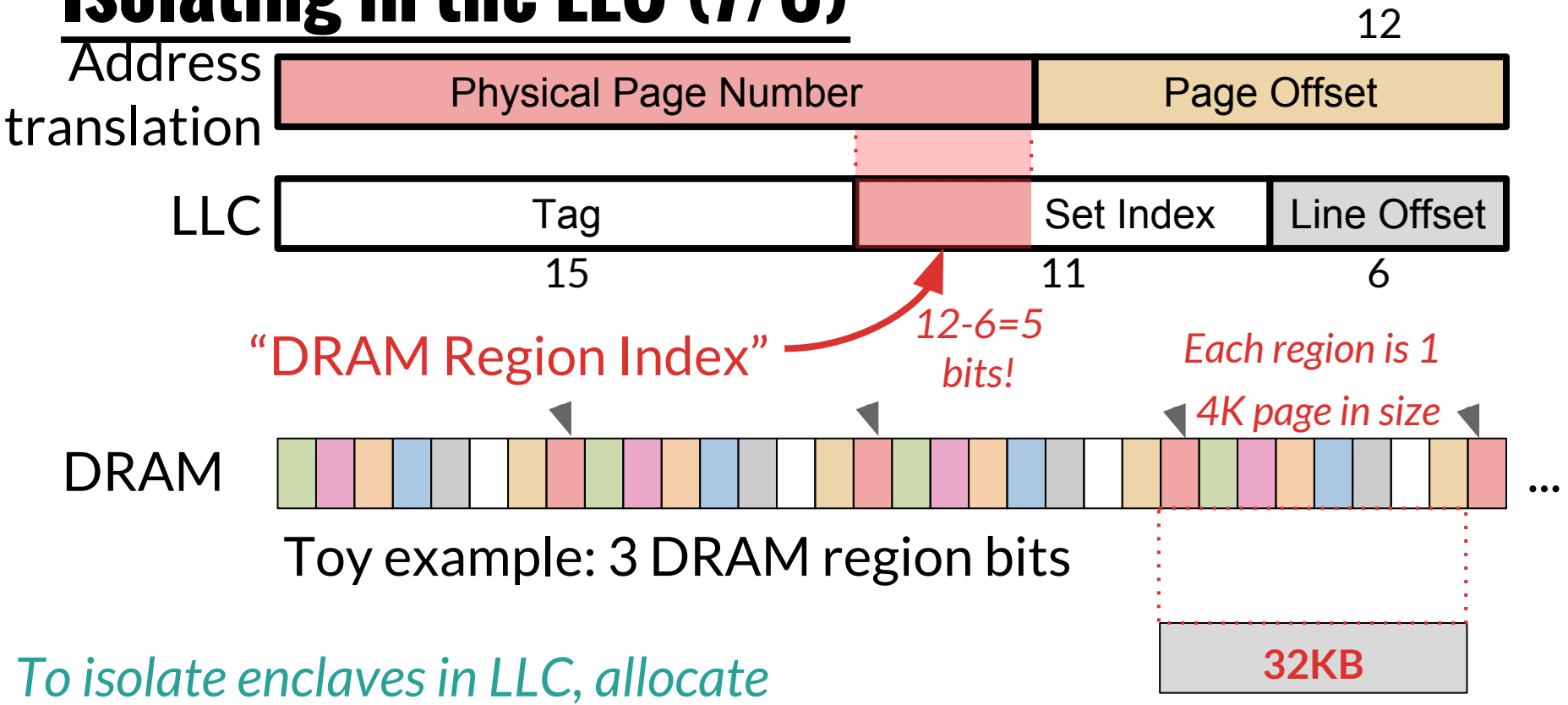


Isolating in the LLC (6/8)



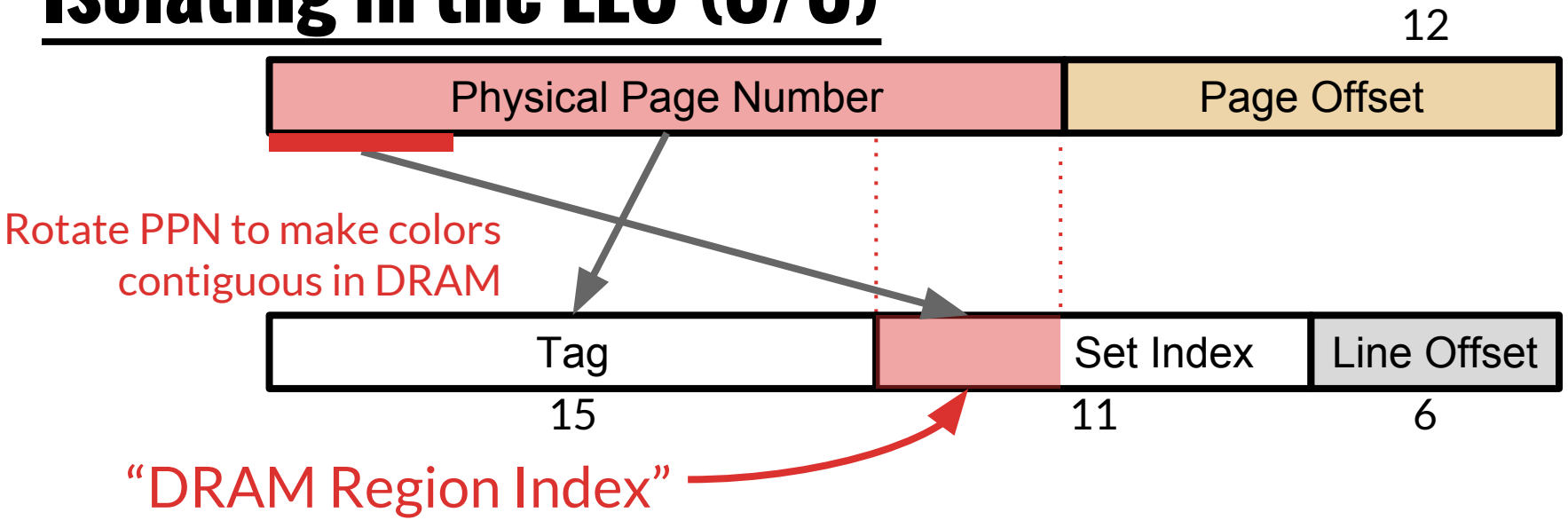
To isolate enclaves in LLC, allocate exclusively, at region granularity!

Isolating in the LLC (7/8)



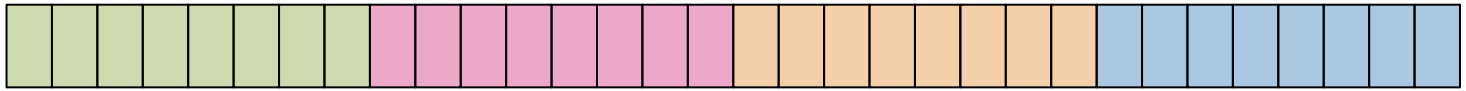
To isolate enclaves in LLC, allocate exclusively, at region granularity!

