SEC RISC-V

Enclave Computing on RISC-V: A Brighter Future for Security?

Ghada Dessouky, Ahmad-Reza Sadeghi, Emmanuel Stapf

Technical University of Darmstadt

TECHNISCHE UNIVERSITÄT DARMSTADT

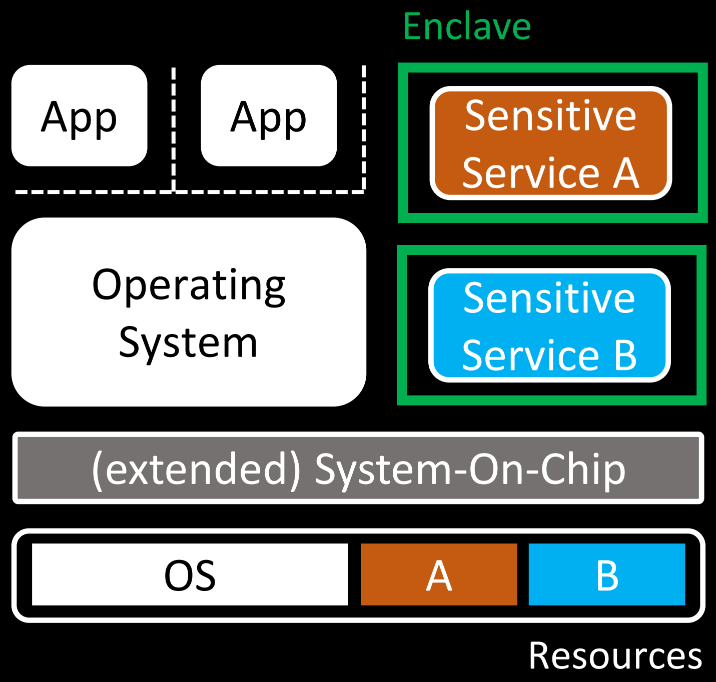
First International Workshop on Secure RISC-V Architecture Design Exploration (SECRISC-V‘20)

Enclave Computing

- Enclaves prominent approach for protecting sensitive services

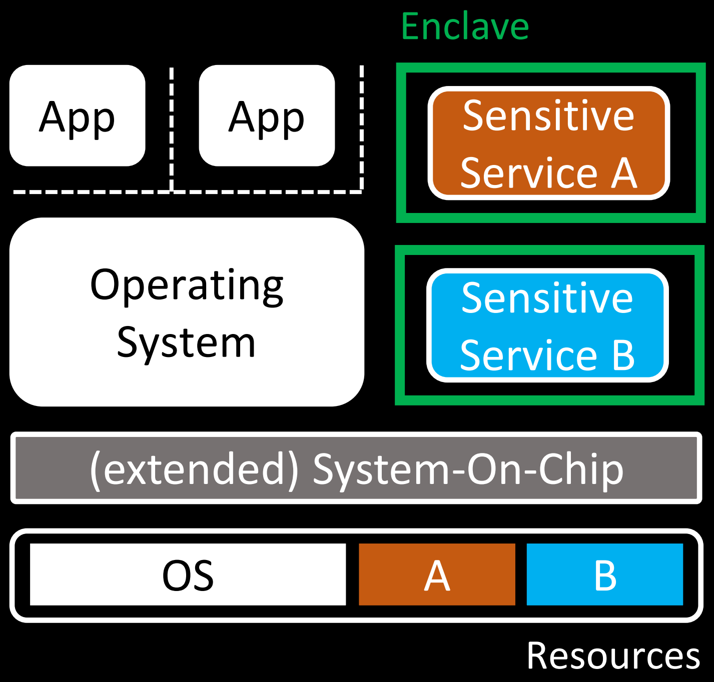
Enclave Computing

- Enclaves prominent approach for protecting sensitive services
- Isolated execution environment, backed by hardware-assisted security mechanisms, configured by trusted SW

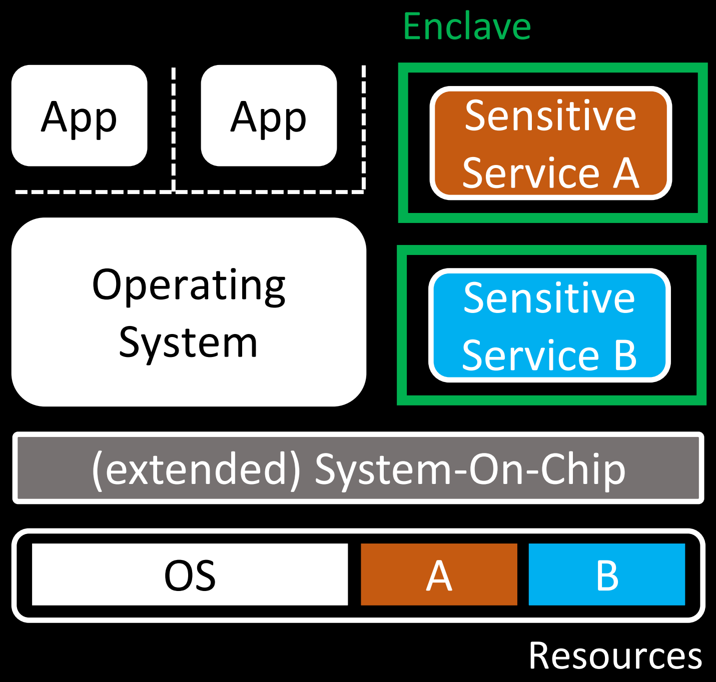


Enclave Computing

- Enclaves prominent approach for protecting sensitive services
- Isolated execution environment, backed by hardware-assisted security mechanisms, configured by trusted SW
- Industry solutions (SGX, SEV, TrustZone) vulnerable to side-channel attacks and miss features (I/O)

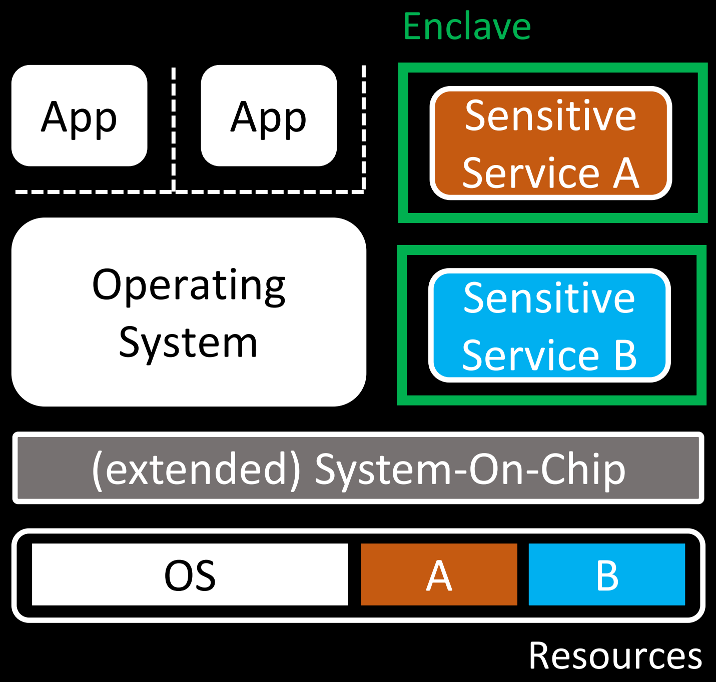


Enclave Computing



- Enclaves prominent approach for protecting sensitive services
- Isolated execution environment, backed by hardware-assisted security mechanisms, configured by trusted SW
- Industry solutions (SGX, SEV, TrustZone) vulnerable to side-channel attacks and miss features (I/O)
- Solutions (often) require HW modifications

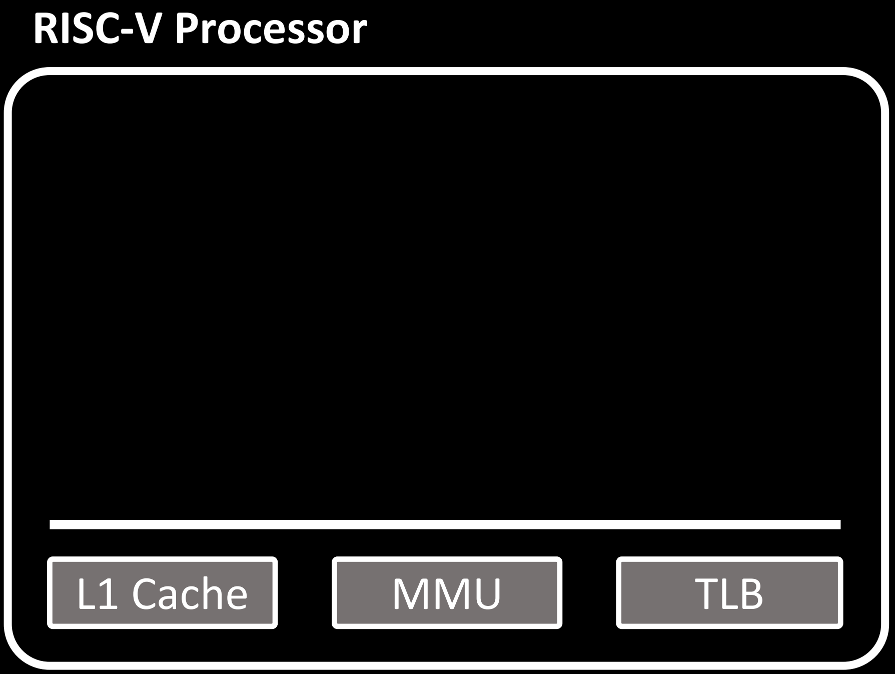
Enclave Computing



- Enclaves prominent approach for protecting sensitive services
- Isolated execution environment, backed by hardware-assisted security mechanisms, configured by trusted SW
- Industry solutions (SGX, SEV, TrustZone) vulnerable to side-channel attacks and miss features (I/O)
- Solutions (often) require HW modifications
- Open HW concept of RISC-V propels research on enclave computing

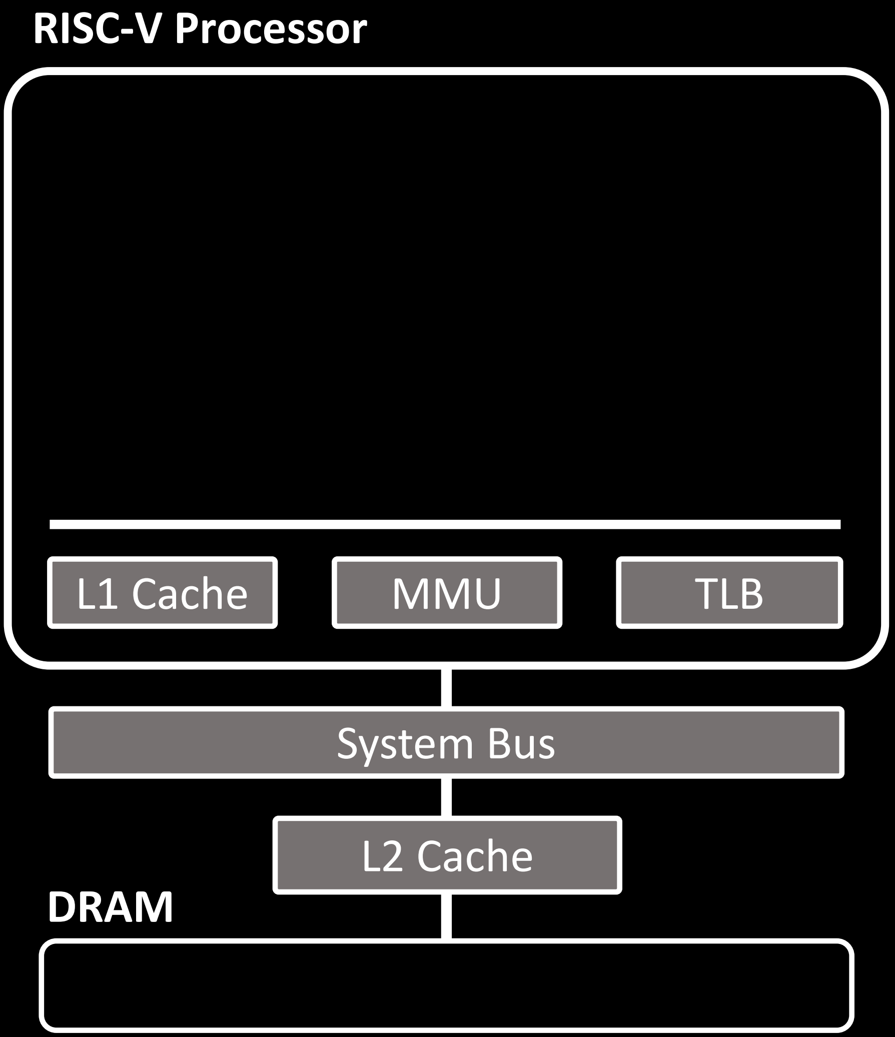
Enclave Security Architectures on RISC-V

