





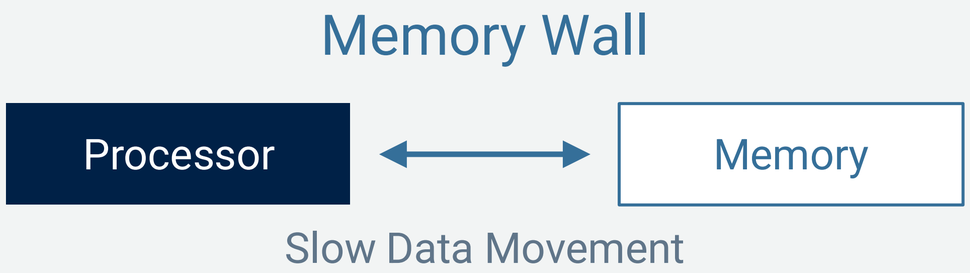
SecPIMlet: Confidential Computing for Spatio-temporally Shared Processing-in-Memory

Sunho Lee* (University of Oxford / KAIST)
Minwoo Noh, Jeongwon Choi, Kwanghoon Choi, Igjae Kim (KAIST)
Chang Hyun Park (Uppsala University), Jaehyuk Huh (KAIST)

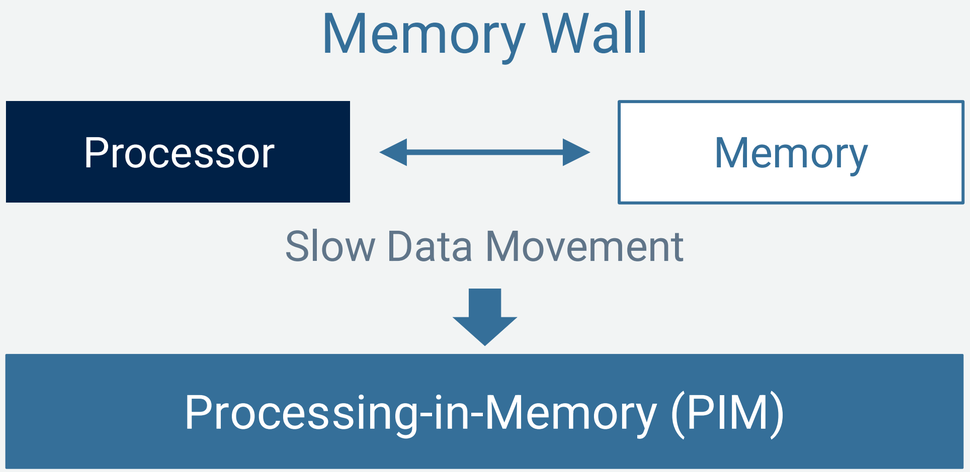
*This work was done at KAIST

SecPIMlet: Breaking the AI memory wall securely

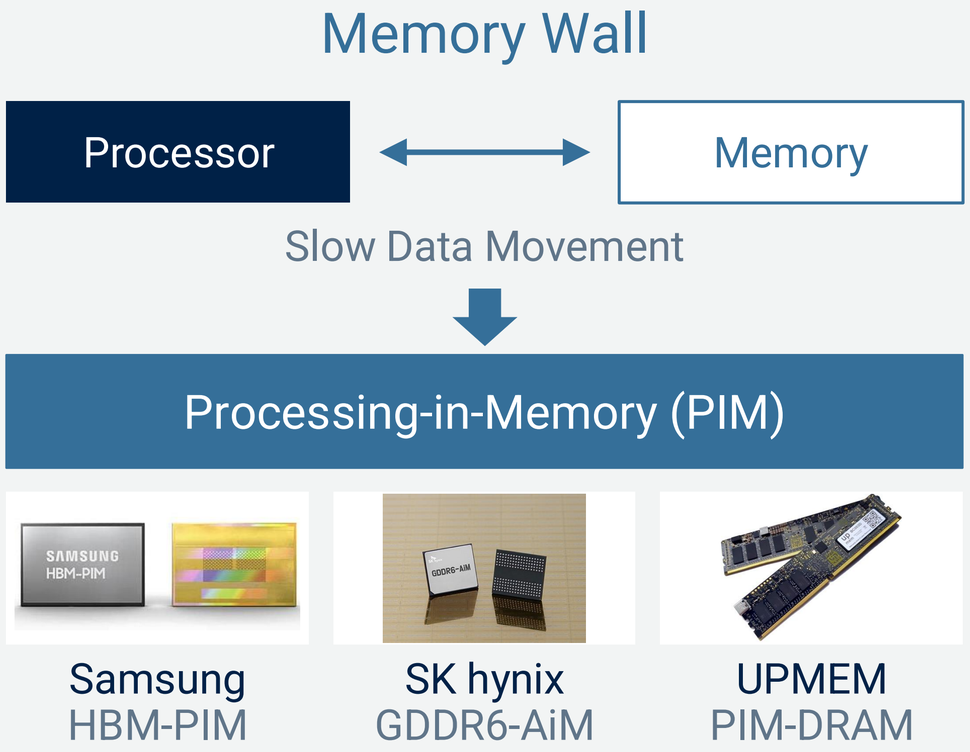
SecPIMlet: Breaking the AI memory wall securely



SecPIMlet: Breaking the AI memory wall securely

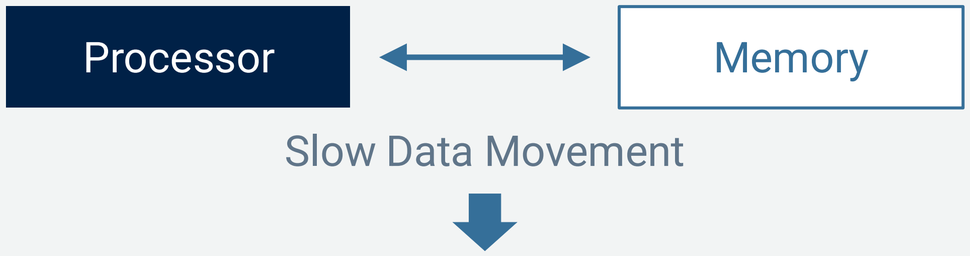


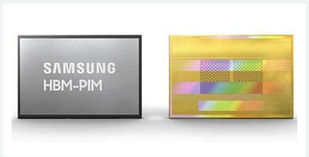
SecPIMlet: Breaking the AI memory wall securely



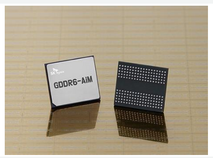
SecPIMlet: Breaking the AI memory wall securely

Memory Wall





Samsung
HBM-PIM

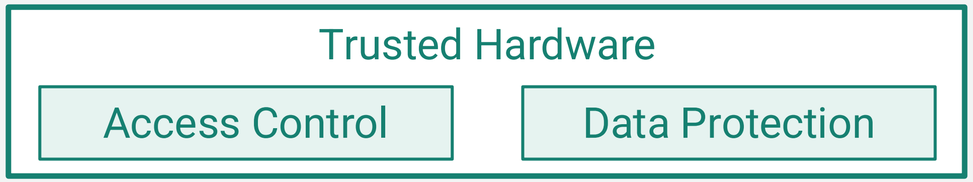


SK hynix
GDDR6-AiM



UPMEM
PIM-DRAM

Confidential AI System



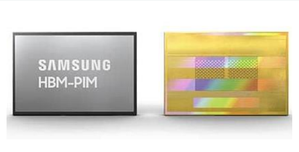
SecPIMlet: Breaking the AI memory wall securely

Memory Wall

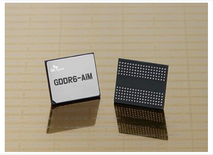
Processor ⟷ Memory

Slow Data Movement

Processing-in-Memory (PIM)



Samsung
HBM-PIM



SK hynix
GDDR6-AiM



UPMEM
PIM-DRAM

Confidential AI System

Trusted Hardware

Access Control | Data Protection

Hardware-based Confidential Computing

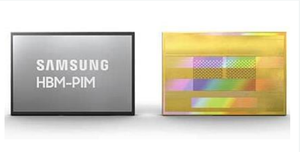
SecPIMlet: Breaking the AI memory wall securely

Memory Wall

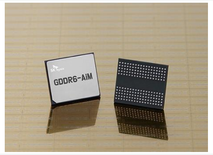
Processor ⟷ Memory

Slow Data Movement

Processing-in-Memory (PIM)



Samsung
HBM-PIM



SK hynix
GDDR6-AiM



UPMEM
PIM-DRAM

Confidential AI System

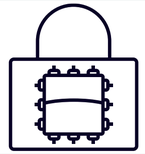
Trusted Hardware

Access Control | Data Protection

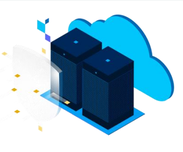
Hardware-based Confidential Computing



NVIDIA
Confidential Computing



Arm
CCA



Intel
TDX

SecPIMlet: Breaking the AI memory wall securely

Memory Wall

Processor

Memory

Slow Data Movement

Confidential AI System

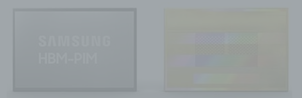
Trusted Hardware

Access Control

Data Protection

***SecPIMlet*: Confidential PIM**

PIM + Hardware-based Confidential Computing



Samsung
HBM-PIM



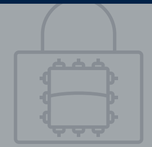
SK hynix
GDDR6-AiM



UPMEM
PIM-DRAM



NVIDIA
Confidential
Computing



Arm
CCA



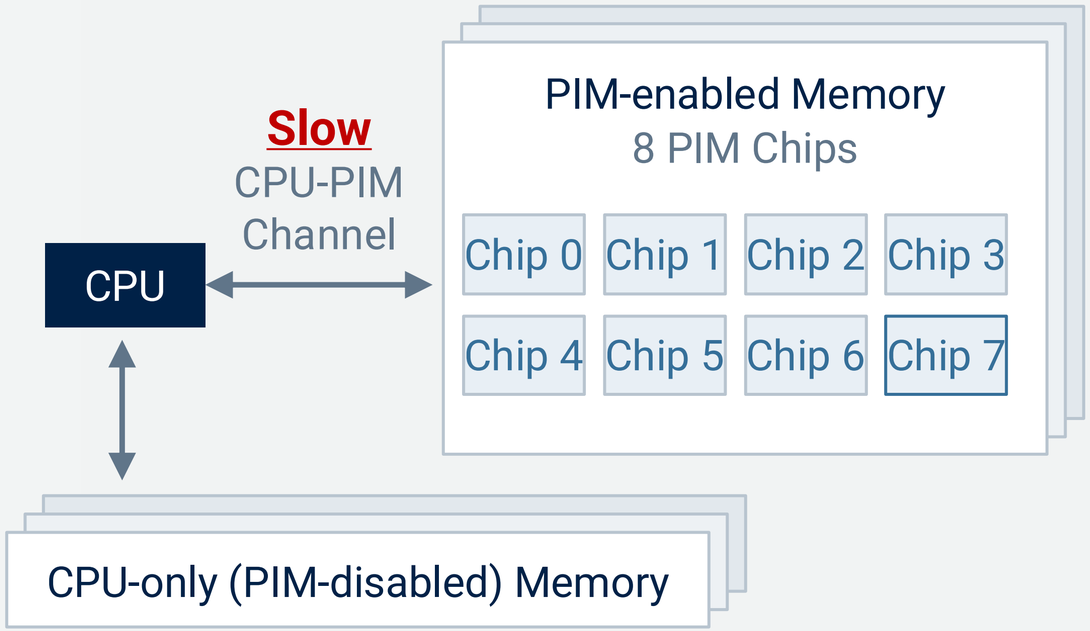
Intel
TDX

General-purpose PIM

UPMEM architecture

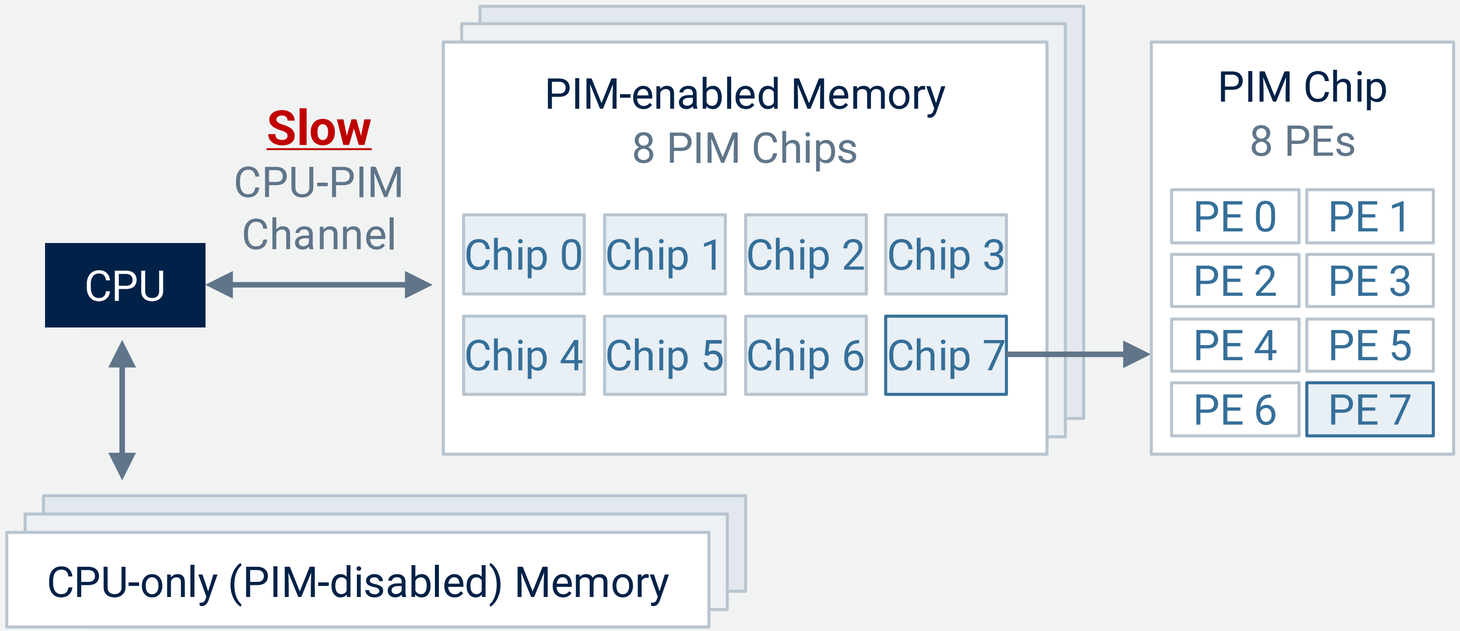
General-purpose PIM

UPMEM architecture



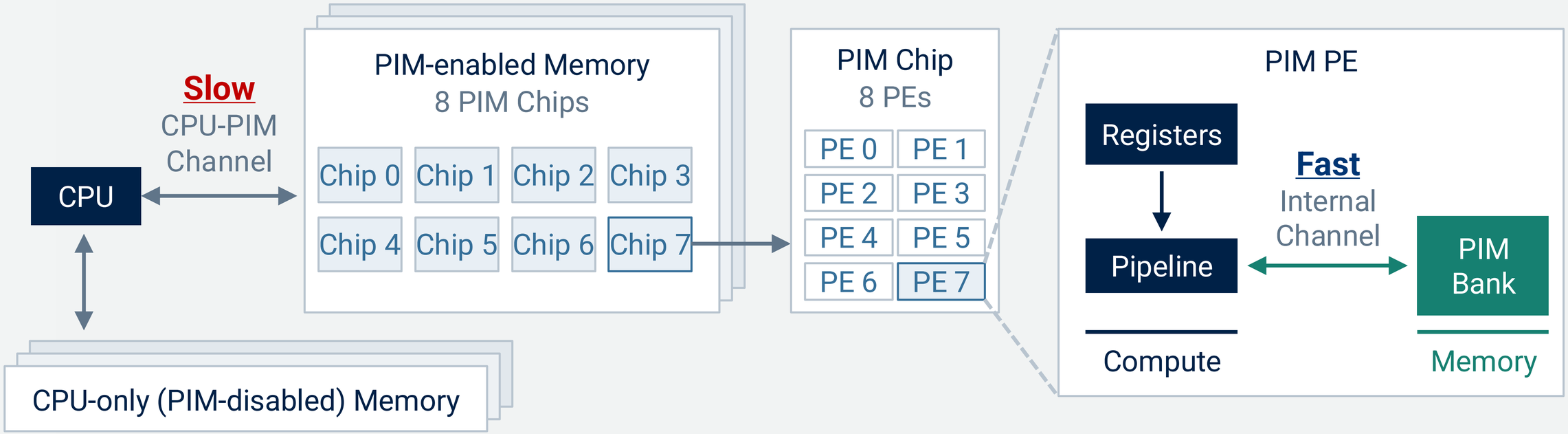
General-purpose PIM

UPMEM architecture



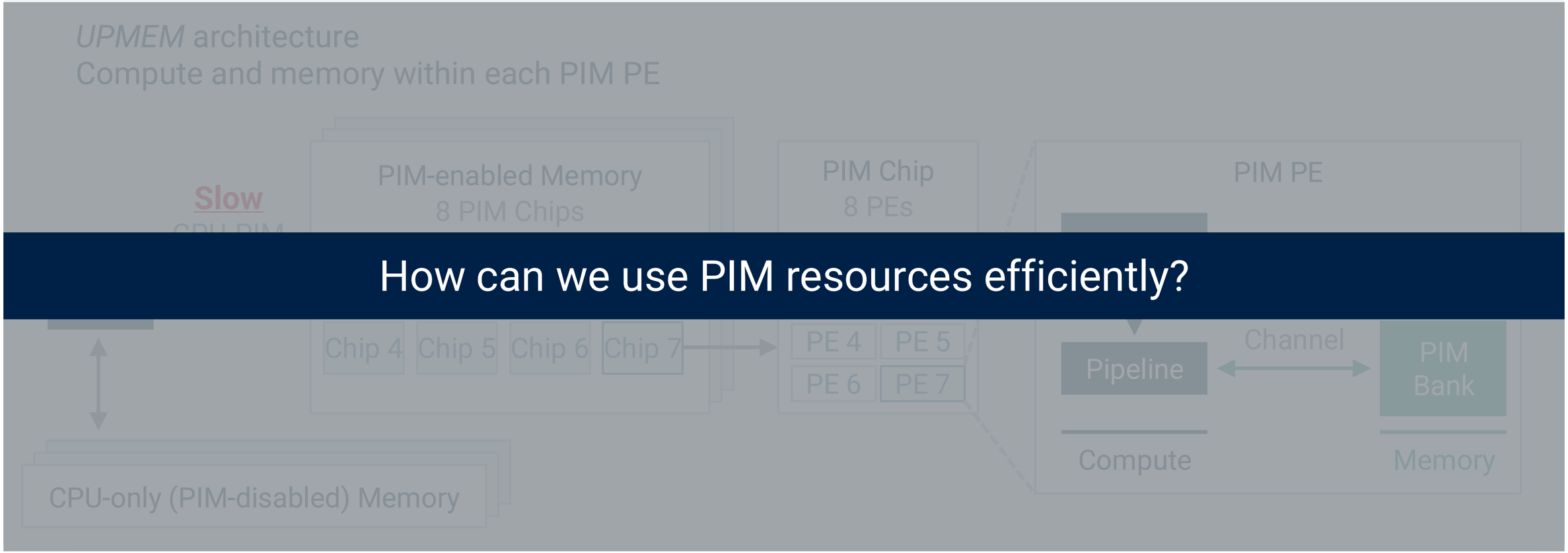
General-purpose PIM

UPMEM architecture
Compute and memory within each PIM PE



General-purpose PIM

UPMEM architecture
Compute and memory within each PIM PE

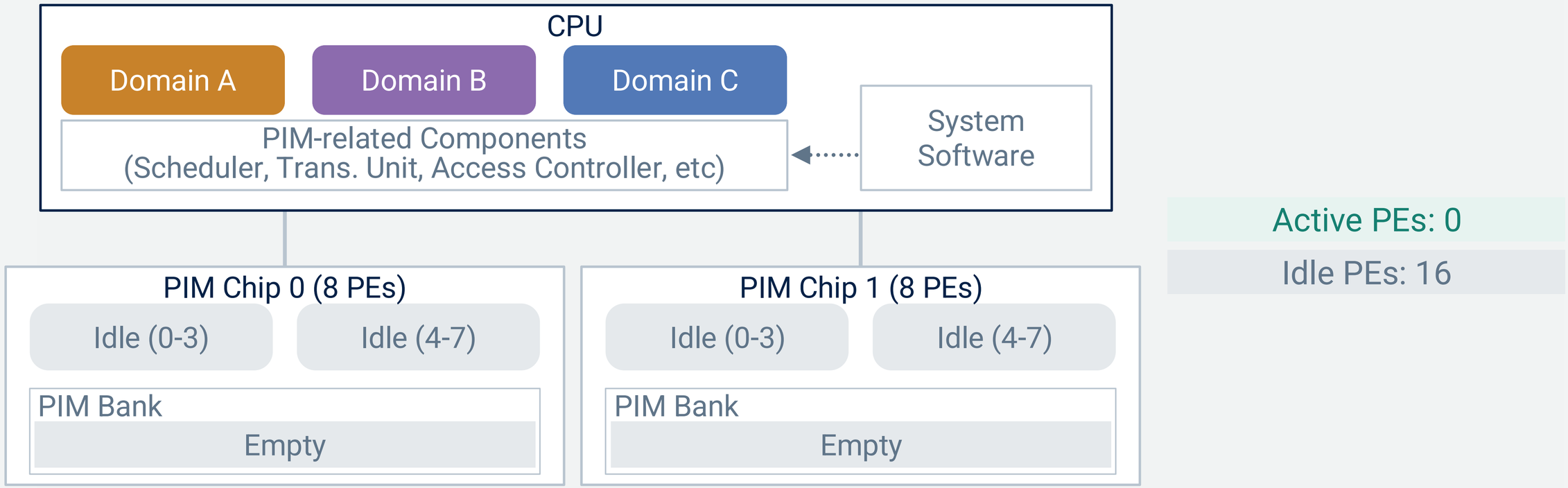


How can we use PIM resources efficiently?

Multi-domain PIM with spatial & temporal sharing

Multi-domain PIM with spatial & temporal sharing

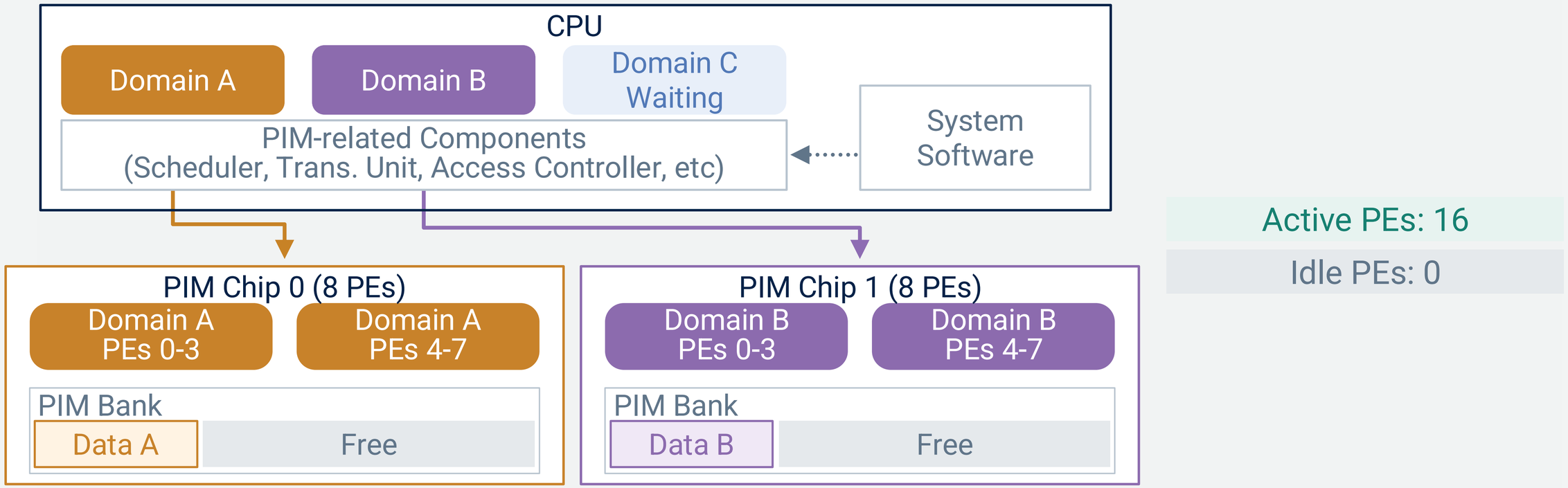
Three domains with equal-size kernels



Multi-domain PIM with spatial & temporal sharing

Prior 1: Spatial-only sharing (*Virtual PIM* [*PACT 2023*])