Isolating in the LLC (8/8)



Now top PA bits determine DRAM region

DRAM



...

Toy example: 3 DRAM region bits

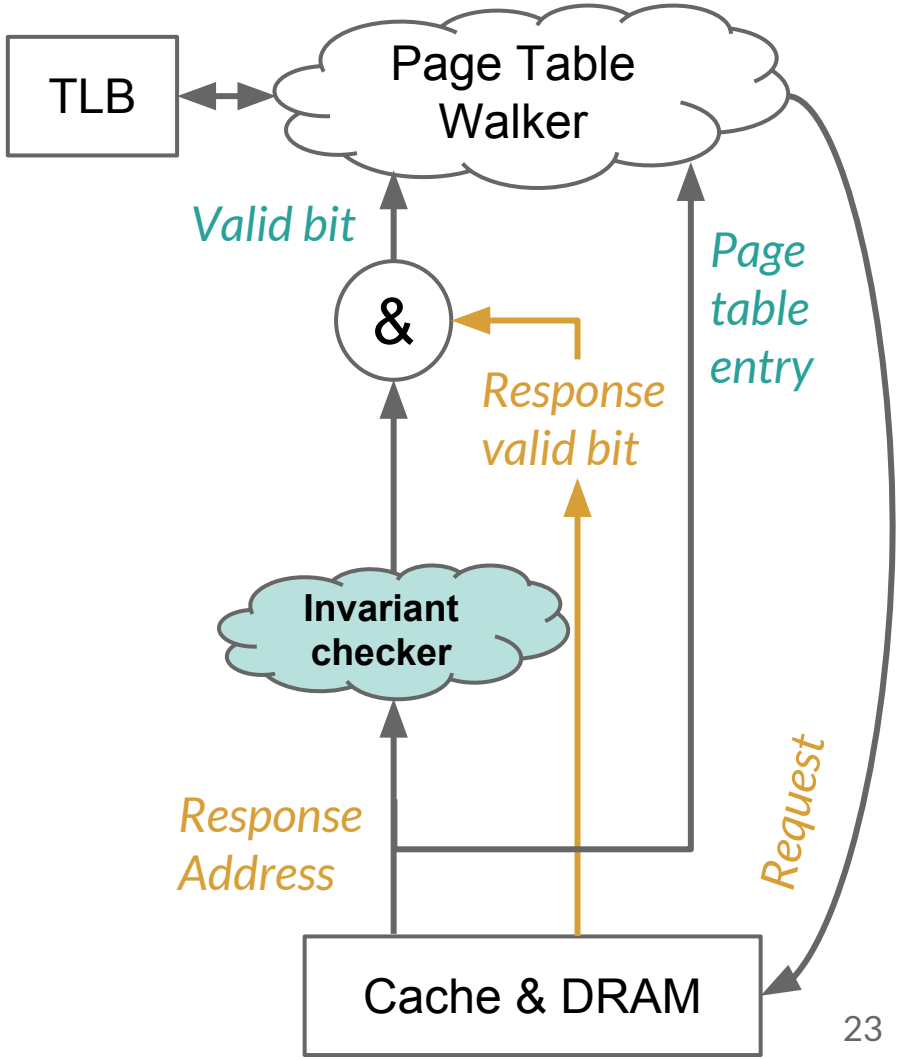
Each region is contiguous, 32KB

32KB

Hardware-assisted Isolation

Maintain an invariant:
TLB entries are safe!

HW enforces invariants at page walk



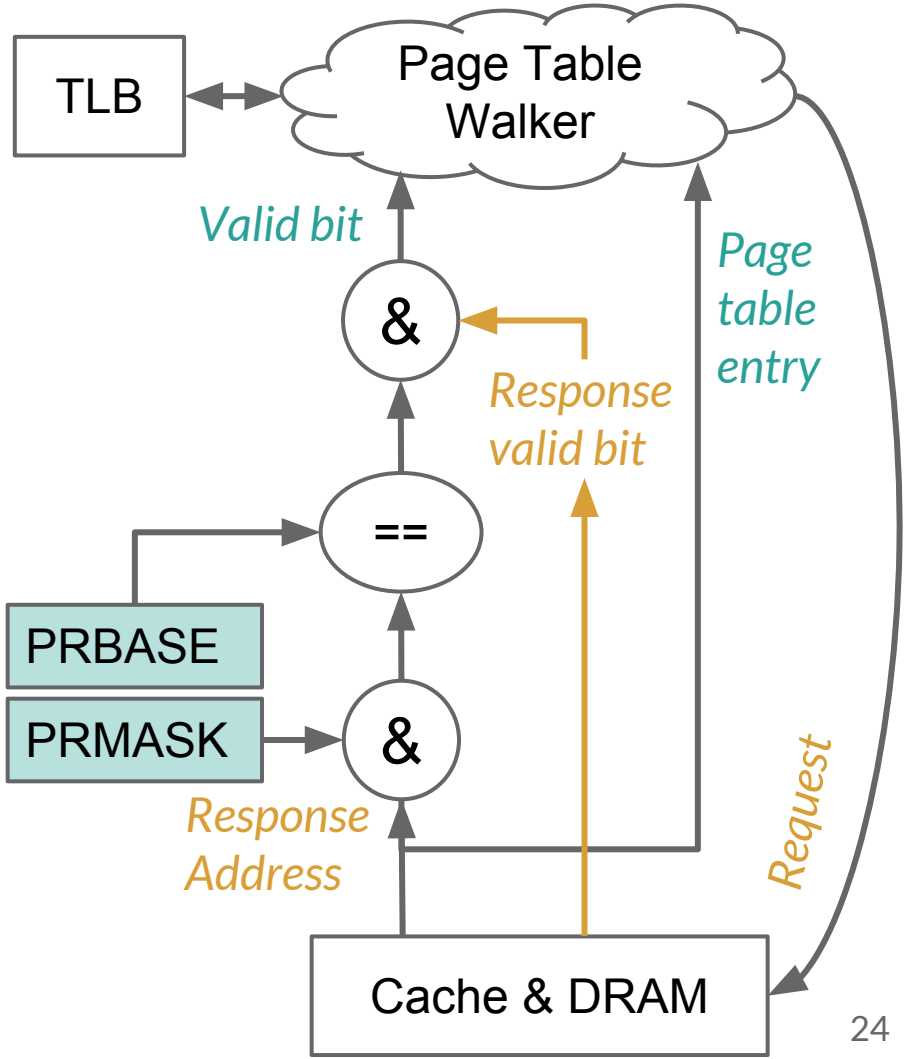
Protect SM memory from everyone

OS could rewrite S.M. code, do evil

fix by...

