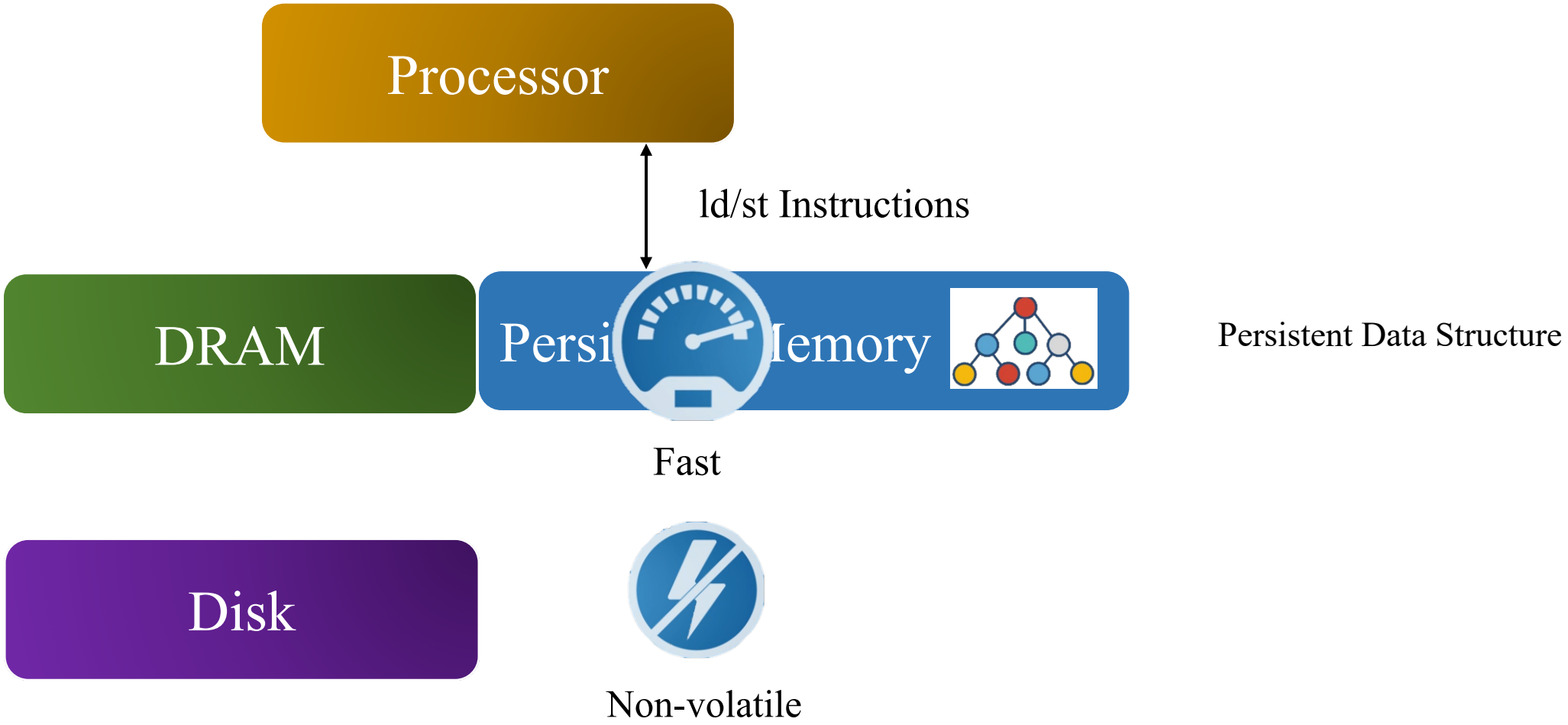
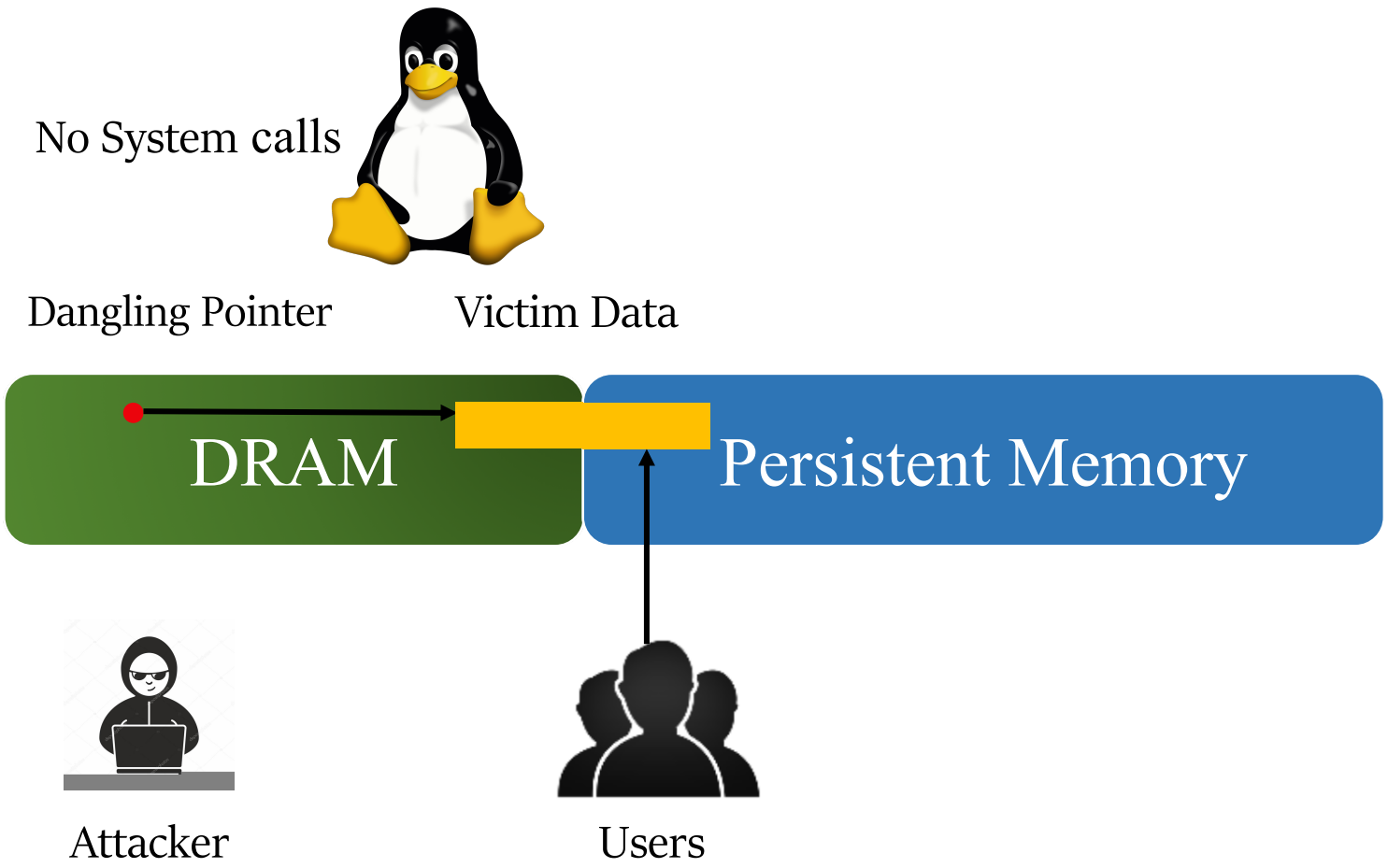
MERR: Improving Security of Persistent Memory Objects via Efficient Memory Exposure Reduction and Randomization

Yuanchao Xu, Yan Solihin, Xipeng Shen

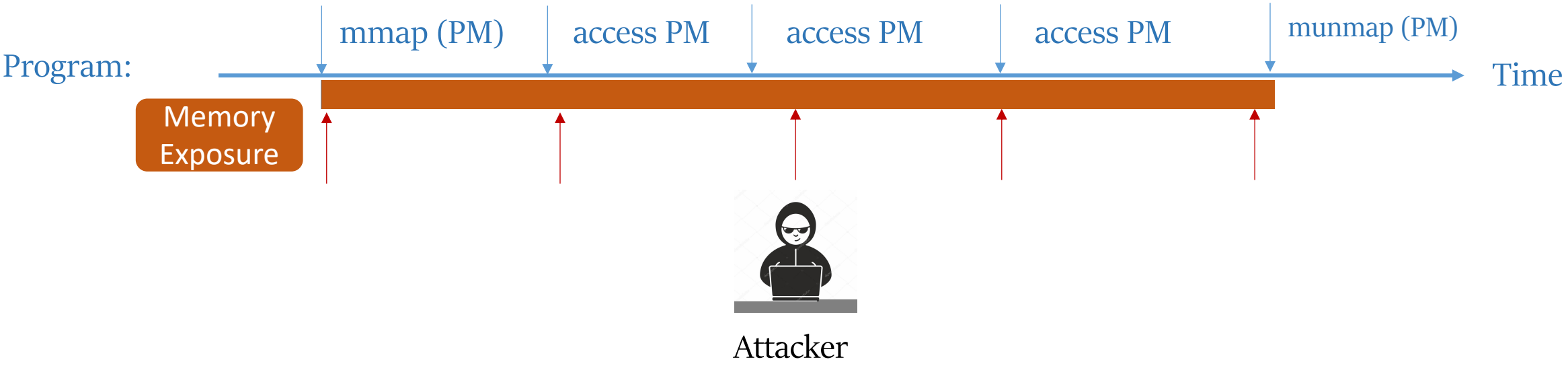
Persistent Memory (PM)