Time (t) = 1



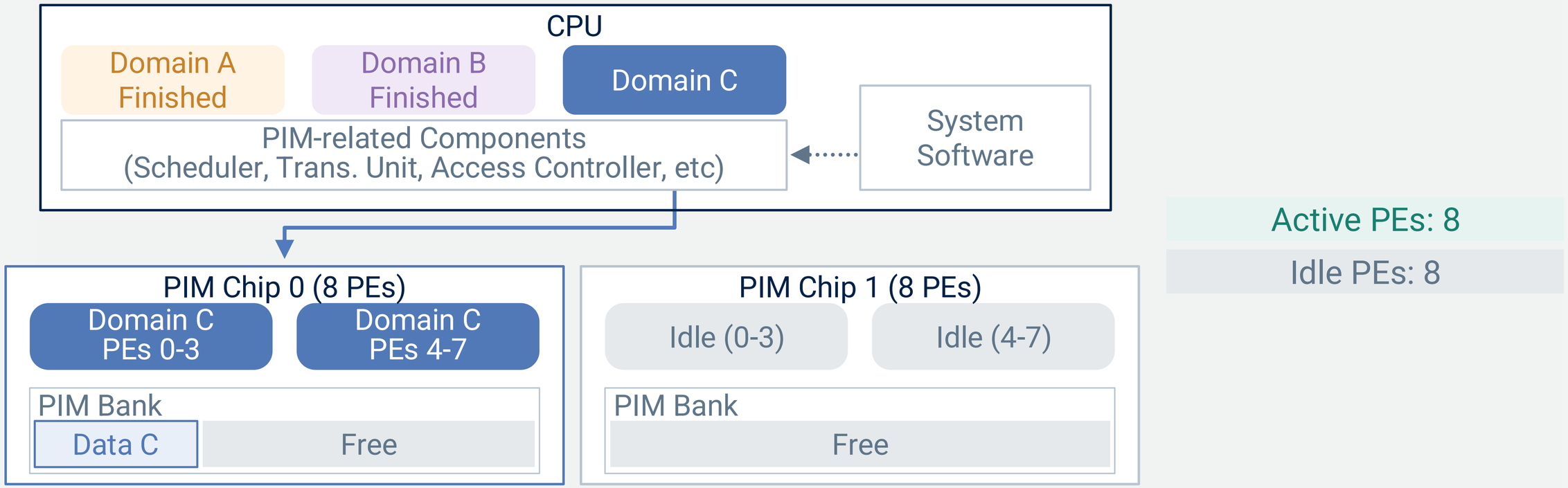
Active PEs: 16

Idle PEs: 0

Multi-domain PIM with spatial & temporal sharing

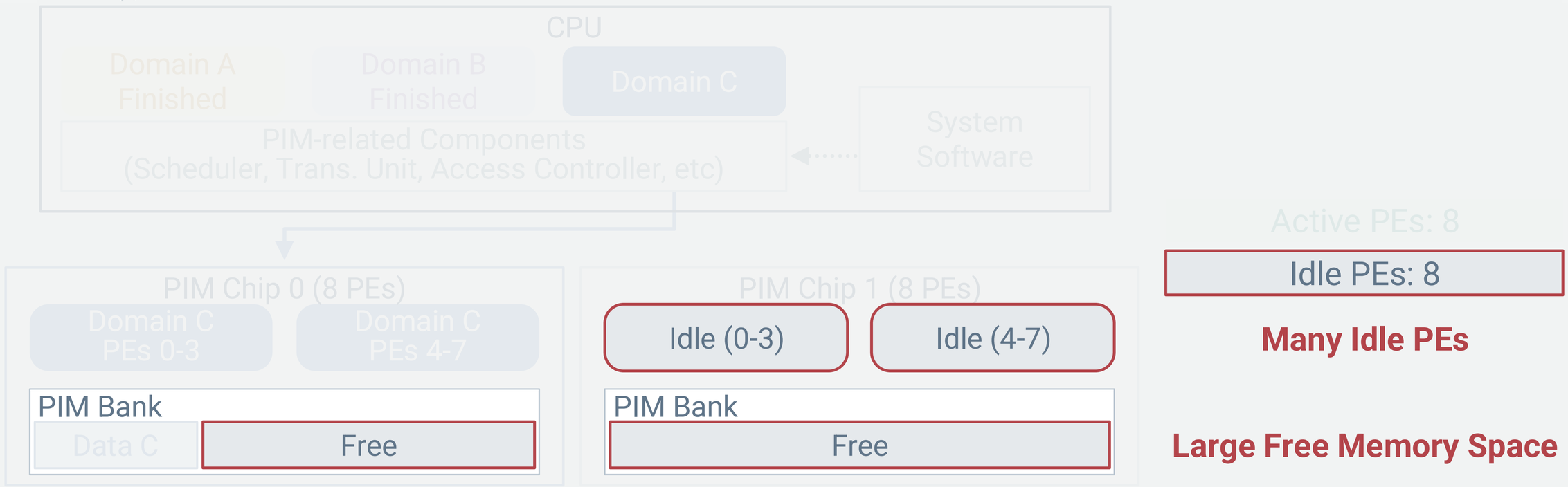
Prior 1: Spatial-only sharing (*Virtual PIM* [*PACT 2023*])

Time (t) = 2



Multi-domain PIM with spatial & temporal sharing

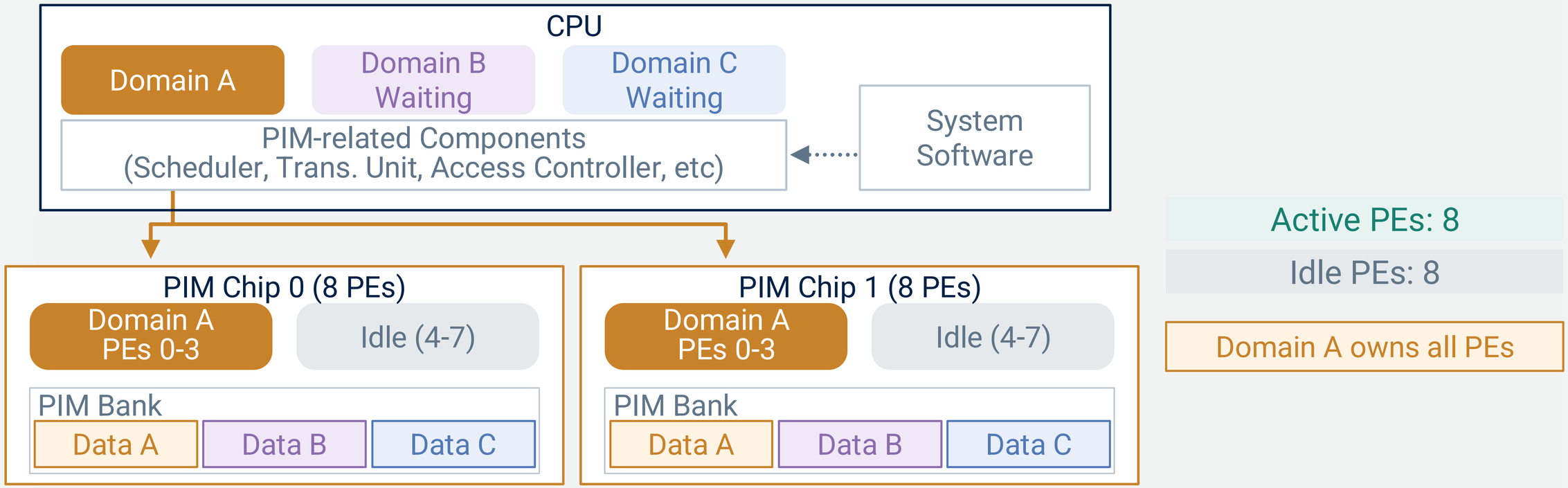
Prior 1: Spatial-only sharing (*Virtual PIM* [*PACT 2023*])



Multi-domain PIM with spatial & temporal sharing

Prior 2: Temporal-only sharing (*UM-PIM* [*ISCA 2024*])

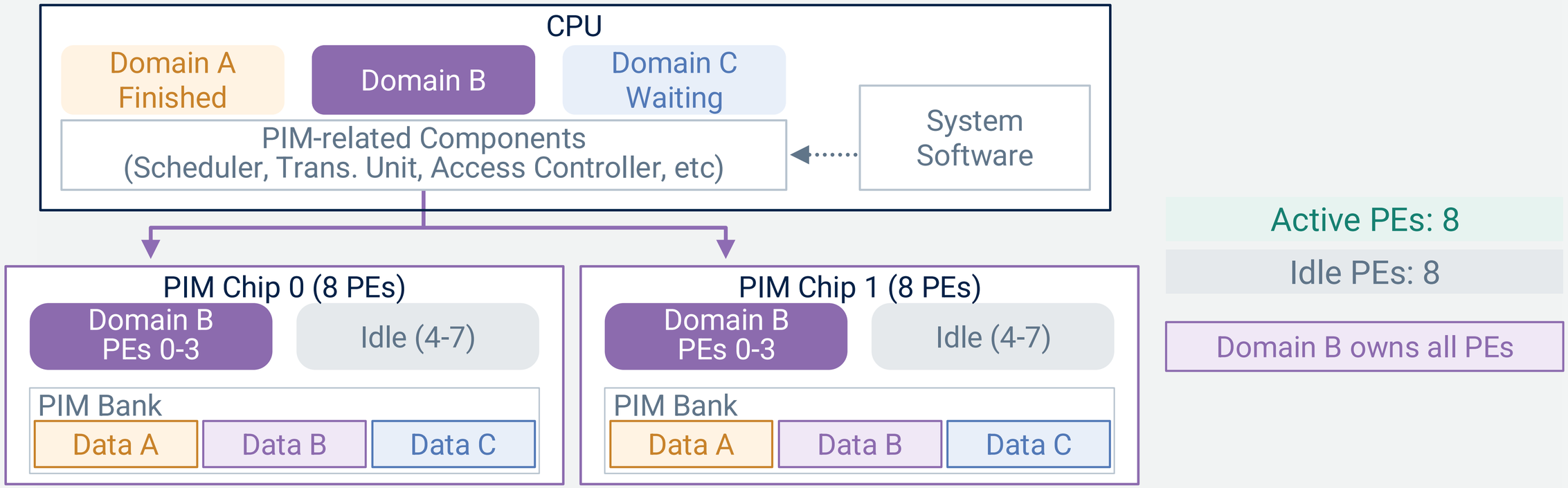
Time (t) = 1



Multi-domain PIM with spatial & temporal sharing

Prior 2: Temporal-only sharing (*UM-PIM* [*ISCA 2024*])

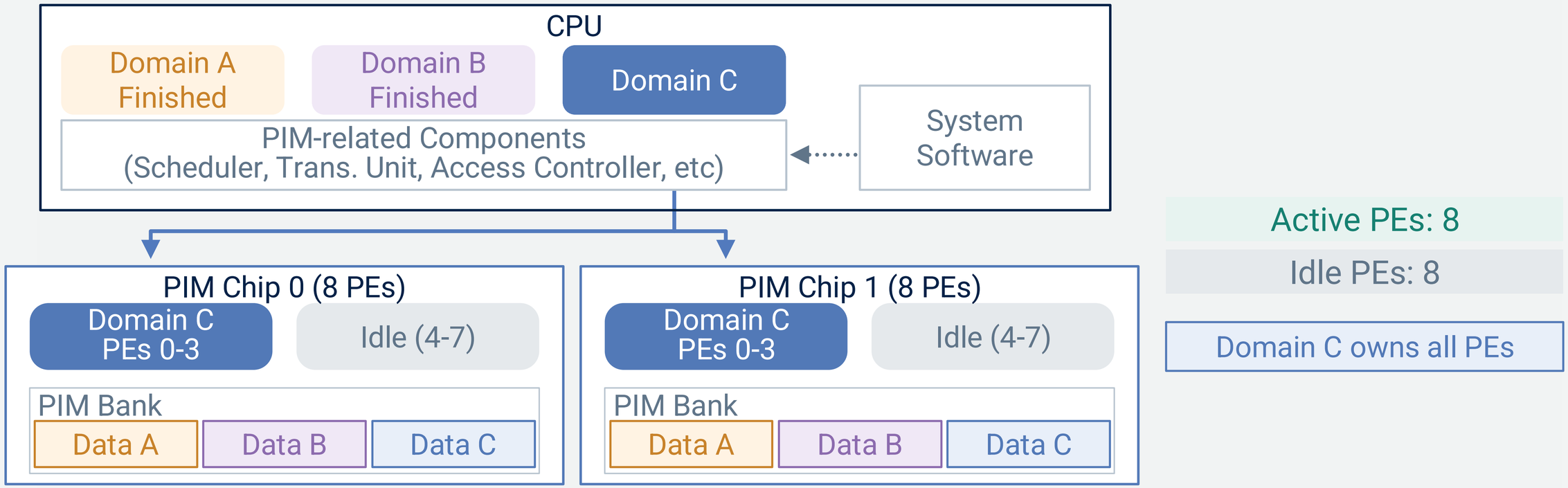
Time (t) = 2



Multi-domain PIM with spatial & temporal sharing

Prior 2: Temporal-only sharing (*UM-PIM* [*ISCA 2024*])

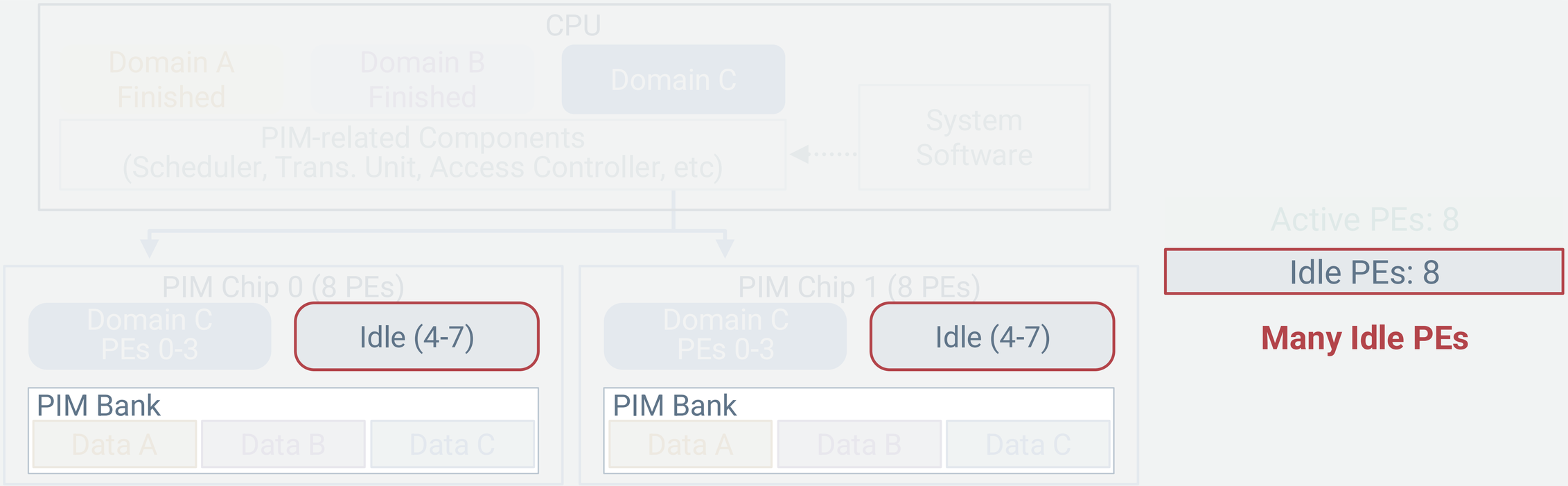
Time (t) = 3



Multi-domain PIM with spatial & temporal sharing

Prior 2: Temporal-only sharing (*UM-PIM* [*ISCA 2024*])

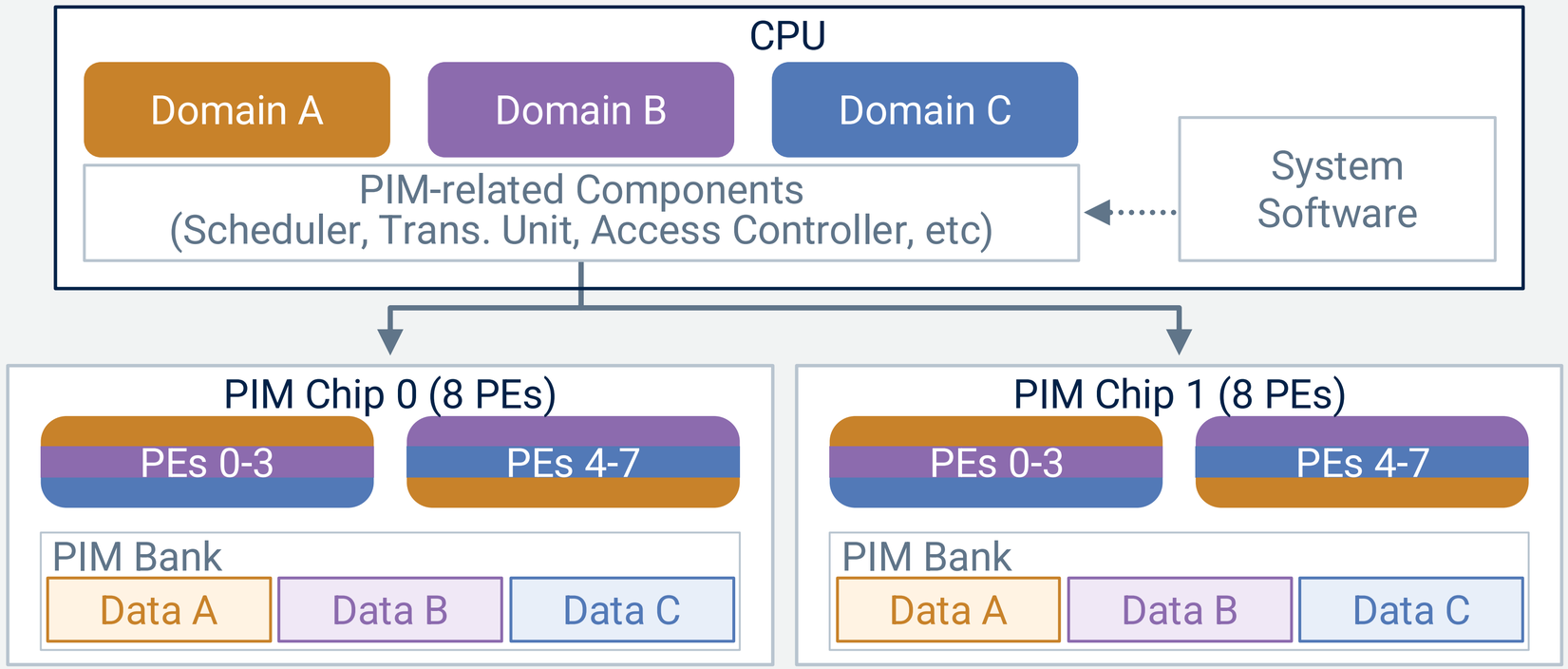
Time (t) = 3



Multi-domain PIM with spatial & temporal sharing

Ours: Spatio-temporal sharing (*SecPIMlet*)

Across time slices



Active PEs: 16

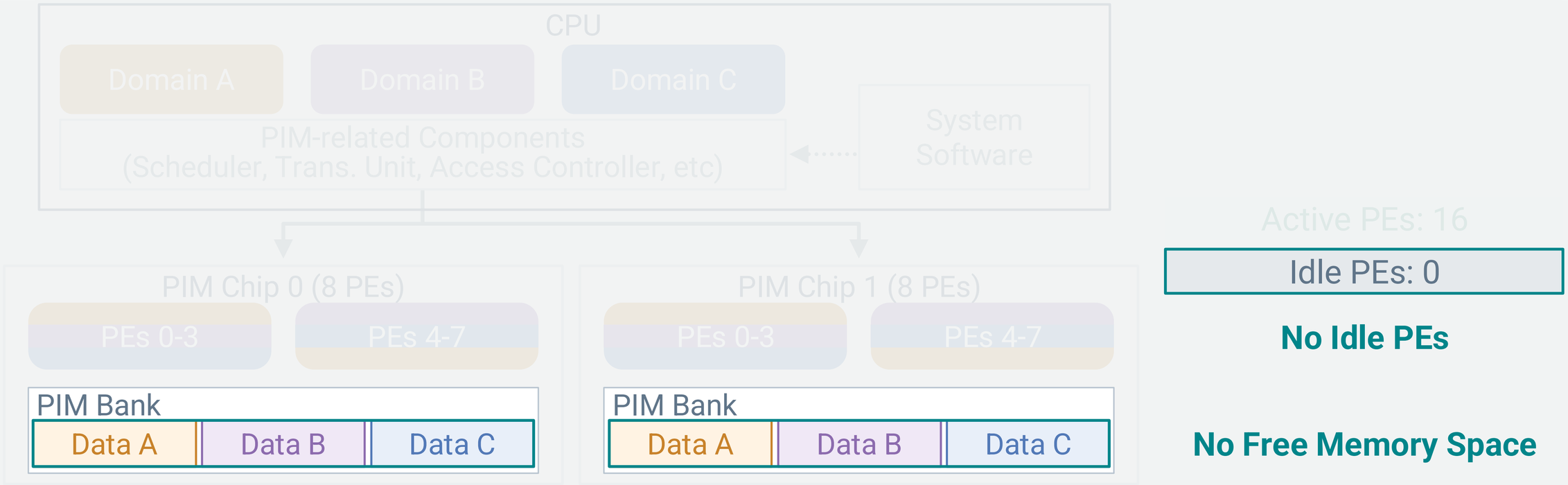
Idle PEs: 0

Domains share PEs over time

Multi-domain PIM with spatial & temporal sharing

Ours: Spatio-temporal sharing (*SecPIMlet*)

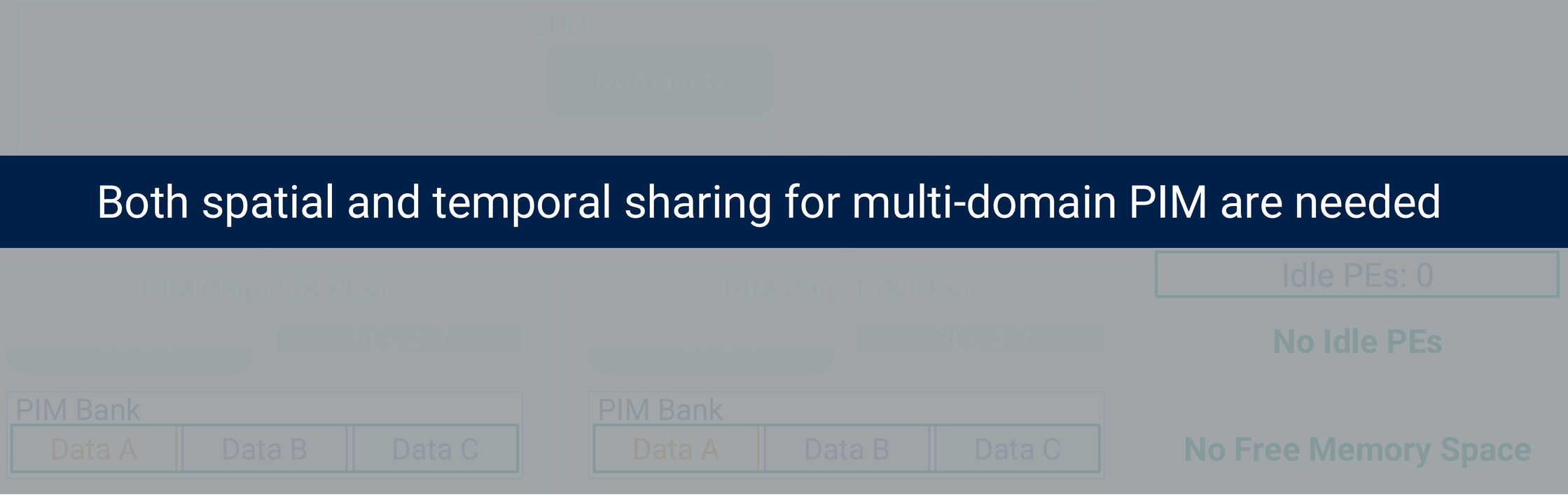
Across time slices



Multi-domain PIM with spatial & temporal sharing

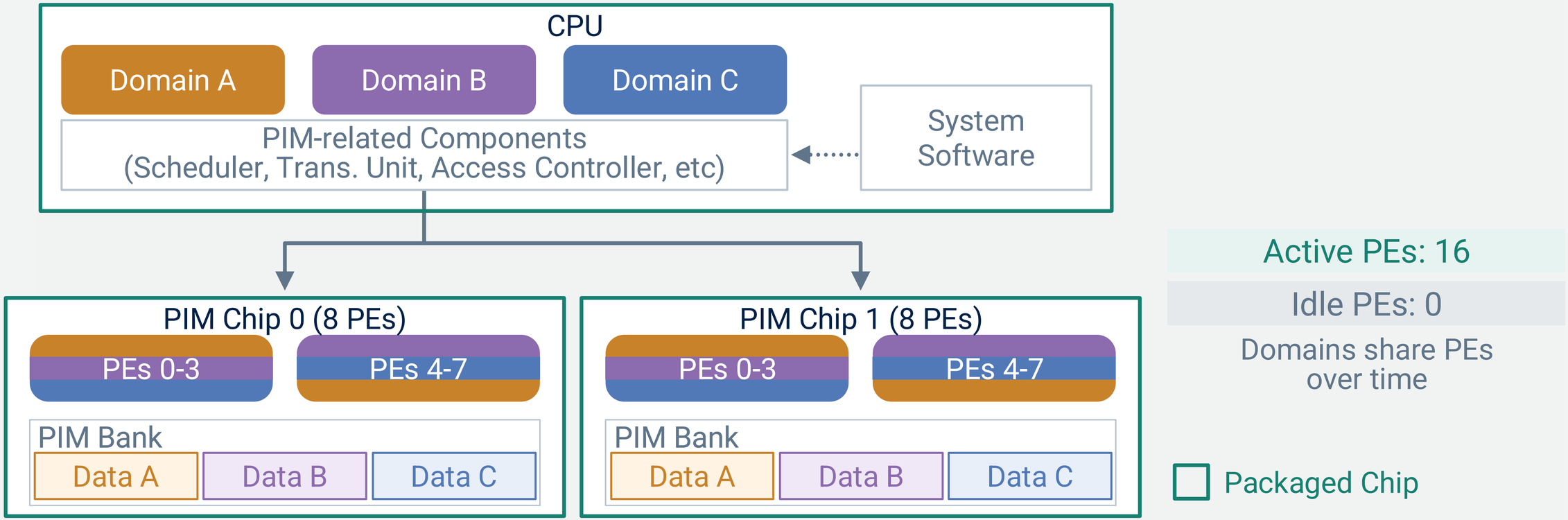
Ours: Spatio-temporal sharing (*SecPIMlet*)

Across time slices



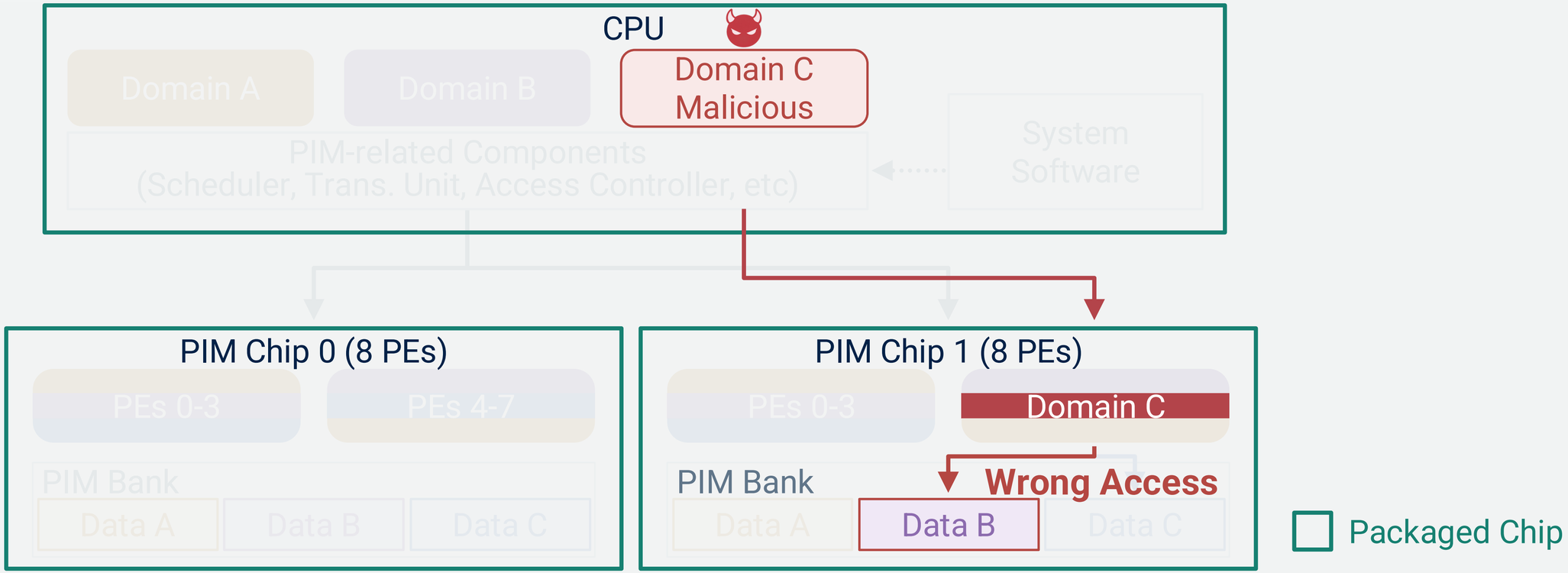
Both spatial and temporal sharing for multi-domain PIM are needed