Never map VAddr to SM memory

S.M. sanitizes mode switch



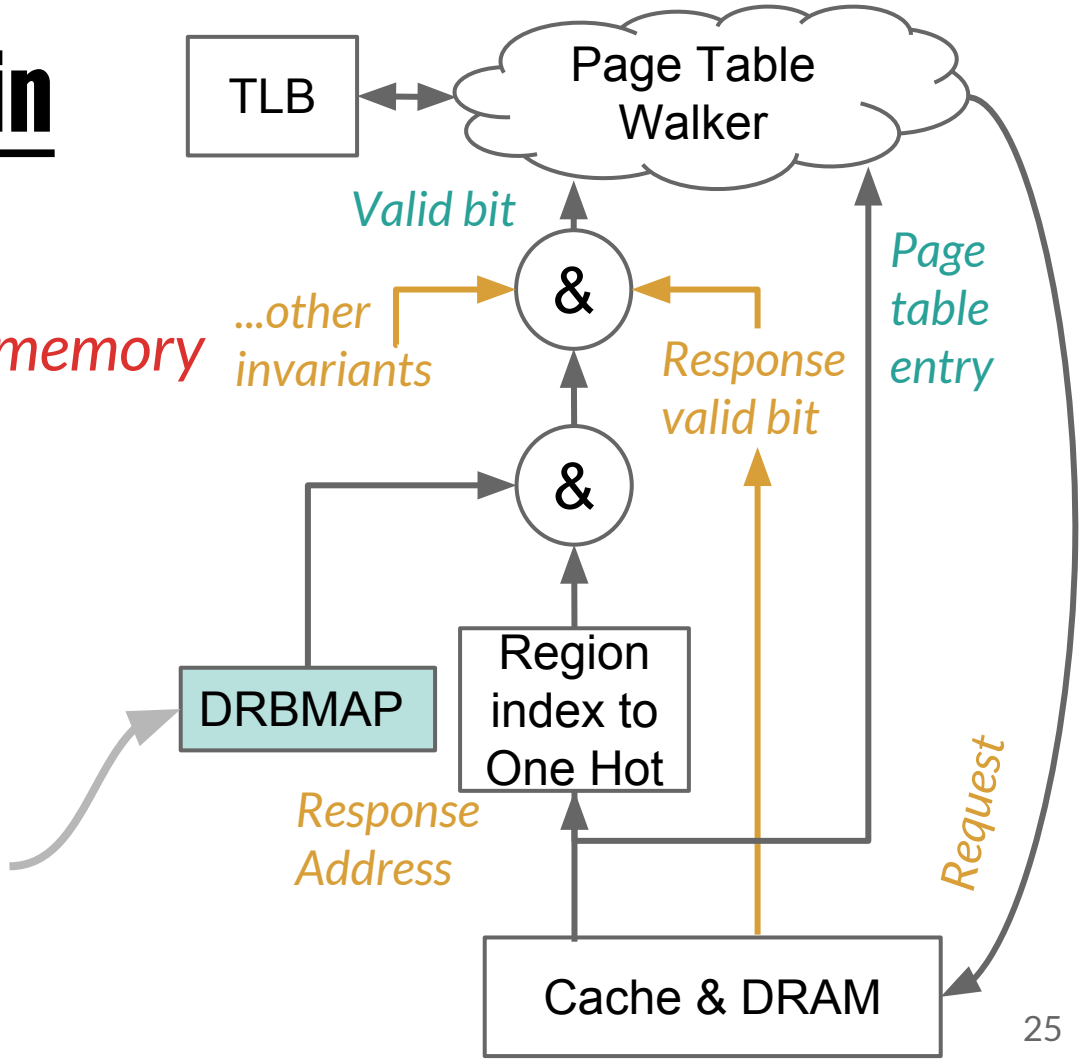
Isolating Enclaves in Physical Memory

OS could read/write Enclave memory

fix by...

Enforce DRAM Region permissions to at page walk

S.M. updates permissions when scheduling enclaves

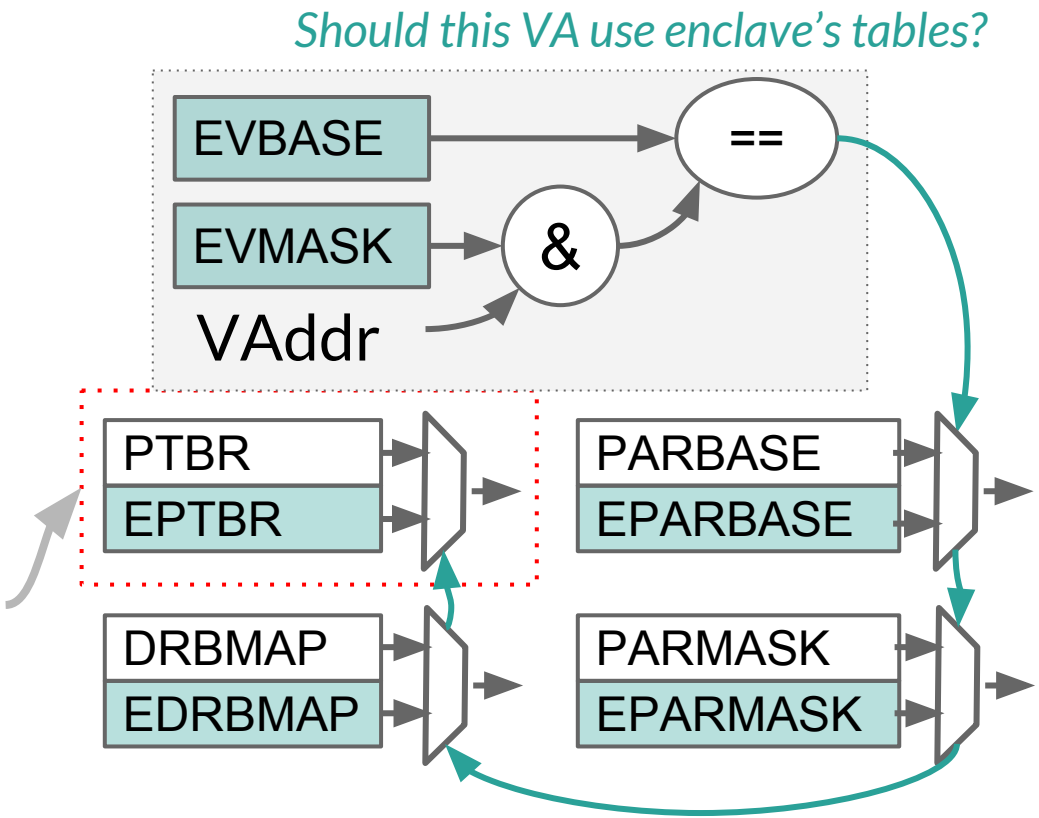


Isolating enclave page tables

OS could spy on enclave's page table entries

fix by...