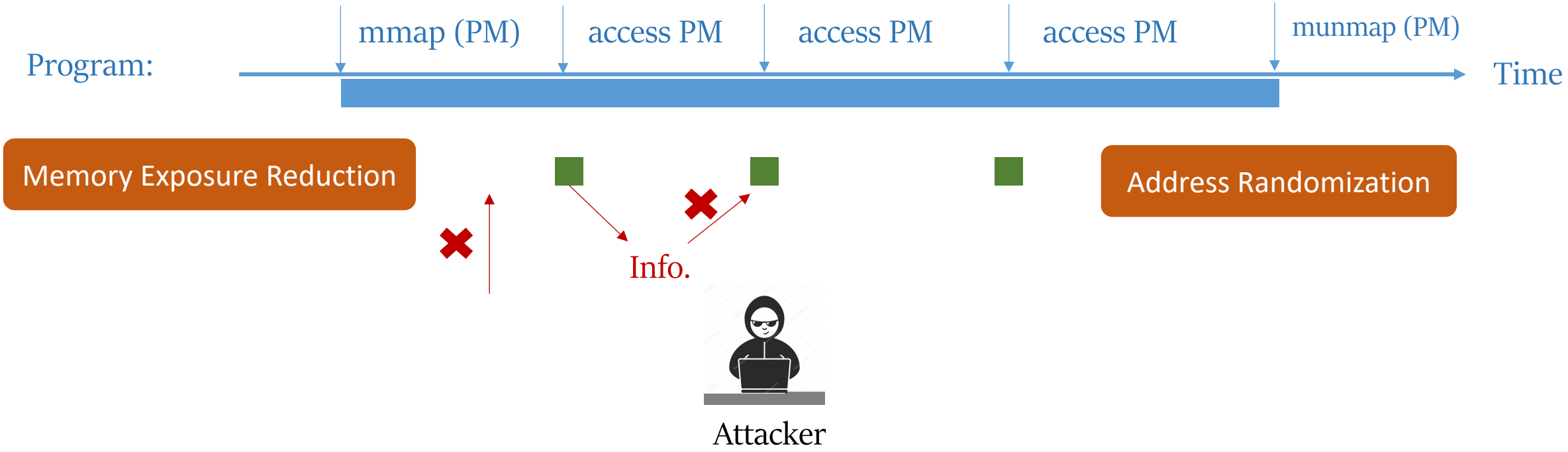
Security is more Important for PM



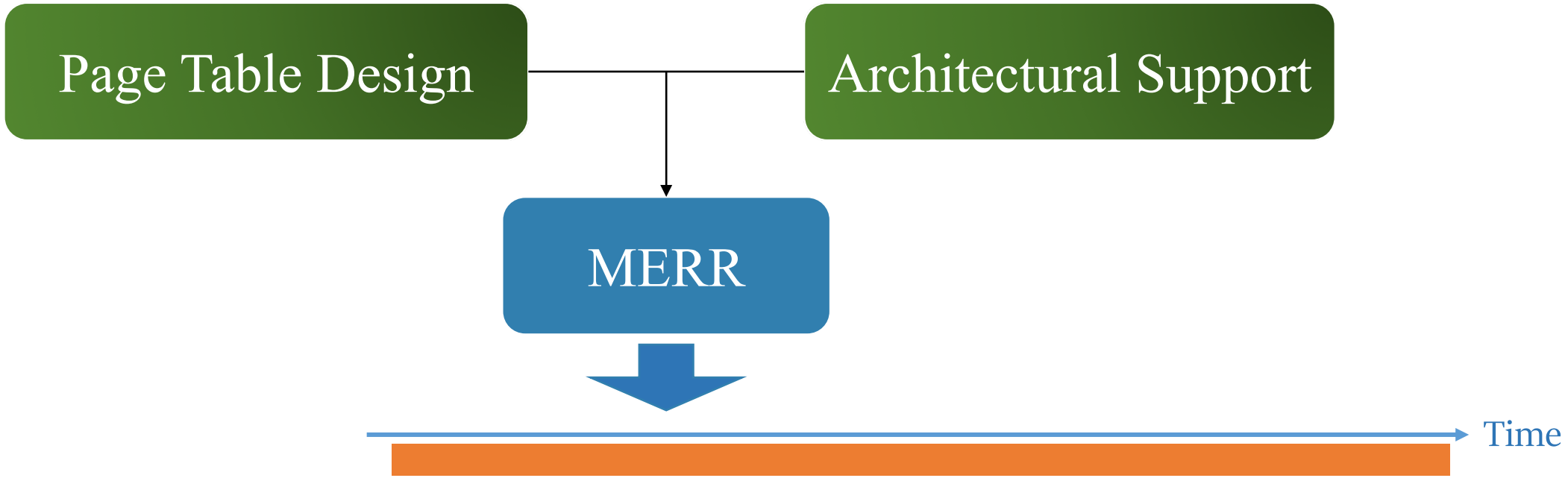
Idea: Memory Exposure Reduction & Randomization



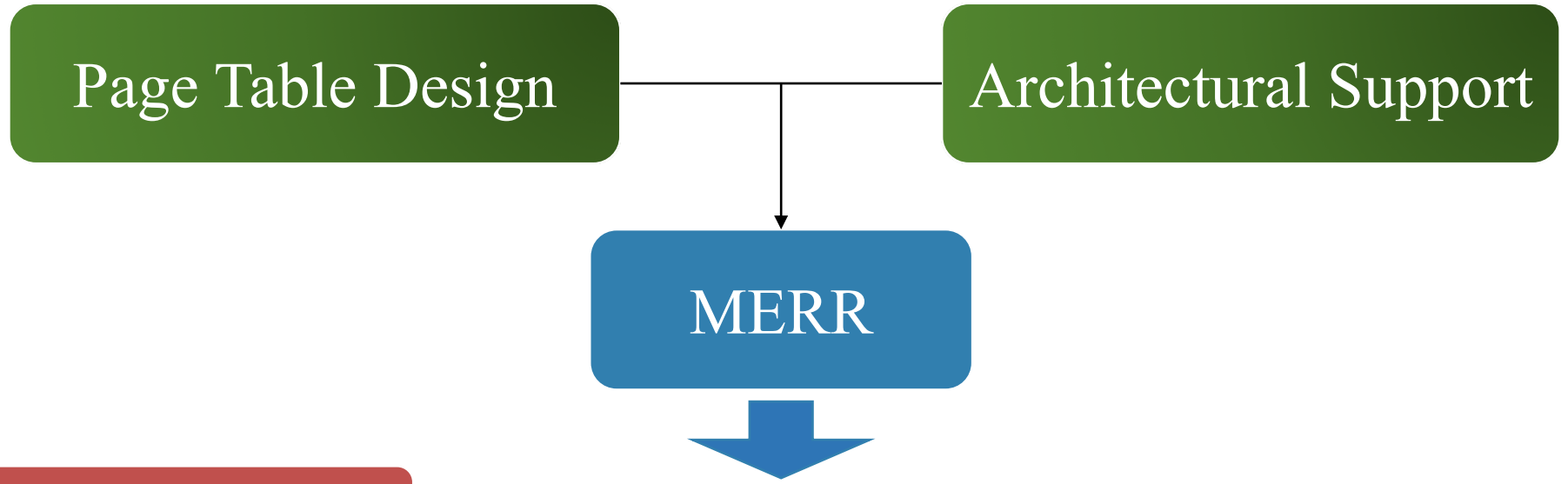
Idea: Memory Exposure Reduction & Randomization