Sanctum



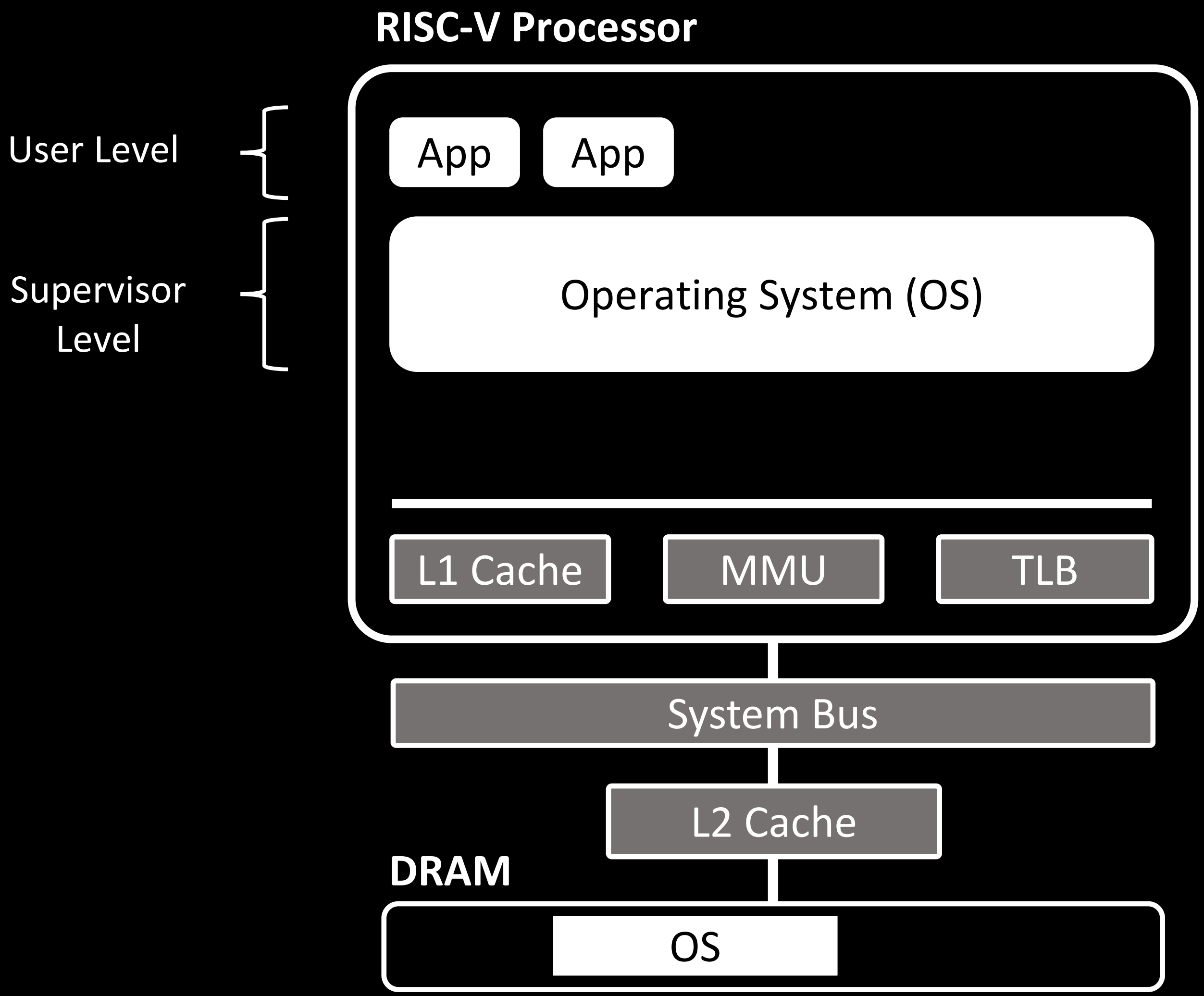
[Costan et al., USENIX Security 2016]

Sanctum



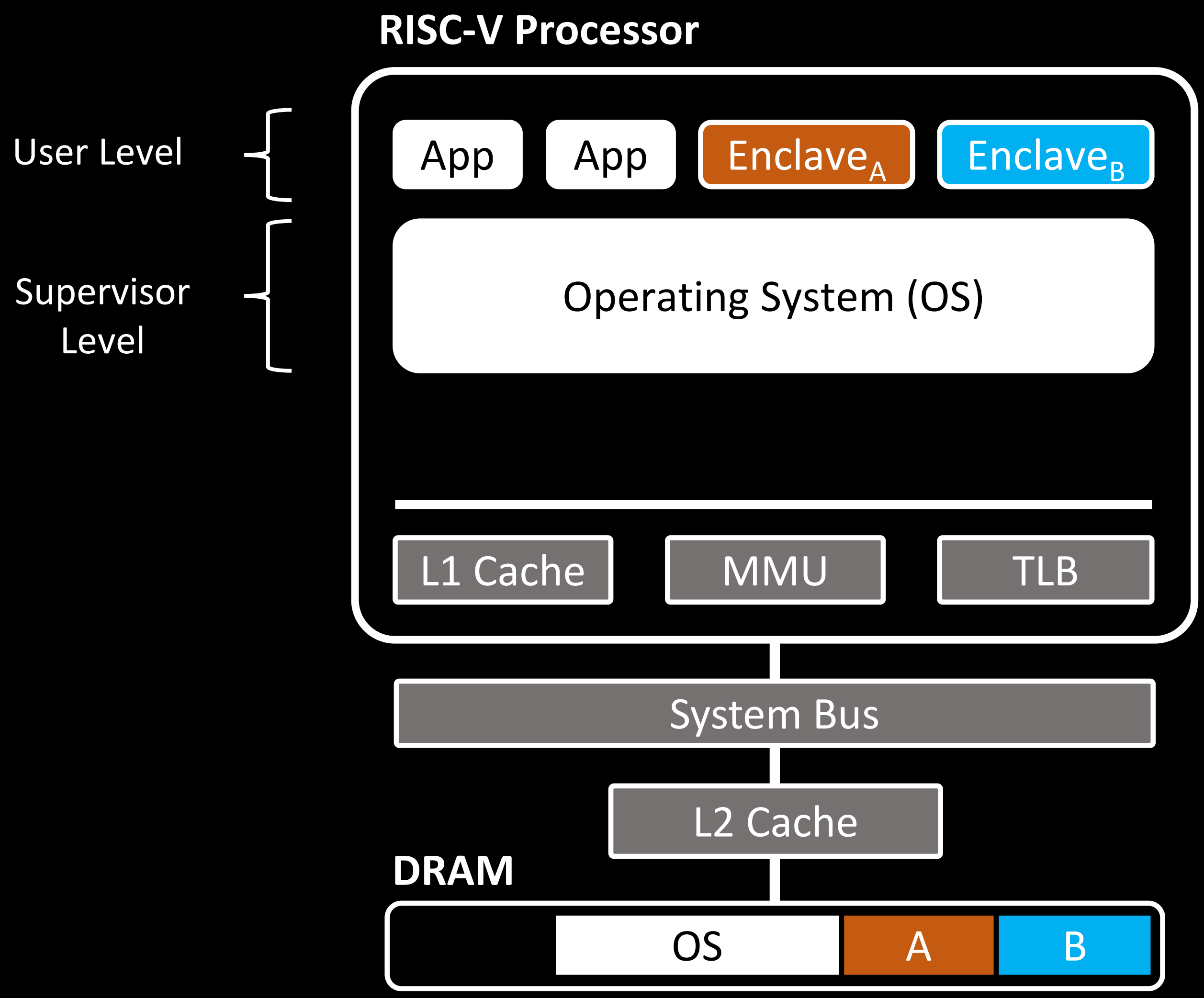
[Costan et al., USENIX Security 2016]

Sanctum



[Costan et al., USENIX Security 2016]

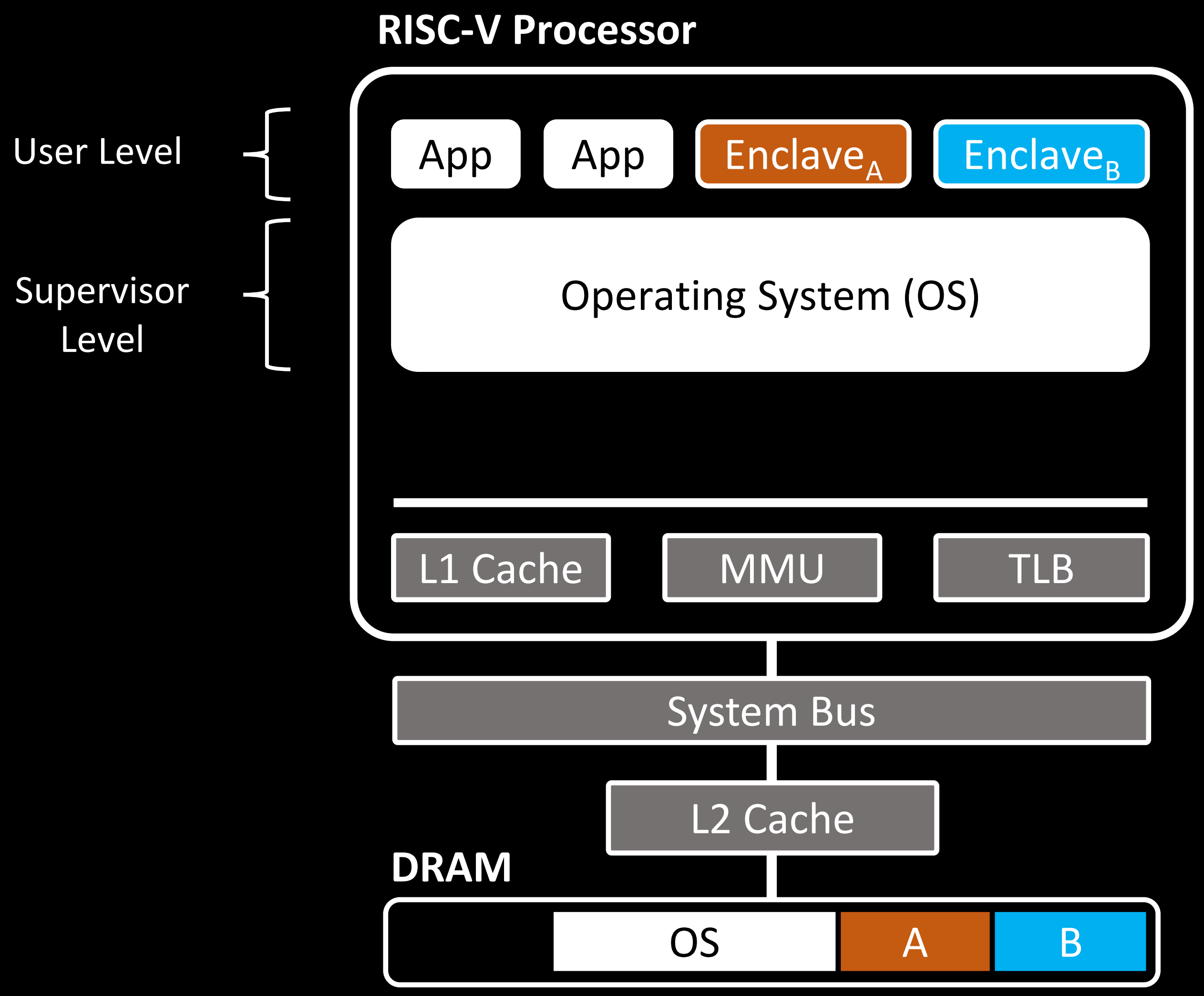
Sanctum



- Enclaves in user level

[Costan et al., USENIX Security 2016]

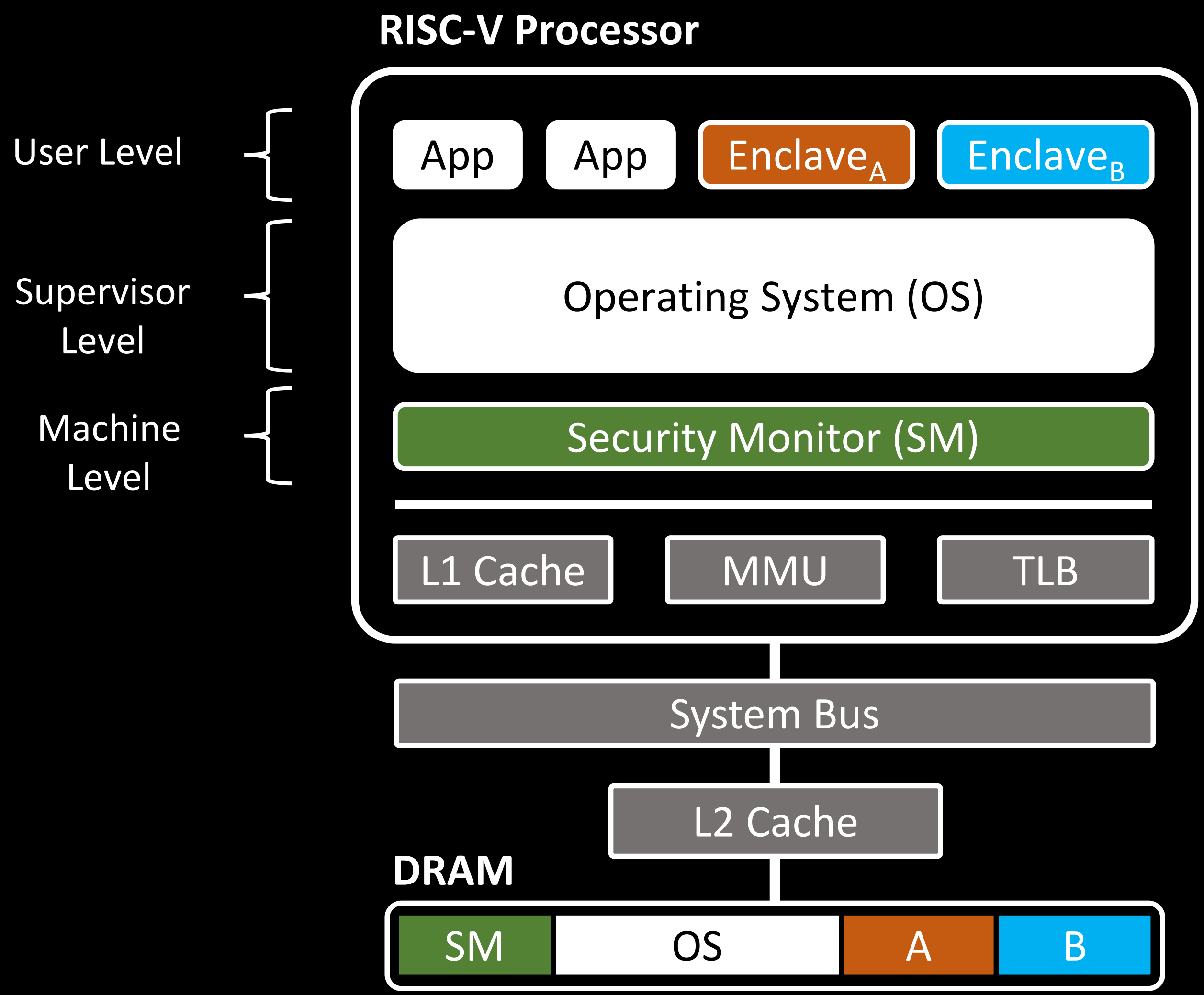
Sanctum



- Enclaves in user level
- OS manages enclaves and provides services (e.g., I/O, interrupt handling)

[Costan et al., USENIX Security 2016]