Security risks in spatio-temporal multi-domain PIM



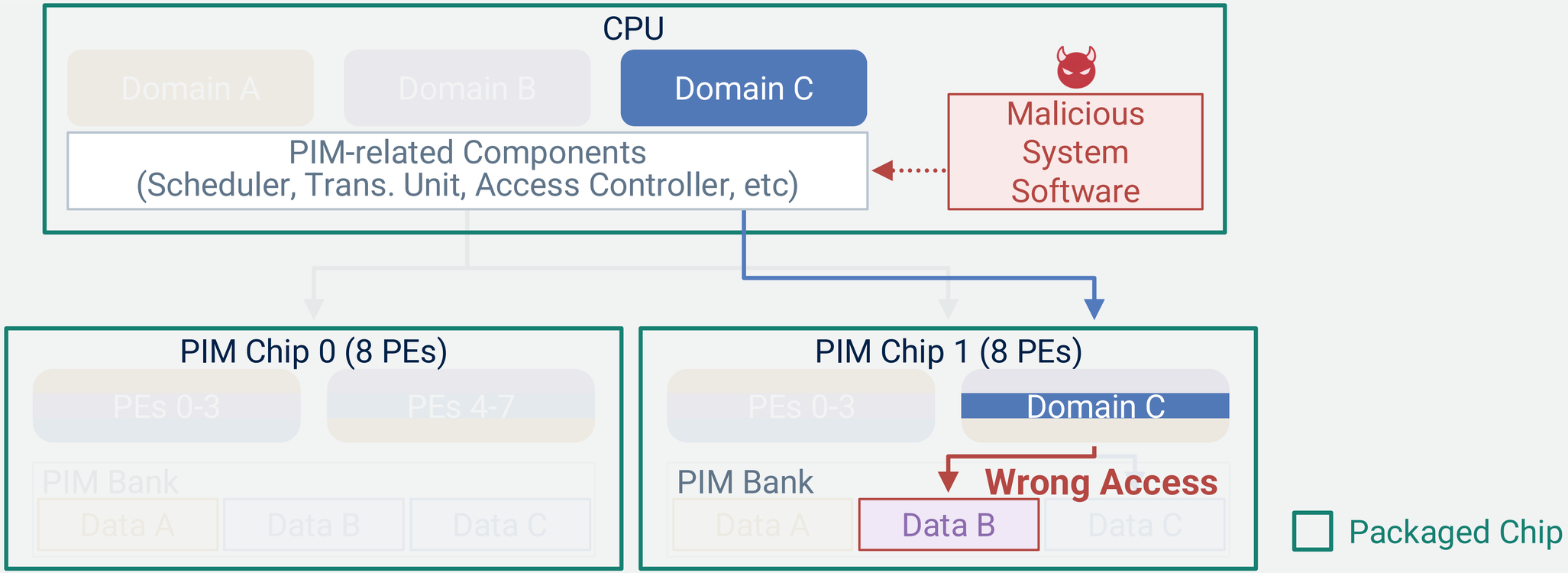
Security risks in spatio-temporal multi-domain PIM

Security risk 1: No domain isolation



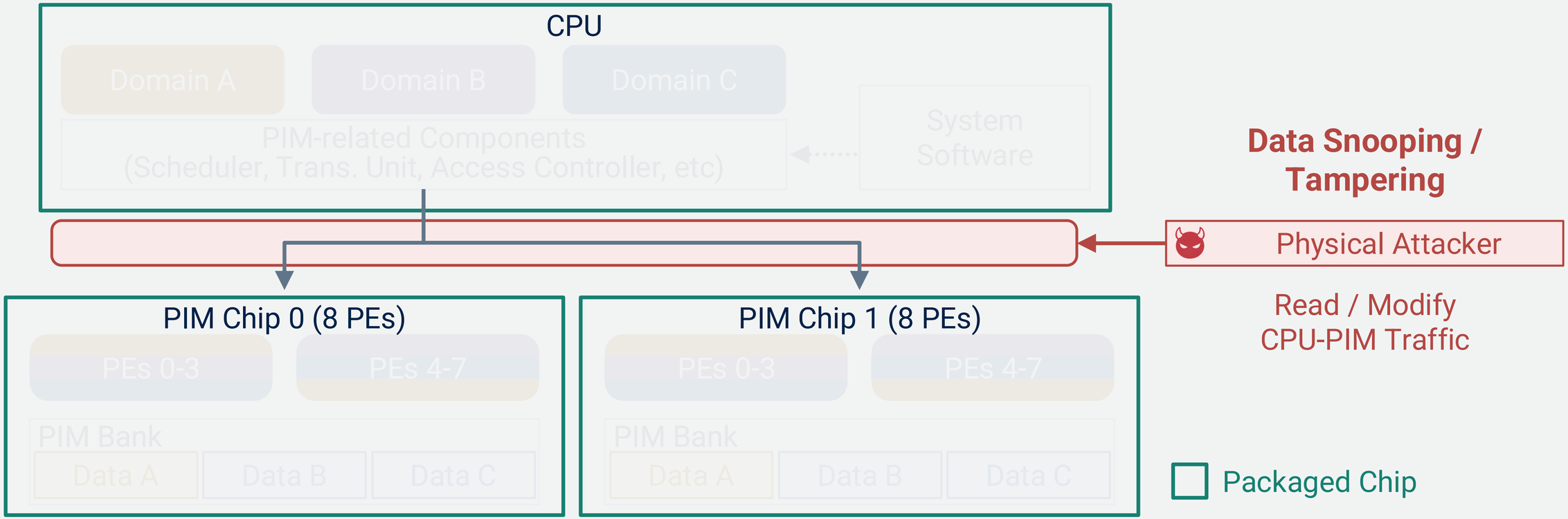
Security risks in spatio-temporal multi-domain PIM

Security risk 2: Compromised system software

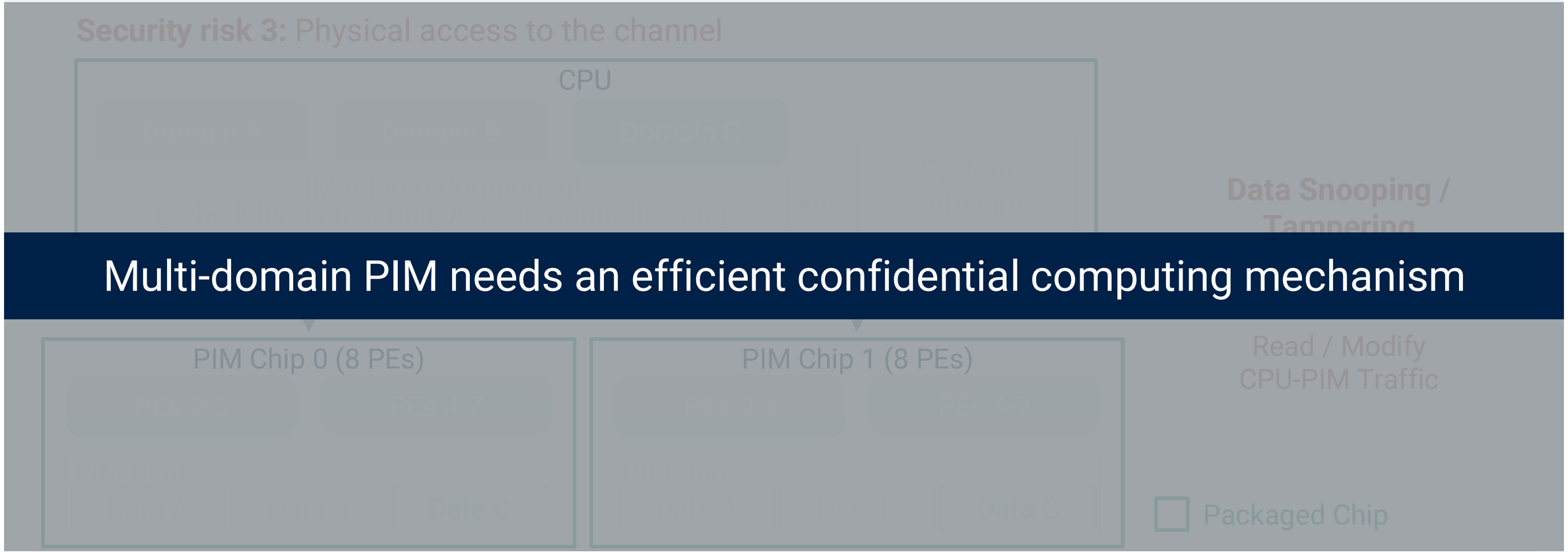


Security risks in spatio-temporal multi-domain PIM

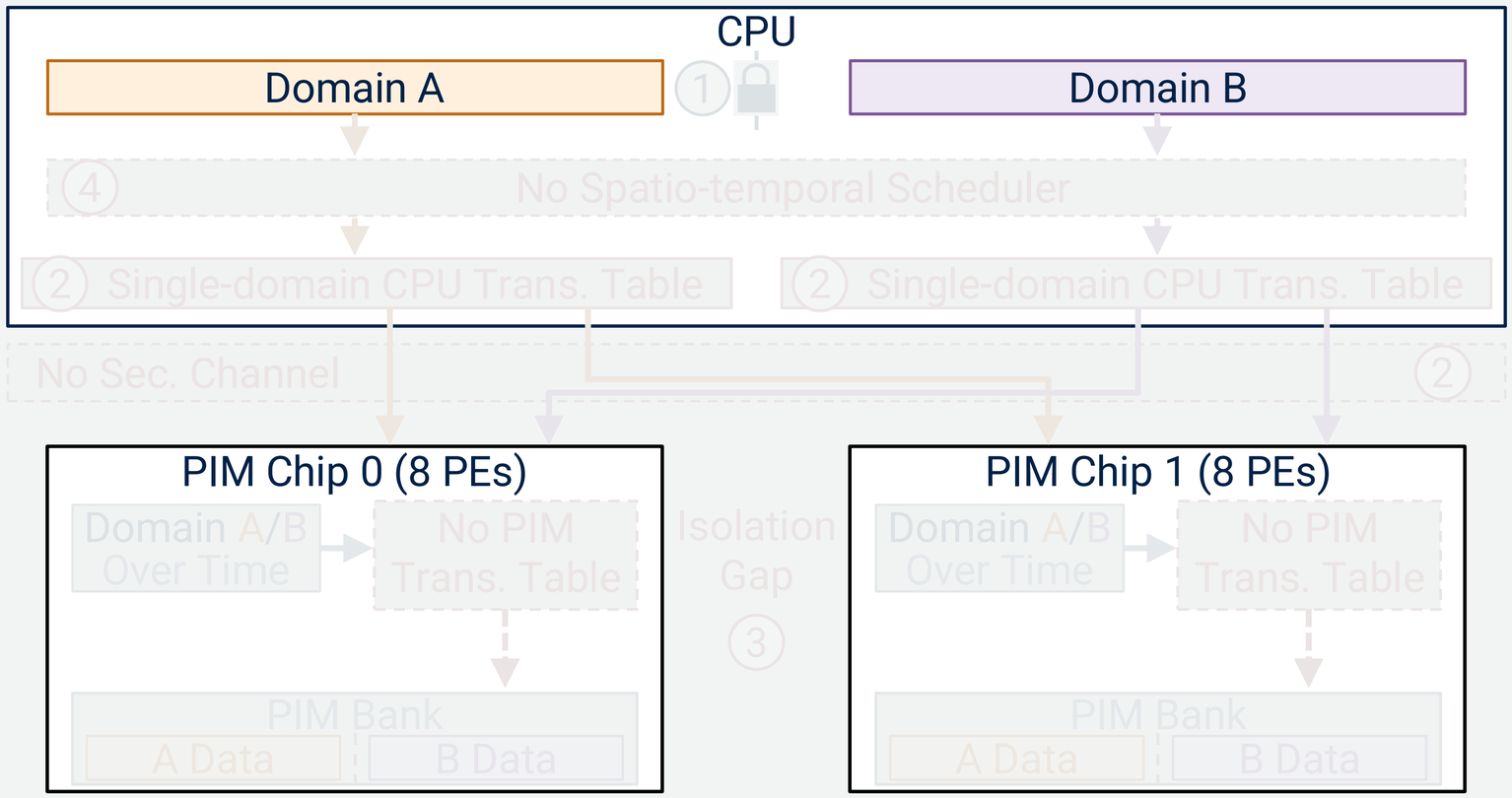
Security risk 3: Physical access to the channel



Security risks in spatio-temporal multi-domain PIM



Missing support for multi-domain PIM



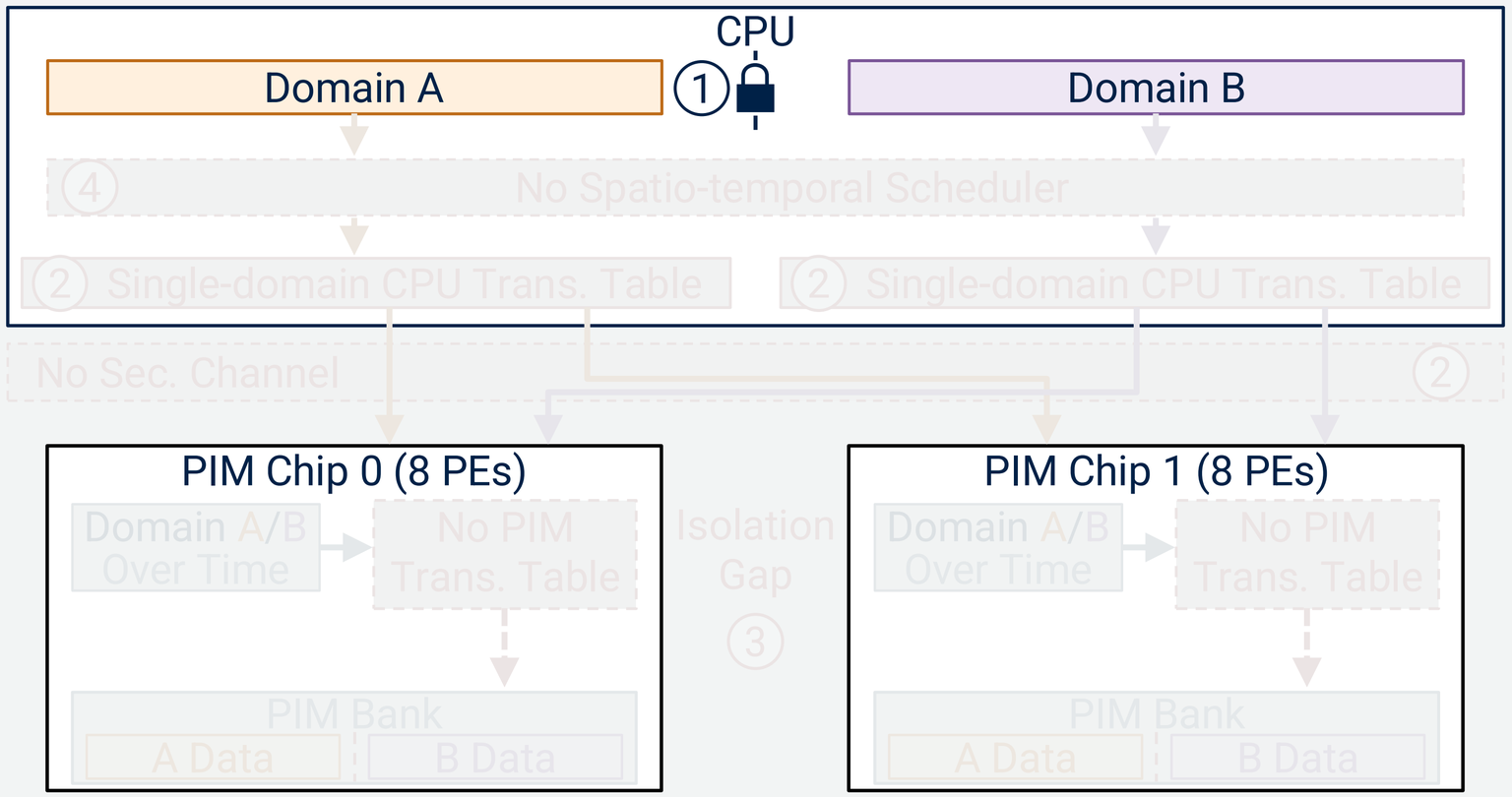
Security

1. CPU-CPU
   Isolation via existing solution (e.g., Arm CCA)
2. CPU-PIM
   No spatio-temporal isolation
   No secure channel
3. PIM-PIM
   No domain isolation

Efficient Scheduling

4. Spatio-temporal Scheduler
   Not available

Missing support for multi-domain PIM



Security

① **CPU-CPU**

Isolation via existing solution (e.g., Arm CCA)

② CPU-PIM

No spatio-temporal isolation
No secure channel

③ PIM-PIM

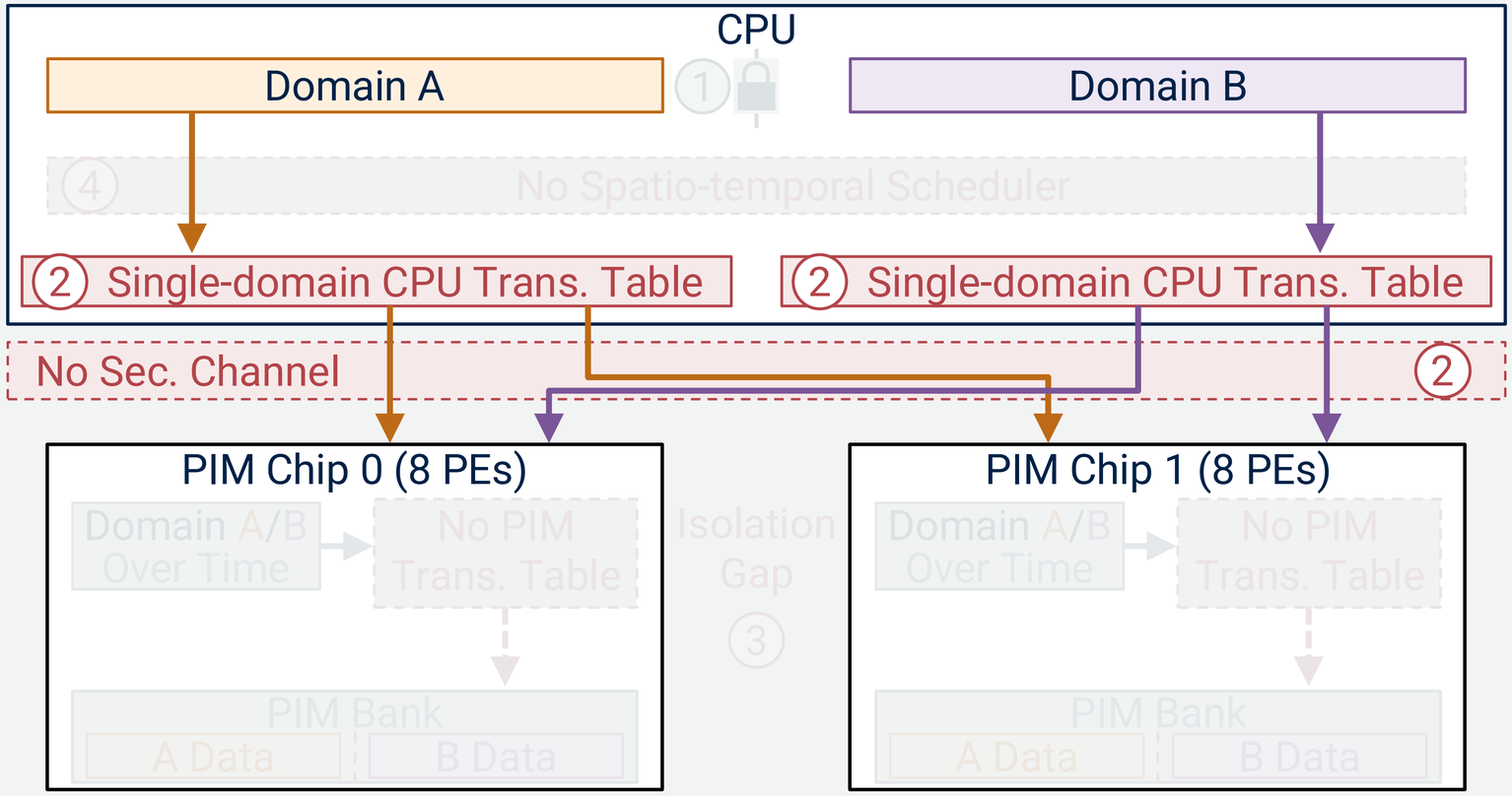
No domain isolation

Efficient Scheduling

④ Spatio-temporal Scheduler

Not available

Missing support for multi-domain PIM



Security

1 CPU-CPU

Isolation via existing solution (e.g., Arm CCA)

2 CPU-PIM

No spatio-temporal isolation
No secure channel

3 PIM-PIM

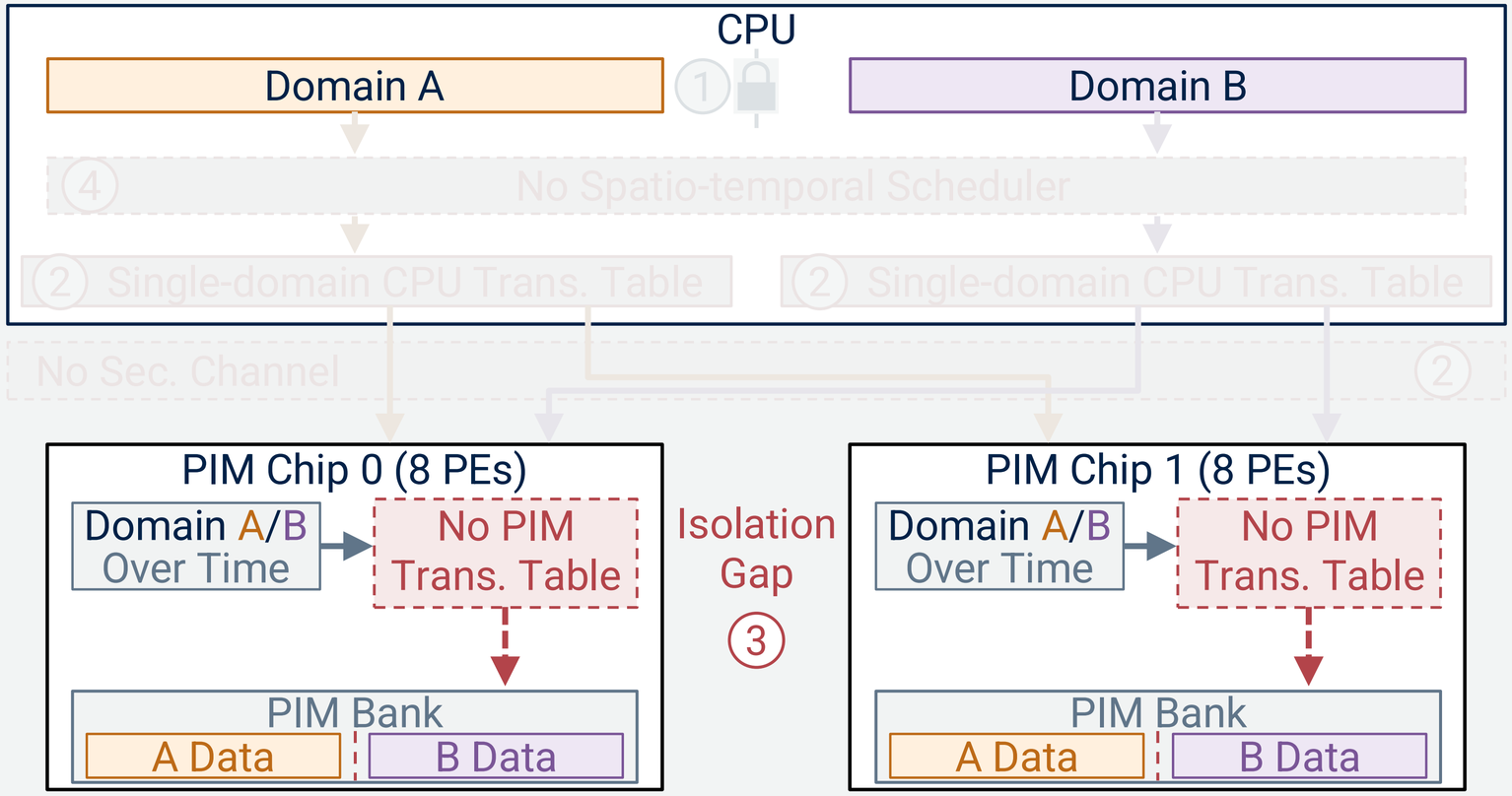
No domain isolation

Efficient Scheduling

4 Spatio-temporal Scheduler

Not available

Missing support for multi-domain PIM



Security

① CPU-CPU

Isolation via existing solution (e.g., Arm CCA)

② CPU-PIM

No spatio-temporal isolation
No secure channel

③ PIM-PIM

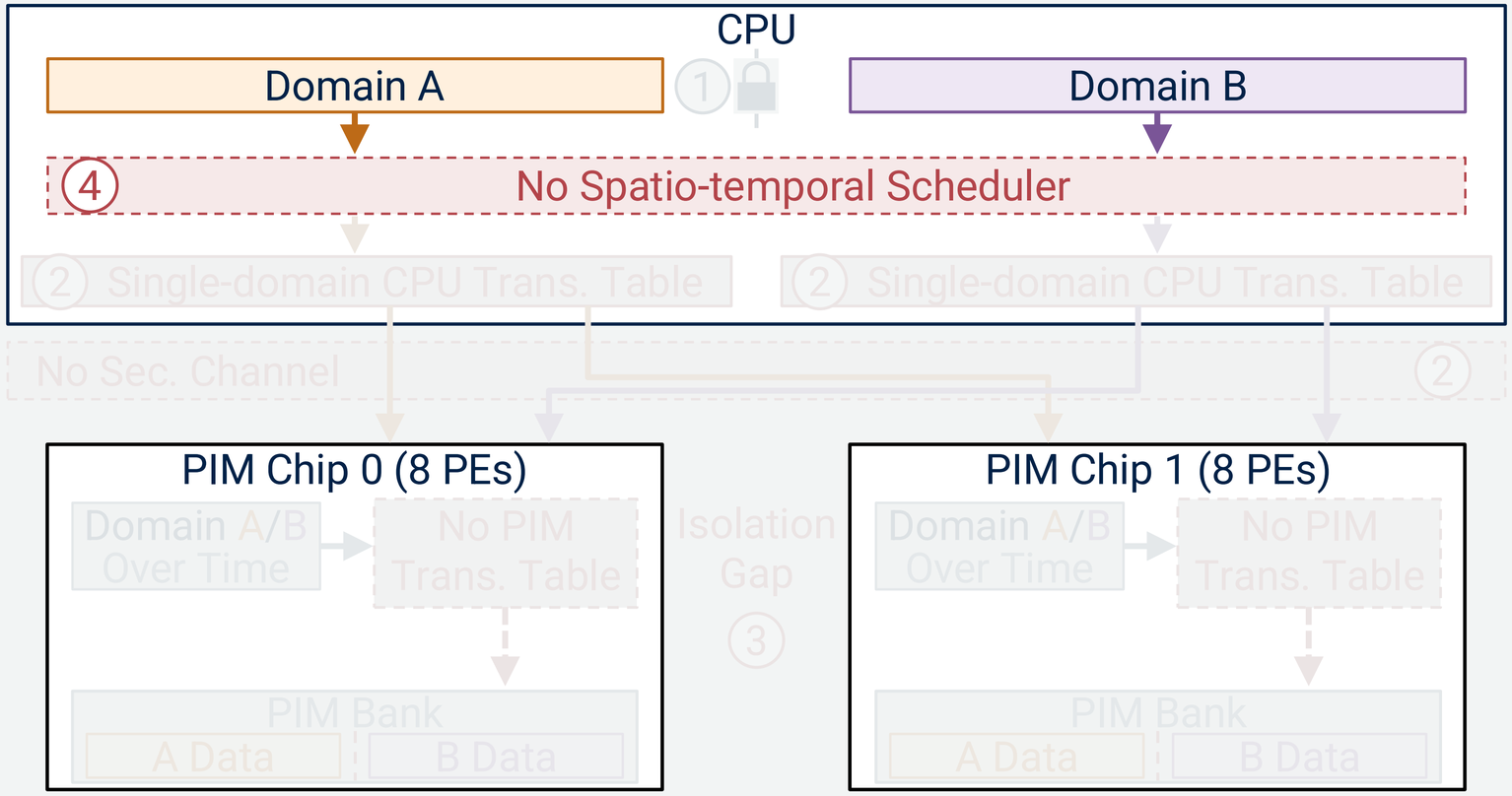
No domain isolation

Efficient Scheduling

④ Spatio-temporal Scheduler

Not available

Missing support for multi-domain PIM



Security

① CPU-CPU
Isolation via existing solution (e.g., Arm CCA)