Implement enclave-private page tables

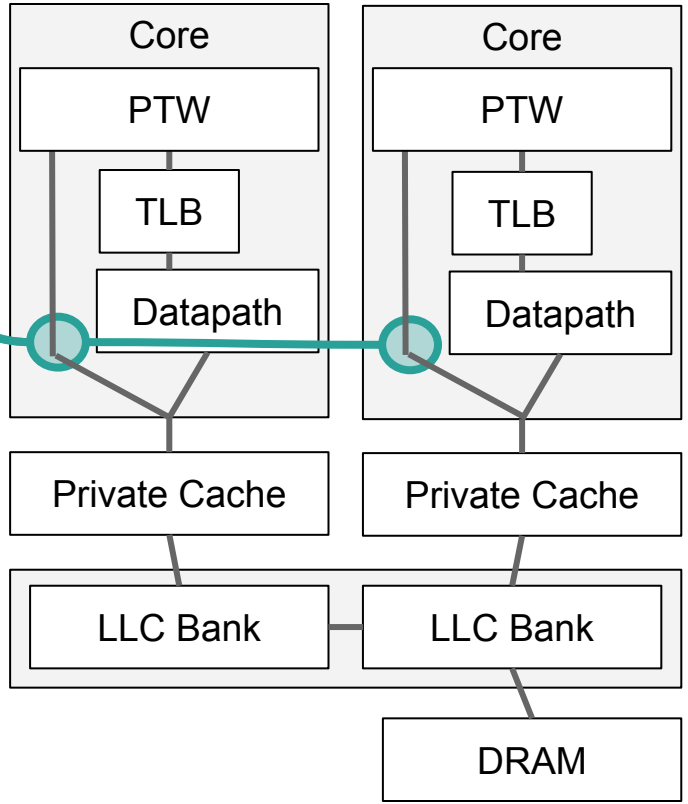


Sanctum modifies RISCV Rocket Chip (1/3)

Add hardware at interfaces only!

~500 Gates, ~700 FFs Per core

- Enclave PTs
- PTW invariant checker



Sanctum modifies RISCV Rocket Chip (2/3)

Add hardware at interfaces only!

Enclave PTs

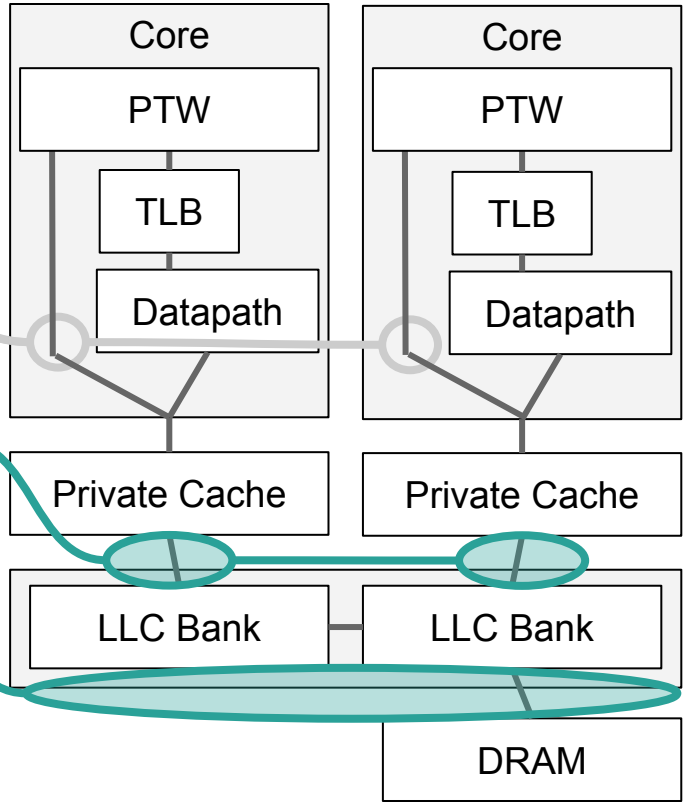
PTW invariant checker

~50 Gates LLC addr. Rotation

600 Gates, 128 FFs DMA whitelist

(hacks upon hacks)

~6% overall performance overhead!



Sanctum modifies RISCV Rocket Chip (3/3)

Add hardware at interfaces only!