MERR



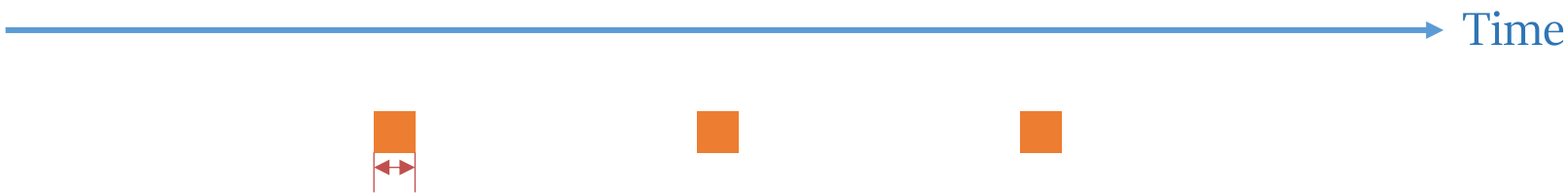
MERR



70% Reduction 10% Overhead

80us Temporal Window

10^5 Randomization Frequency



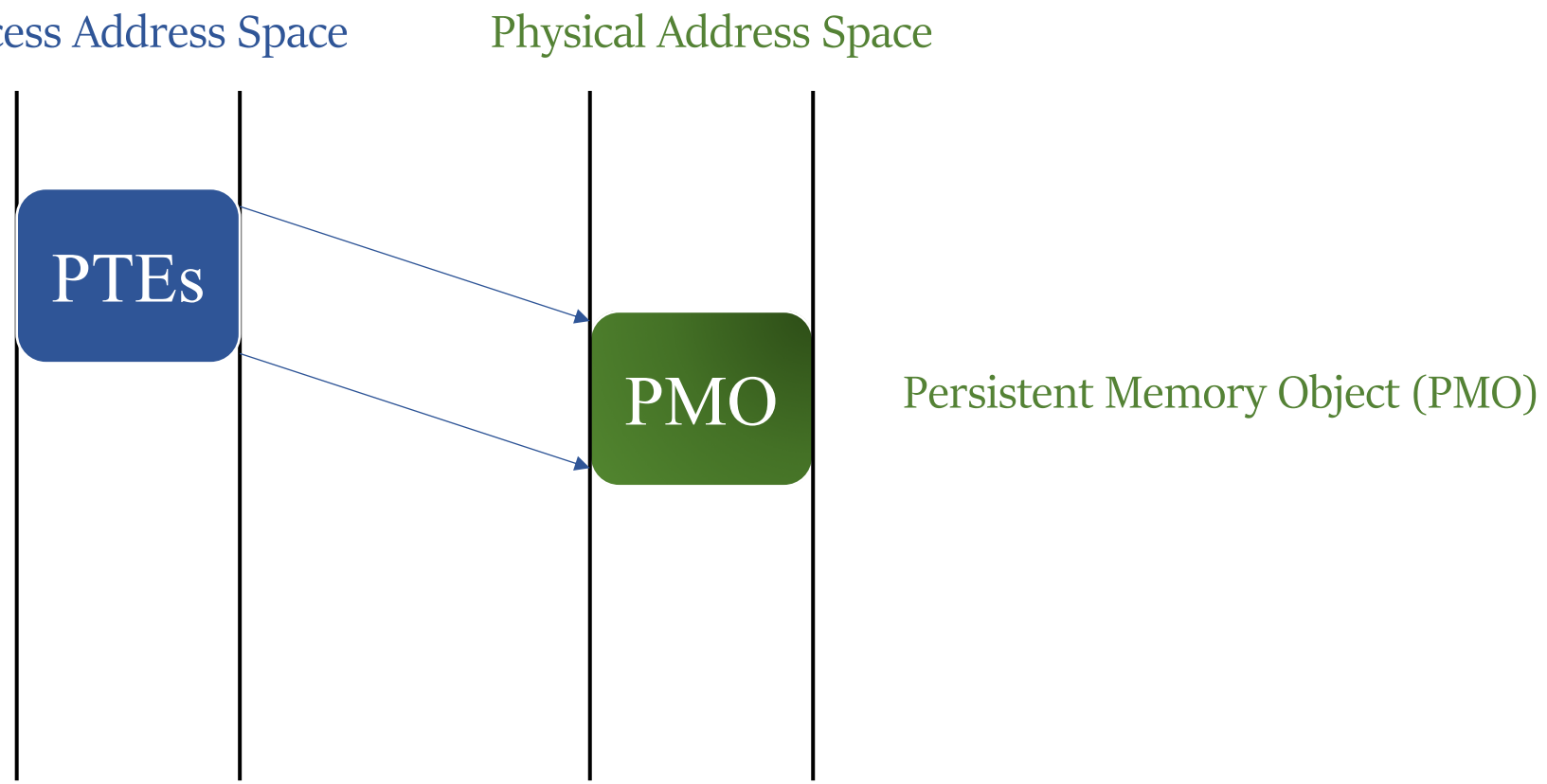
Outline

- Attach and Detach system calls
- Page table design
- Randomization
- Architecture support
- Evaluation

Attach & Detach

Attach
(PMO ID, Permission)

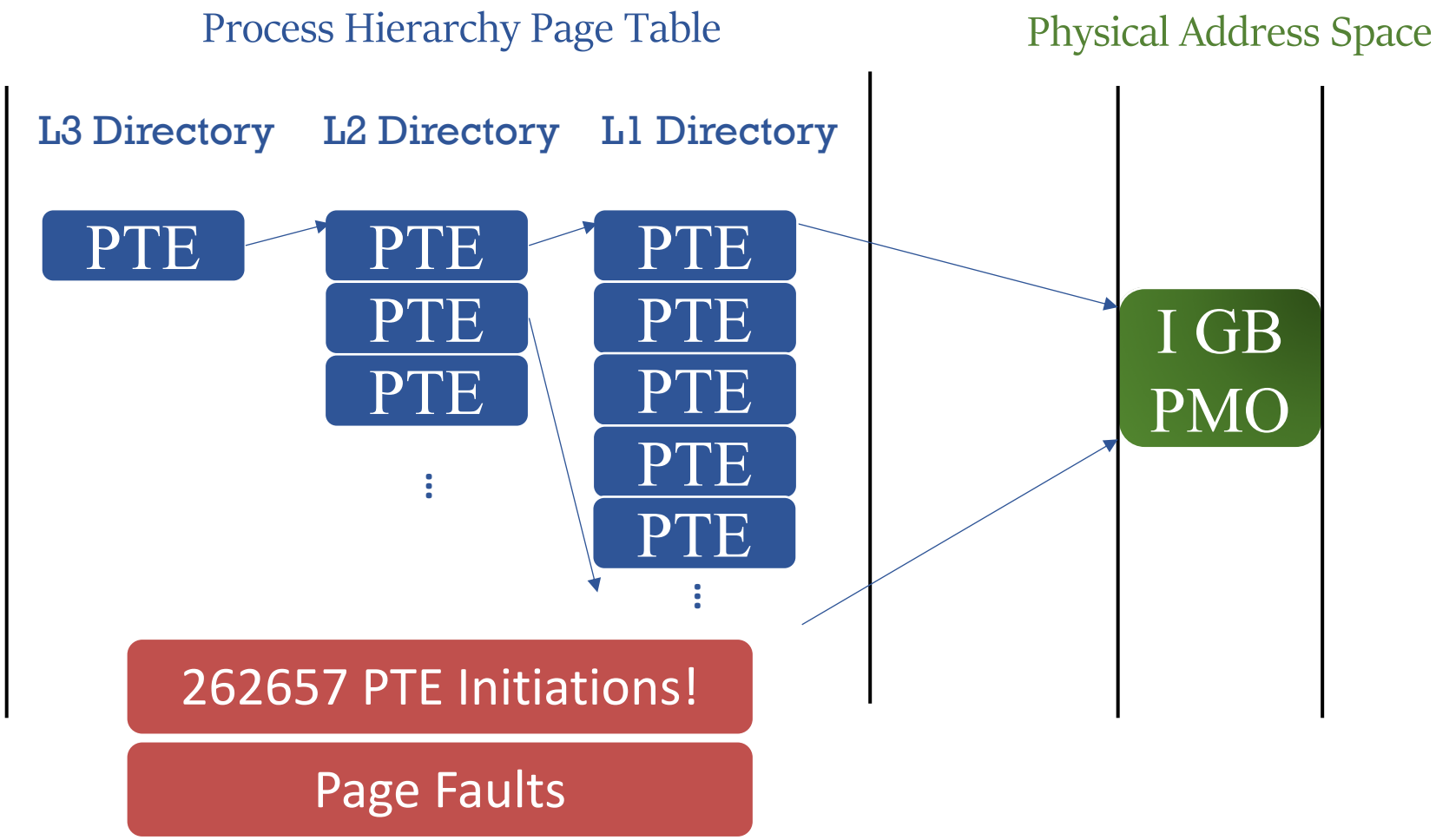
Detach(PMO ID)