Sanctum

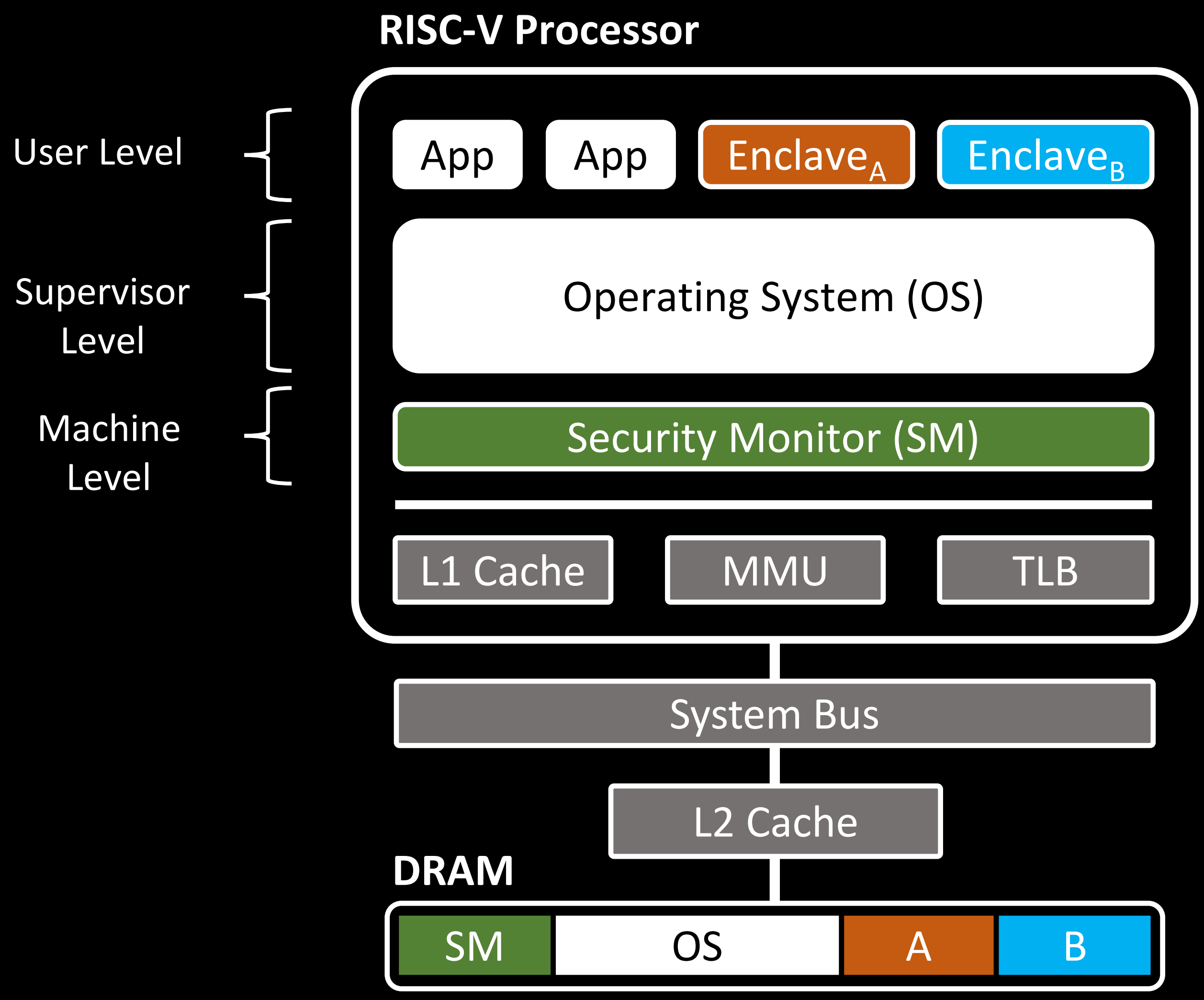


- Enclaves in user level
- OS manages enclaves and provides services (e.g., I/O, interrupt handling)
- SM checks OS management decisions

🟩 Software TCB

[Costan et al., USENIX Security 2016]

Sanctum

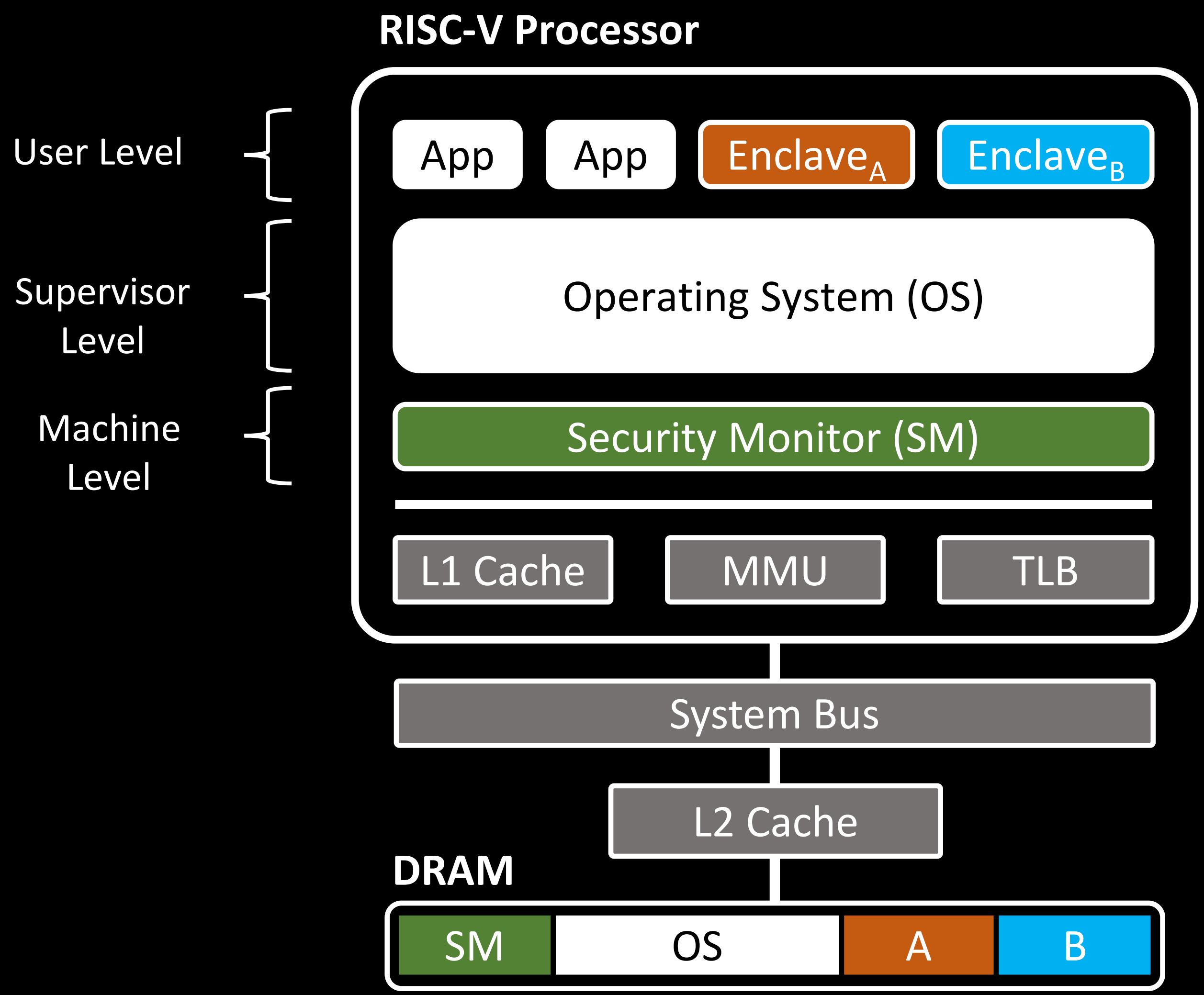


- Enclaves in user level
- OS manages enclaves and provides services (e.g., I/O, interrupt handling)
- SM checks OS management decisions
- Custom circuitry in MMU protects enclaves and SM memory

🟩 Software TCB

[Costan et al., USENIX Security 2016]

Sanctum

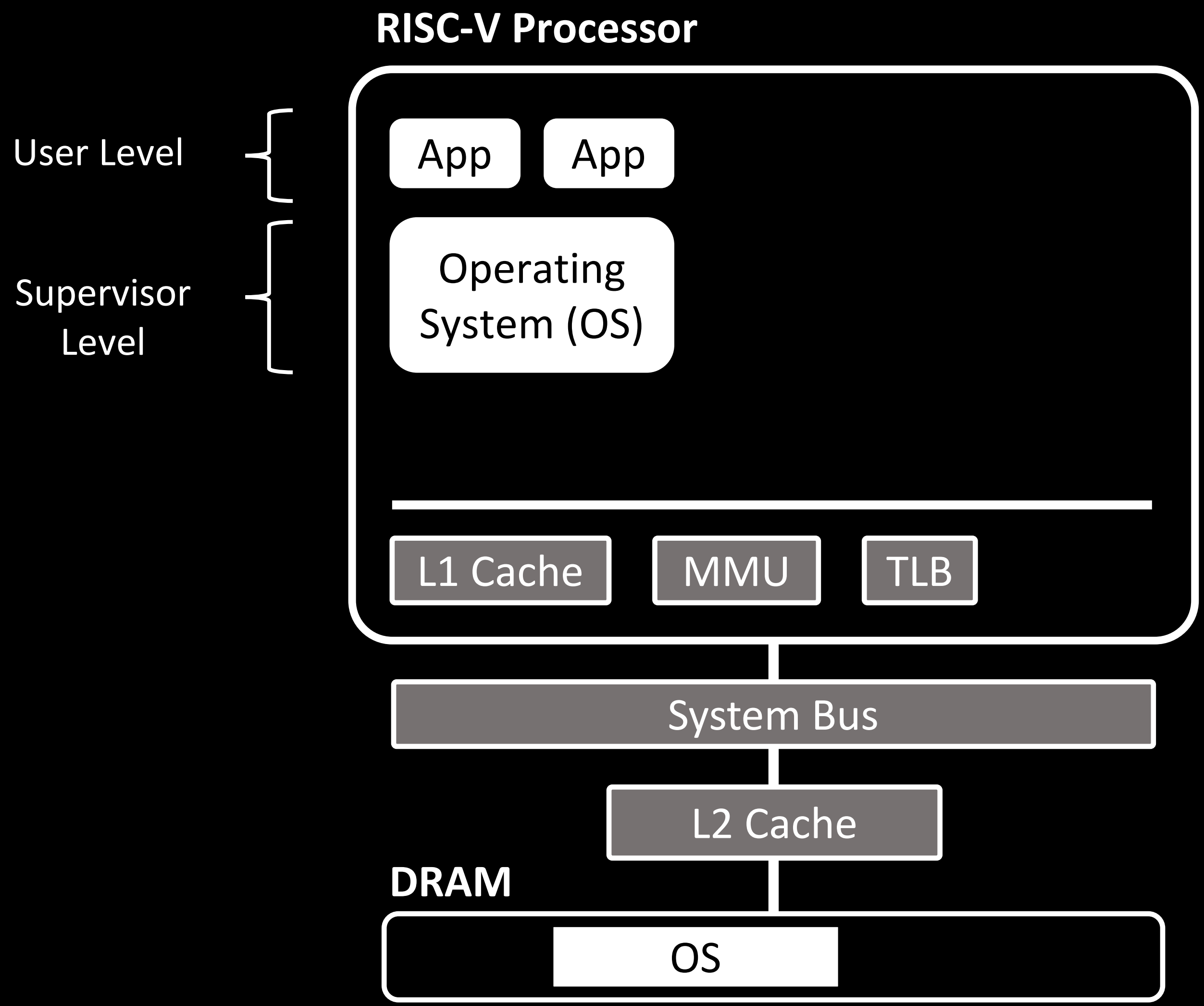


- Enclaves in user level
- OS manages enclaves and provides services (e.g., I/O, interrupt handling)
- SM checks OS management decisions
- Custom circuitry in MMU protects enclaves and SM memory
- Cache side-channel protection through page coloring, influences OS memory layout

🟩 Software TCB

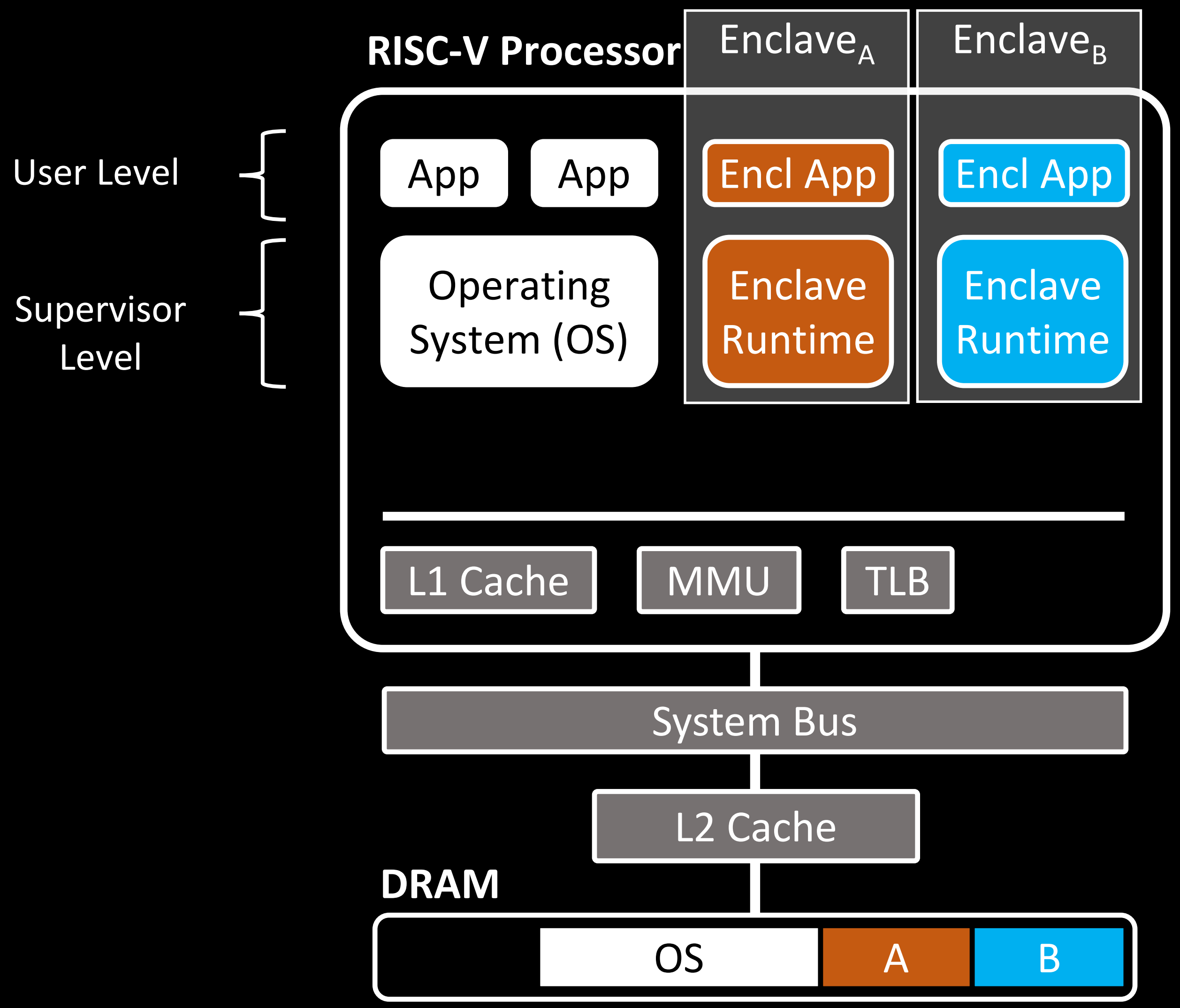
[Lee et al., EuroSys 2020]