Enclave PTs

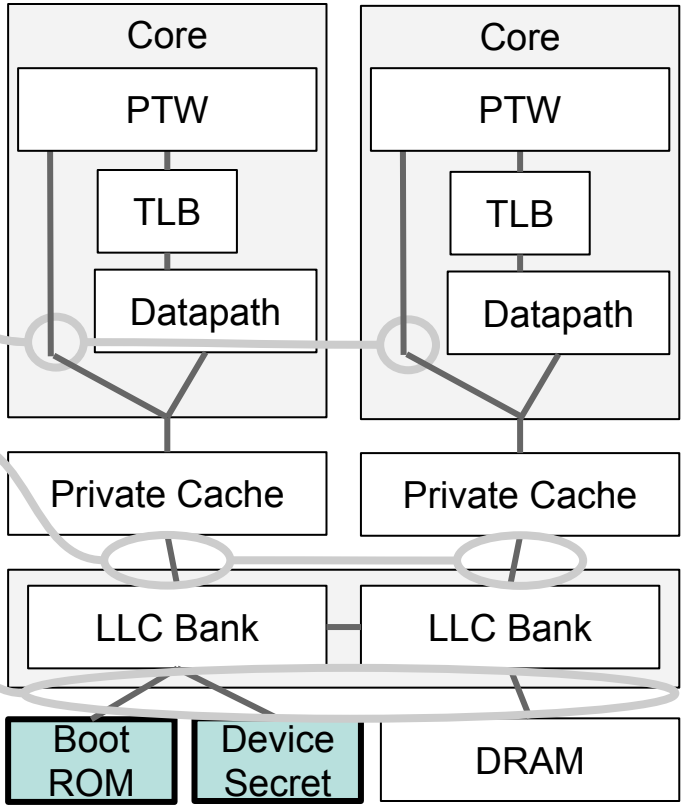
PTW invariant checker

LLC addr. Rotation

DMA whitelist

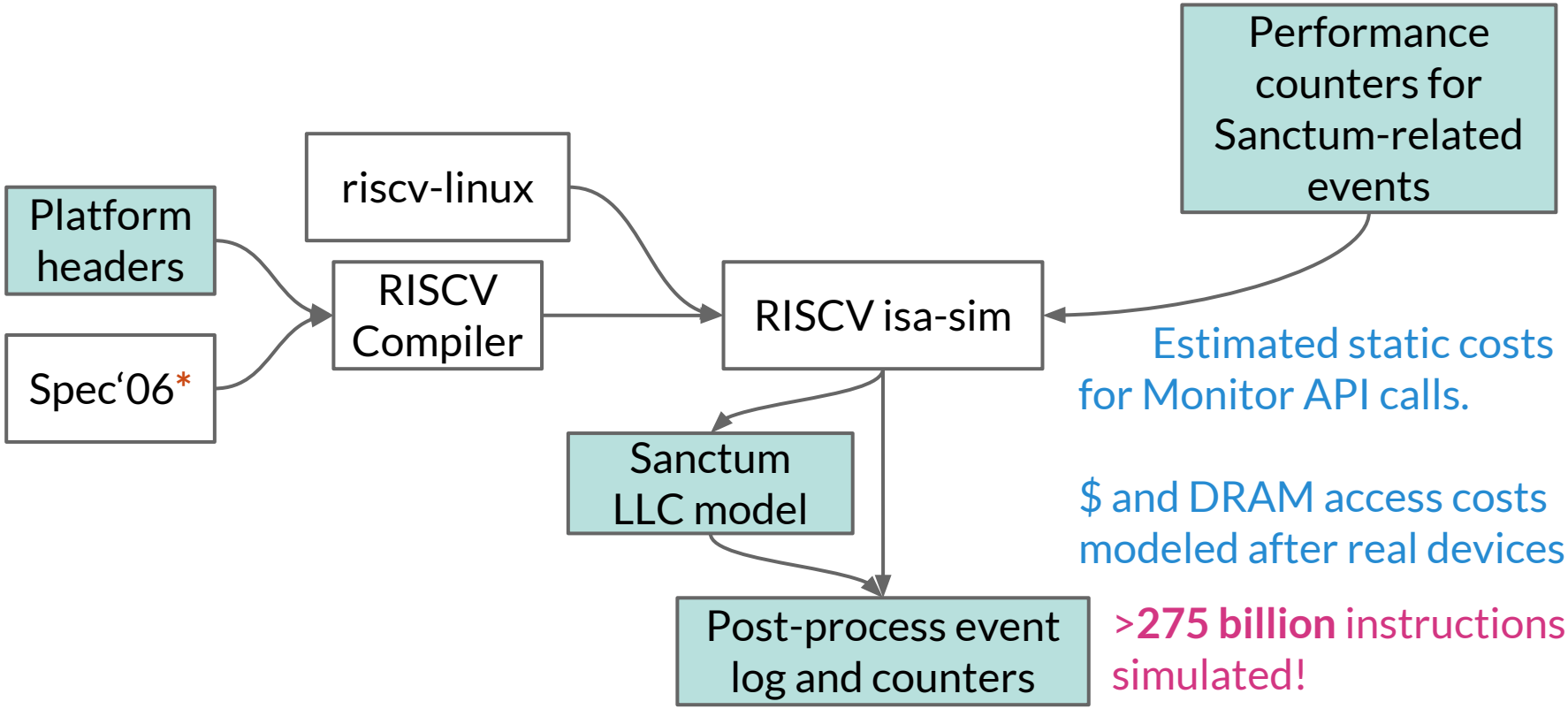
128 FFs Device secret

Boot ROM



~2% area increase in total!