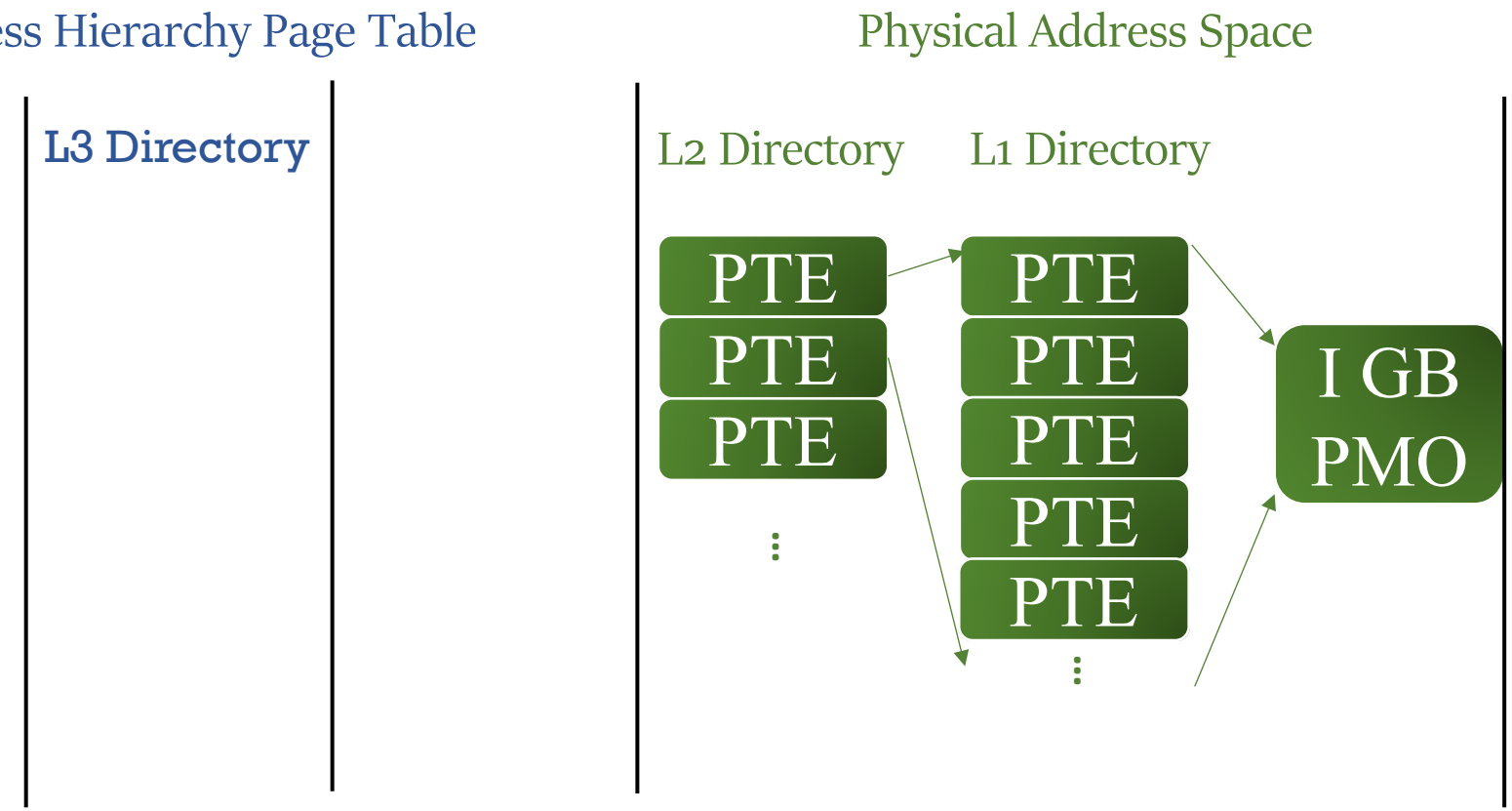
Efficiency Challenge

Attach
(PMO ID, Permission)

PMO Access



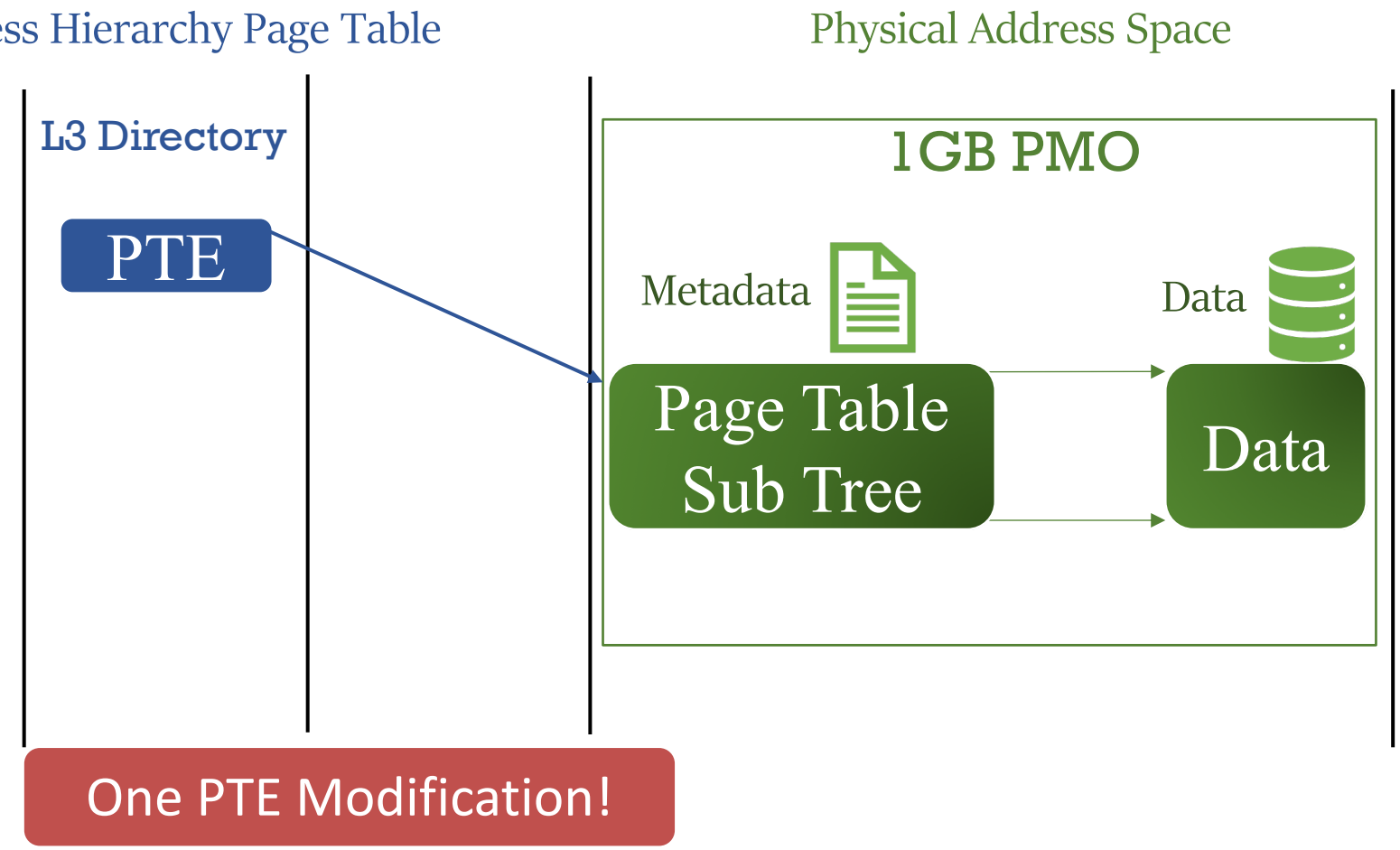
Embedding Page Table Subtree



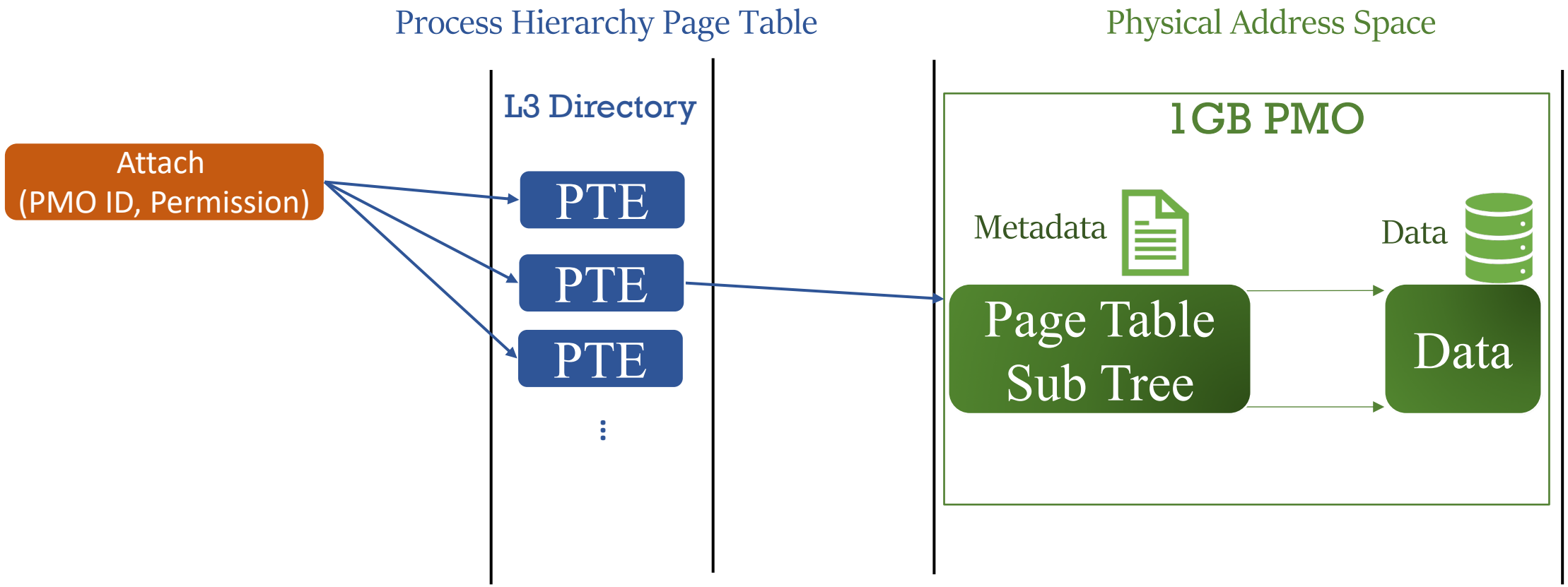
Embedding Page Table Subtree

Attach
(PMO ID, Permission)