Keystone



[Lee et al., EuroSys 2020]

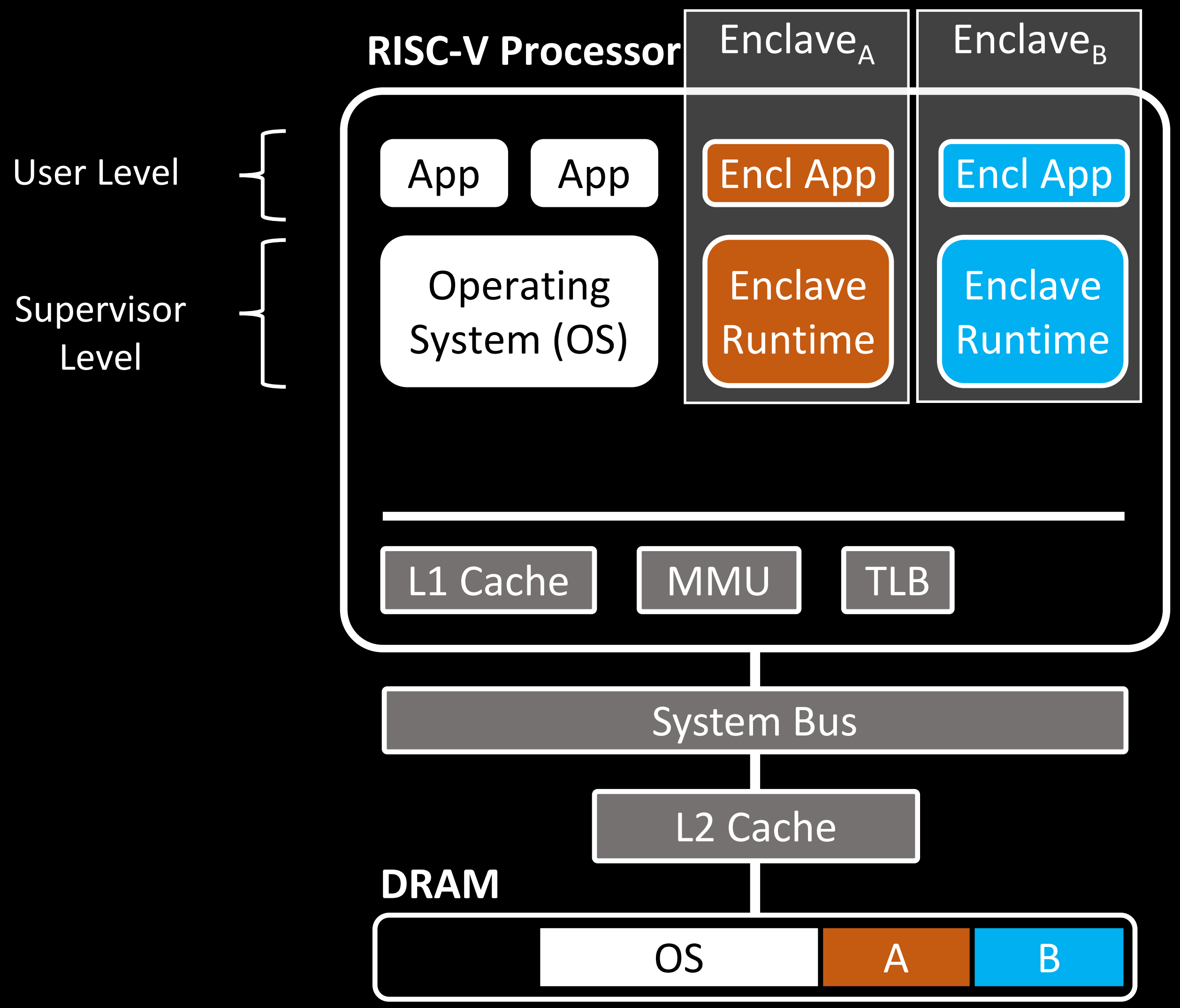
Keystone



- Enclaves contain user and supervisor level

[Lee et al., EuroSys 2020]

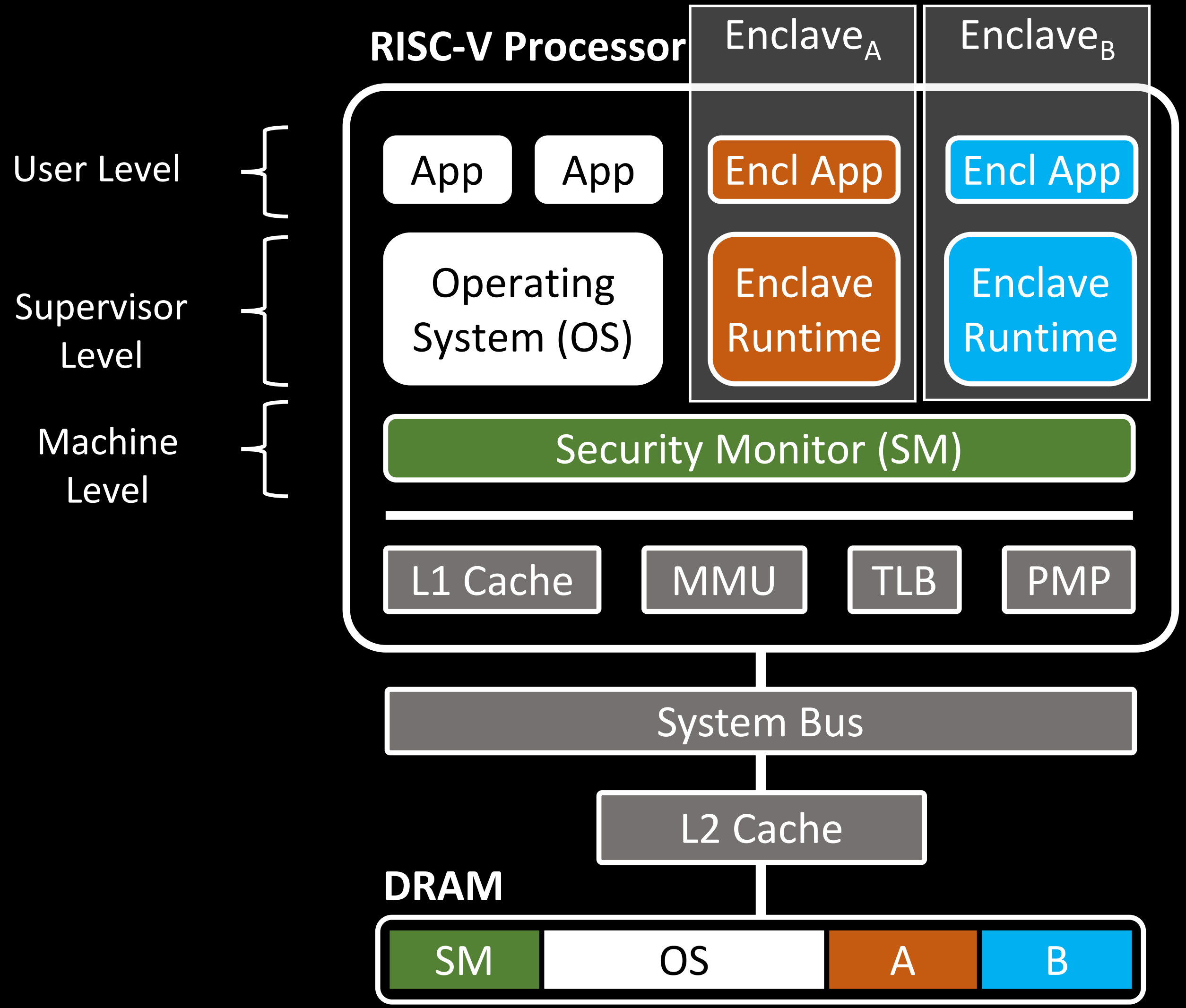
Keystone



- Enclaves contain user and supervisor level
- Enclave runtime provides thread and page table management

[Lee et al., EuroSys 2020]

Keystone

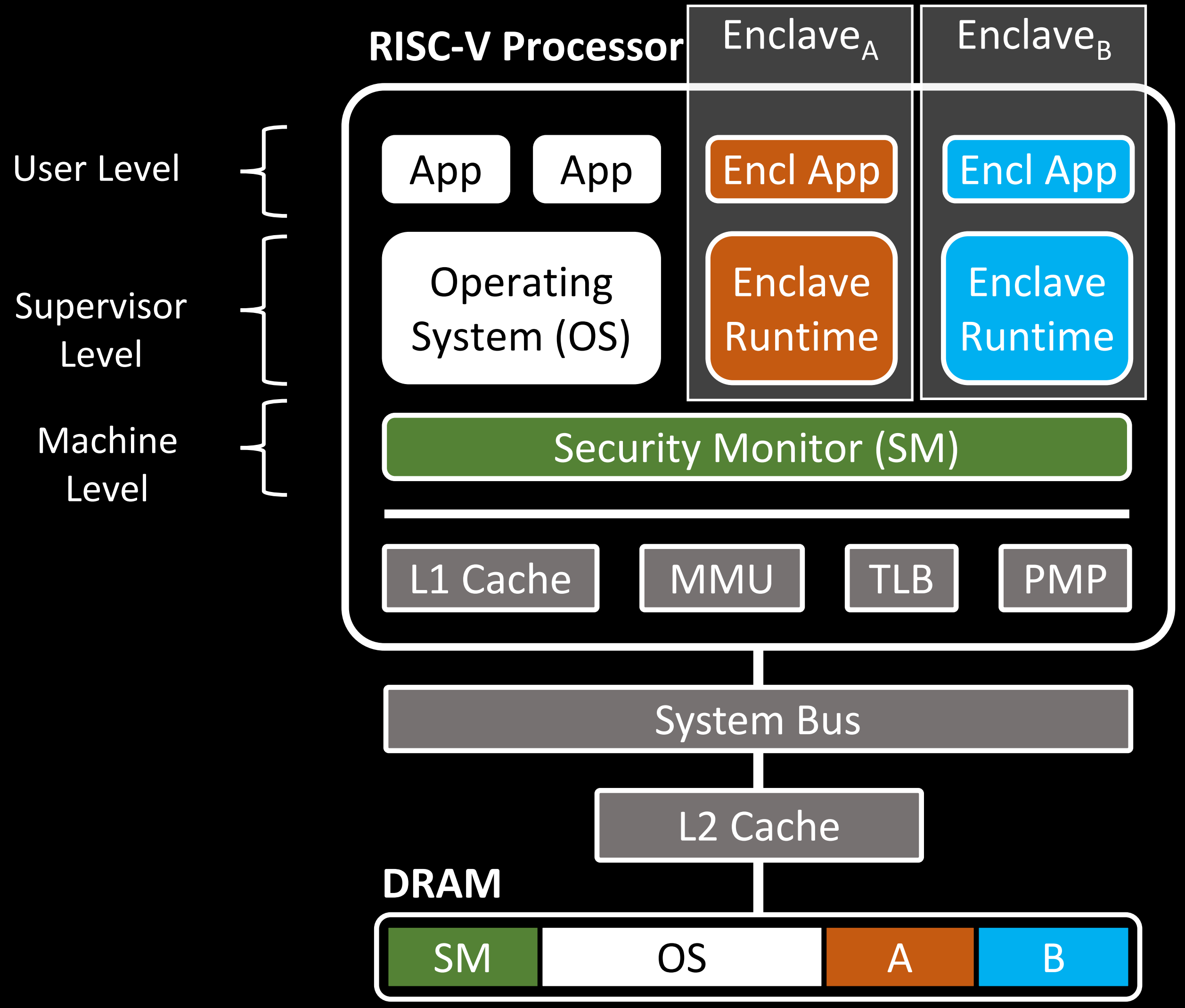


- Enclaves contain user and supervisor level
- Enclave runtime provides thread and page table management
- Enclaves protected by Physical Memory Protection (PMP), configured by SM

🟩 Software TCB

[Lee et al., EuroSys 2020]

Keystone

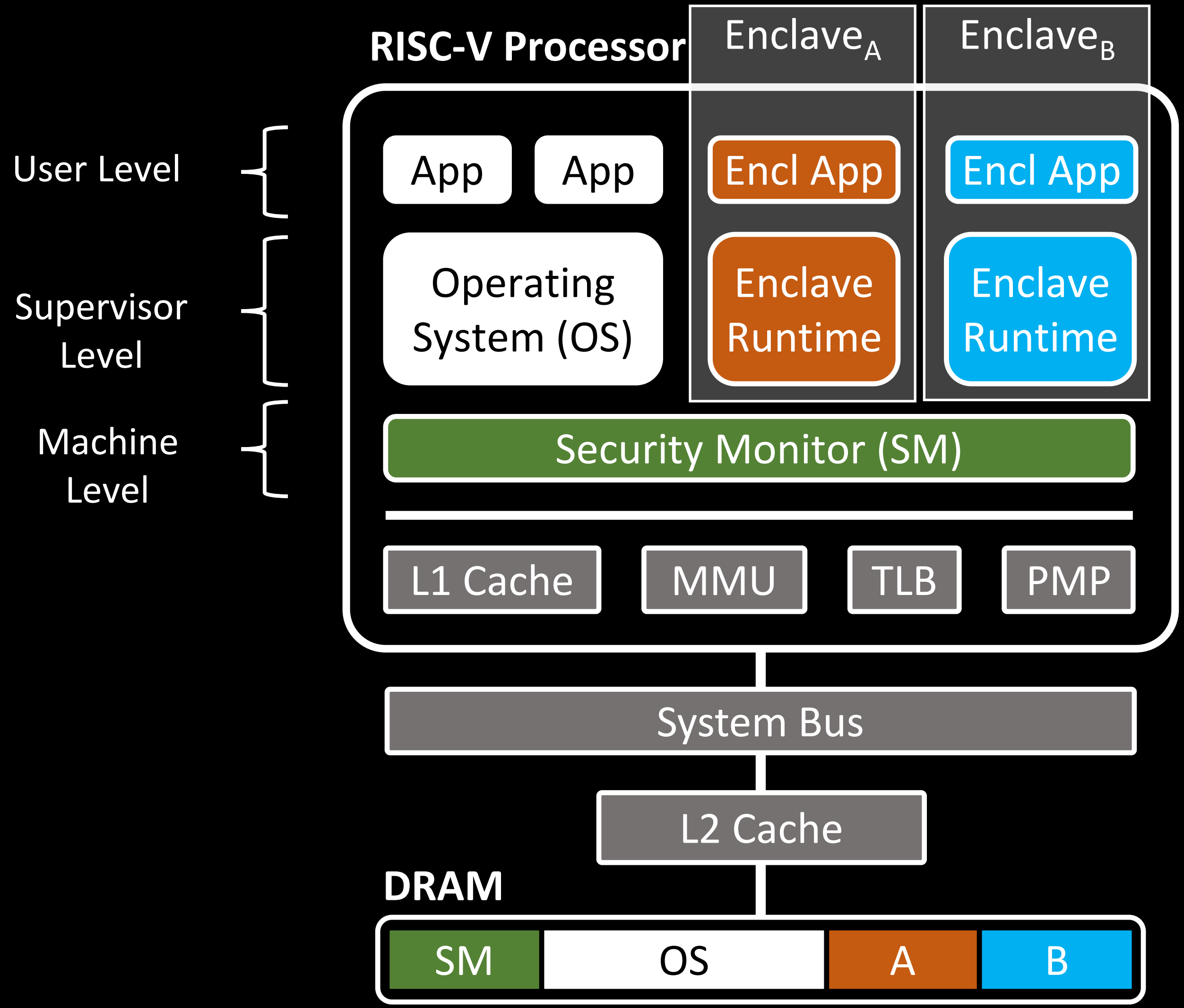


- Enclaves contain user and supervisor level
- Enclave runtime provides thread and page table management
- Enclaves protected by Physical Memory Protection (PMP), configured by SM
- One PMP region reserved for each active enclave