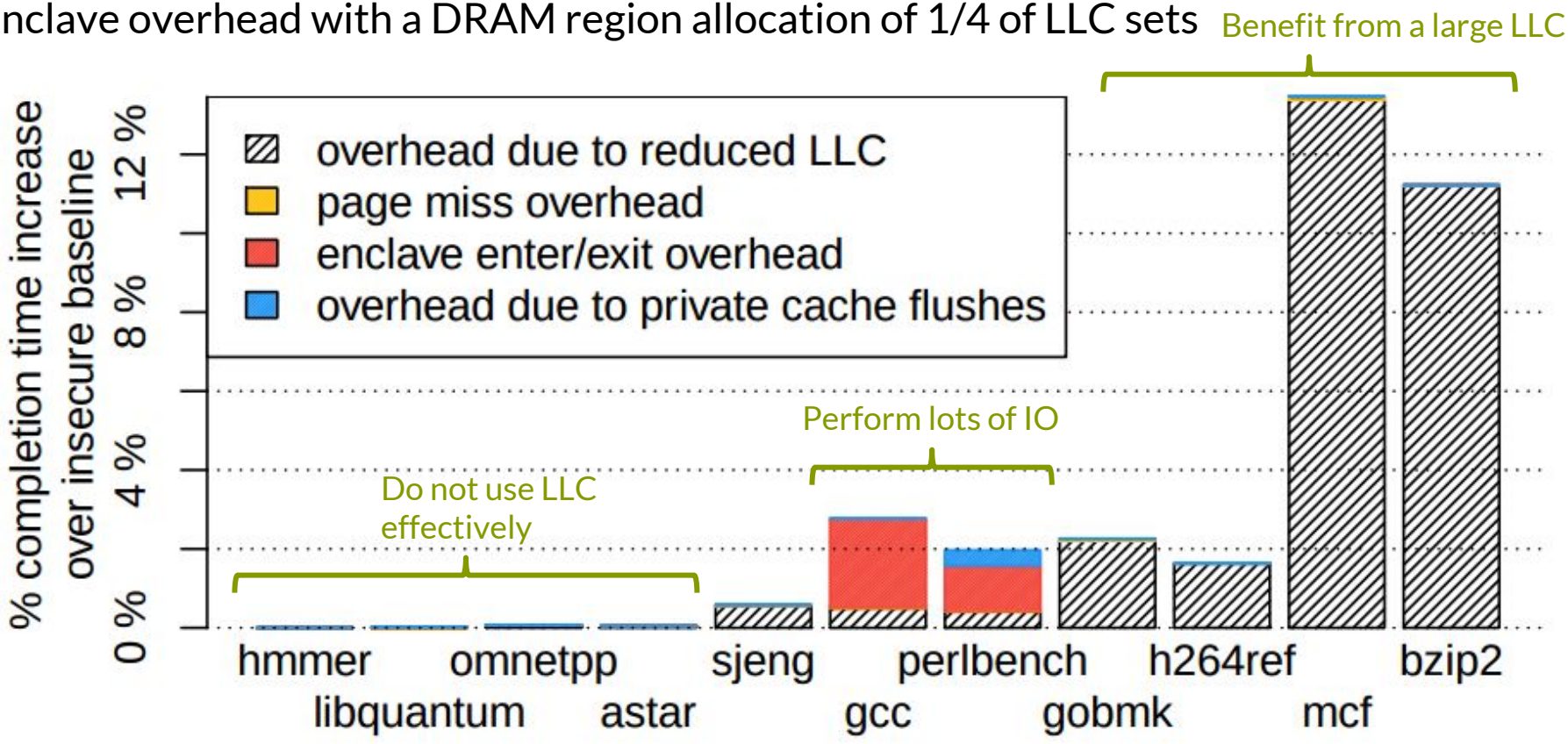
Measuring Overheads: Experimental setup



this slide is intentionally left blank

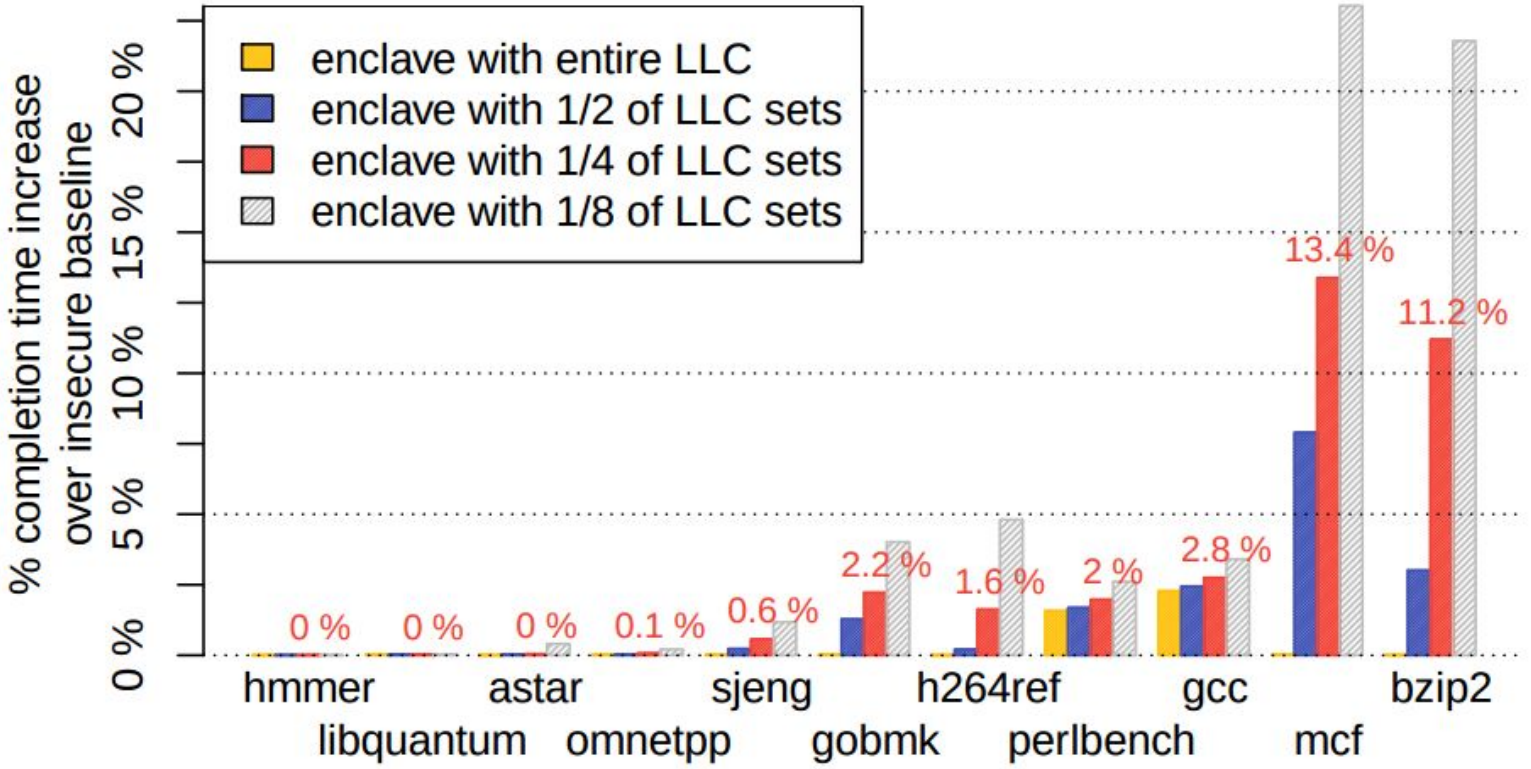
A Detailed Look at Sanctum's Overheads (2/3)

Enclave overhead with a DRAM region allocation of 1/4 of LLC sets

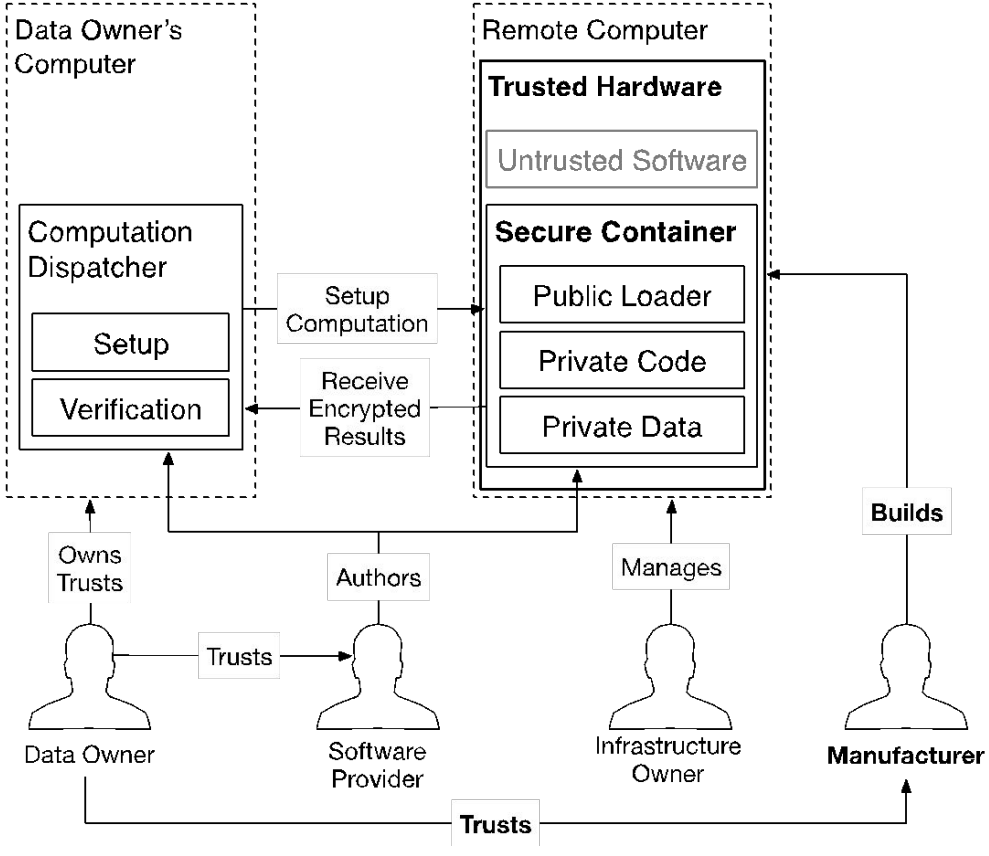


A Detailed Look at Sanctum's Overheads (3/3)