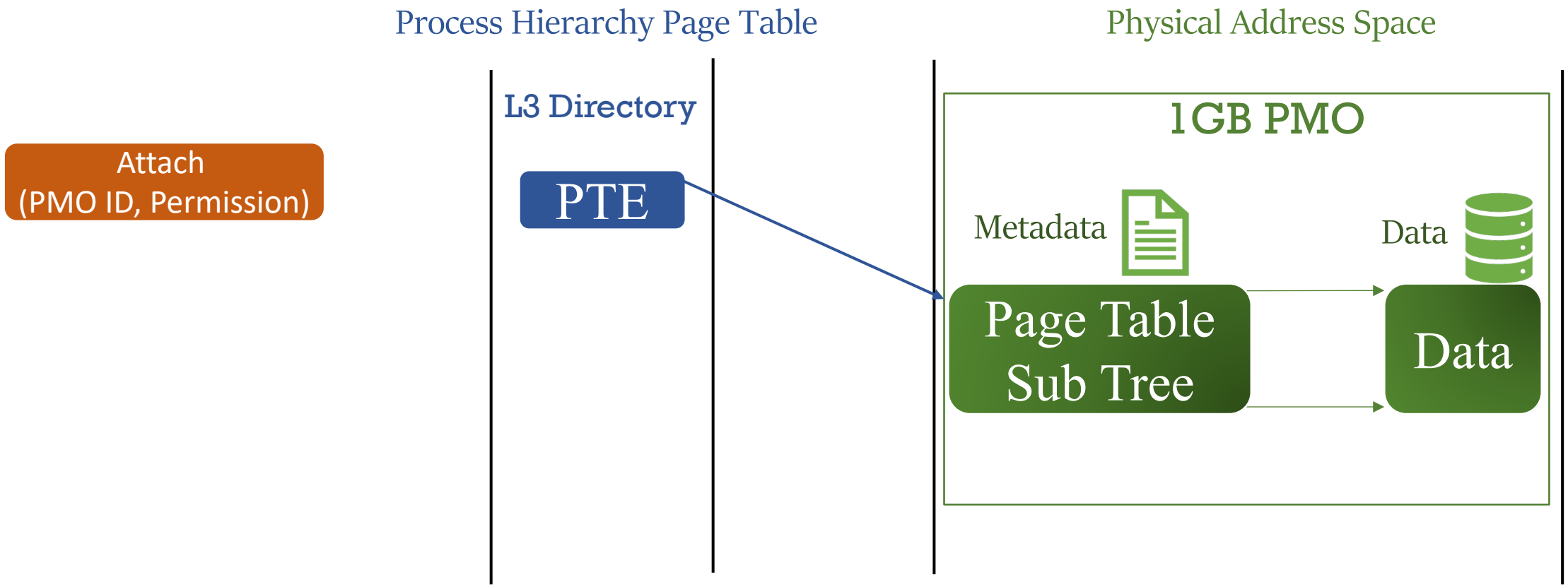
Detach (PMO ID)