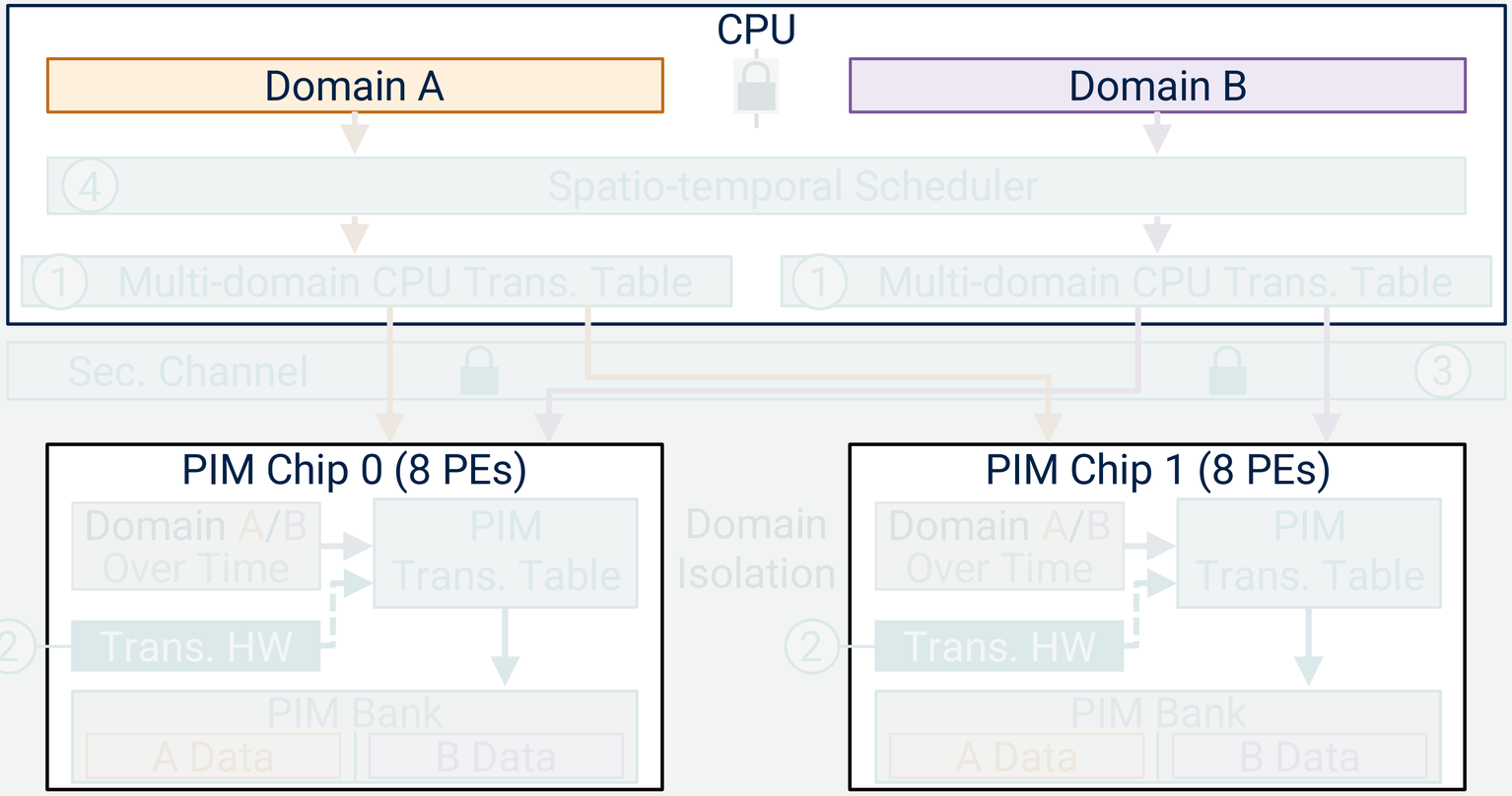
② CPU-PIM
No spatio-temporal isolation
No secure channel

③ PIM-PIM
No domain isolation

Efficient Scheduling

④ Spatio-temporal Scheduler
Not available

SecPIMlet: Secure, efficient multi-domain PIM



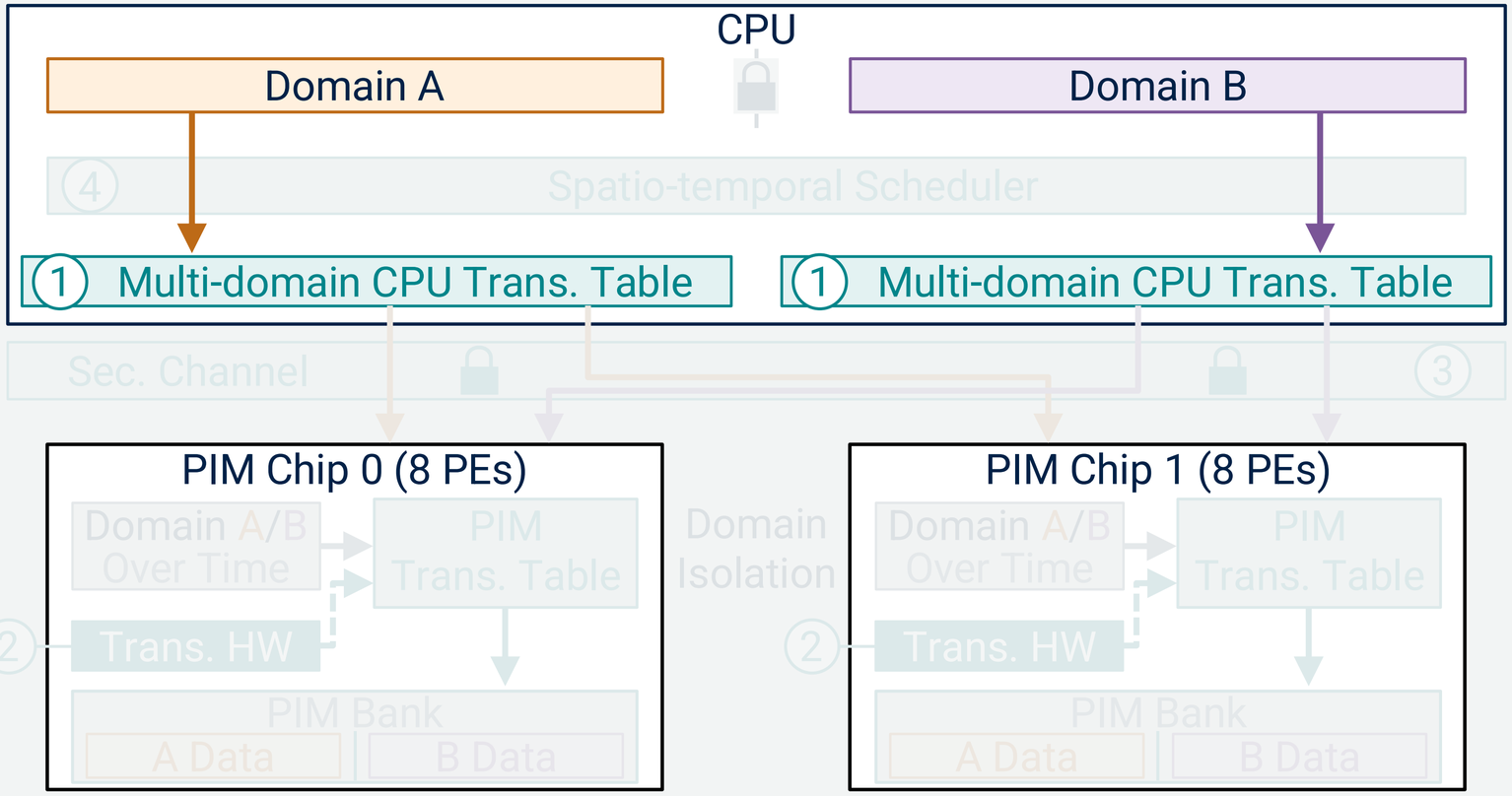
Security

1. Chip&Page-granular Translation
   Spatial + temporal isolation
2. PIM-side Translation HW
   PIM-PIM isolation with low trans. table-switch overhead
3. Secure CPU-PIM Channel
   Per-chip, CPU/PIM layout-aware secure engine

Efficient Scheduling

4. Spatio-temporal Scheduler
   PE allocation and time slicing

SecPIMlet: Secure, efficient multi-domain PIM



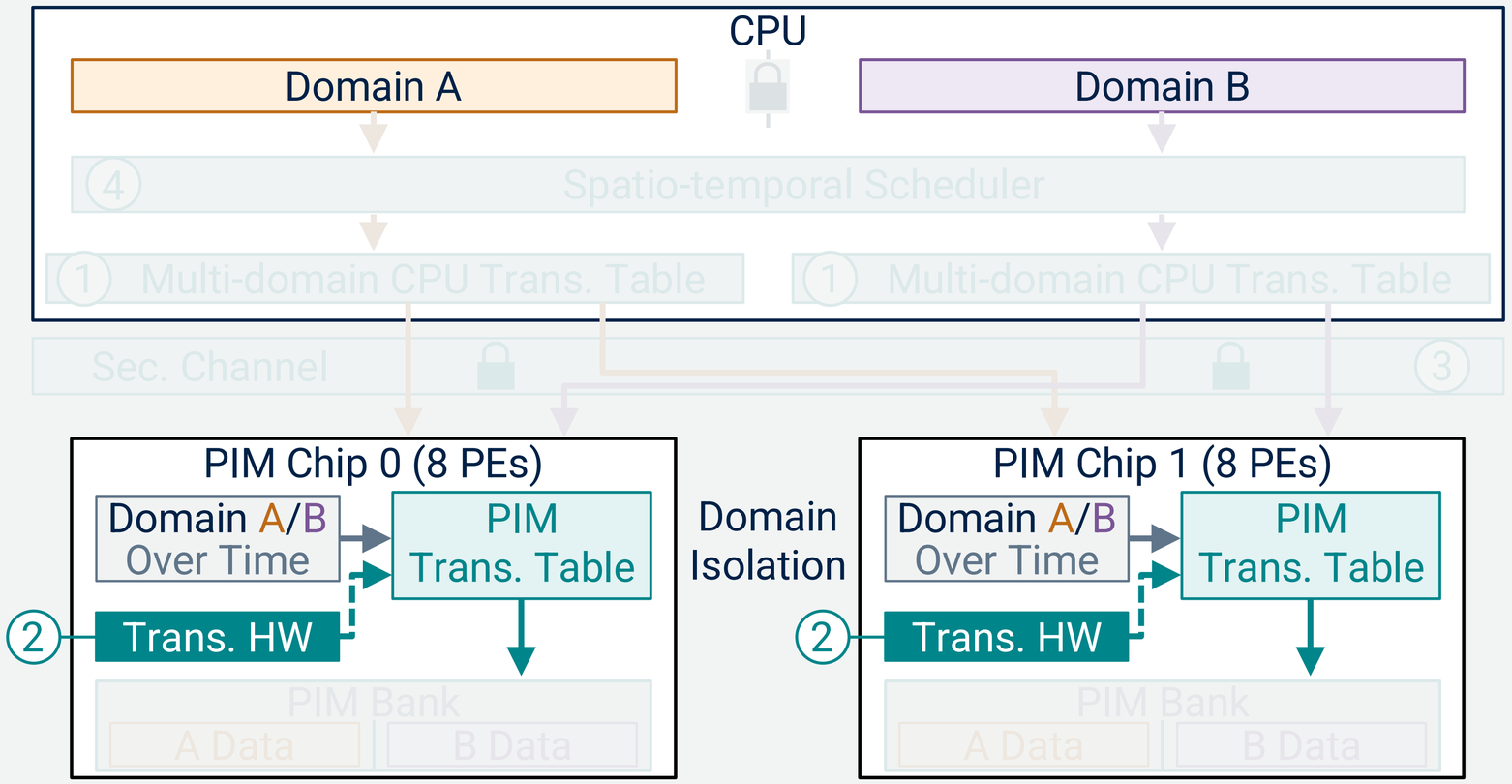
Security

1. **Chip&Page-granular Translation**
   Spatial + temporal isolation
2. PIM-side Translation HW
   PIM-PIM isolation with low trans. table-switch overhead
3. Secure CPU-PIM Channel
   Per-chip, CPU/PIM layout-aware secure engine

Efficient Scheduling

4. Spatio-temporal Scheduler
   PE allocation and time slicing

SecPIMlet: Secure, efficient multi-domain PIM



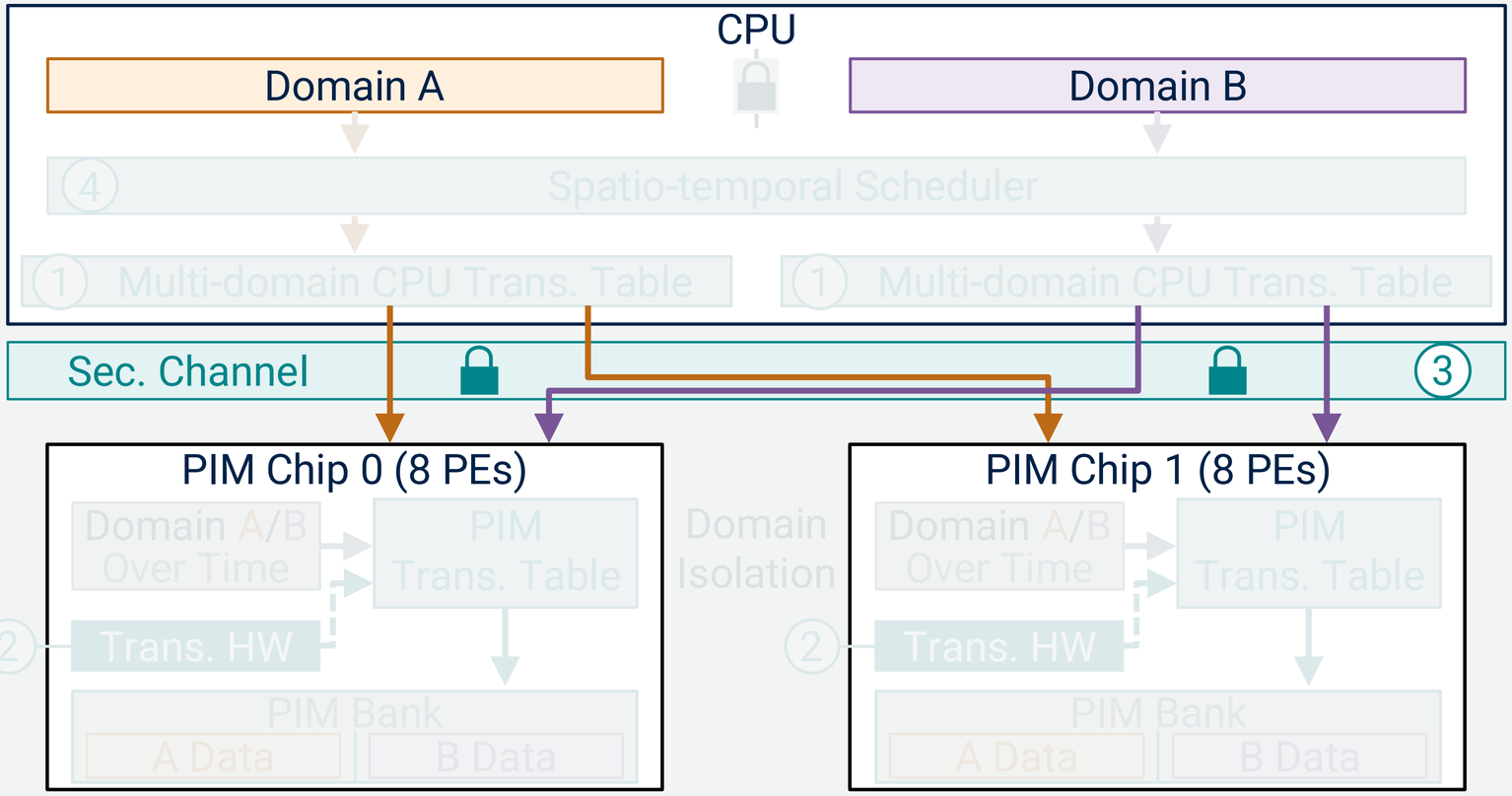
Security

1. Chip&Page-granular Translation
   Spatial + temporal isolation
2. **PIM-side Translation HW**
   PIM-PIM isolation with low trans. table-switch overhead
3. Secure CPU-PIM Channel
   Per-chip, CPU/PIM layout-aware secure engine

Efficient Scheduling

4. Spatio-temporal Scheduler
   PE allocation and time slicing

SecPIMlet: Secure, efficient multi-domain PIM



Security

1 Chip&Page-granular Translation
Spatial + temporal isolation

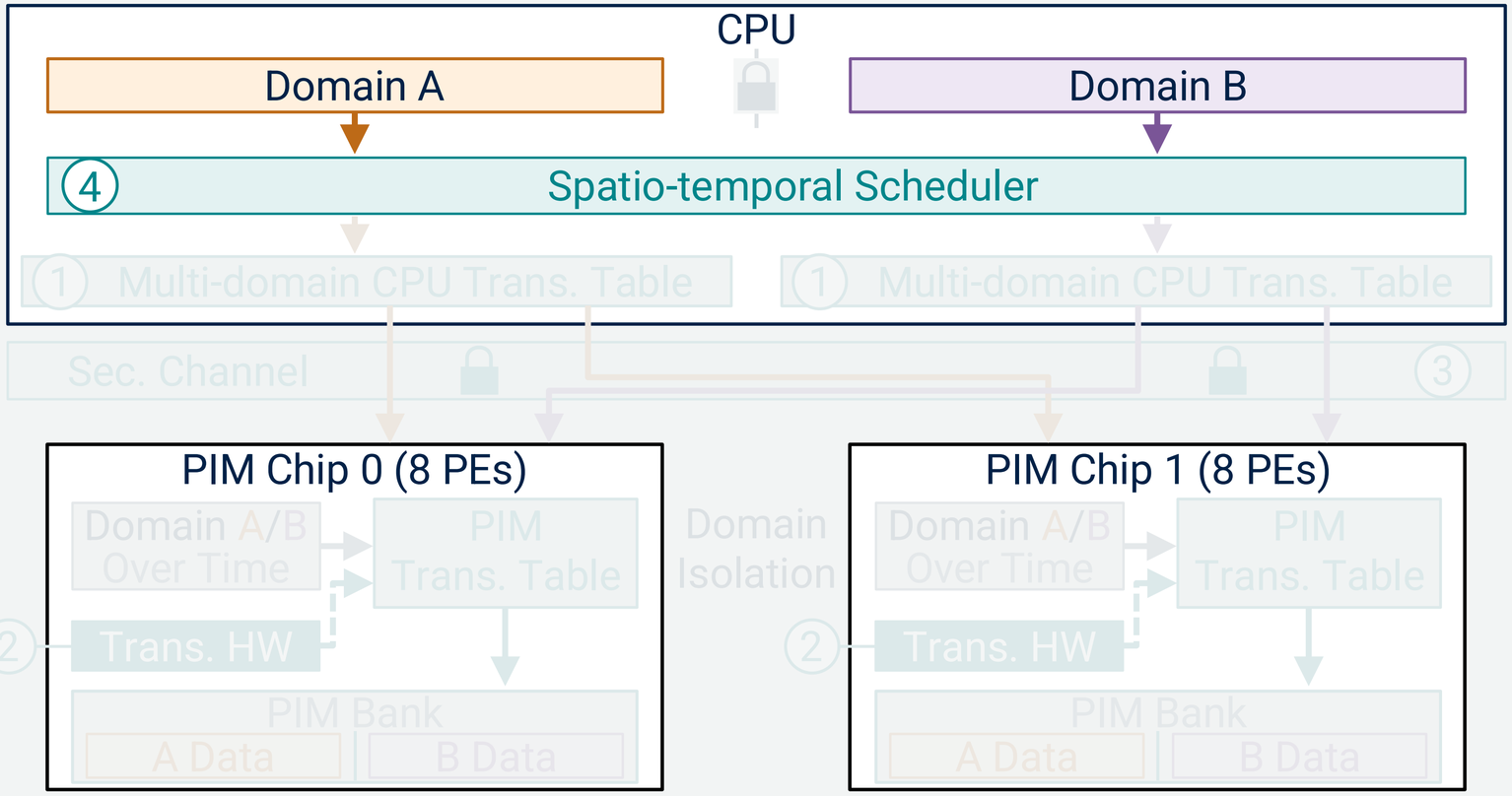
2 PIM-side Translation HW
PIM-PIM isolation with low trans. table-switch overhead

3 Secure CPU-PIM Channel
Per-chip, CPU/PIM layout-aware secure engine

Efficient Scheduling

4 Spatio-temporal Scheduler
PE allocation and time slicing

SecPIMlet: Secure, efficient multi-domain PIM



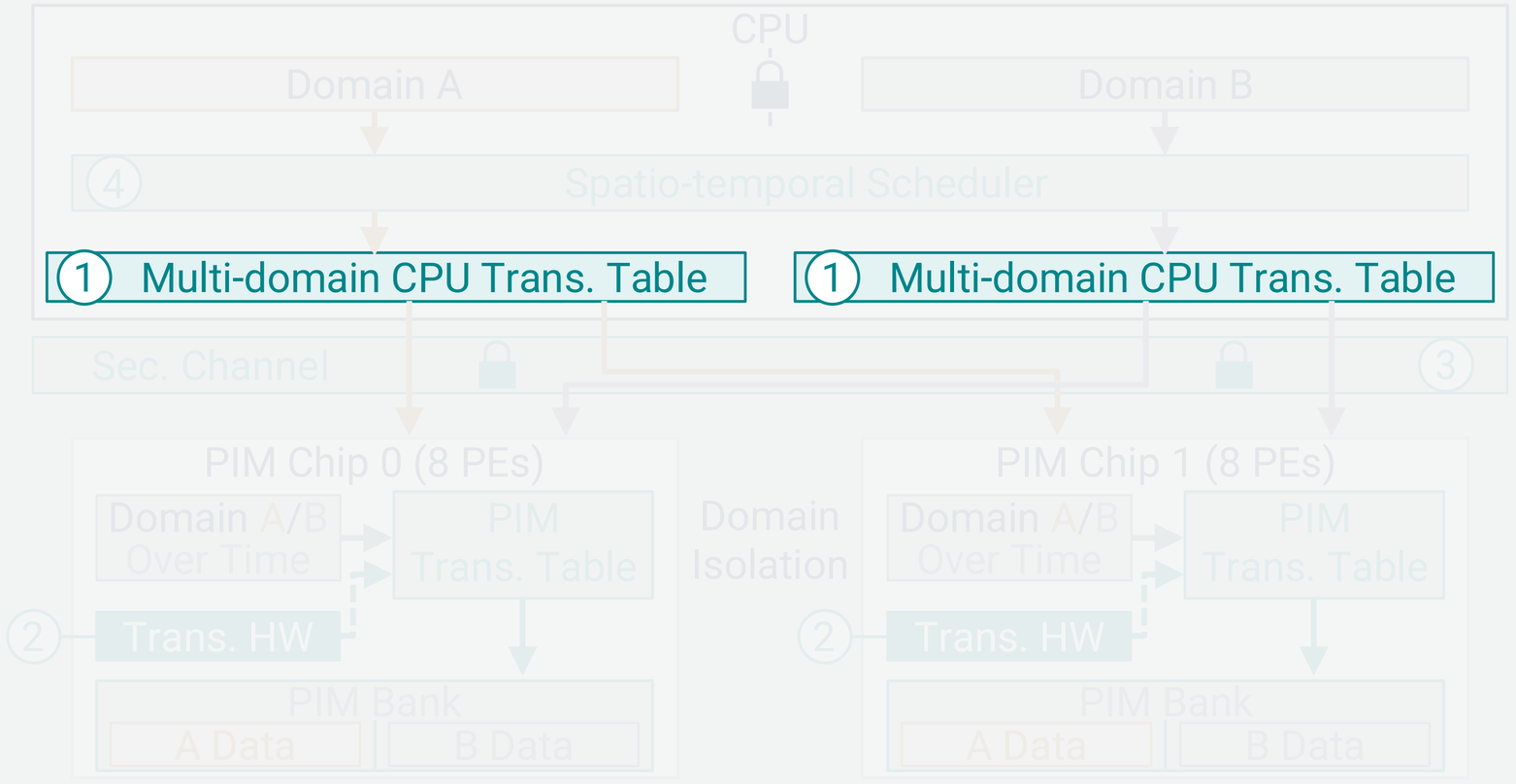
Security

1. Chip&Page-granular Translation
   Spatial + temporal isolation
2. PIM-side Translation HW
   PIM-PIM isolation with low trans. table-switch overhead
3. Secure CPU-PIM Channel
   Per-chip, CPU/PIM layout-aware secure engine

Efficient Scheduling

4. Spatio-temporal Scheduler
   PE allocation and time slicing

Chip&Page-granular translation



Security

1. **Chip&Page-granular Translation**
   Spatial + temporal isolation
2. PIM-side Translation HW
   PIM-PIM isolation with low trans. table-switch overhead
3. Secure CPU-PIM Channel
   Per-chip, CPU/PIM layout-aware secure engine

Efficient Scheduling

4. Spatio-temporal Scheduler
   PE allocation and time slicing

Inter-PE and intra-PE isolation

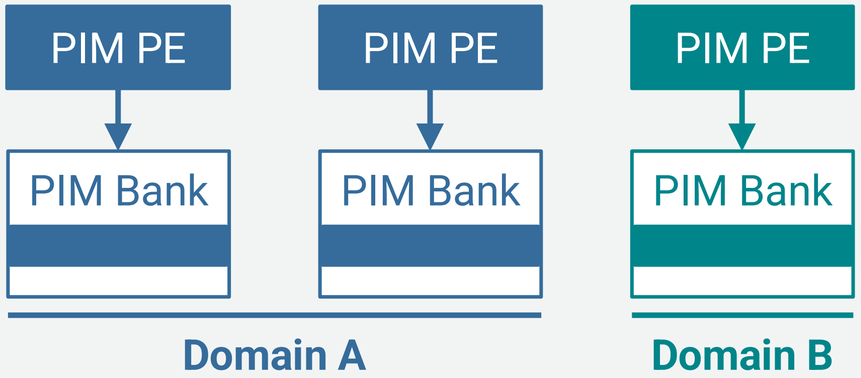
Spatial sharing needs inter-PE isolation while temporal sharing requires intra-PE isolation.