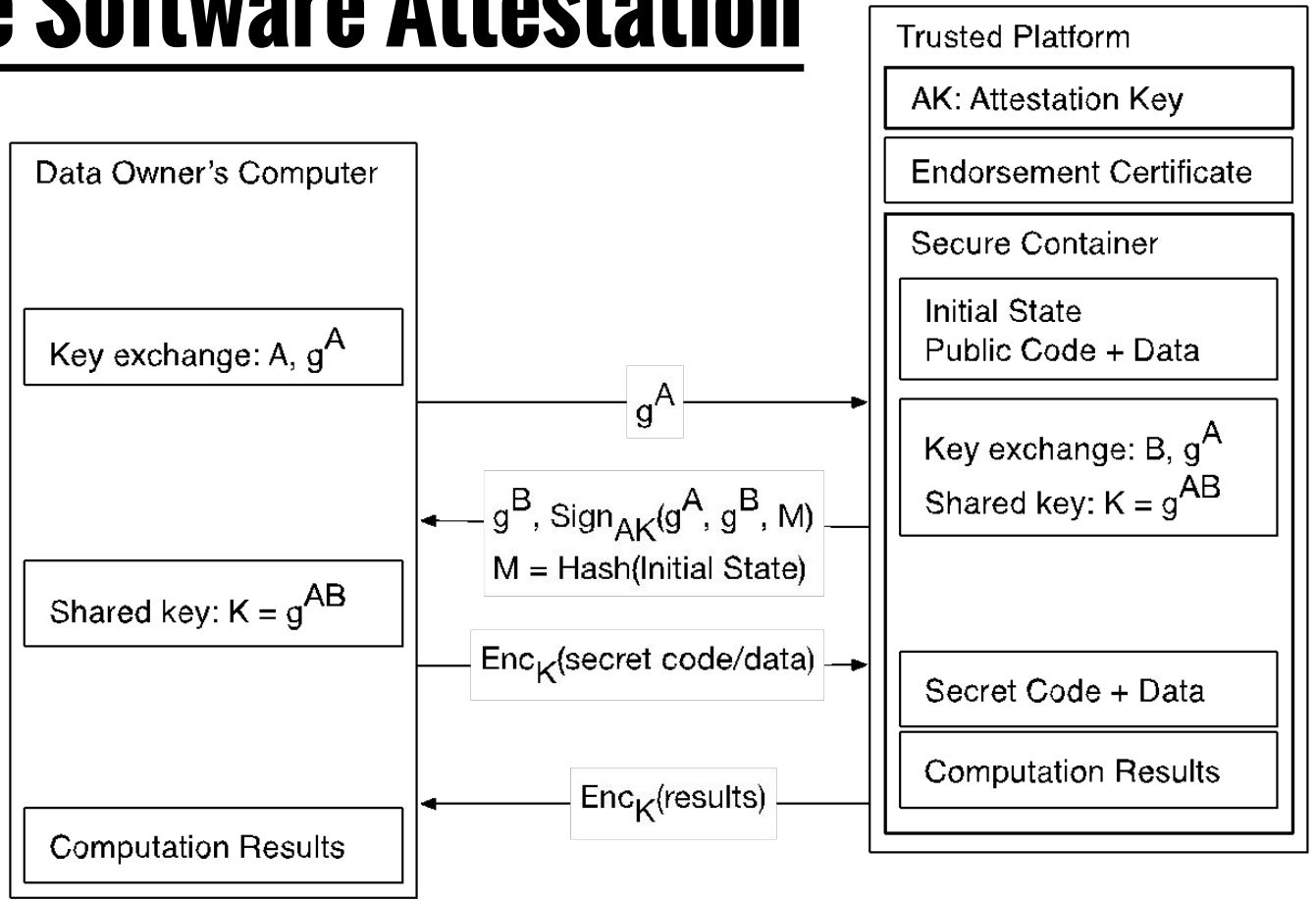
Enclave overhead for various enclave sizes