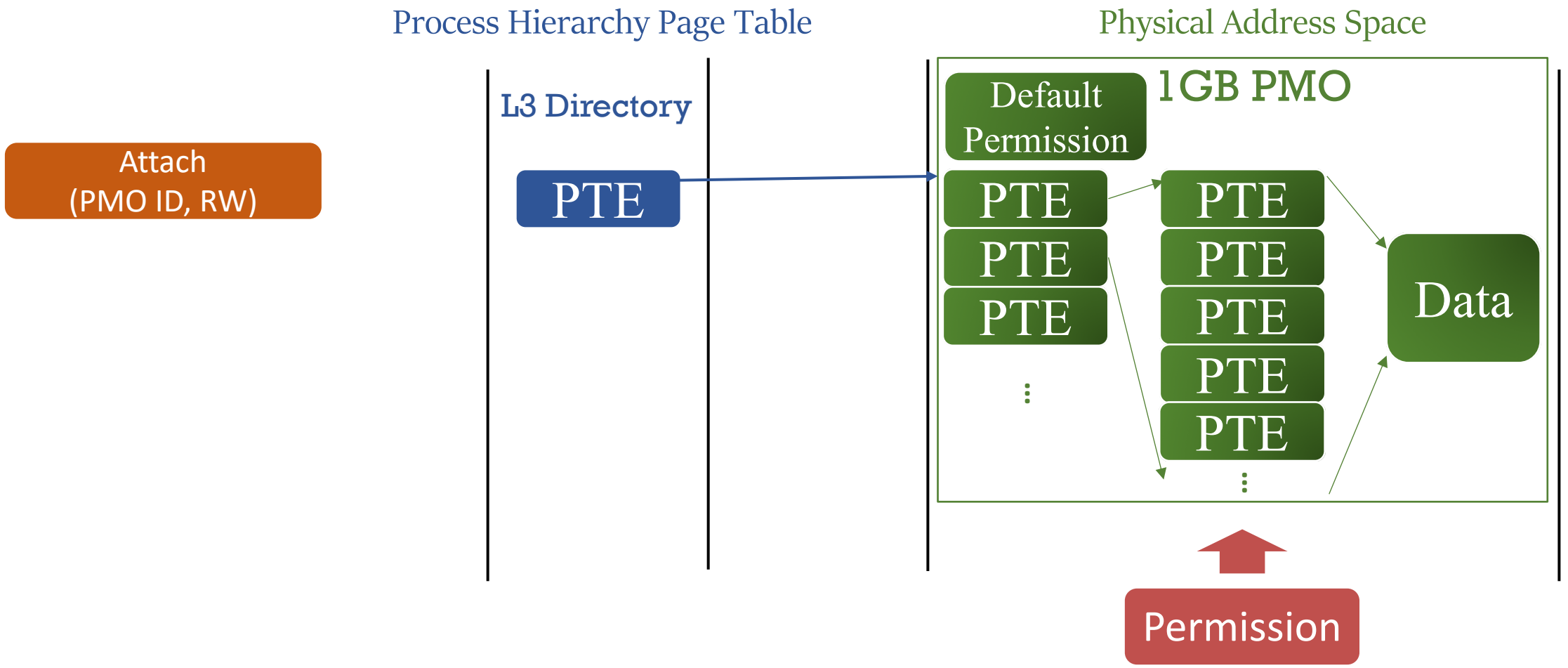
PMO Space Layout Randomization



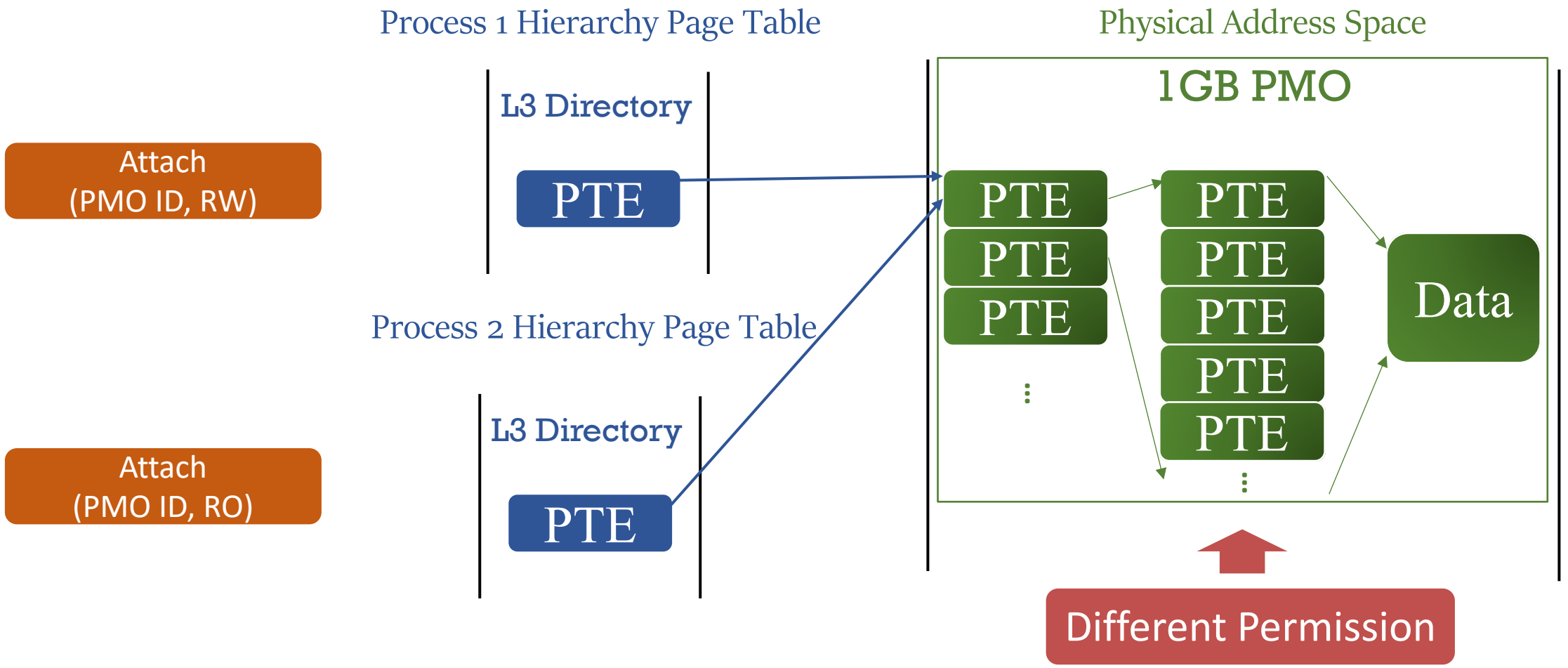
Permission Control Challenges



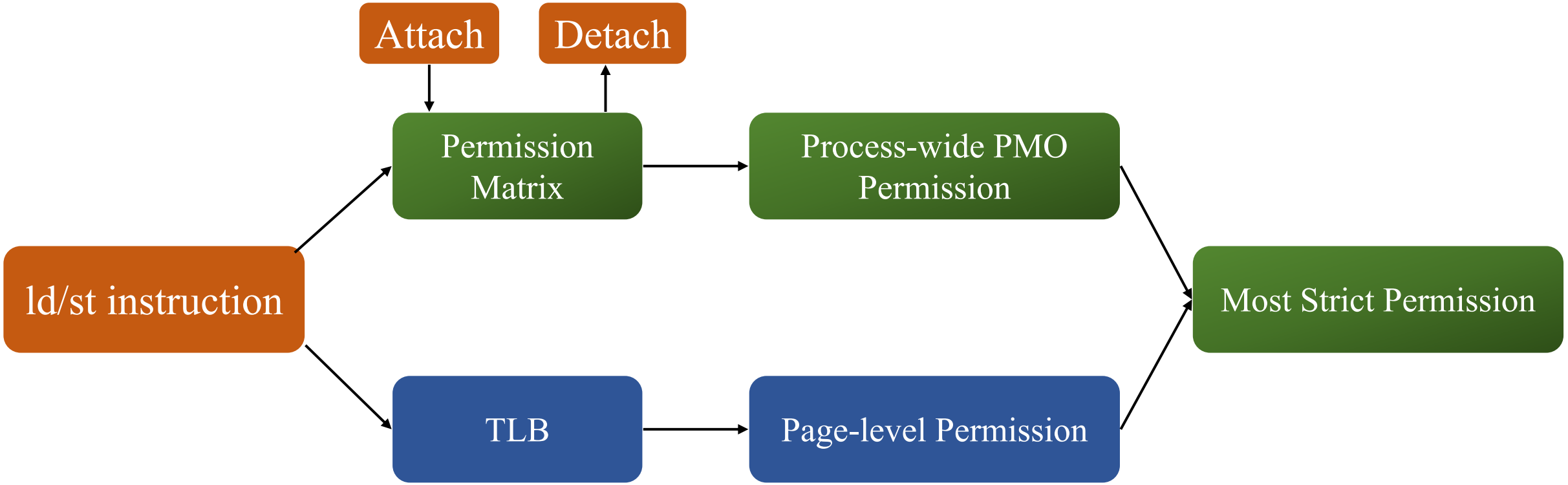
Permission Control Challenges



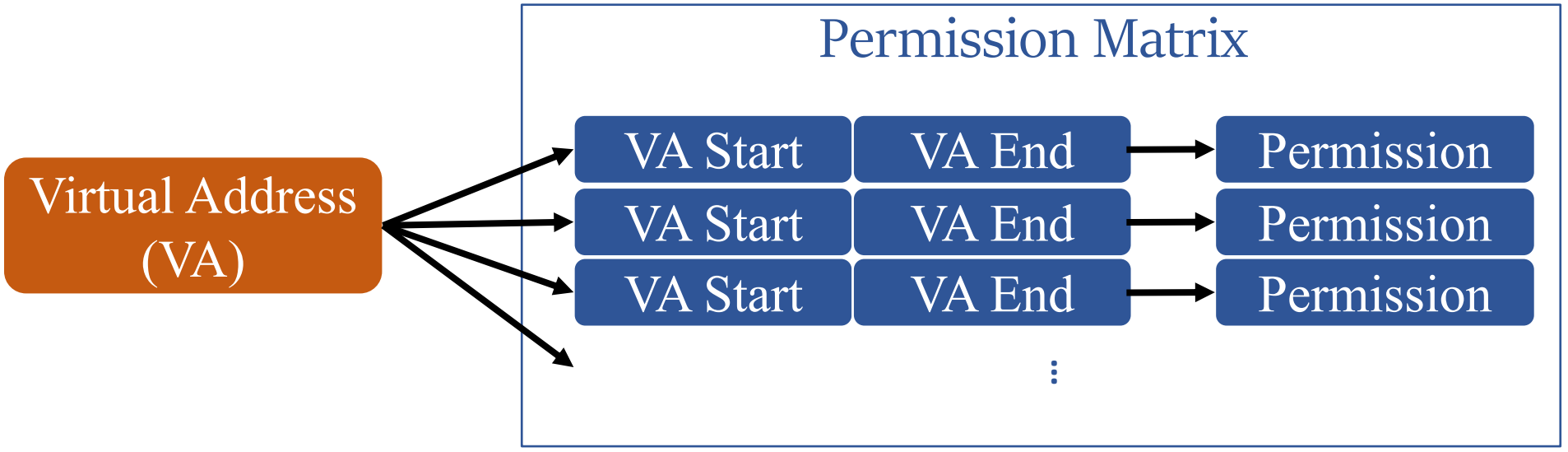
Permission Control Challenges



Permission Matrix

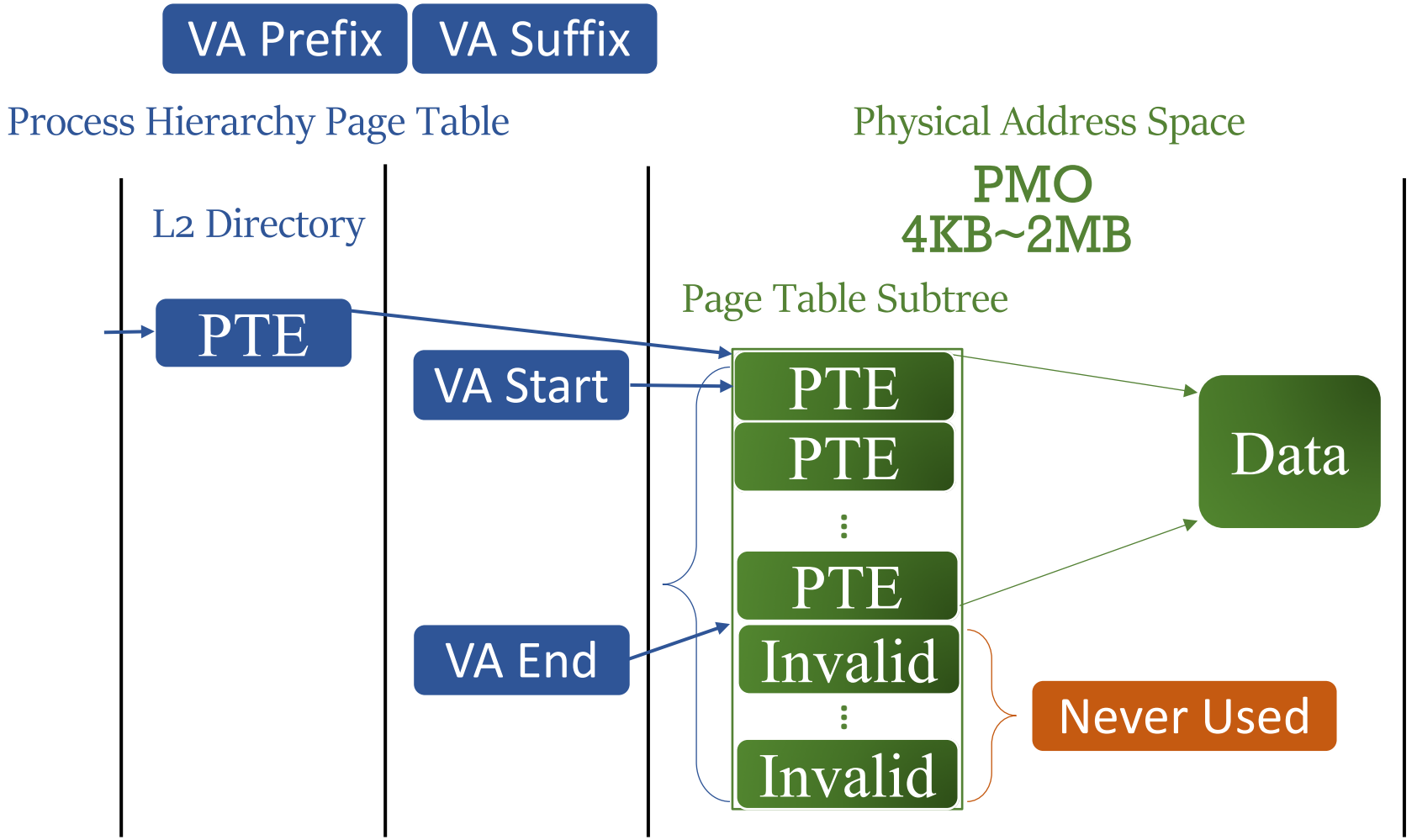


Permission Matrix Basic Design



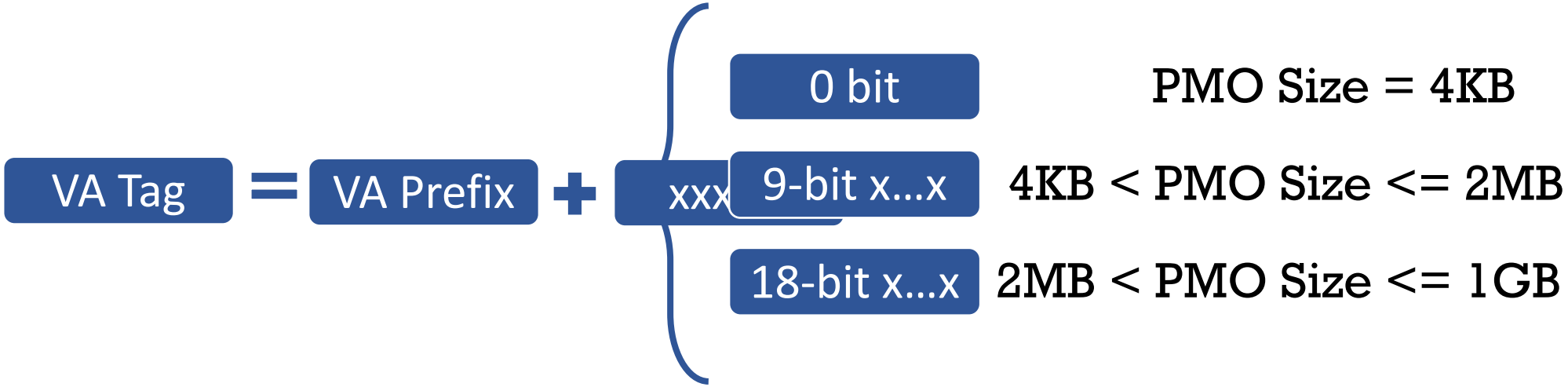
High Overhead in Critical Path!

Virtual Address Tag in Permission Matrix



Virtual Address Tag in Permission Matrix