Inter-PE and intra-PE isolation

Spatial sharing needs inter-PE isolation while temporal sharing requires intra-PE isolation.

Inter-PE isolation (for spatial sharing)

Virtual PIM [PACT 2023]

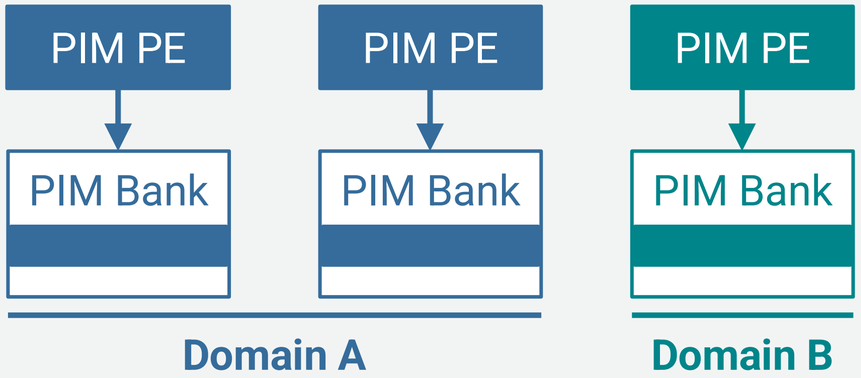


Inter-PE and intra-PE isolation

Spatial sharing needs inter-PE isolation while temporal sharing requires intra-PE isolation.

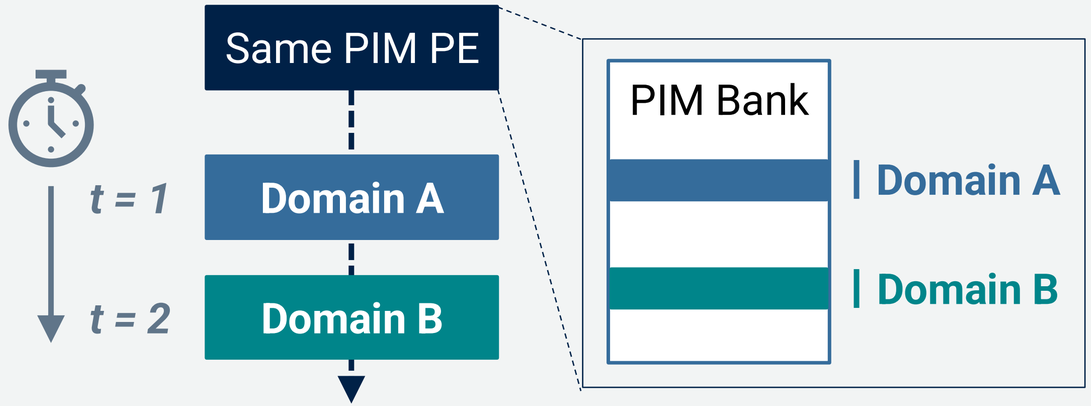
Inter-PE isolation (for spatial sharing)

Virtual PIM [PACT 2023]



Intra-PE isolation (for temporal sharing)

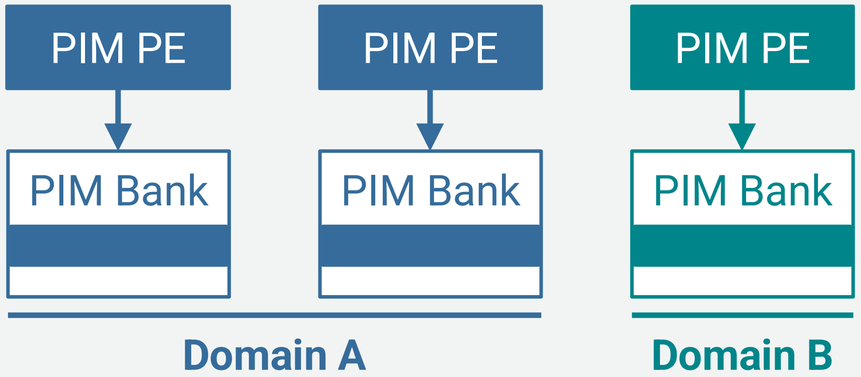
UM-PIM [ISCA 2024]



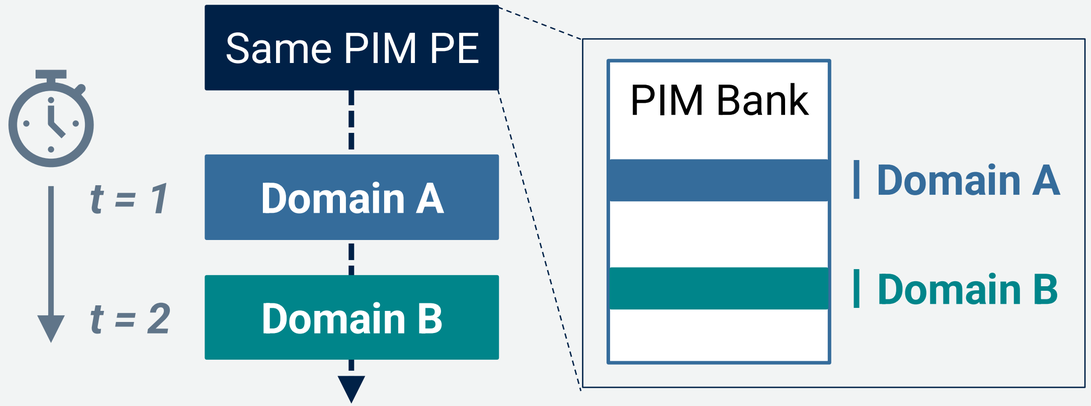
Inter-PE and intra-PE isolation

Spatial sharing needs inter-PE isolation while temporal sharing requires intra-PE isolation.

Inter-PE isolation (for spatial sharing)
Virtual PIM [PACT 2023]



Intra-PE isolation (for temporal sharing)
UM-PIM [ISCA 2024]



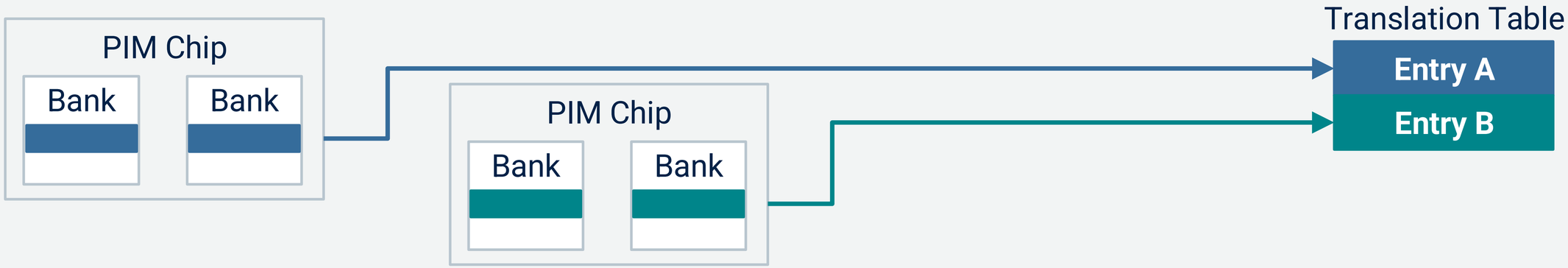
Unlike prior studies, spatio-temporally shared PIM requires **both inter- and intra-PE** isolation

Chip&Page-granular translation

Chip&page-granular translation units enable both inter- and intra-PE isolation.

Chip&Page-granular translation

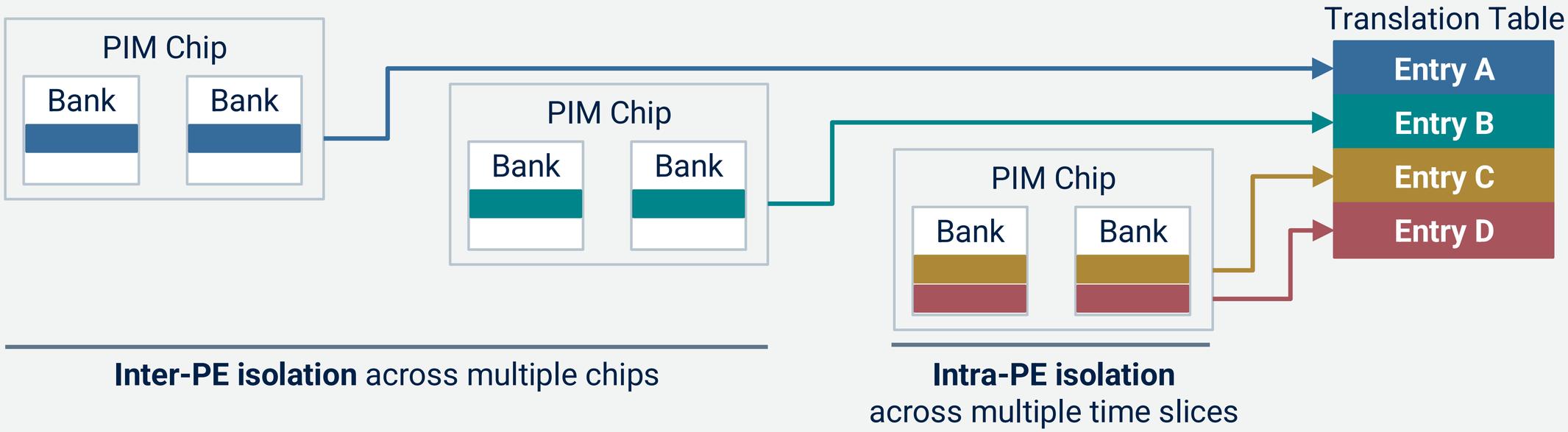
Chip&page-granular translation units enable both inter- and intra-PE isolation.



Inter-PE isolation across multiple chips

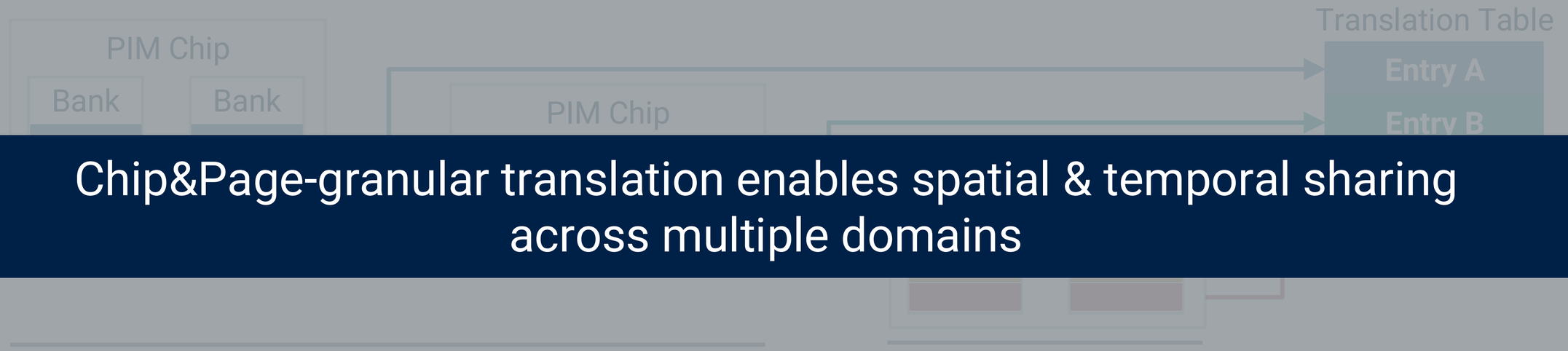
Chip&Page-granular translation

Chip&page-granular translation units enable both inter- and intra-PE isolation.



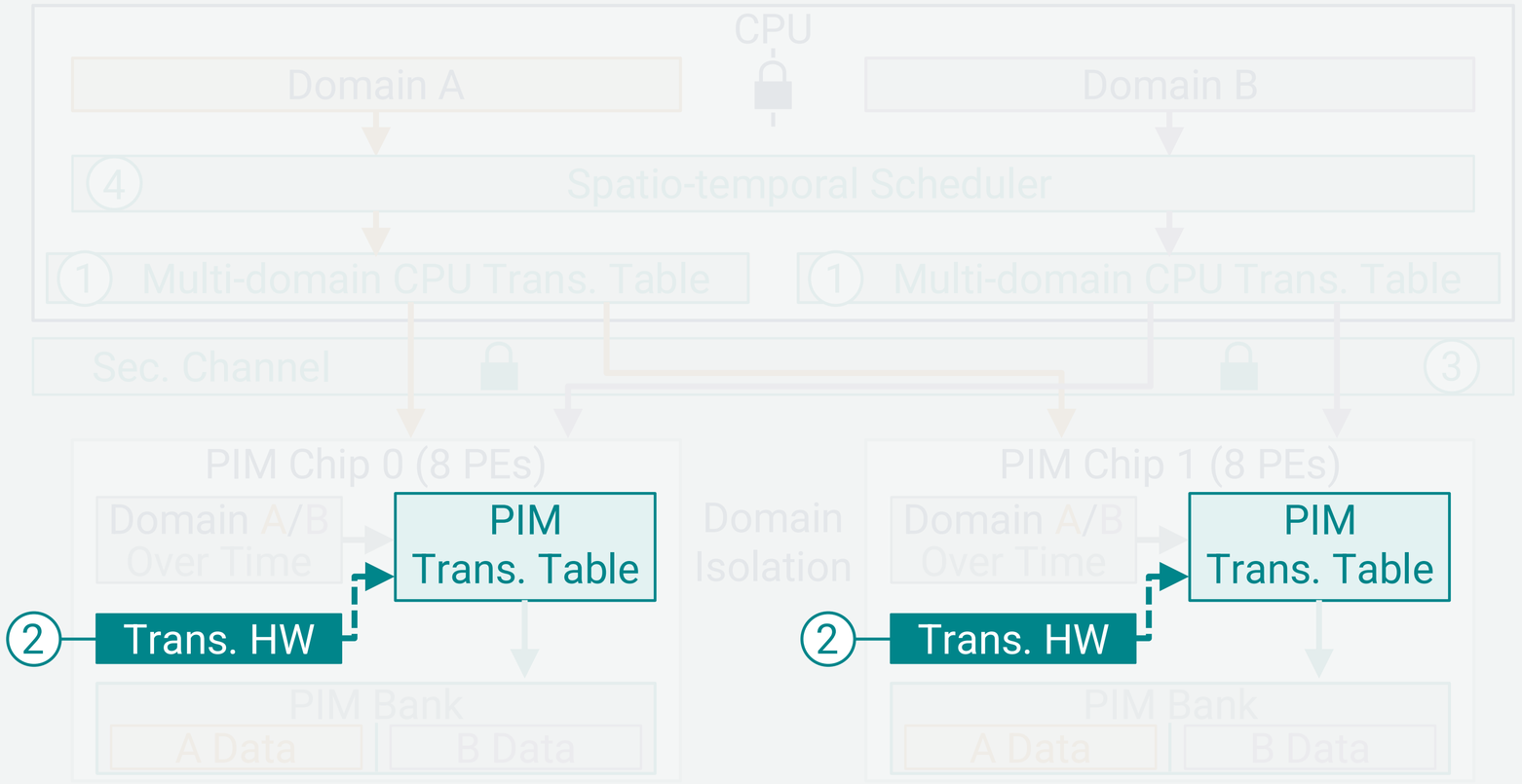
Chip&Page-granular translation

Chip&page-granular translation units enable both inter- and intra-PE isolation.



Chip&Page-granular translation enables spatial & temporal sharing across multiple domains

PIM-side translation hardware



Security

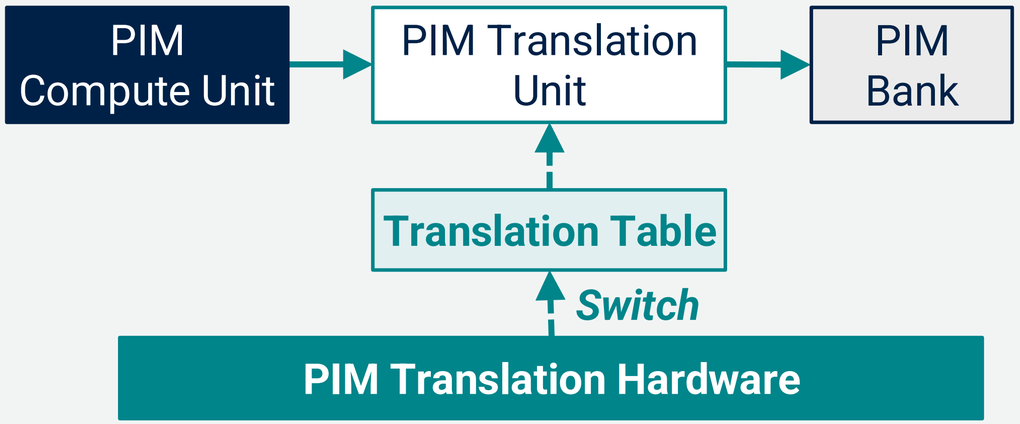
1. Chip&Page-granular Translation
   Spatial + temporal isolation
2. **PIM-side Translation HW**
   PIM-PIM isolation with low trans. table-switch overhead
3. Secure CPU-PIM Channel
   Per-chip, CPU/PIM layout-aware secure engine

Efficient Scheduling

4. Spatio-temporal Scheduler
   PE allocation and time slicing

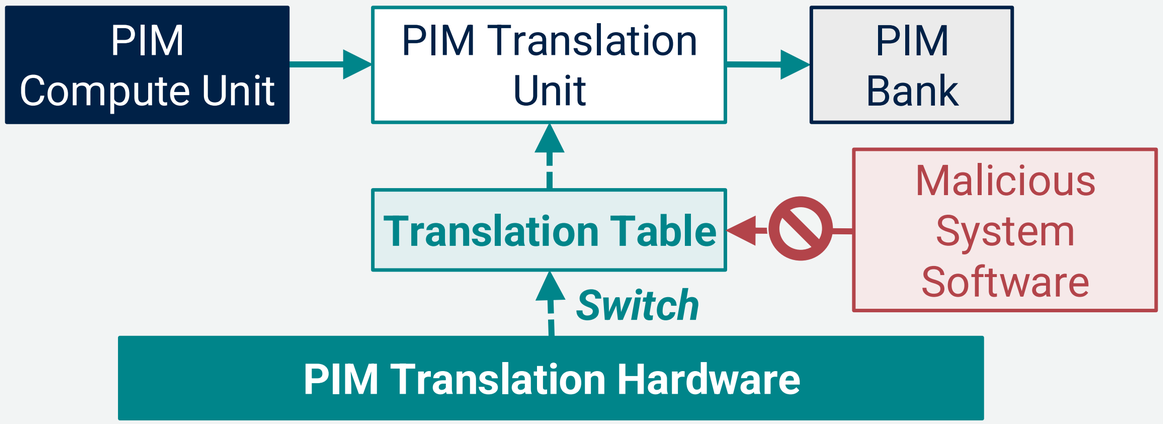
Translation HW for PIM trans. table management

Translation HW **exclusively switches the PIM translation table** to enforce isolation.



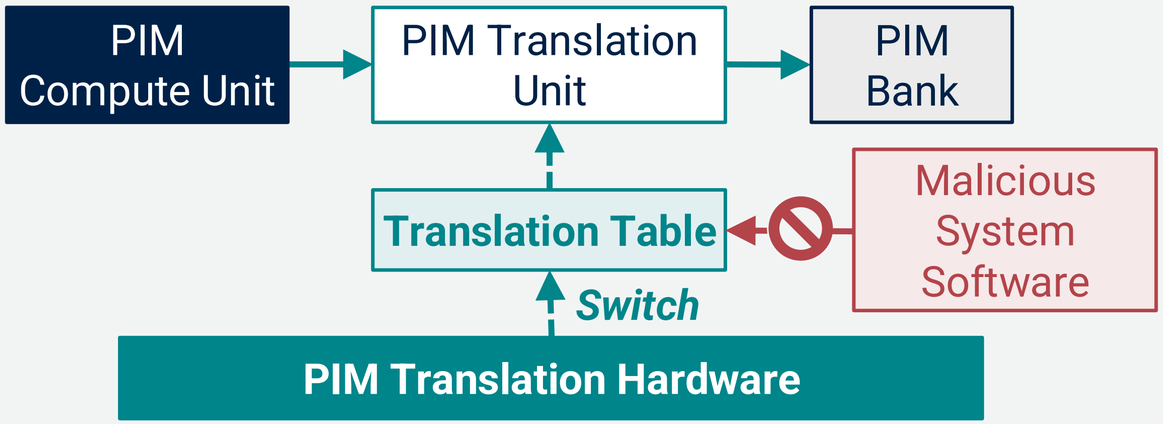
Translation HW for PIM trans. table management

Translation HW **exclusively switches the PIM translation table** to enforce isolation.
This prevents malicious system software from accessing the data of other domains.



Translation HW for PIM trans. table management

Translation HW **exclusively switches the PIM translation table** to enforce isolation.
This prevents malicious system software from accessing the data of other domains.

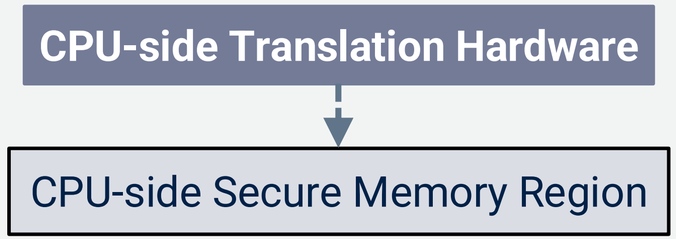


Where is PIM translation hardware located?

PIM-side translation hardware

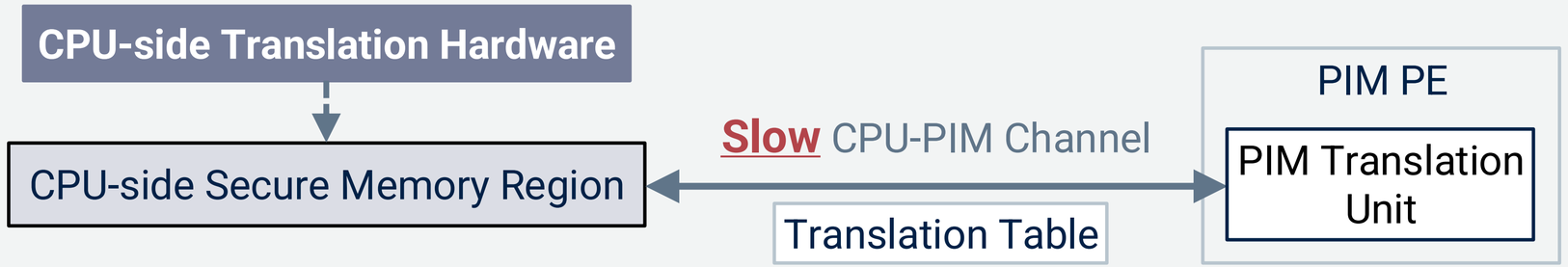
PIM-side translation hardware

CPU-side HW

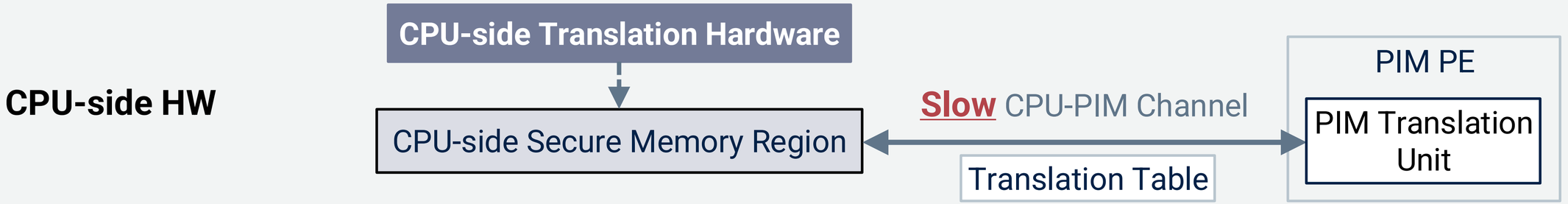


PIM-side translation hardware

CPU-side HW



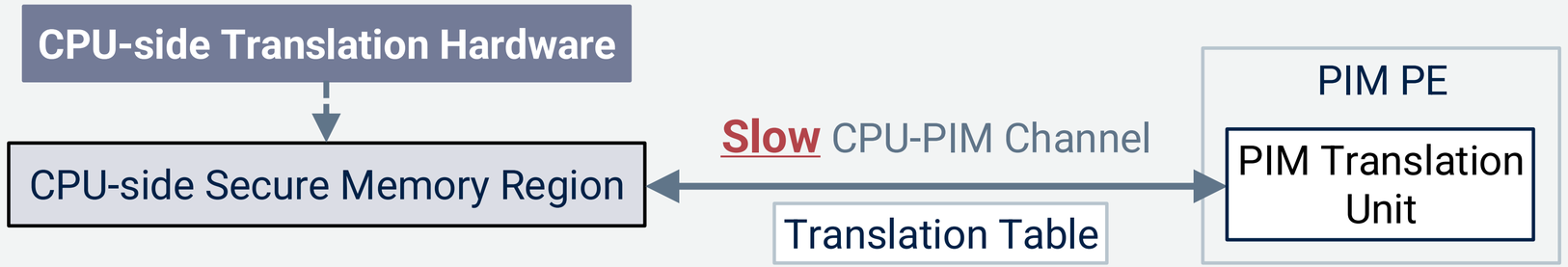
PIM-side translation hardware



Severe table-switching performance overhead due to **slow** CPU-PIM channel

PIM-side translation hardware

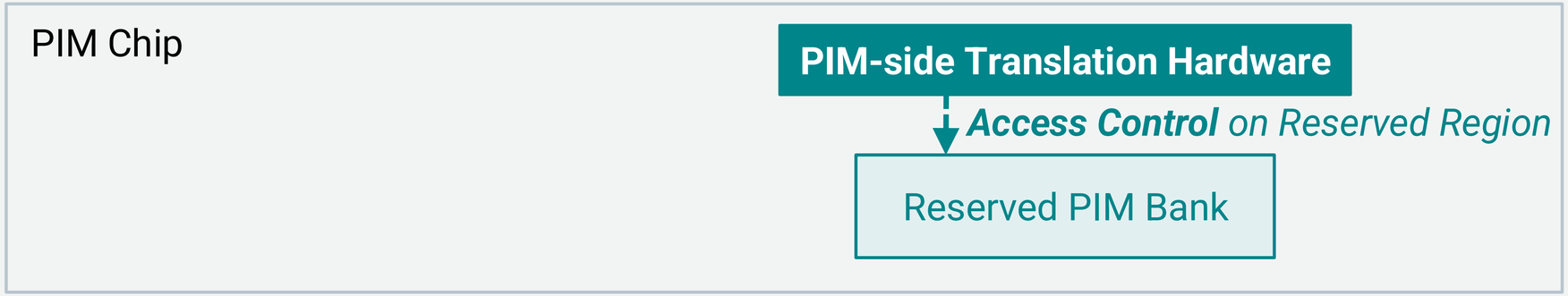
CPU-side HW



Severe table-switching performance overhead due to **slow** CPU-PIM channel

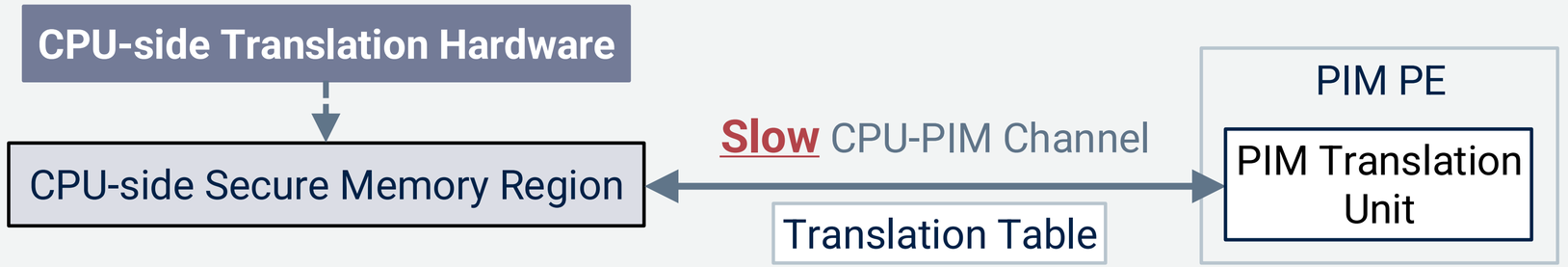
PIM-side HW (Ours; *SecPIMlet*)