🟩 Software TCB

[Lee et al., EuroSys 2020]

Keystone

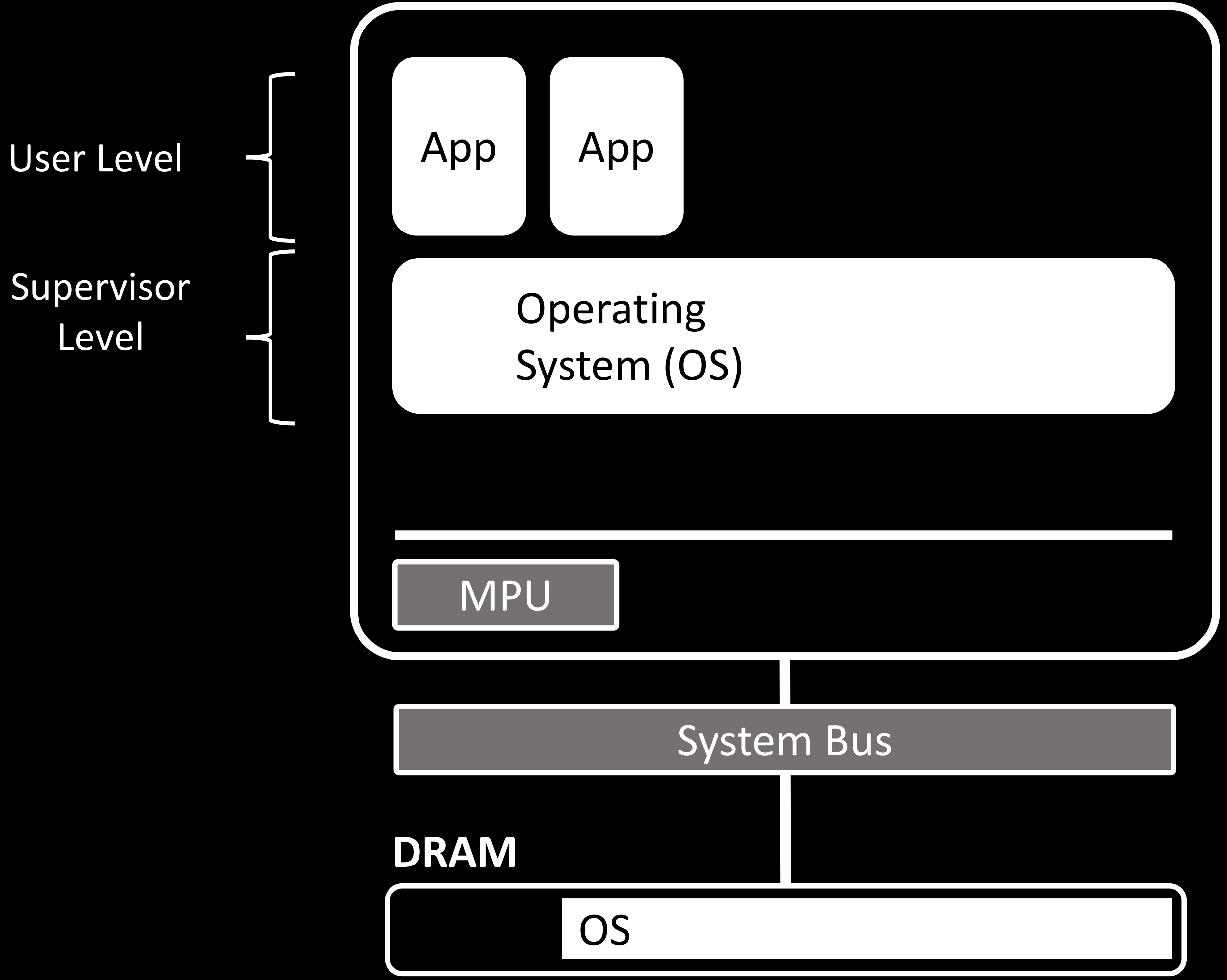


- Enclaves contain user and supervisor level
- Enclave runtime provides thread and page table management
- Enclaves protected by Physical Memory Protection (PMP), configured by SM
- One PMP region reserved for each active enclave
- Assigns cache ways to processor cores

🟩 Software TCB

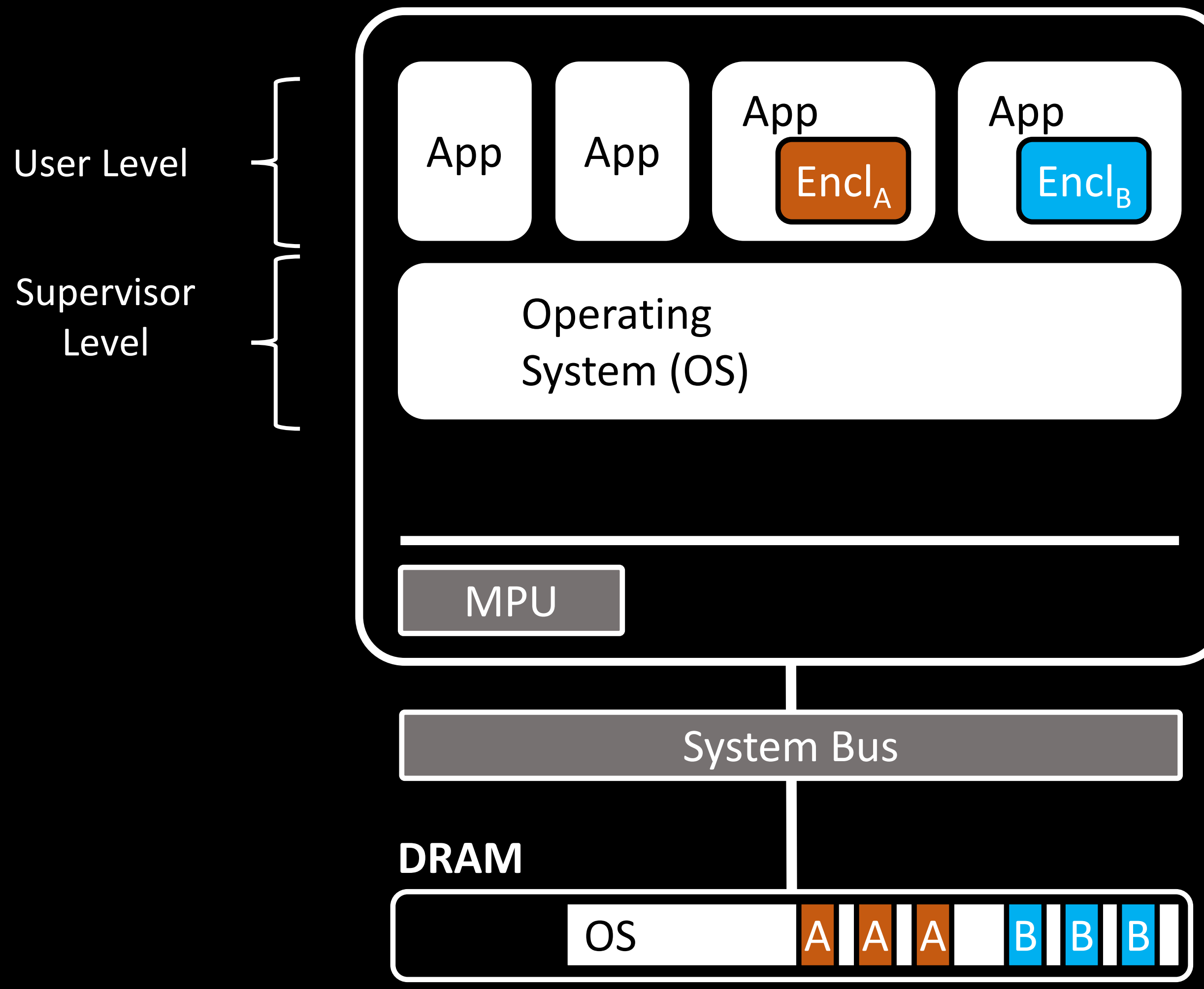
[Weiser et al., NDSS 2019]

TIMBER-V



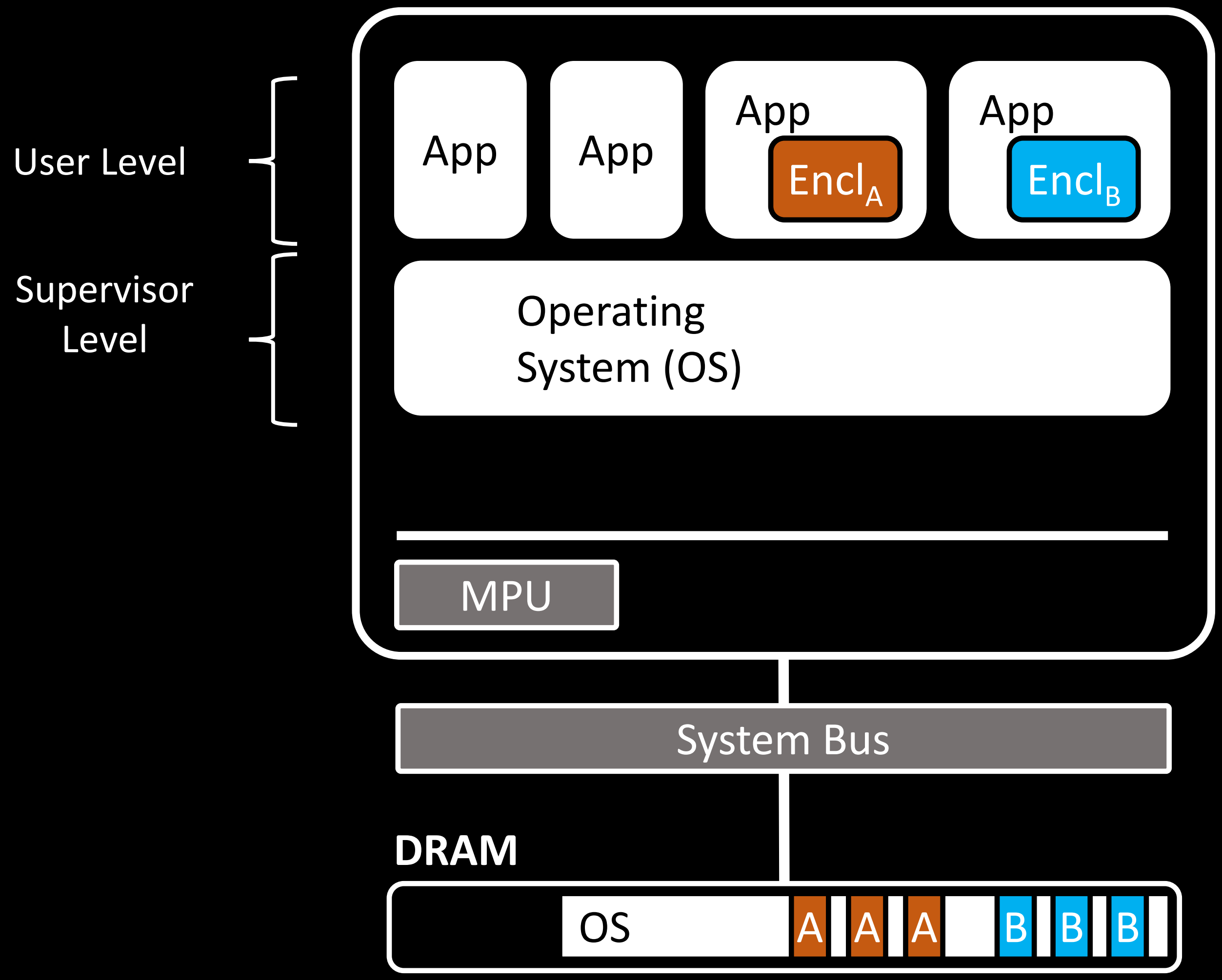
[Weiser et al., NDSS 2019]

TIMBER-V



[Weiser et al., NDSS 2019]