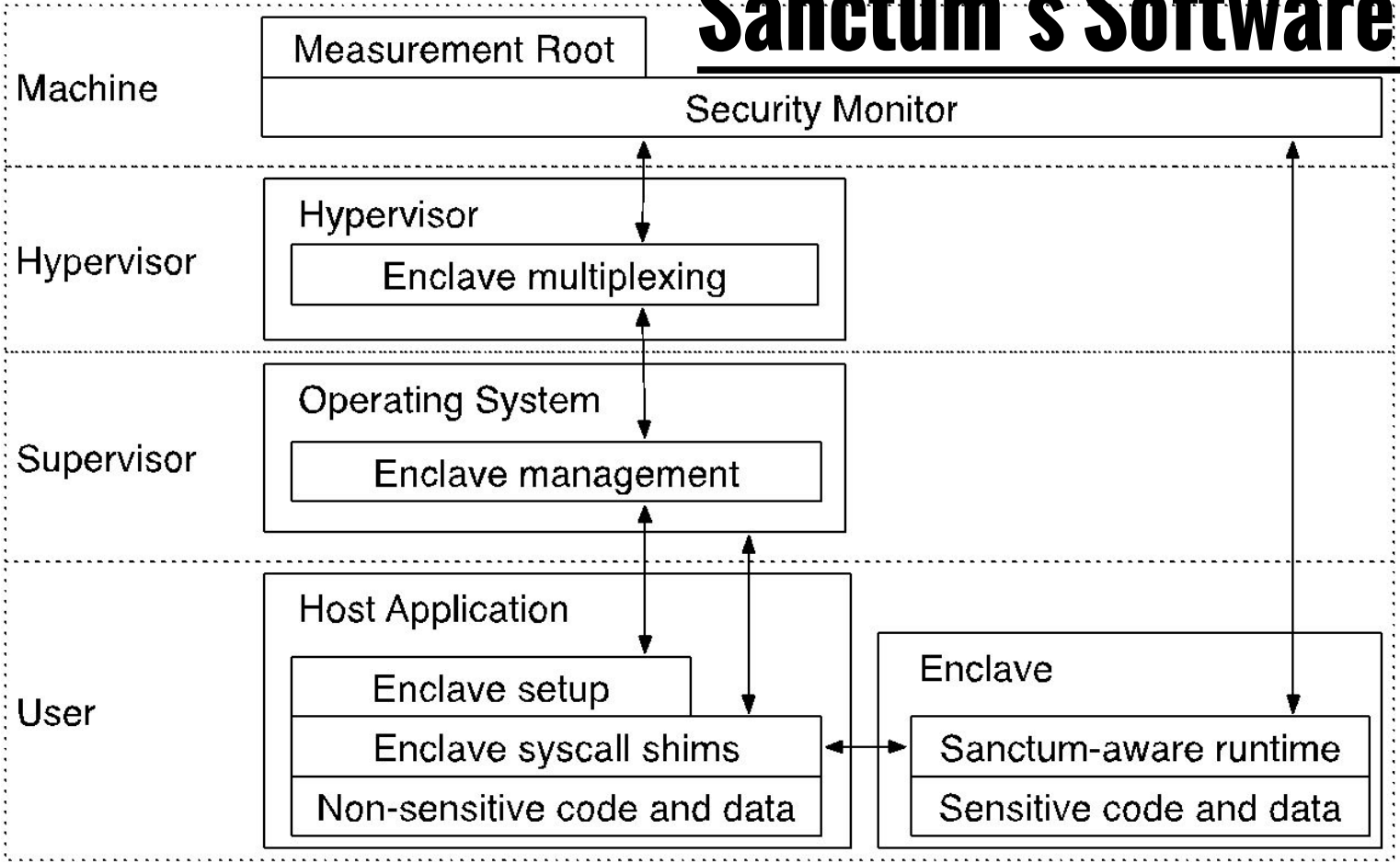
Trusted Manufacturer as CA



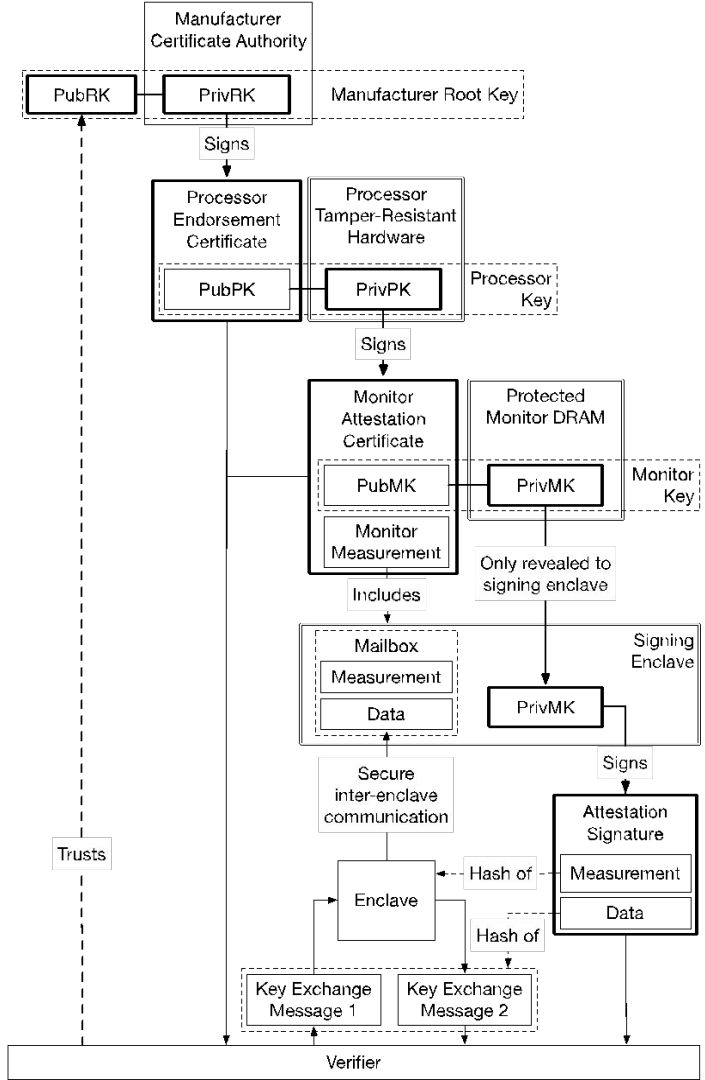
Remote Software Attestation



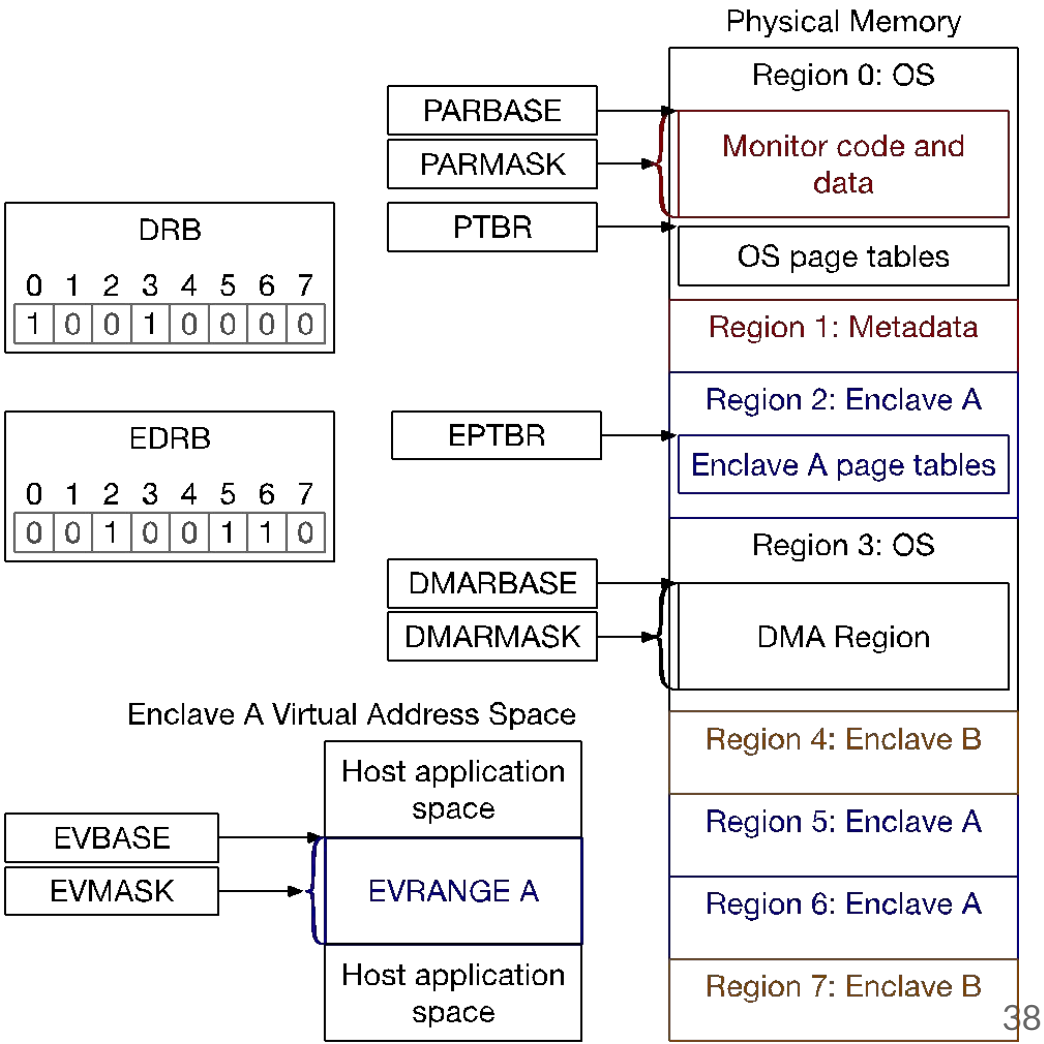
Sanctum's Software Stack



Attestation in Sanctum



Physical Memory in Sacntum (1/2)



Physical Memory in Sacntum (1/2)