Table contexts reside in PIM banks by trusted HW



PIM-side translation hardware

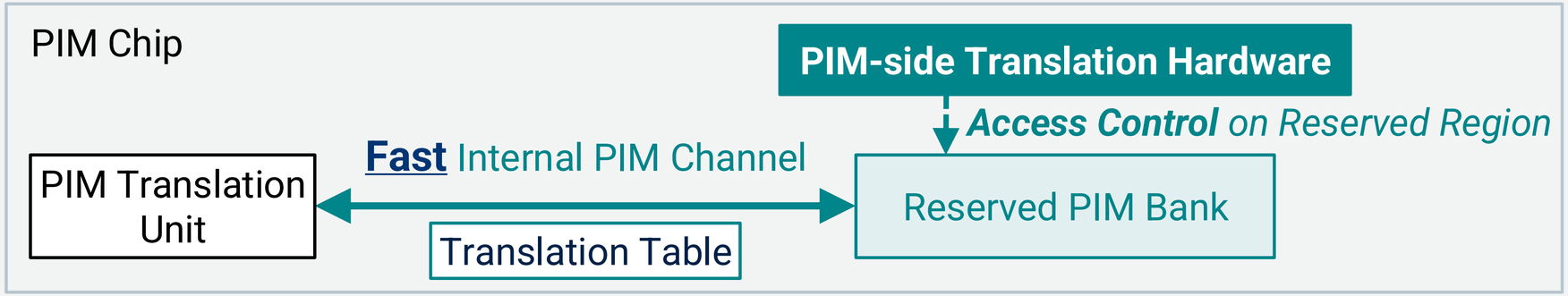
CPU-side HW



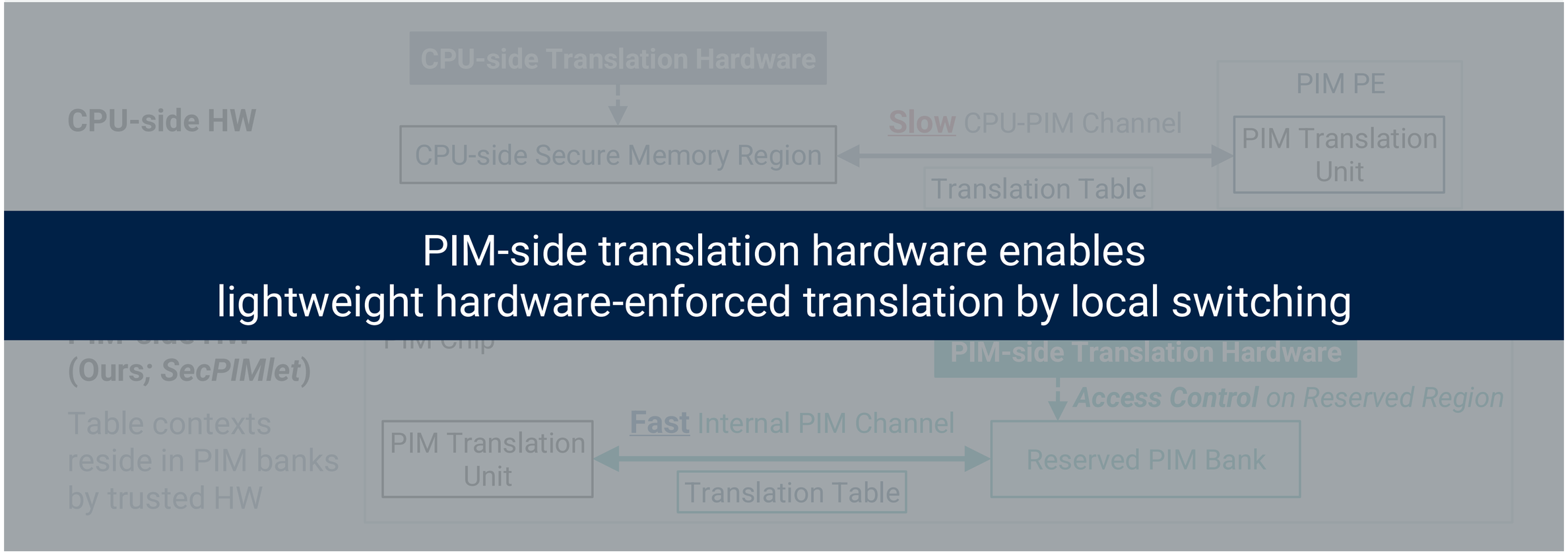
Severe table-switching performance overhead due to **slow** CPU-PIM channel

PIM-side HW (Ours; *SecPIMlet*)

Table contexts reside in PIM banks by trusted HW

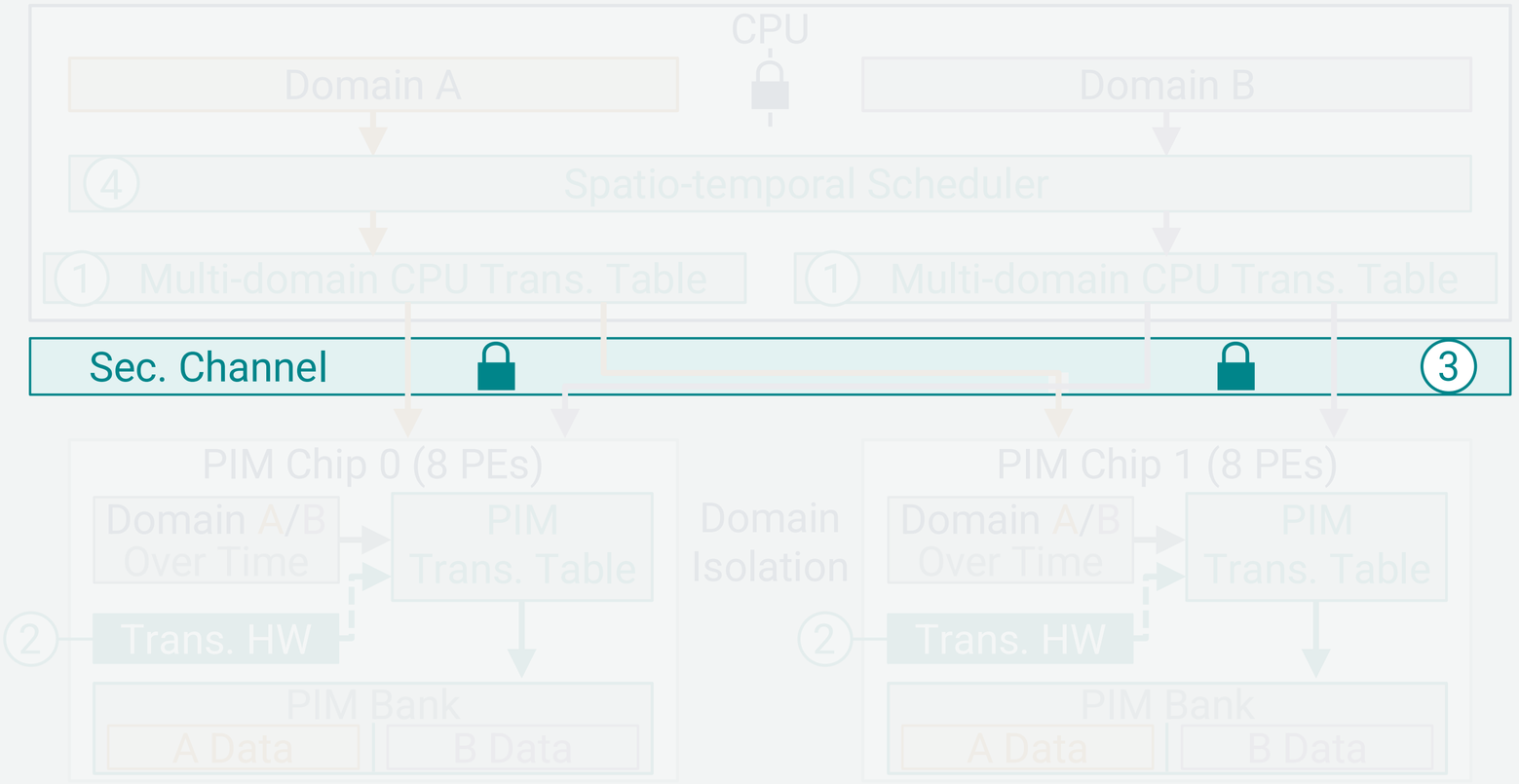


PIM-side translation hardware



PIM-side translation hardware enables lightweight hardware-enforced translation by local switching

Secure CPU-PIM channel



Security

1. Chip&Page-granular Translation
   Spatial + temporal isolation
2. PIM-side Translation HW
   PIM-PIM isolation with low trans. table-switch overhead
3. **Secure CPU-PIM Channel**
   Per-chip, CPU/PIM layout-aware secure engine

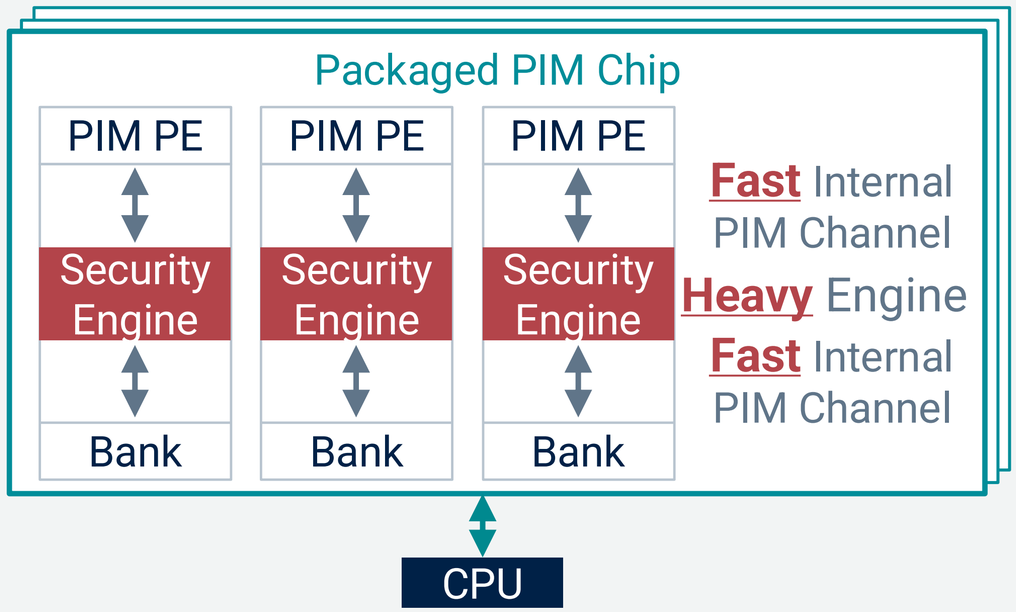
Efficient Scheduling

4. Spatio-temporal Scheduler
   PE allocation and time slicing

Lightweight per-chip security engine for CPU-PIM

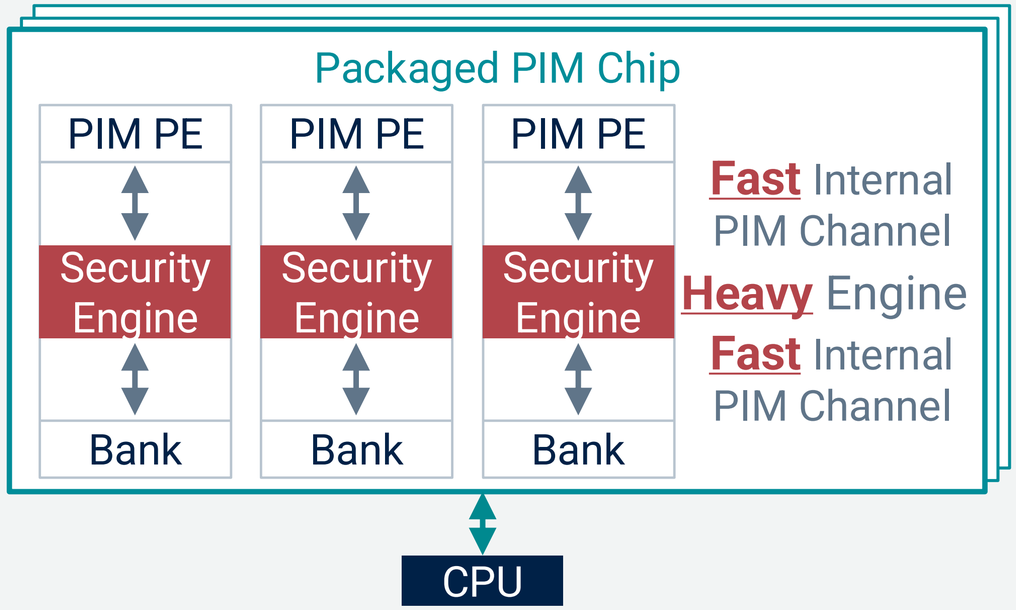
Lightweight per-chip security engine for CPU-PIM

Per-PE engine (*SE-PIM* [*TCC 2022*])

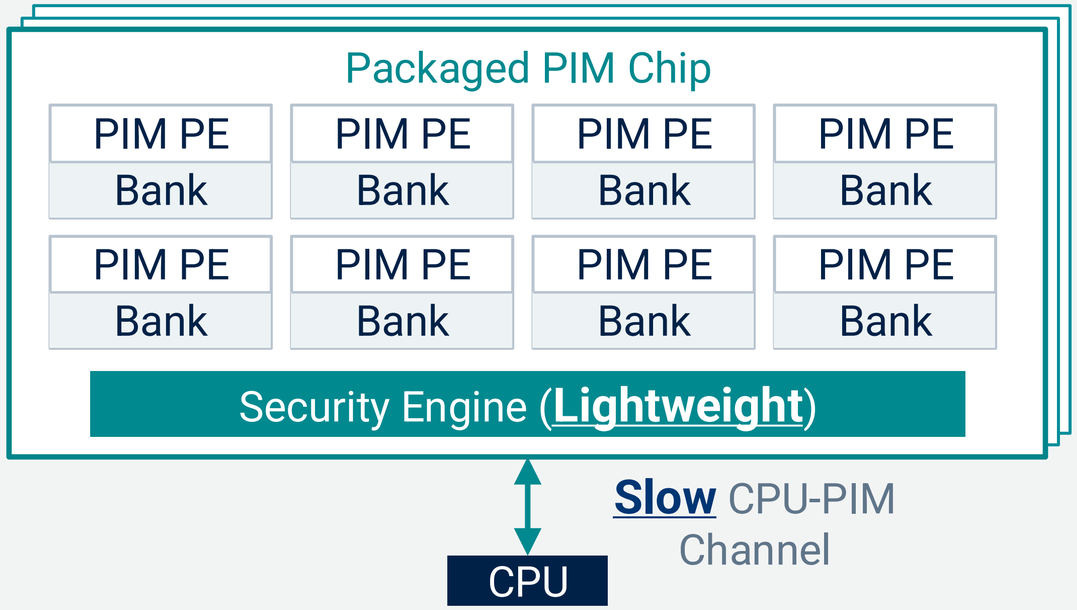


Lightweight per-chip security engine for CPU-PIM

Per-PE engine (*SE-PIM* [*TCC 2022*])



Per-chip (=8 PEs) engine (Ours; *SecPIMlet*)

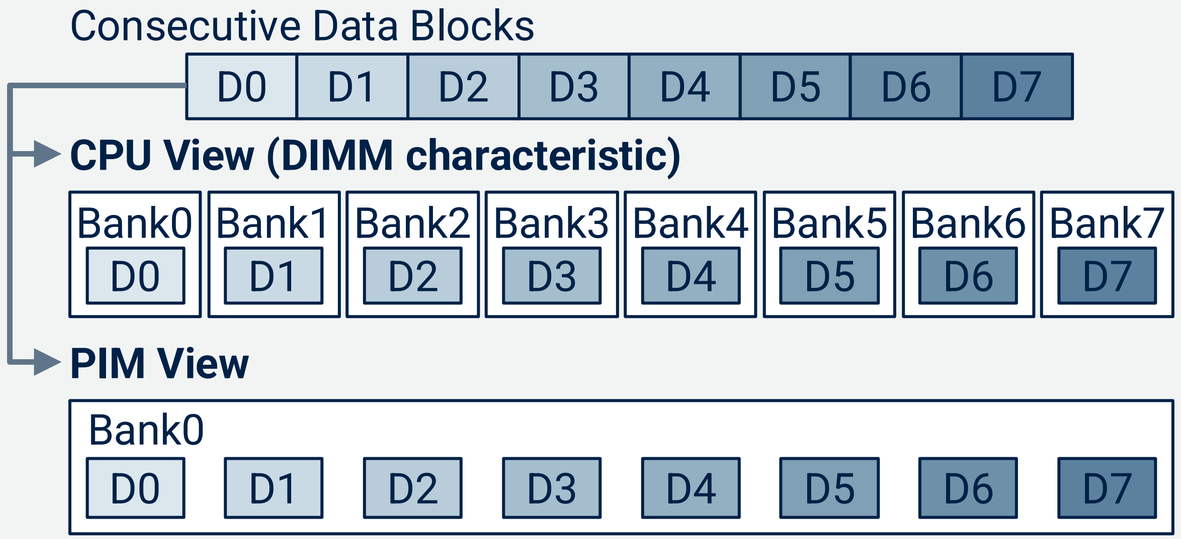


Layout-aware security engine for CPU-PIM

Layout-aware security engine for CPU-PIM

Different CPU and PIM layouts (due to DIMM arch.)

CPU: Interleaved, PIM: Sequential bank mapping

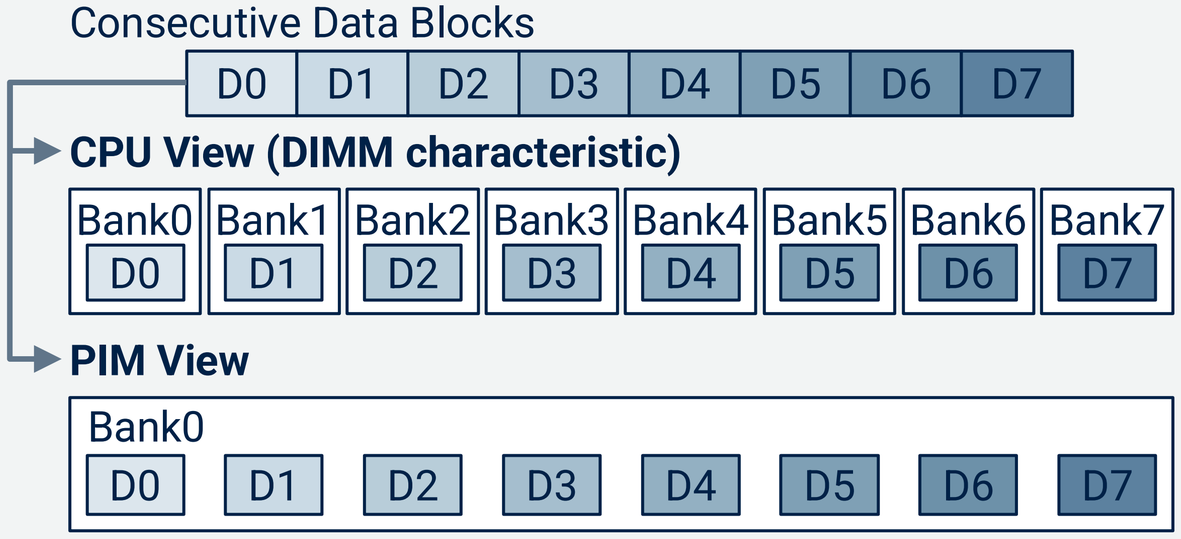


Misordering breaks encryption & integrity protection

Layout-aware security engine for CPU-PIM

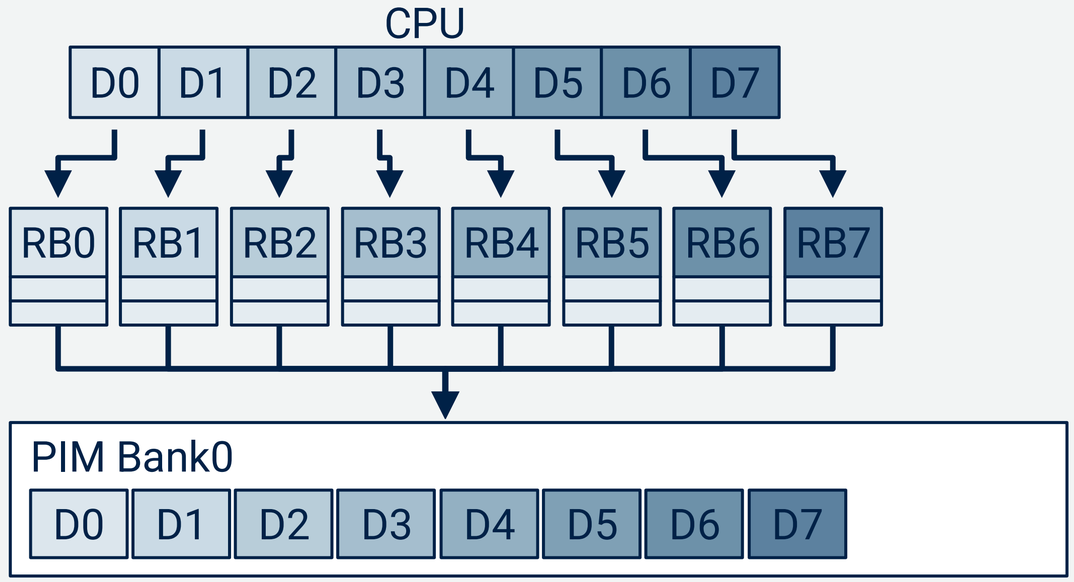
Different CPU and PIM layouts (due to DIMM arch.)

CPU: Interleaved, PIM: Sequential bank mapping



Misordering breaks encryption & integrity protection

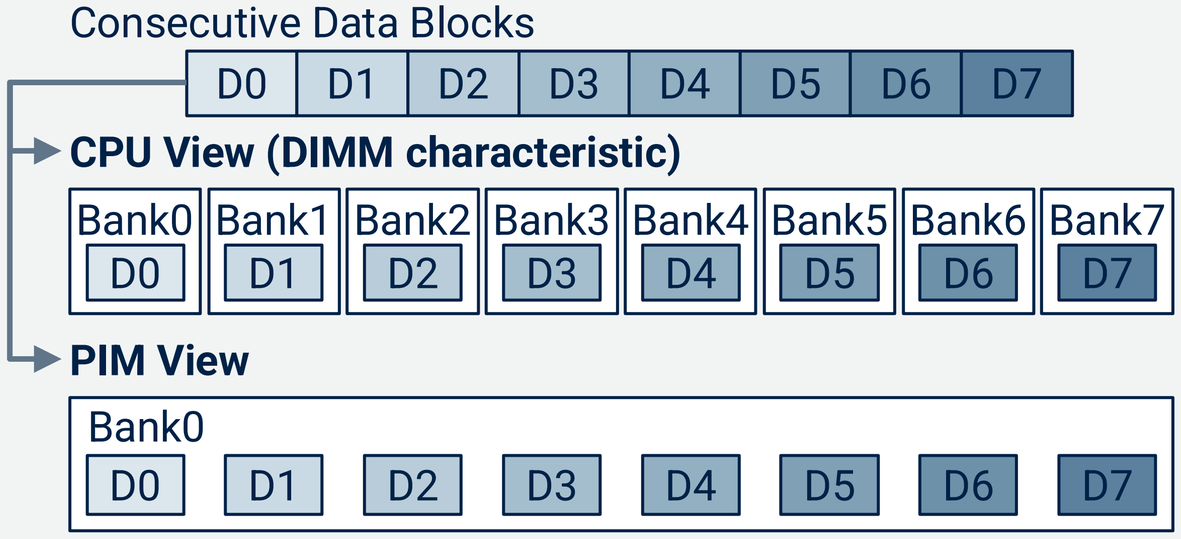
Layout-aware security engine



Layout-aware security engine for CPU-PIM

Different CPU and PIM layouts (due to DIMM arch.)