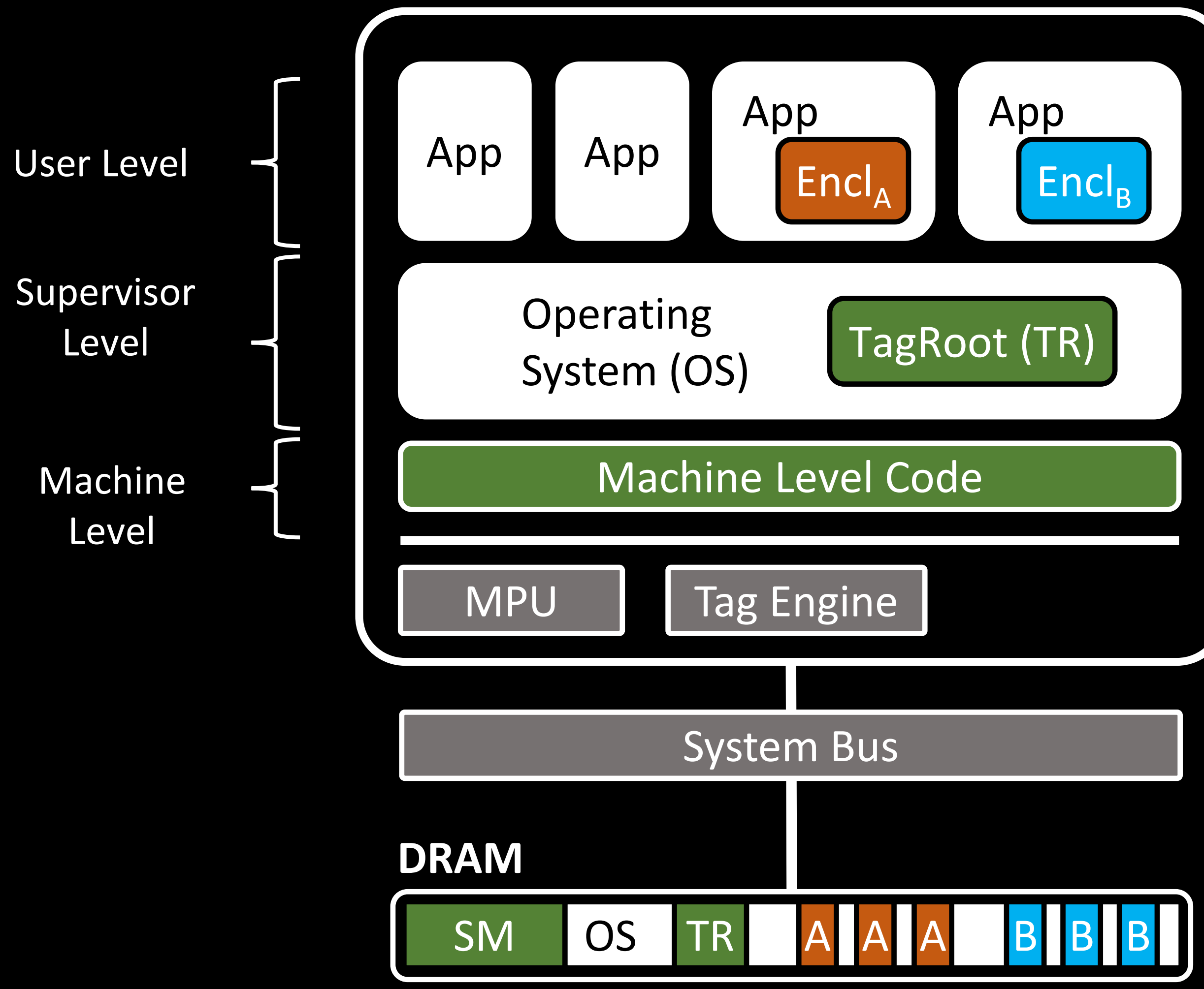
TIMBER-V



- Isolates sensitive part of app using memory tagging (in-process enclave)

[Weiser et al., NDSS 2019]

TIMBER-V

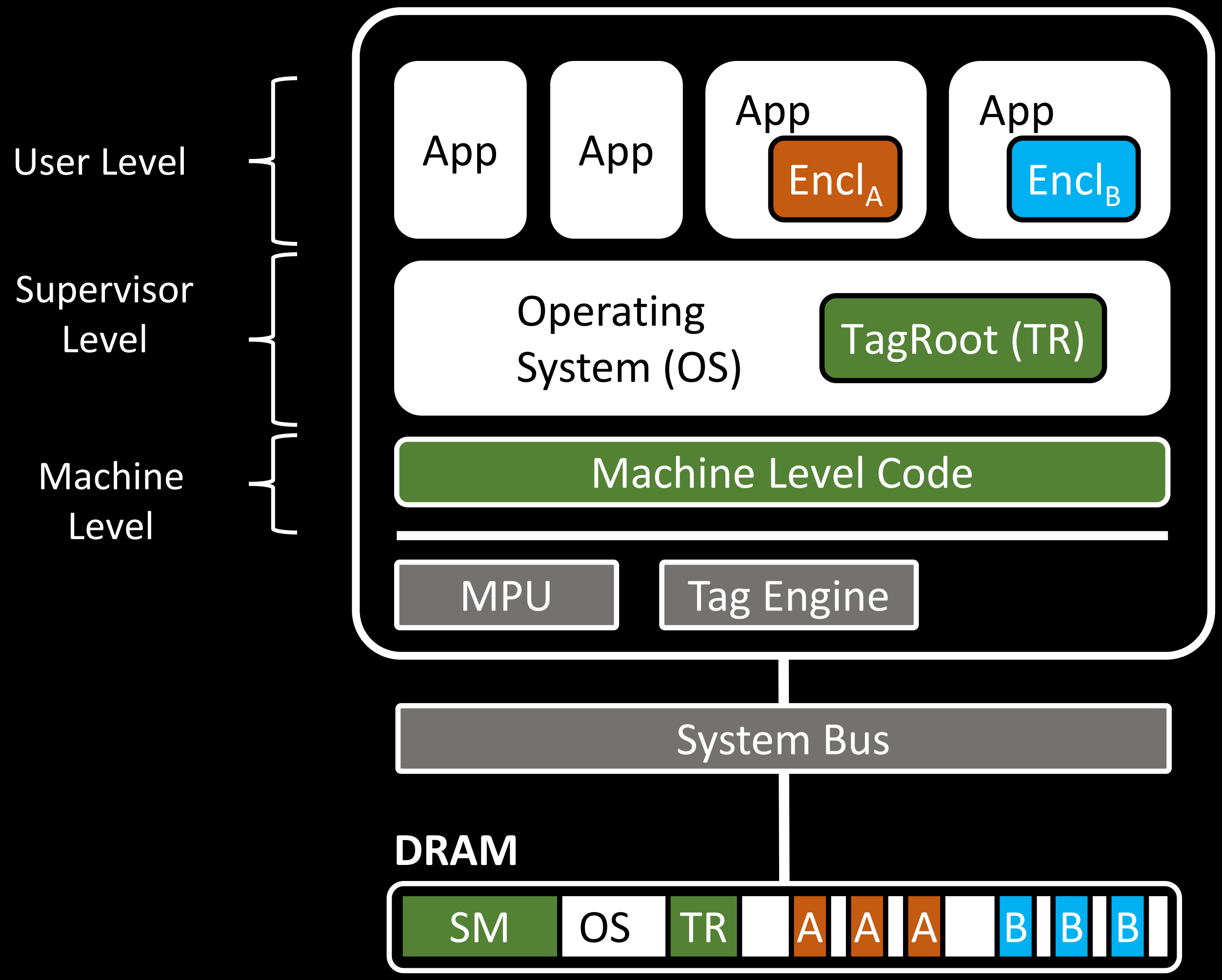


- Isolates sensitive part of app using memory tagging (in-process enclave)
- TagRoot provides trusted services to enclaves and OS (e.g., sealing)

🟩 Software TCB

[Weiser et al., NDSS 2019]

TIMBER-V

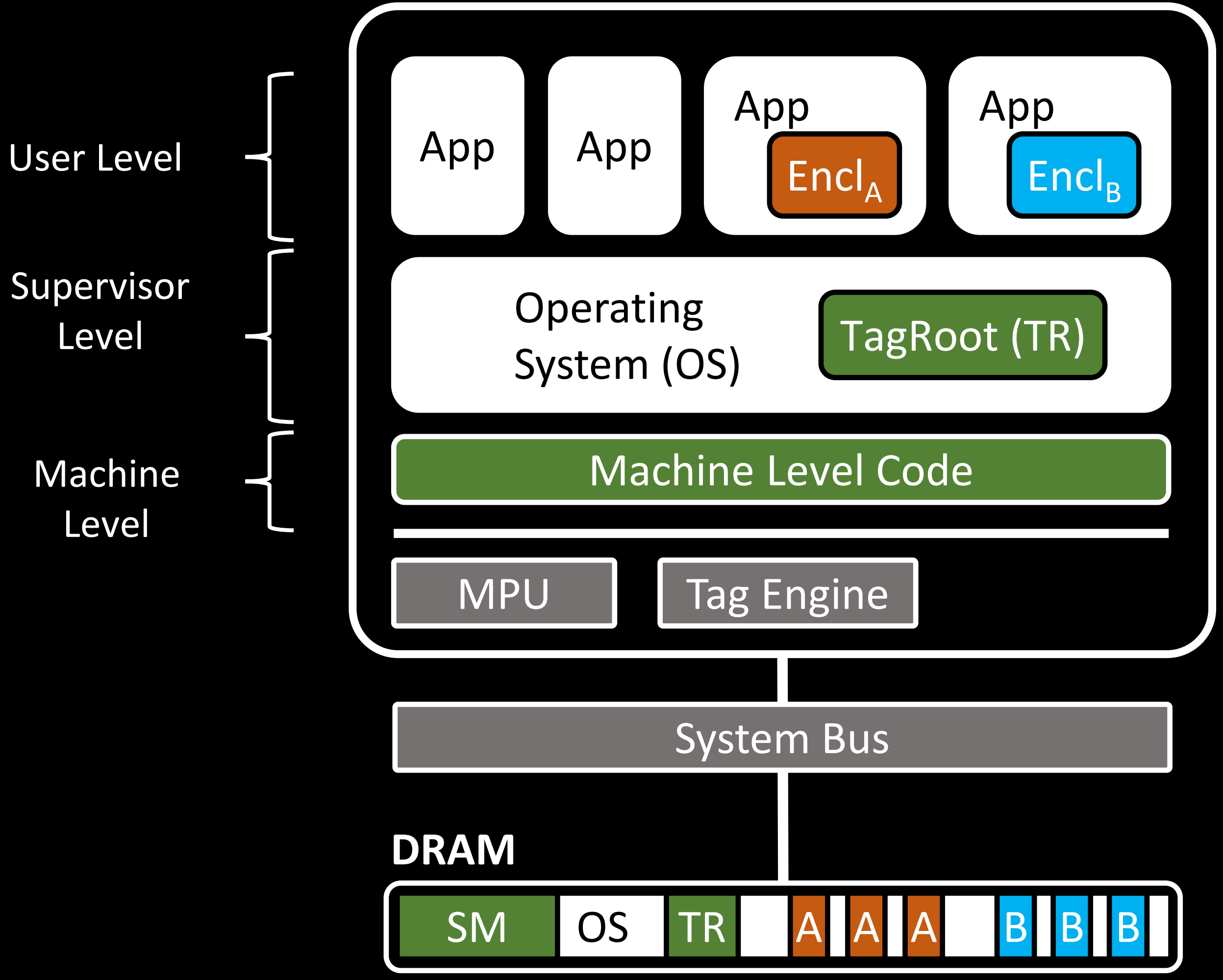


- Isolates sensitive part of app using memory tagging (in-process enclave)
- TagRoot provides trusted services to enclaves and OS (e.g., sealing)
- TagRoot configures custom Memory Protection Unit (MPU)

🟩 Software TCB

[Weiser et al., NDSS 2019]

TIMBER-V

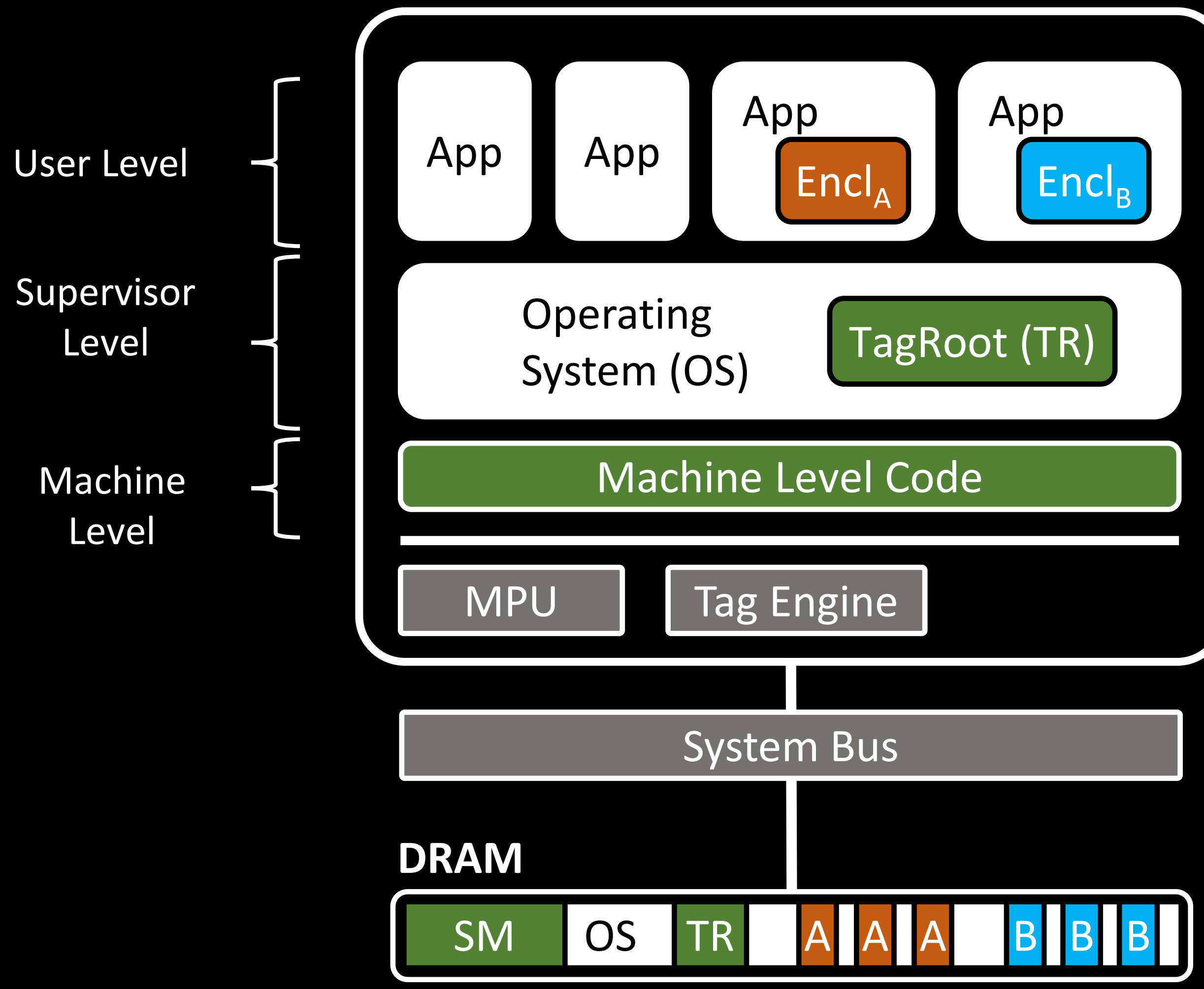


- Isolates sensitive part of app using memory tagging (in-process enclave)
- TagRoot provides trusted services to enclaves and OS (e.g., sealing)
- TagRoot configures custom Memory Protection Unit (MPU)
- Memory access controlled by Tag Engine