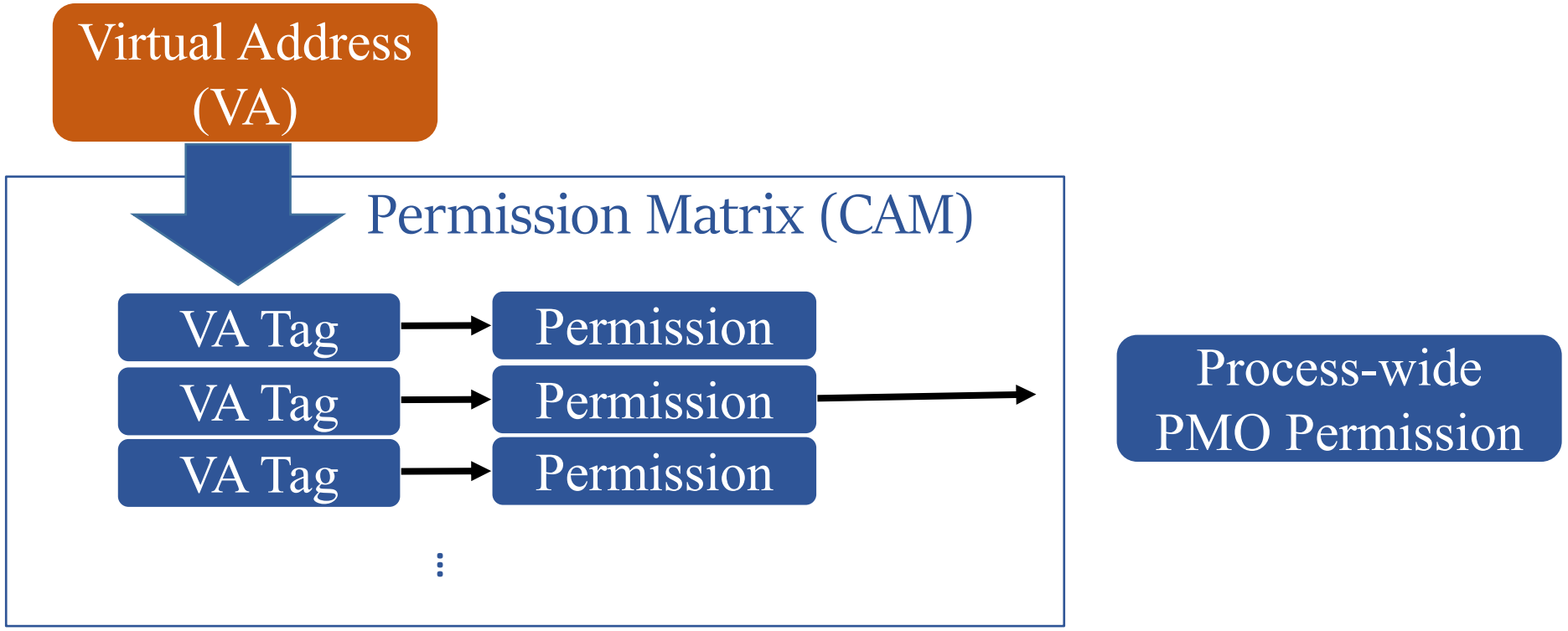
Virtual Address Tag in Permission Matrix



Content Addressable Memory (CAM) Locate Exact one Entry From VA!

x means all match in Content Addressable Memory

Permission Matrix Design



Evaluation Methodology

- Workloads:
  - WHISPER benchmarks
- Operating System Overhead:
  - Implement attach and detach library to replace mmap() and munmap()
- Architectural Overhead:
  - Intel Pin toolkit
  - Trace-Driven Simulation

Evaluation Metrics

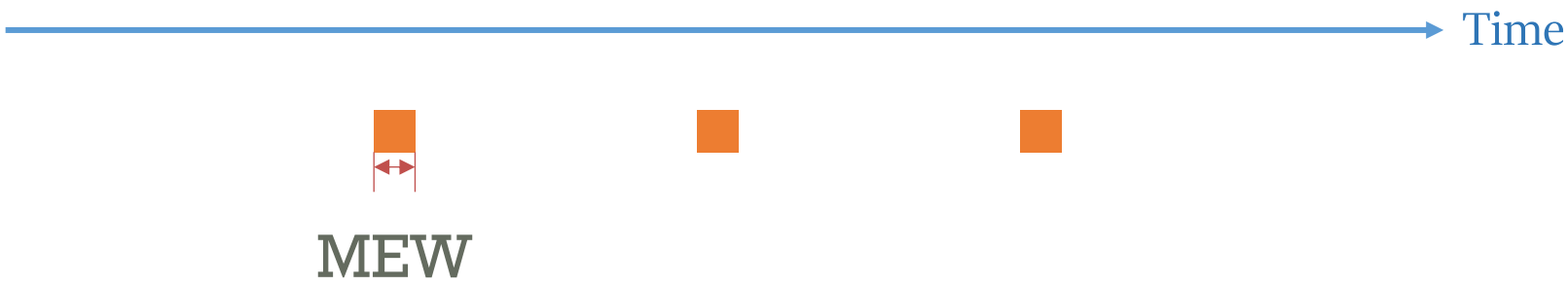
- Attached Memory Exposure Rate (AMER)