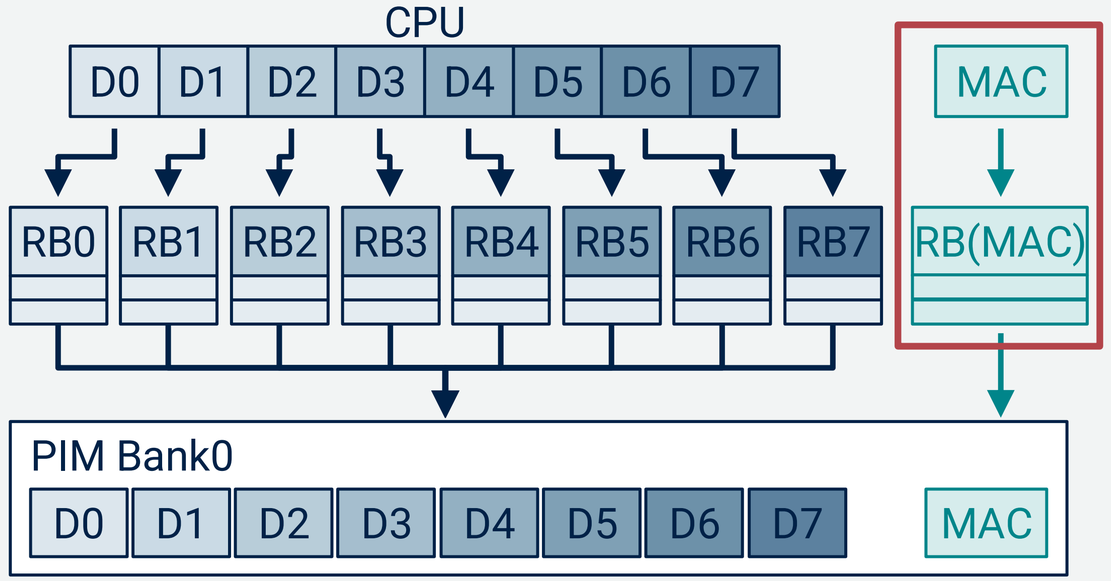
CPU: Interleaved, PIM: Sequential bank mapping



Misordering breaks encryption & integrity protection

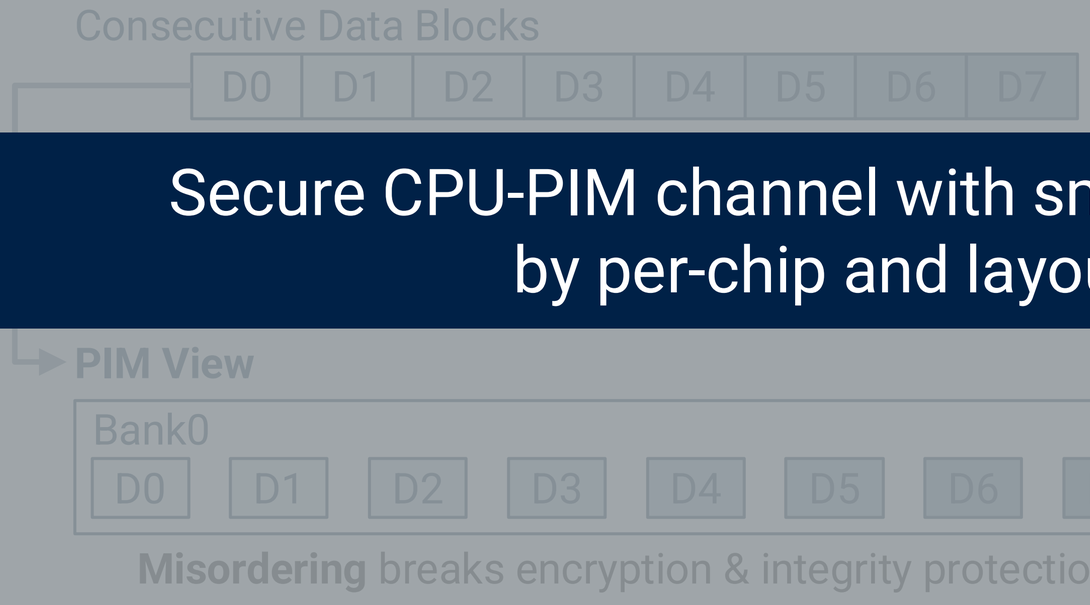
Layout-aware security engine

Additional reorder buffer (RB) for security metadata



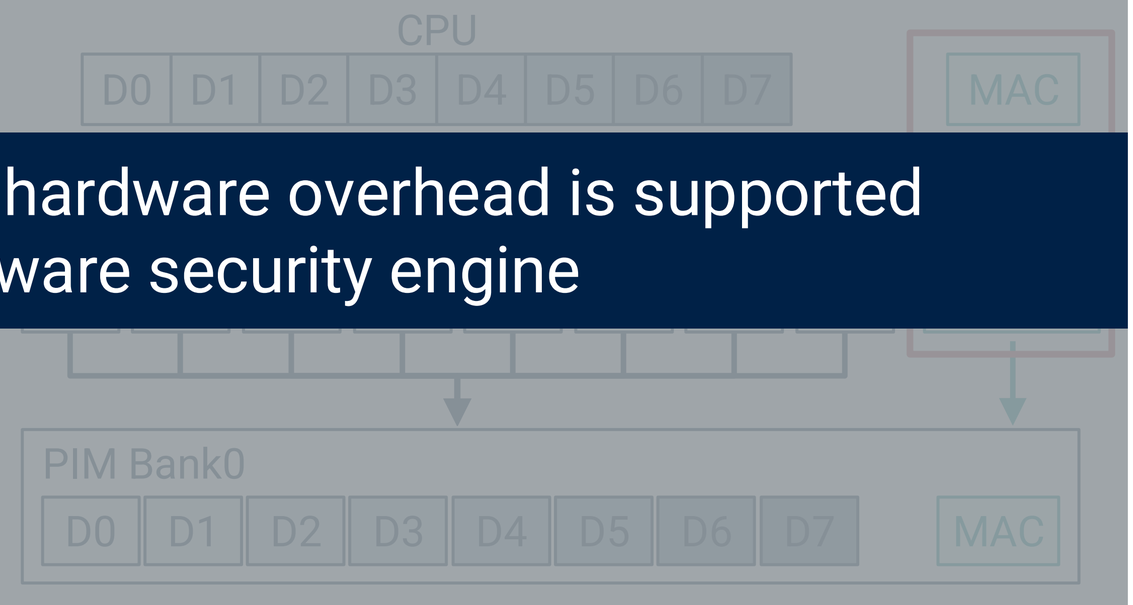
Layout-aware security engine for CPU-PIM

Different CPU and PIM layouts (due to DIMM arch.)
CPU: Interleaved, PIM: Sequential bank mapping



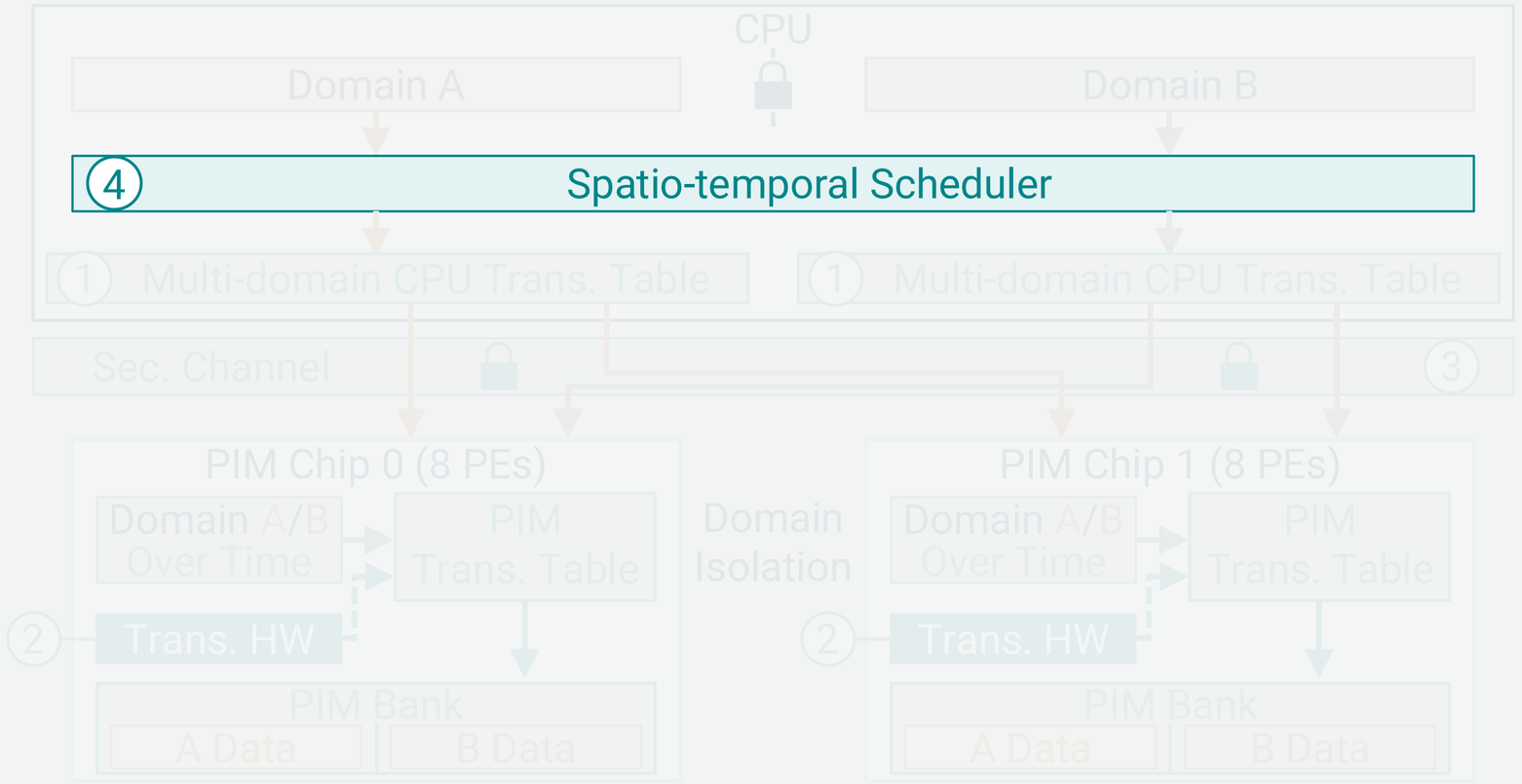
Misordering breaks encryption & integrity protection

Layout-aware security engine
Additional reorder buffer (RB) for security metadata



Secure CPU-PIM channel with small hardware overhead is supported by per-chip and layout-aware security engine

Spatio-temporal scheduling



Security

1. Chip&Page-granular Translation
   Spatial + temporal isolation
2. PIM-side Translation HW
   PIM-PIM isolation with low trans. table-switch overhead
3. Secure CPU-PIM Channel
   Per-chip, CPU/PIM layout-aware secure engine

Efficient Scheduling

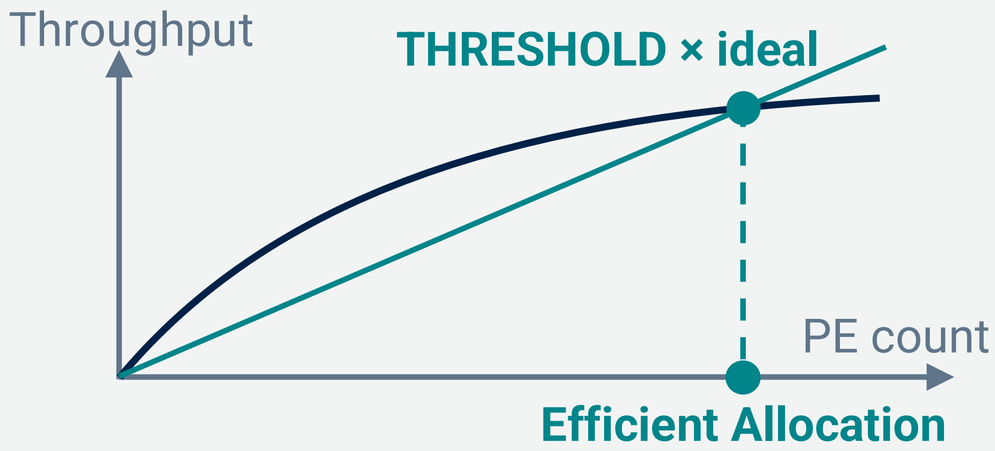
4. Spatio-temporal Scheduler
   PE allocation and time slicing

Throughput & small time slice-based scheduling

Throughput & small time slice-based scheduling

How many PEs? (Spatial sharing)

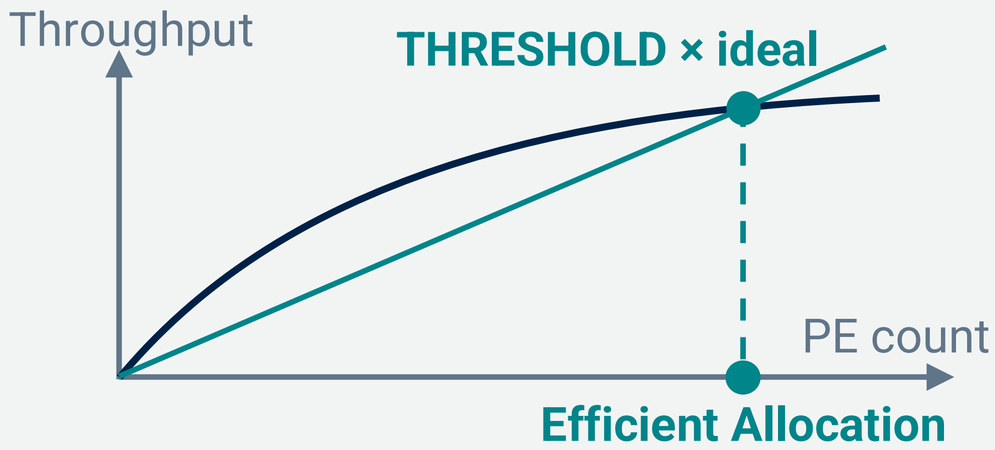
Allocate PEs while scaling remains efficient



Throughput & small time slice-based scheduling

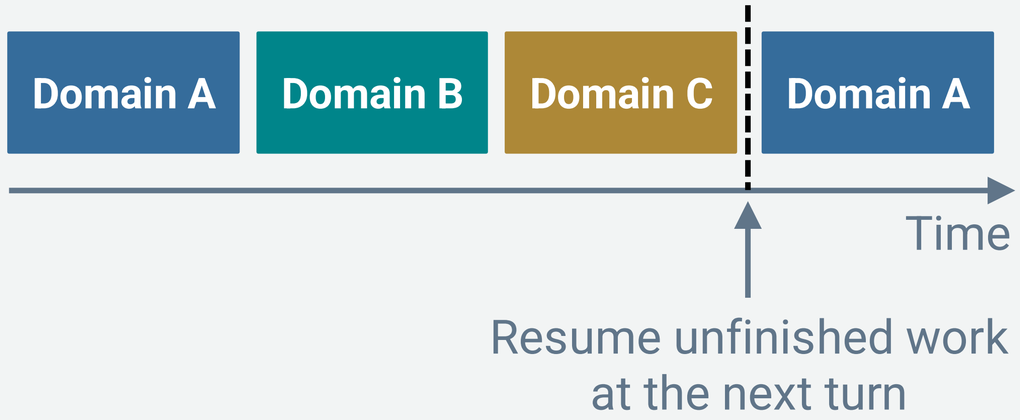
How many PEs? (Spatial sharing)

Allocate PEs while scaling remains efficient



When to switch? (Temporal sharing)

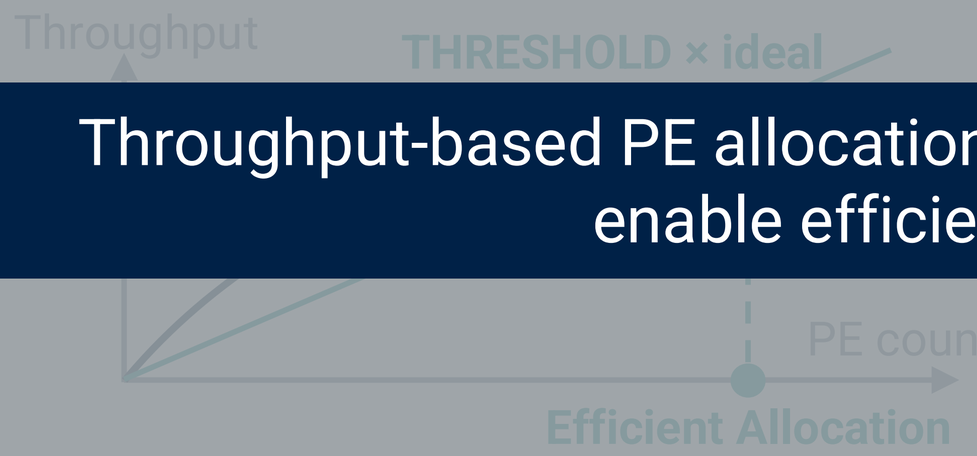
Round-robin with a small (*1 ms*) time slice



Throughput & small time slice-based scheduling

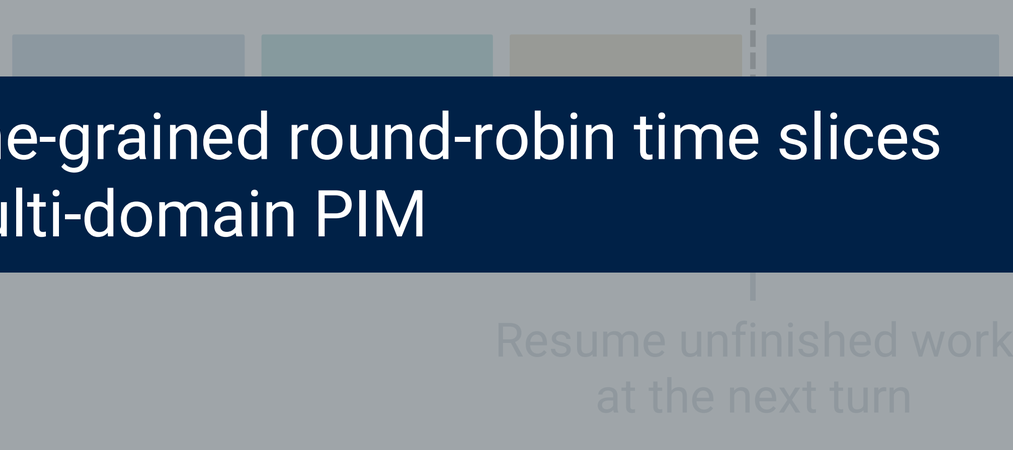
How many PEs? (Spatial sharing)

Allocate PEs while scaling remains efficient



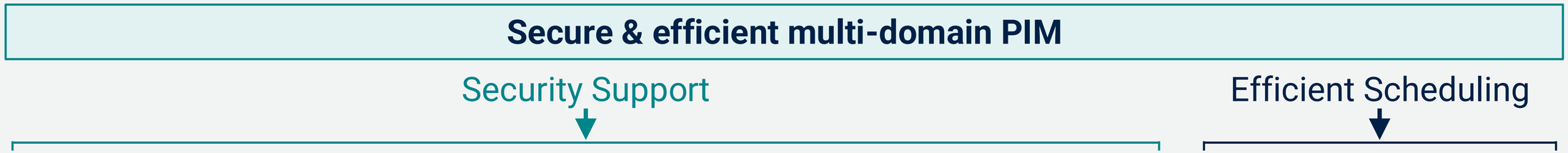
When to switch? (Temporal sharing)

Round-robin with a small (*1 ms*) time slice



Throughput-based PE allocation + fine-grained round-robin time slices enable efficient multi-domain PIM

Summary: Secure & efficient multi-domain PIM



Summary: Secure & efficient multi-domain PIM

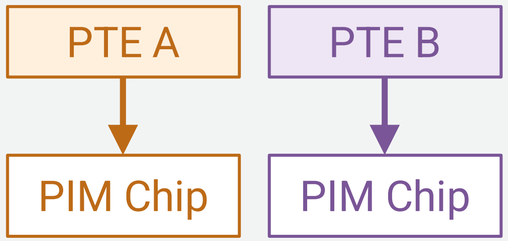
Secure & efficient multi-domain PIM

Security Support

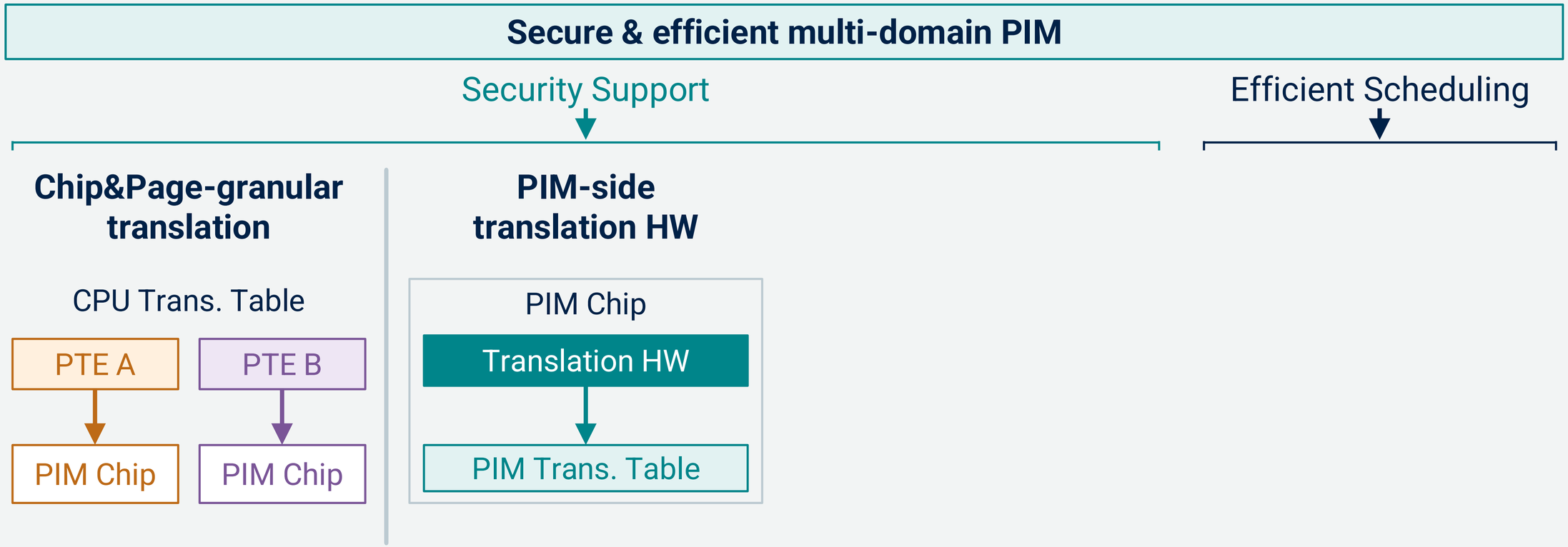
Efficient Scheduling

Chip&Page-granular translation

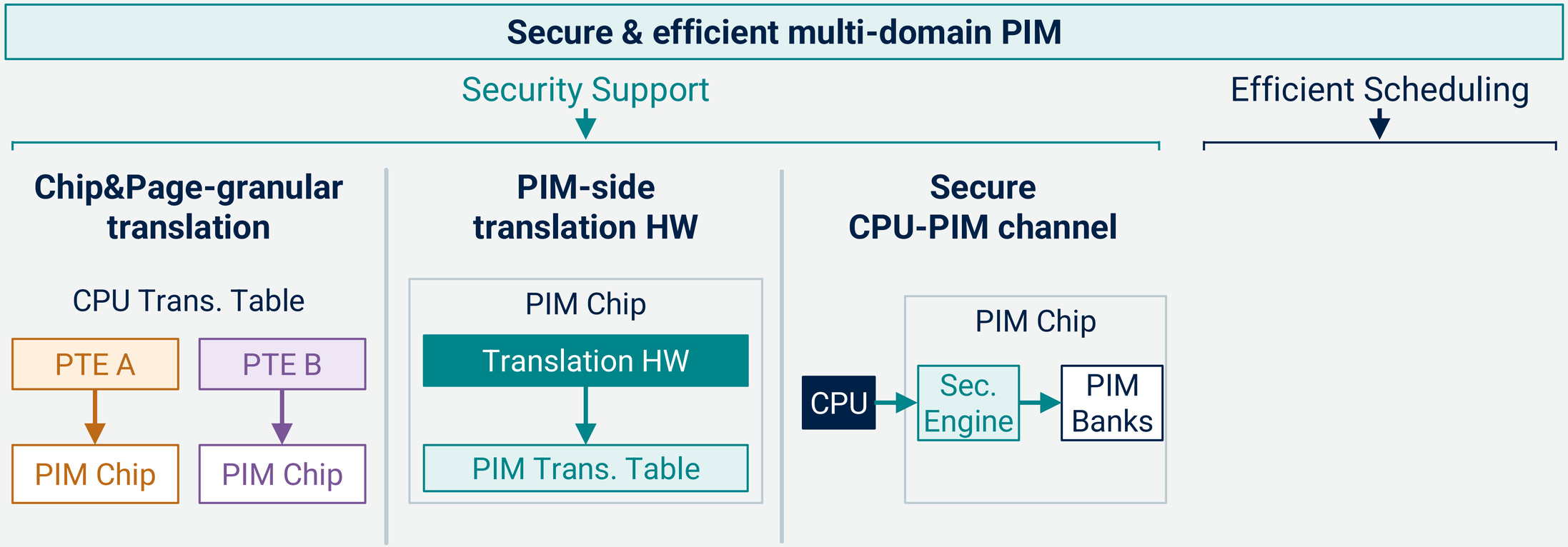
CPU Trans. Table



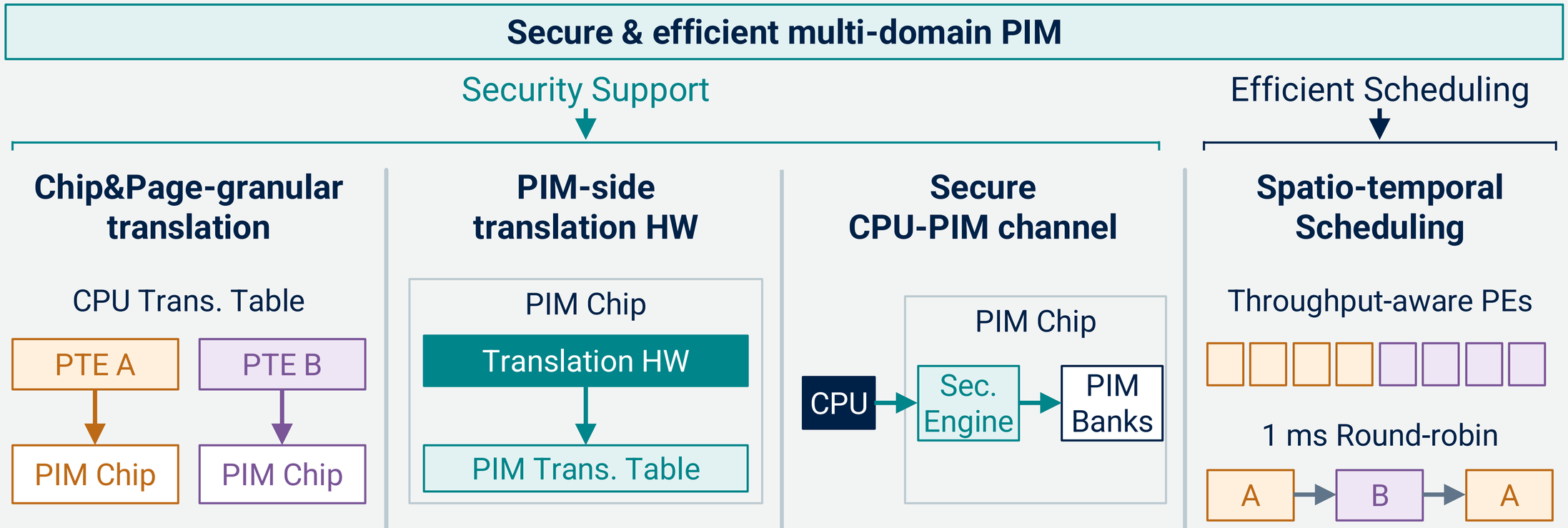
Summary: Secure & efficient multi-domain PIM



Summary: Secure & efficient multi-domain PIM



Summary: Secure & efficient multi-domain PIM



Evaluation

Evaluation

UPMEM model w/ 1) req. generator, 2) scheduler, 3) translation, 4) CPU-PIM secure channel

Evaluation

UPMEM model w/ 1) req. generator, 2) scheduler, 3) translation, 4) CPU-PIM secure channel

Three-domain scenarios with 22 workloads

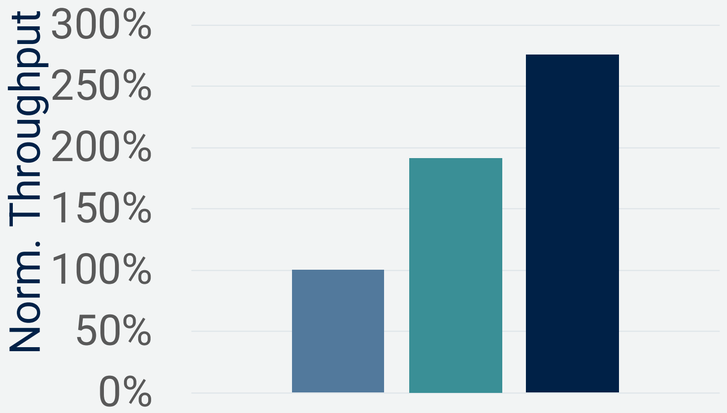
Workload families	Workloads
PrIM kernels (*PrIM*)	*TS*, *GEMV*, *HST-L*, *HST-S*, *MLP*, *VA*, *SCAN-RSS*, *SCAN-SSA*, *SEL*, *UNI*, *RED*, *BS*, *NW*, *BFS*
Recommendation (*Rec*)	*MOVIE*, *BOOK*
Small language model	*GEMMA*, *TLLAMA*