🟩 Software TCB

[Weiser et al., NDSS 2019]

TIMBER-V



- Isolates sensitive part of app using memory tagging (in-process enclave)
- TagRoot provides trusted services to enclaves and OS (e.g., sealing)
- TagRoot configures custom Memory Protection Unit (MPU)
- Memory access controlled by Tag Engine
- Cache memory not considered in the design

🟩 Software TCB

Comparison RISC-V Enclave Security Architectures

	Sanctum	Keystone	TIMBER-V
User level enclaves			
User/Supervisor level enclaves			
In-process enclaves			
Dynamic cache side-channel resilience			
Controlled side-channel resilience			
Enclave-to-peripheral binding (MMIO/DMA)			

●	Full feature support
◐	Limited feature support
○	Feature not supported

Comparison RISC-V Enclave Security Architectures

	Sanctum	Keystone	TIMBER-V
User level enclaves	●	○	○
User/Supervisor level enclaves	○	●	○
In-process enclaves	○	○	●
Dynamic cache side-channel resilience			
Controlled side-channel resilience			
Enclave-to-peripheral binding (MMIO/DMA)			

●	Full feature support
◐	Limited feature support
○	Feature not supported

Comparison RISC-V Enclave Security Architectures

	Sanctum	Keystone	TIMBER-V
User level enclaves	●	○	○
User/Supervisor level enclaves	○	●	○
In-process enclaves	○	○	●
Dynamic cache side-channel resilience	◐	●	○
Controlled side-channel resilience			
Enclave-to-peripheral binding (MMIO/DMA)			

●	Full feature support
◐	Limited feature support
○	Feature not supported

Comparison RISC-V Enclave Security Architectures

	Sanctum	Keystone	TIMBER-V
User level enclaves	●	○	○
User/Supervisor level enclaves	○	●	○
In-process enclaves	○	○	●
Dynamic cache side-channel resilience	◐	●	○
Controlled side-channel resilience	●	●	◐
Enclave-to-peripheral binding (MMIO/DMA)			

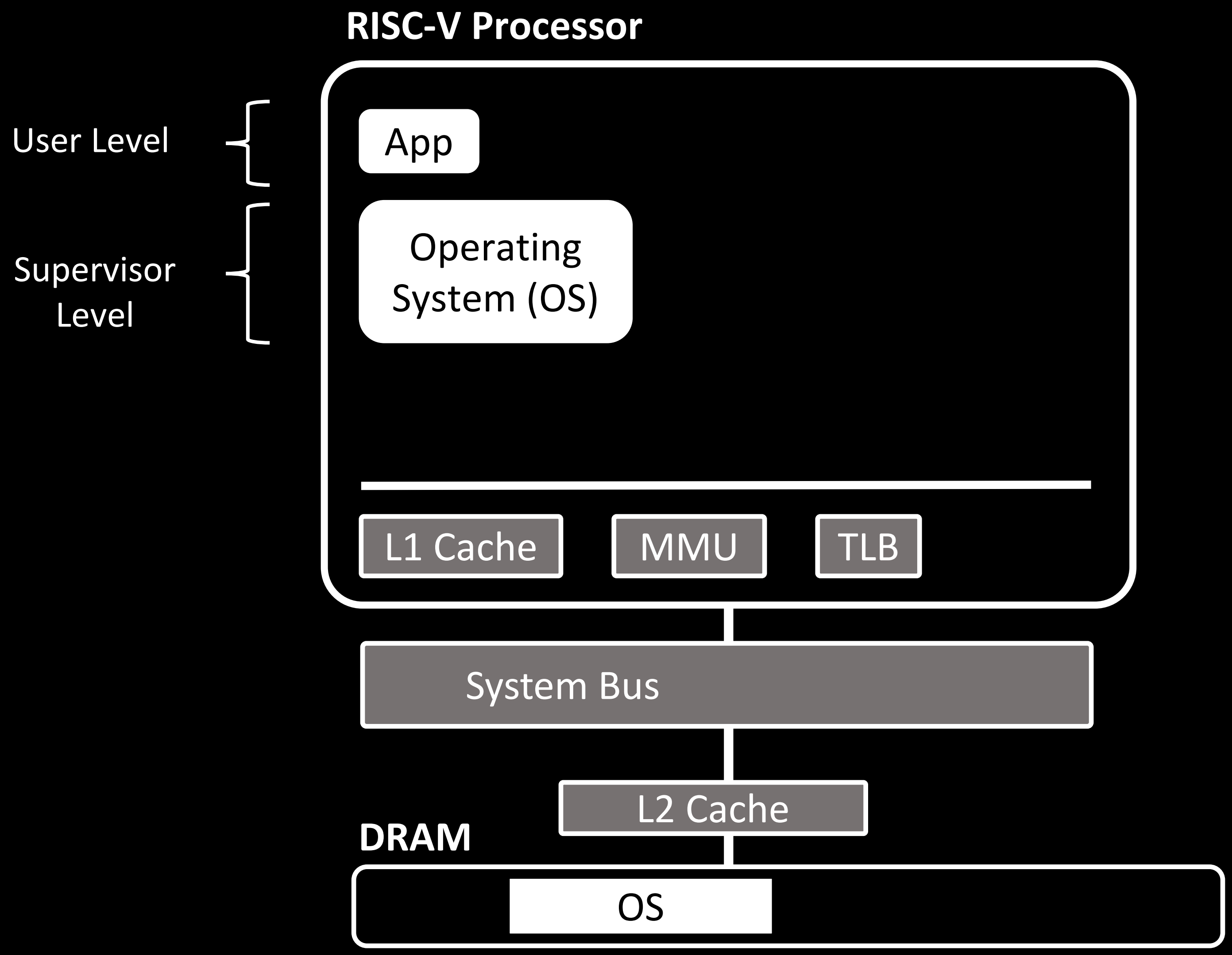
●	Full feature support
◐	Limited feature support
○	Feature not supported

Comparison RISC-V Enclave Security Architectures

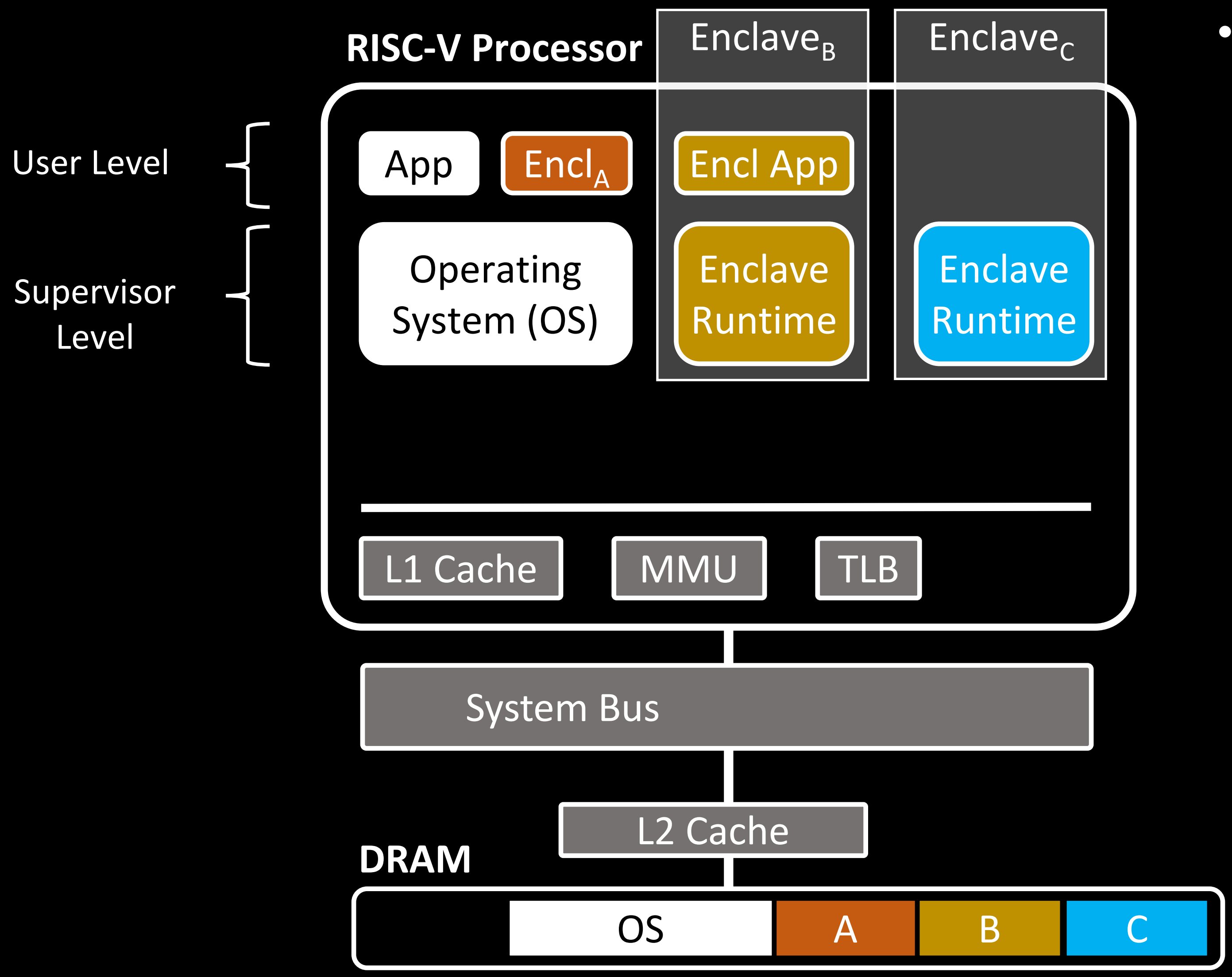
	Sanctum	Keystone	TIMBER-V
User level enclaves	●	○	○
User/Supervisor level enclaves	○	●	○
In-process enclaves	○	○	●
Dynamic cache side-channel resilience	◐	●	○
Controlled side-channel resilience	●	●	◐
Enclave-to-peripheral binding (MMIO/DMA)	○	○	○