$$\text{AMER} = \frac{Reduced\ Time}{Original\ Time}$$

Evaluation Metrics

- Attached Memory Exposure Rate (AMER)
- Memory Exposure Window (MEW)
- PMO Space Layout Randomization Frequency (PSLR Frequency)

$$\text{AMER} = \frac{Reduced\ Time}{Original\ Time}$$

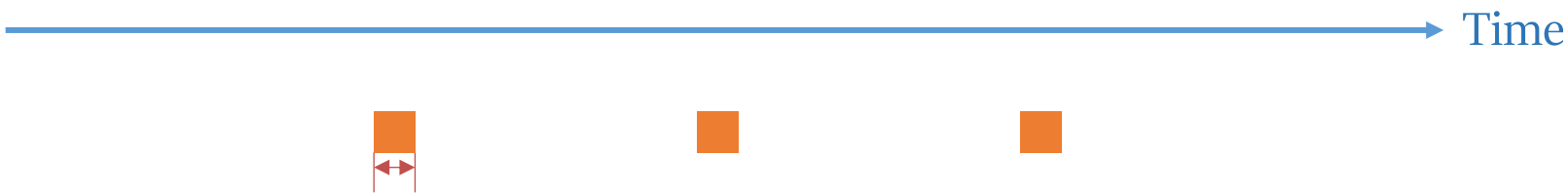


$$\text{PSLR Frequency} = \frac{Total\ Times}{Execution\ Time}$$

Evaluation Metrics

- Attached Memory Exposure Rate (AMER)
- Memory Exposure Window (MEW)
- PMO Space Layout Randomization Frequency (PSLR Frequency)

$$\text{AMER} = \frac{Reduced\ Time}{Original\ Time}$$



MEW

$$\text{PSLR Frequency} = \frac{3}{Execution\ Time}$$

Performance

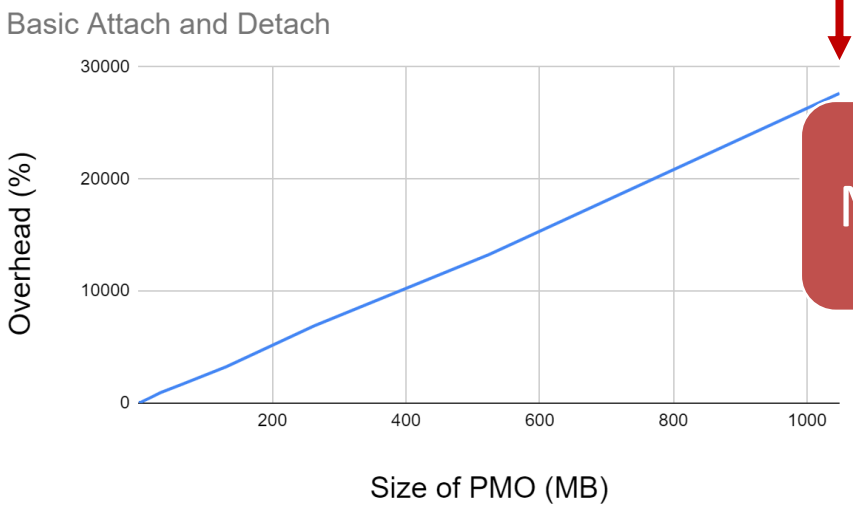
- About 10% overhead

30% AMER (70% Reduction)

80us Maximal MEW

PSLR per 41us

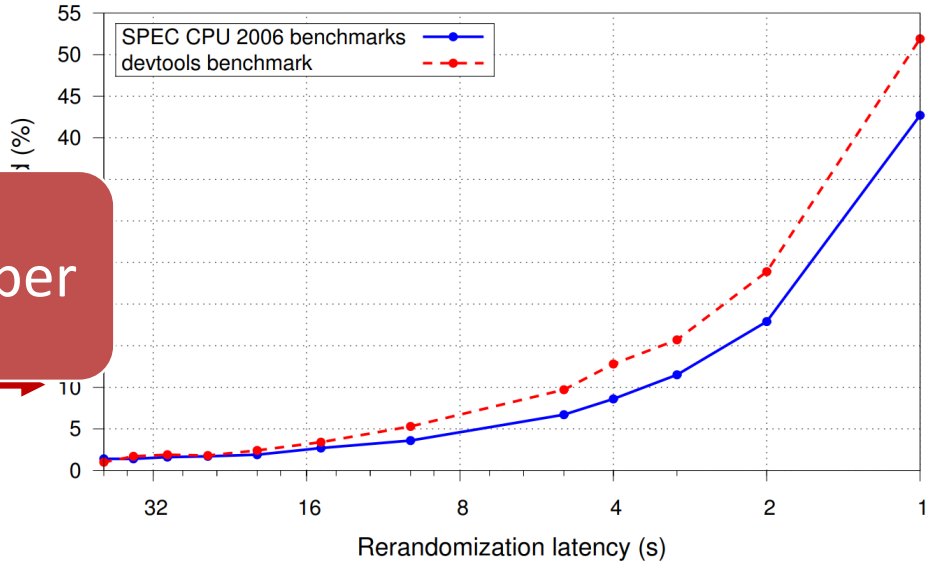
- To achieve above goals



2700x Speedup

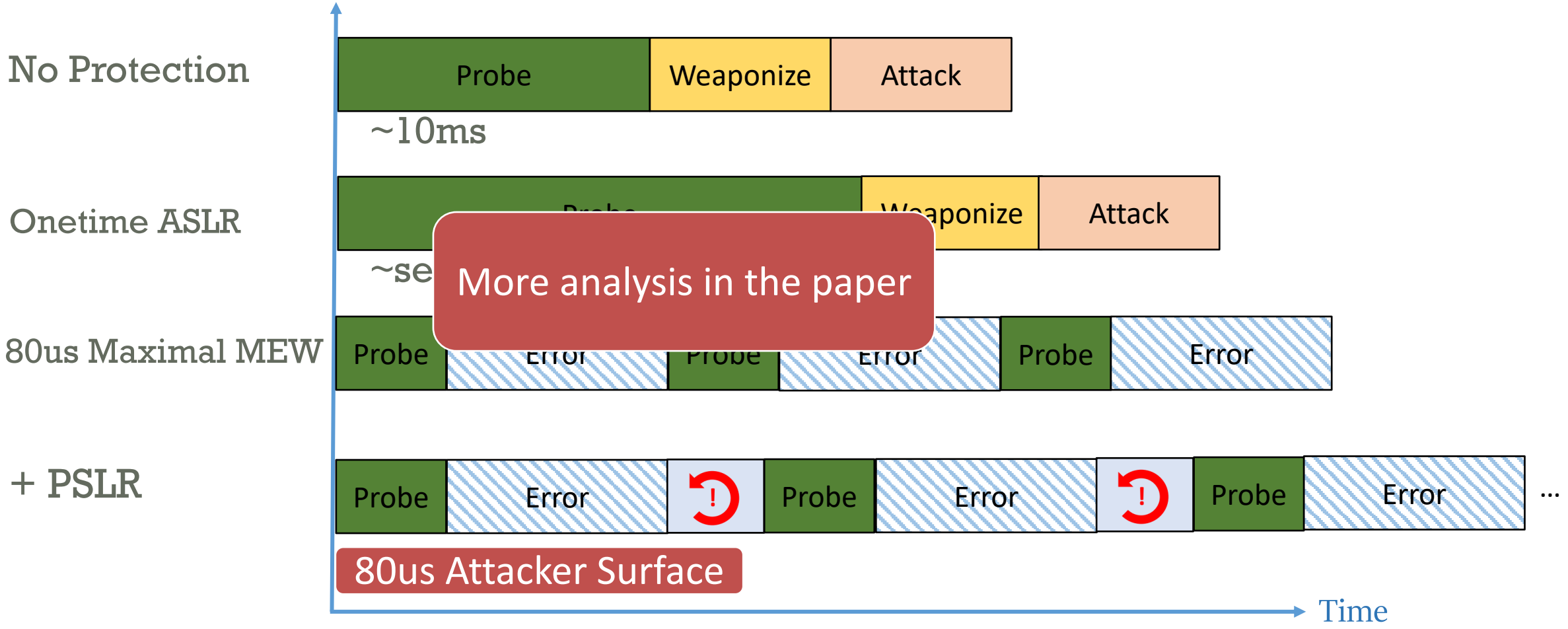
More results in the paper

Enhanced ASLR



10^5 per Second Frequency

Security of MERR



Conclusion

- New angle to improve security through **reduce memory exposure and randomization**
- Improved efficiency by **page table design** and **architectural support**
- **Achieved 70% memory exposure reduction** and **80us memory exposure window** with about **10%** overhead on real world applications
- **Provided An Order-of-magnitude** speedup compared to state-of-art runtime randomization