Scenarios	Workloads	Scenarios	Workloads
LM1	*GEMMA*, *PrIM*, *PrIM*	***LM2***	*TLLAMA*, *PrIM*, *PrIM*
DualLM	*GEMMA*, *TLLAMA*, *PrIM*	***RecMajor***	*Rec*, *PrIM*, *PrIM*
LMRec1	*GEMMA*, *Rec*, *PrIM*	***LMRec2***	*TLLAMA*, *Rec*, *PrIM*

Throughput & latency

175%

higher throughput

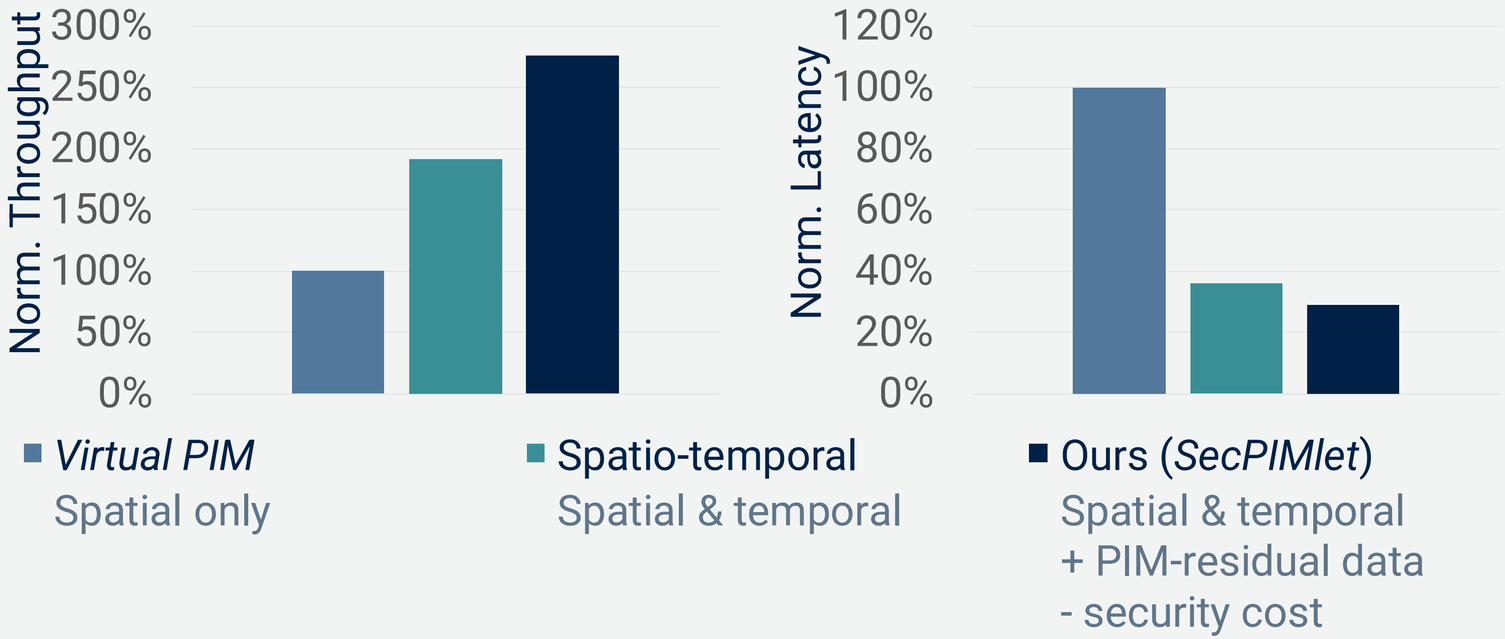


- *Virtual PIM*
  Spatial only
- Spatio-temporal
  Spatial & temporal
- Ours (*SecPIMlet*)
  Spatial & temporal
  + PIM-residual data
  - security cost

Throughput & latency

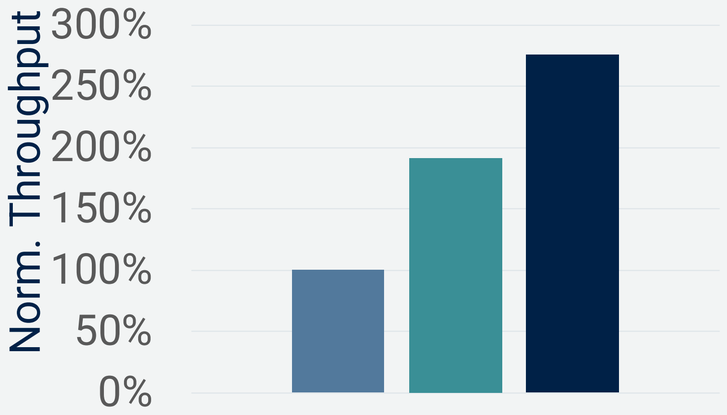
175%
higher throughput

71%
lower latency

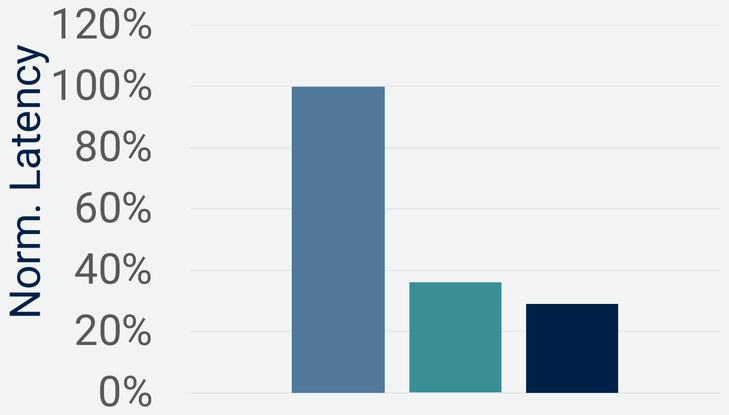


Throughput & latency

175%
higher throughput



71%
lower latency



- *Virtual PIM*
  Spatial only
- Spatio-temporal
  Spatial & temporal
- Ours (*SecPIMlet*)
  Spatial & temporal
  + PIM-residual data
  - security cost

Latency reduction analysis

64%
Spatio-temporal sharing

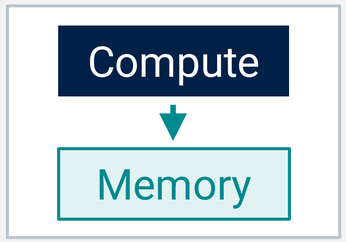
9%
PIM-residual data

-2%
Security cost

SecPIMlet: Breaking the AI memory wall securely

SecPIMlet: Breaking the AI memory wall securely

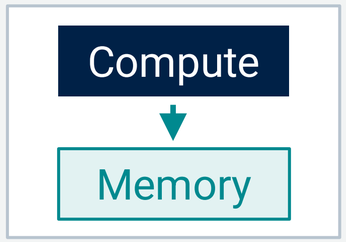
PIM



Breaks the
memory wall

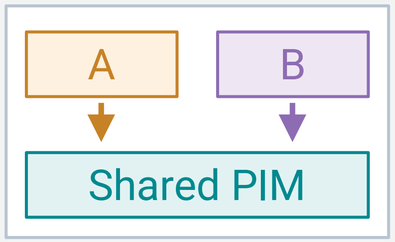
SecPIMlet: Breaking the AI memory wall securely

PIM



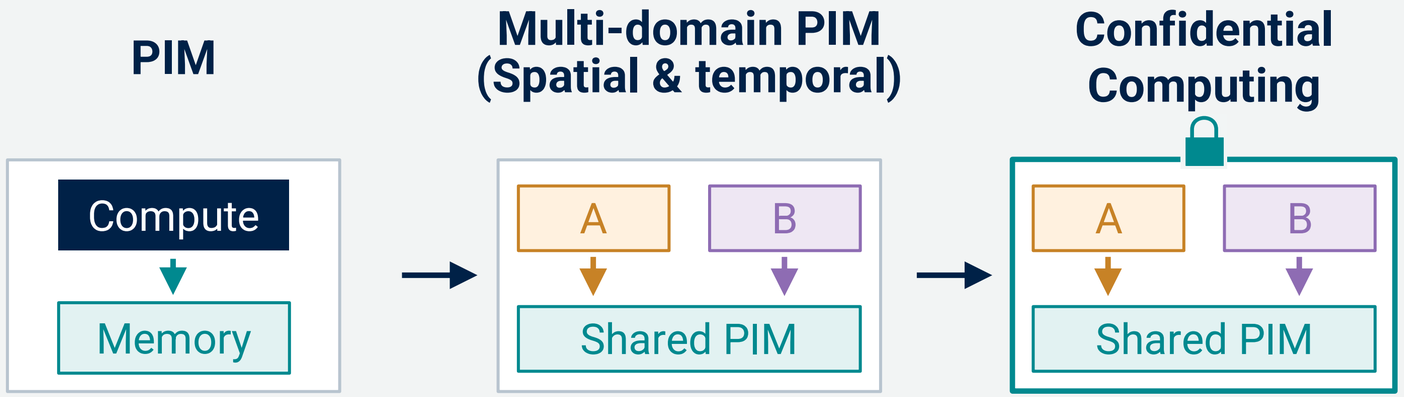
Breaks the
memory wall

Multi-domain PIM
(Spatial & temporal)



Performance gain
in **general** usages

SecPIMlet: Breaking the AI memory wall securely

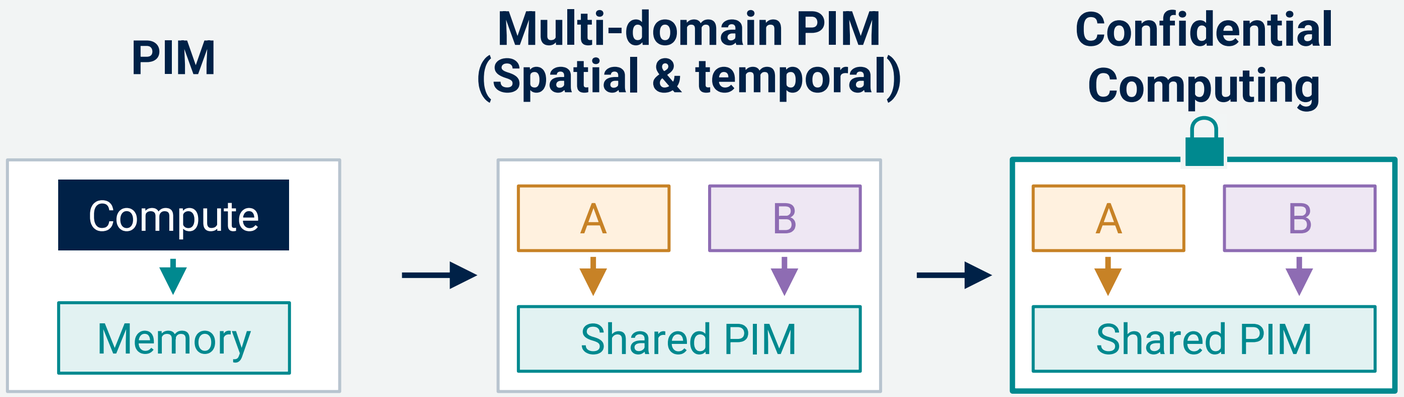


Breaks the **memory wall**

Performance gain in **general** usages

Supports **security** of data & execution

SecPIMlet: Breaking the AI memory wall securely



Breaks the **memory wall**

Performance gain in **general** usages

Supports **security** of data & execution

***SecPIMlet*: First confidential PIM study for spatio-temporal multi-domain sharing**

Result

Confidential & efficient multi-domain PIM

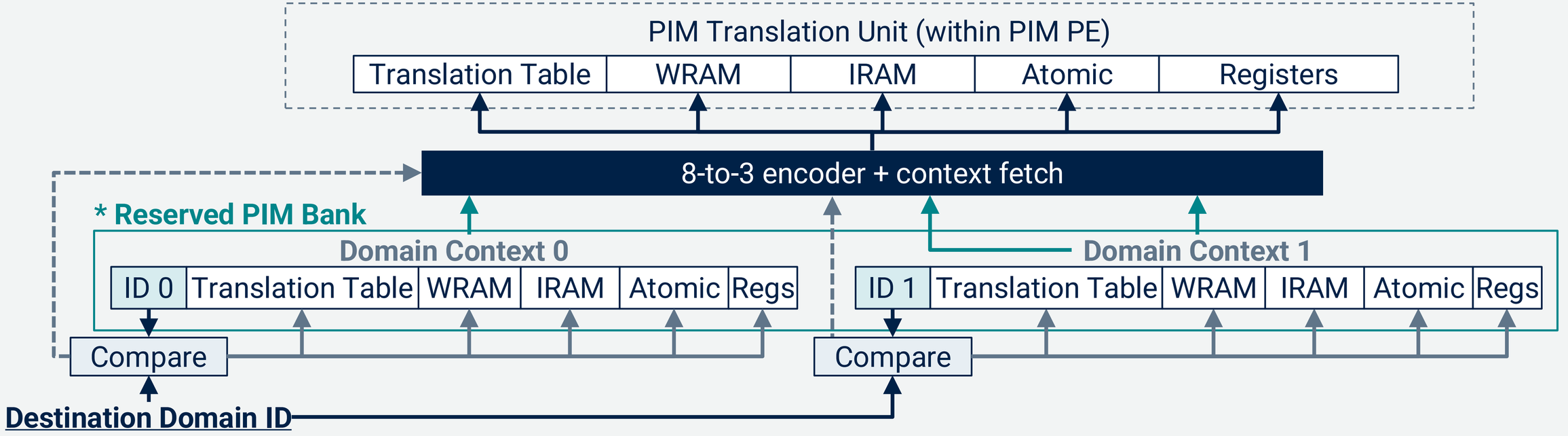
Proposed Mechanism

1) Chip&Page-level translation
2) PIM-side translation HW
3) CPU-PIM secure channel
4) Efficient scheduling



Thank you

Architecture of PIM translation HW



*** Reserved Bank protection against PA tampering**

1) CPU-side: Existing CPU access control + cmd encryption
2) PIM-side: HW-only access + packaged-chip physical protection

Security scope and trusted initialization

Area	Scope
Trusted computing base	Packaged CPU/PIM hardware and verified RMM
Initialization	Secure boot, attestation, and session-key exchange
Channel guarantees	Confidentiality, integrity, and freshness
Out-of-scope	Rowhammer, side-channel, and availability attacks

Plaintext inside a packaged chip follows the physical-protection assumption.

Evaluation detail configurations

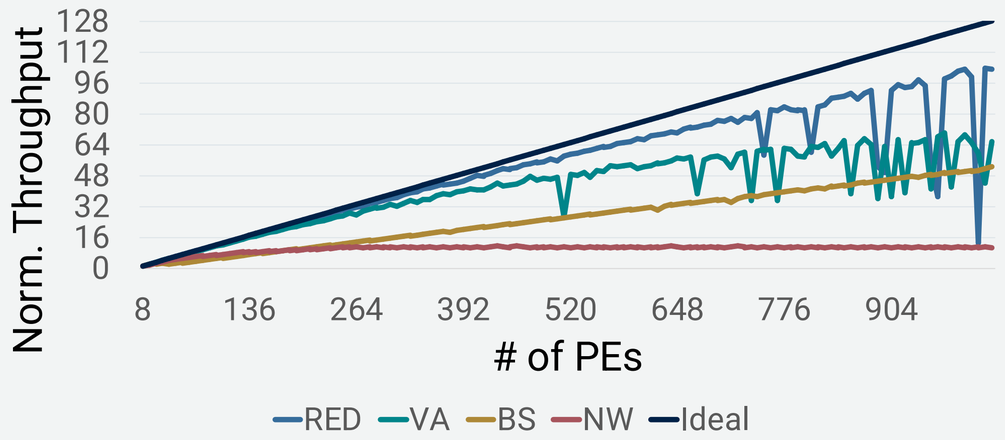
Component	Configuration
PIM-side translation	2-cycle penalty
CPU-side translation	4-level page table
AES-GCM	11-cycle scheme, 8B counter and 8B MAC
Workloads	22 workloads in 3 workload types
Scenarios	6 three-domain scenarios

Why multi-domain PIM?

General-purpose PIM under-utilizes PIM resources in client AI scenarios.

Low PE utilization

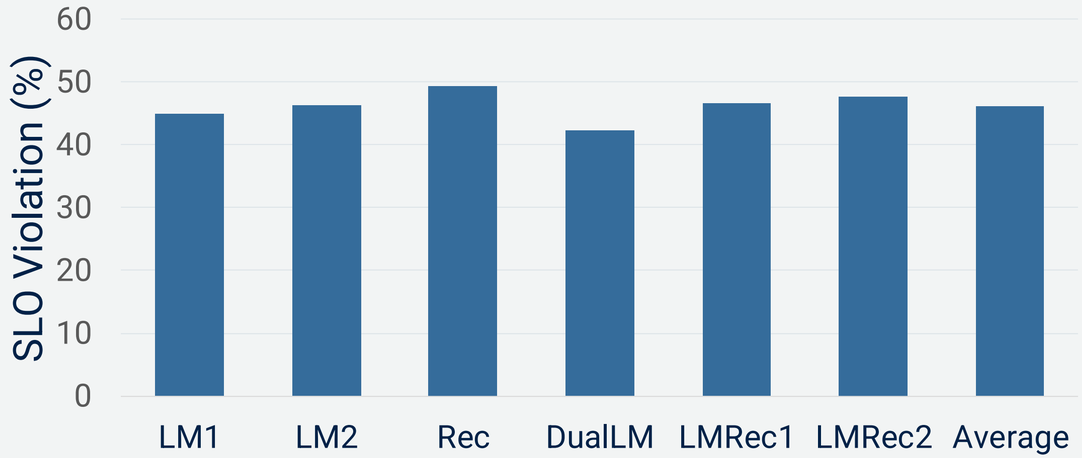
1024 PEs: 19-91% shortfall vs. ideal



Spatial sharing

High SLO violation

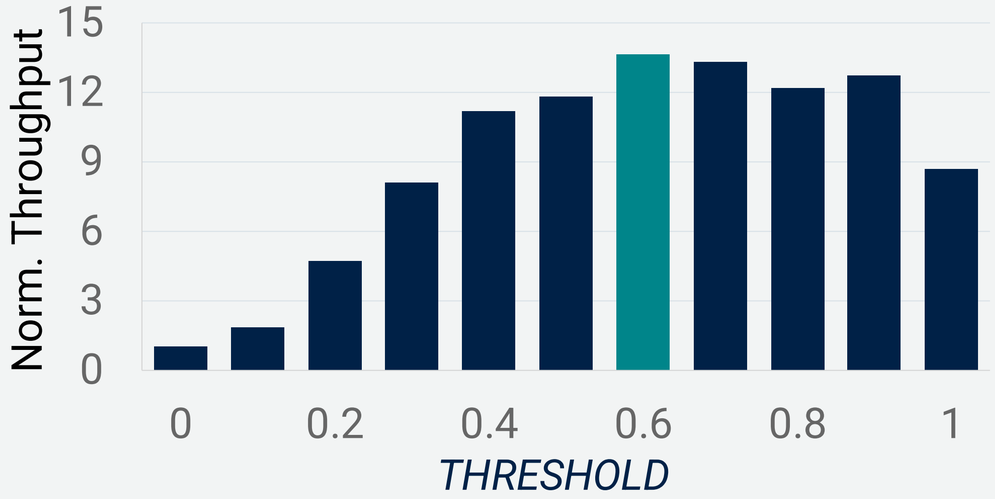
42.3-49.3% SLO violation

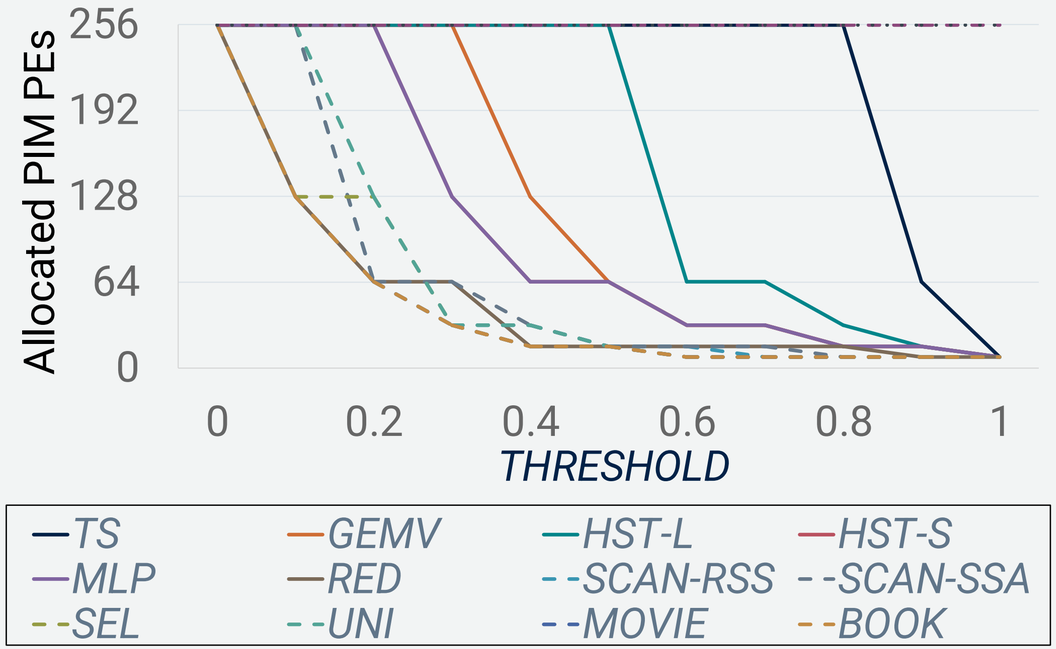


Temporal sharing

THRESHOLD analysis

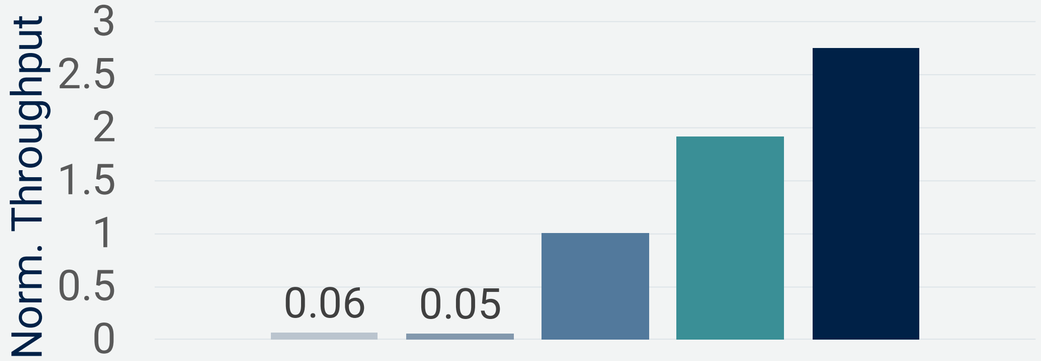
Select the largest throughput *THRESHOLD*, and its allocation distribution



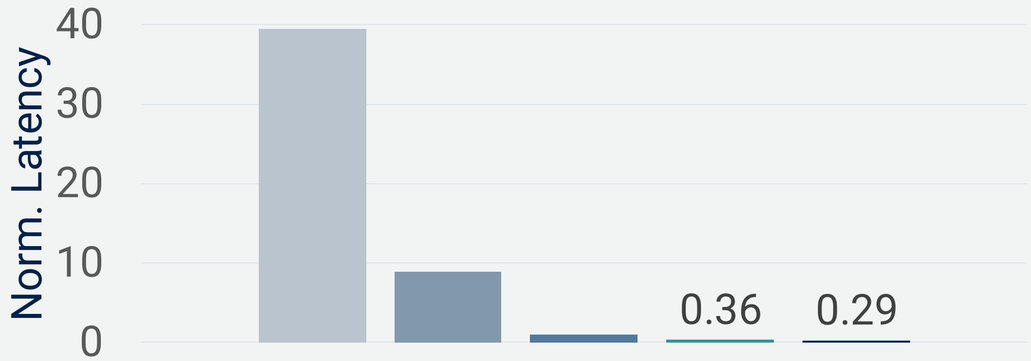


Throughput & latency

175.4% higher throughput



70.5% lower latency



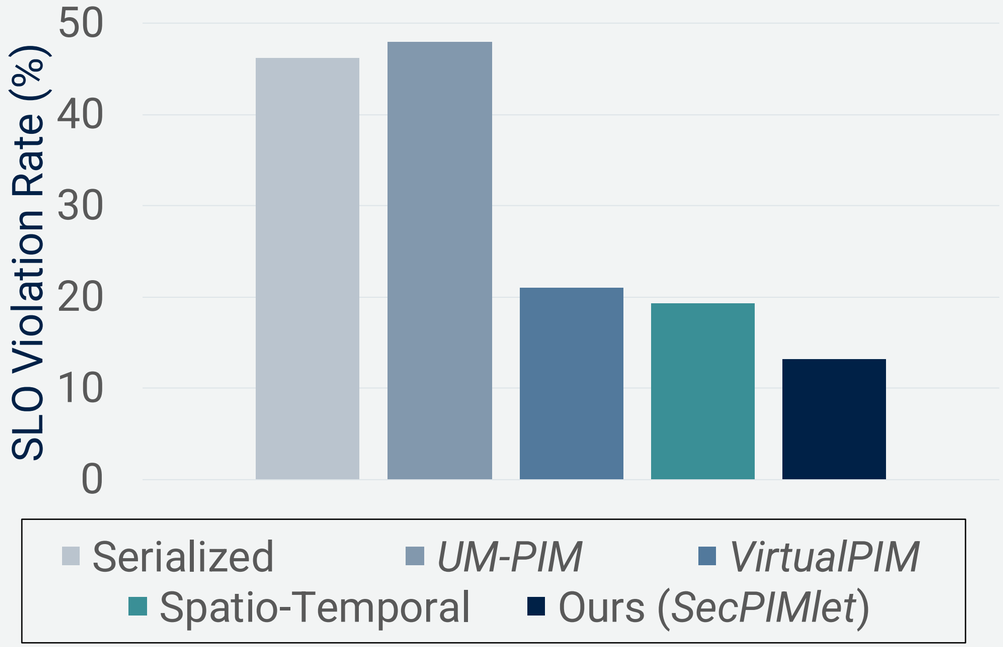
- Serialized — Serial execution
- *UM-PIM* — Temporal only + PIM-residual data
- *VirtualPIM* — Spatial only
- Spatio-temporal — Spatial & temporal
- Ours (*SecPIMlet*) — Spatial & temporal + PIM-residual data − security cost

Latency reduction analysis

1) **63.7%** (spatio-temporal sharing), 2) **8.5%** (PIM-residual data), 3) **-1.7%** (security cost)

SLO violation rate & request-rate scalability study

46.2% → 13.1% SLO violation rate



223% higher throughput in high request rates