●	Full feature support
◐	Limited feature support
○	Feature not supported

CURE: Customizable and Resilient Enclaves

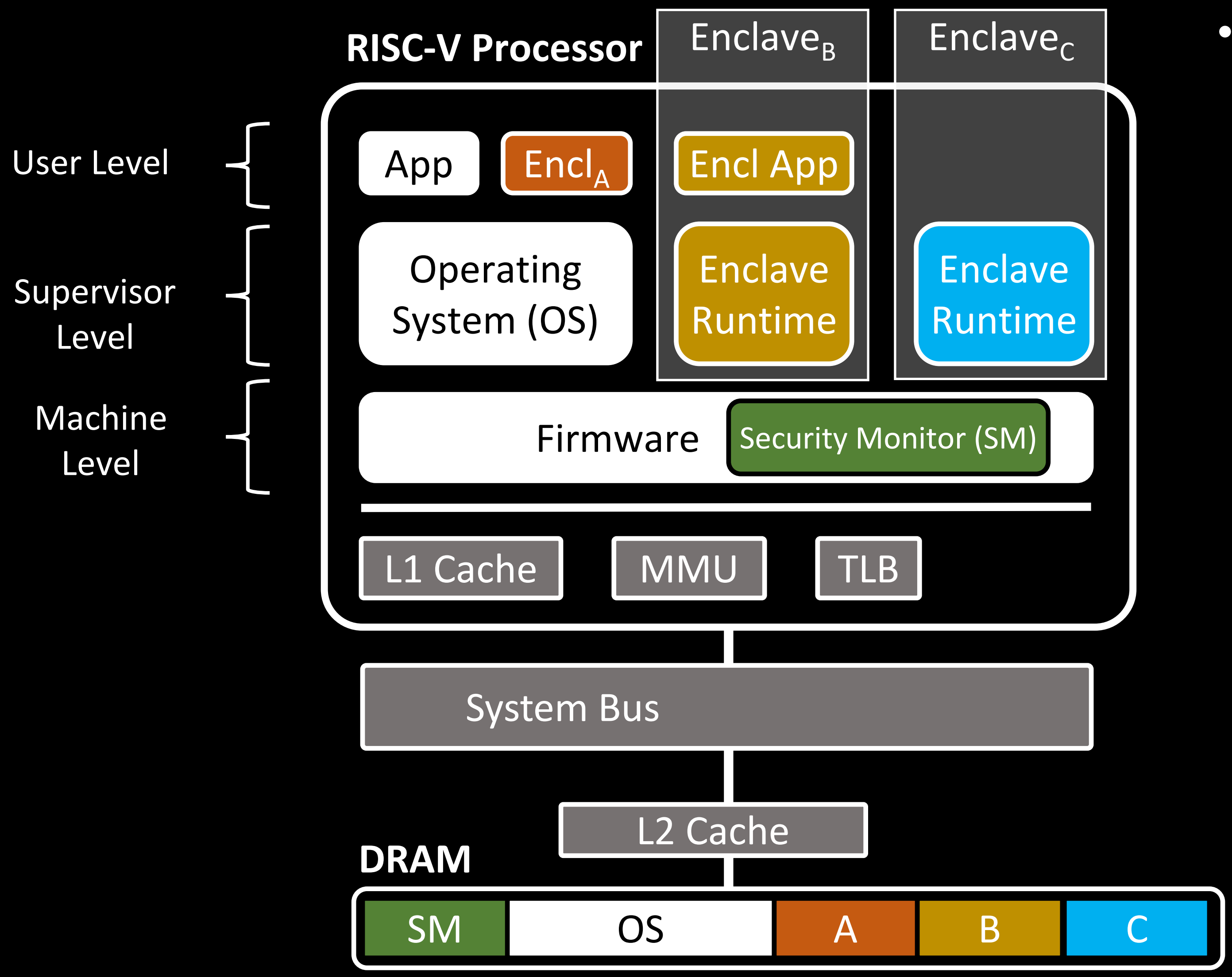


CURE: Customizable and Resilient Enclaves



- Multiple *types* of enclaves, in user level, supervisor level or in-process

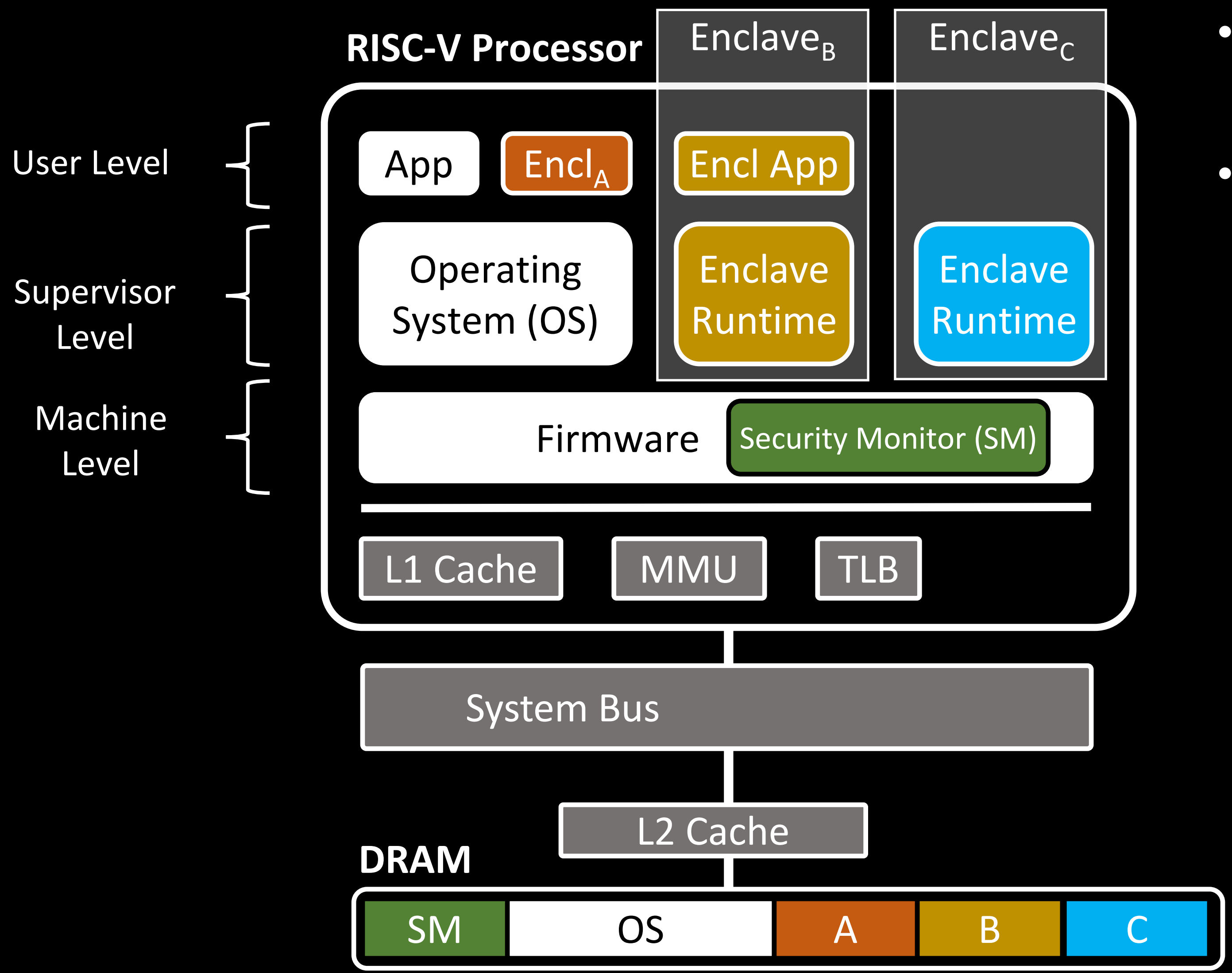
CURE: Customizable and Resilient Enclaves



- Multiple *types* of enclaves, in user level, supervisor level or in-process

🟩 Software TCB

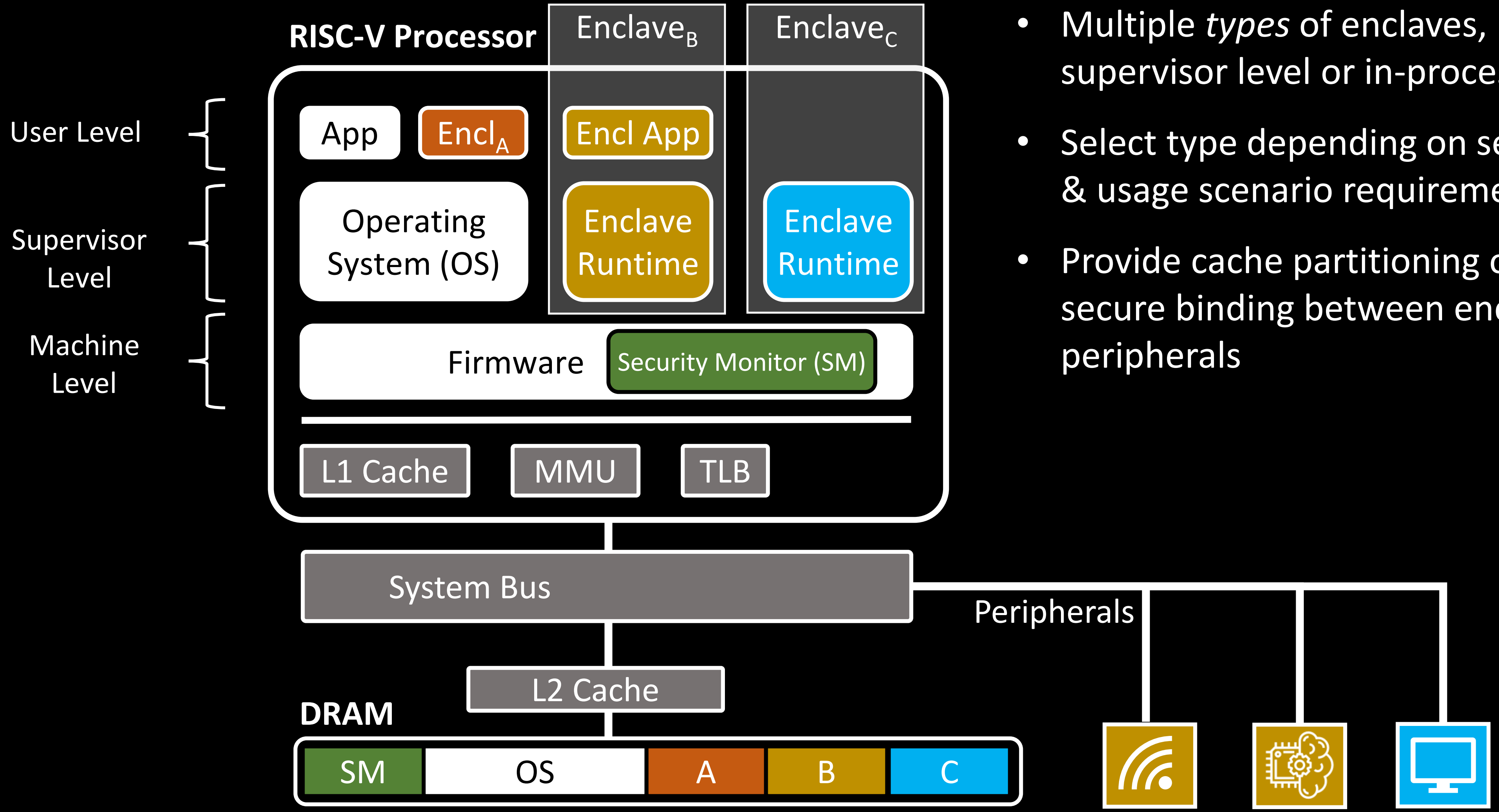
CURE: Customizable and Resilient Enclaves



- Multiple *types* of enclaves, in user level, supervisor level or in-process
- Select type depending on sensitive service & usage scenario requirements

🟩 Software TCB

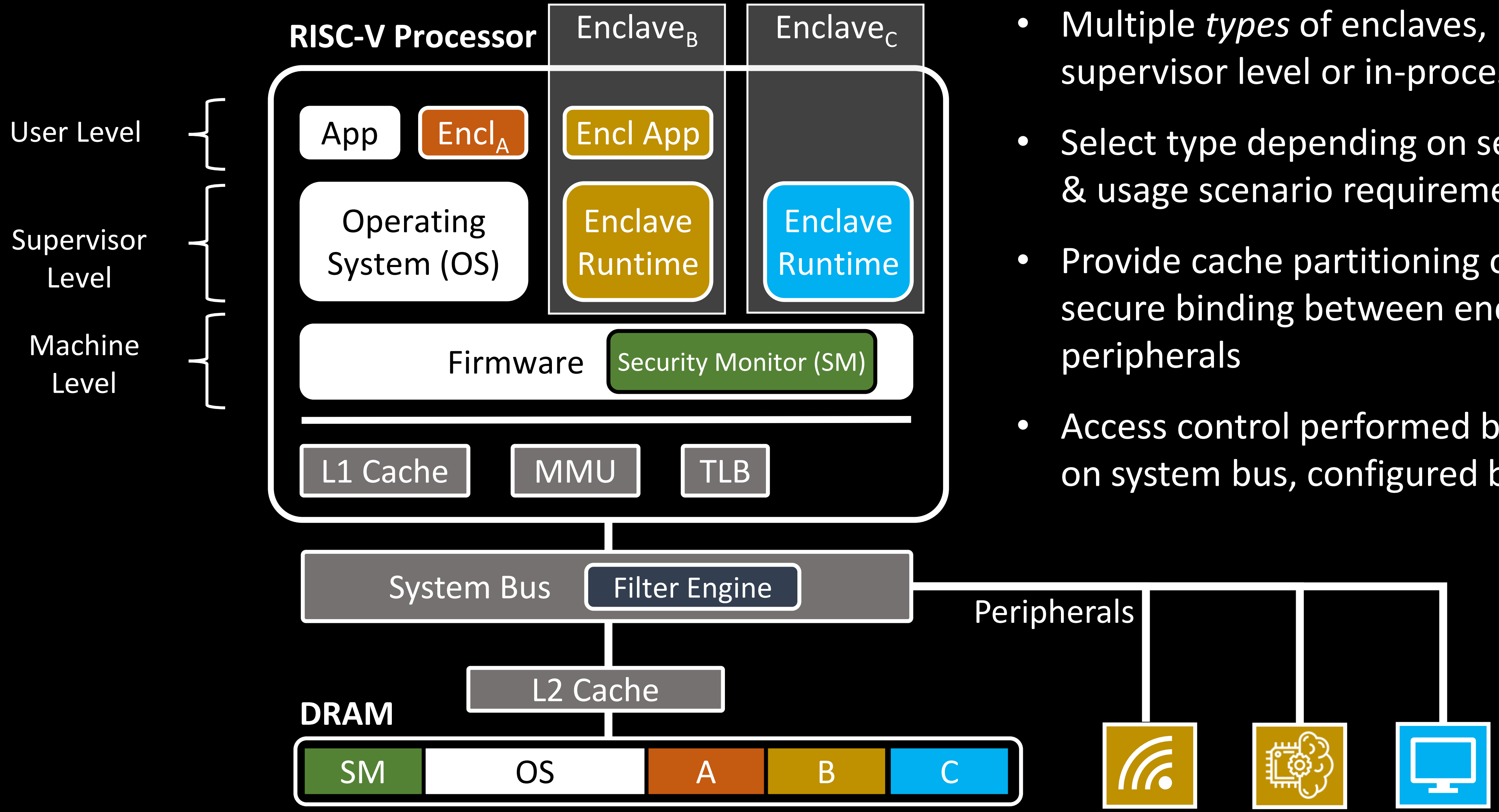
CURE: Customizable and Resilient Enclaves



- Multiple *types* of enclaves, in user level, supervisor level or in-process
- Select type depending on sensitive service & usage scenario requirements
- Provide cache partitioning on L2 cache and secure binding between enclaves and peripherals

🟩 Software TCB

CURE: Customizable and Resilient Enclaves



- Multiple *types* of enclaves, in user level, supervisor level or in-process
- Select type depending on sensitive service & usage scenario requirements
- Provide cache partitioning on L2 cache and secure binding between enclaves and peripherals
- Access control performed by filter engine on system bus, configured by SM

🟩 Software TCB

Conclusion

Conclusion

- RISC-V enabled several novel security architectures

Conclusion

- RISC-V enabled several novel security architectures
- Sensitive services and usage scenarios are highly diverse (threat model, performance, required functionality)

Conclusion

- RISC-V enabled several novel security architectures
- Sensitive services and usage scenarios are highly diverse (threat model, performance, required functionality)
- Goal: Customizable enclave adapt to sensitive services

Conclusion

- RISC-V enabled several novel security architectures
- Sensitive services and usage scenarios are highly diverse
  (threat model, performance, required functionality)
- Goal: Customizable enclave adapt to sensitive services
- We tackled this problem with CURE by providing multiple enclave *types*

Conclusion

- RISC-V enabled several novel security architectures
- Sensitive services and usage scenarios are highly diverse
  (threat model, performance, required functionality)
- Goal: Customizable enclave adapt to sensitive services
- We tackled this problem with CURE by providing multiple enclave *types*
- Research proposals don't ignore side-channel attacks

Conclusion

- RISC-V enabled several novel security architectures
- Sensitive services and usage scenarios are highly diverse (threat model, performance, required functionality)
- Goal: Customizable enclave adapt to sensitive services
- We tackled this problem with CURE by providing multiple enclave *types*
- Research proposals don't ignore side-channel attacks
- Open Challenges:

Conclusion

- RISC-V enabled several novel security architectures
- Sensitive services and usage scenarios are highly diverse
  (threat model, performance, required functionality)
- Goal: Customizable enclave adapt to sensitive services
- We tackled this problem with CURE by providing multiple enclave *types*
- Research proposals don’t ignore side-channel attacks
- Open Challenges:
  - Low-overhead (performance and memory) side-channel resilient cache architecture for enclaves

Conclusion

- RISC-V enabled several novel security architectures
- Sensitive services and usage scenarios are highly diverse (threat model, performance, required functionality)
- Goal: Customizable enclave adapt to sensitive services
- We tackled this problem with CURE by providing multiple enclave *types*
- Research proposals don't ignore side-channel attacks
- Open Challenges:
  - Low-overhead (performance and memory) side-channel resilient cache architecture for enclaves
  - Enclave architectures for Network-on-Chip platforms

Questions ?

emmanuel.stapf@trust.tu-darmstadt.de