ASPLOS 2019 @ Providence, RI, USA

Heterogeneous Isolated Execution for Commodity GPUs

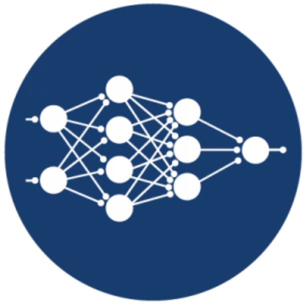
Insu Jang[1], Adrian Tang[2], Taehoon Kim[1],
Simha Sethumadhavan[2], and Jaehyuk Huh[1]

[1] KAIST, School of Computing
[2] Columbia University, Department of Computer Science

KAIST School of Computing
COLUMBIA | ENGINEERING
The Fu Foundation School of Engineering and Applied Science

Architecture Trend: Heterogeneous Computing

- Heterogeneous computing is emerging (**GPUs**, FPGAs, etc)



Machine Learning

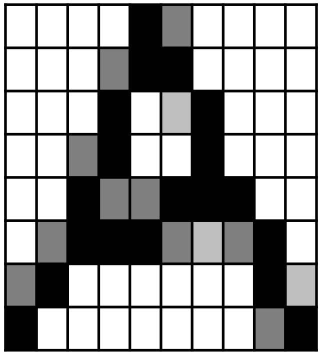
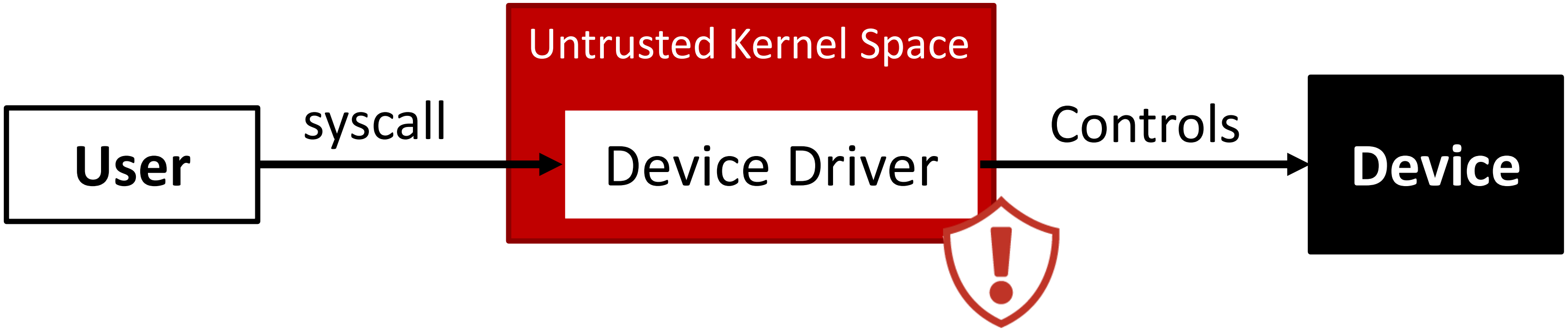


Image Processing

$$\int \frac{dy}{dx}$$

Complex Calculations

- **Problem: lack of trusted execution environment in devices**



Untrusted Kernel Space

User

syscall

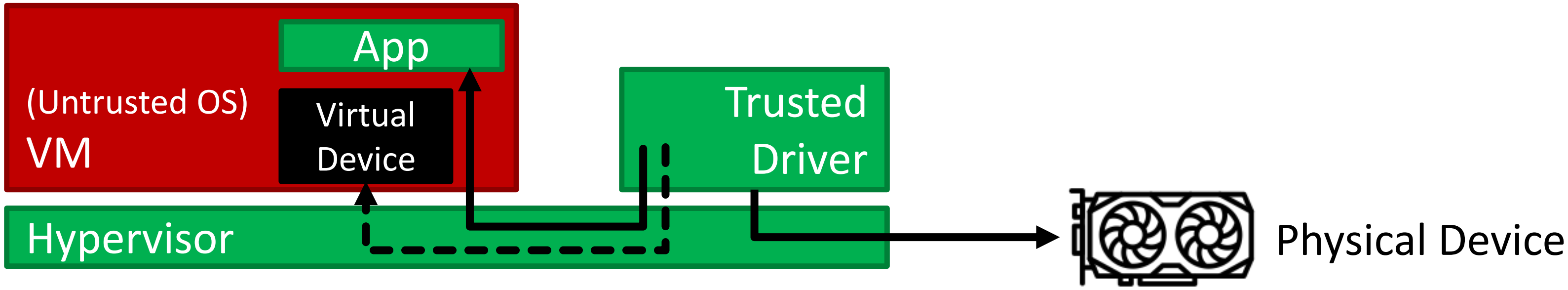
Device Driver

Controls

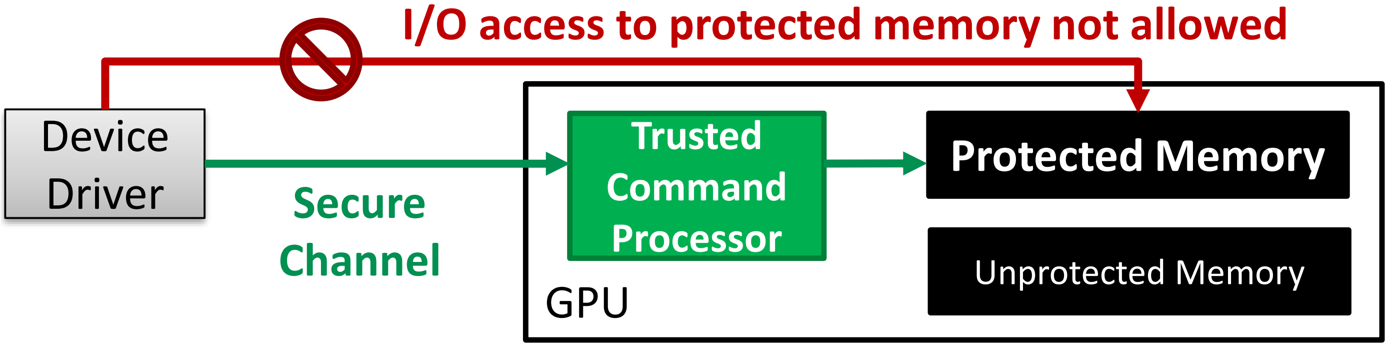
Device

How to Provide TEE to Devices?

- **Problem: lack of trusted execution environment in devices**
- Existing works regarding TEE for peripheral devices
  - **SGXIO** **[Weiser, CODASPY'17]**: use **a trusted hypervisor**

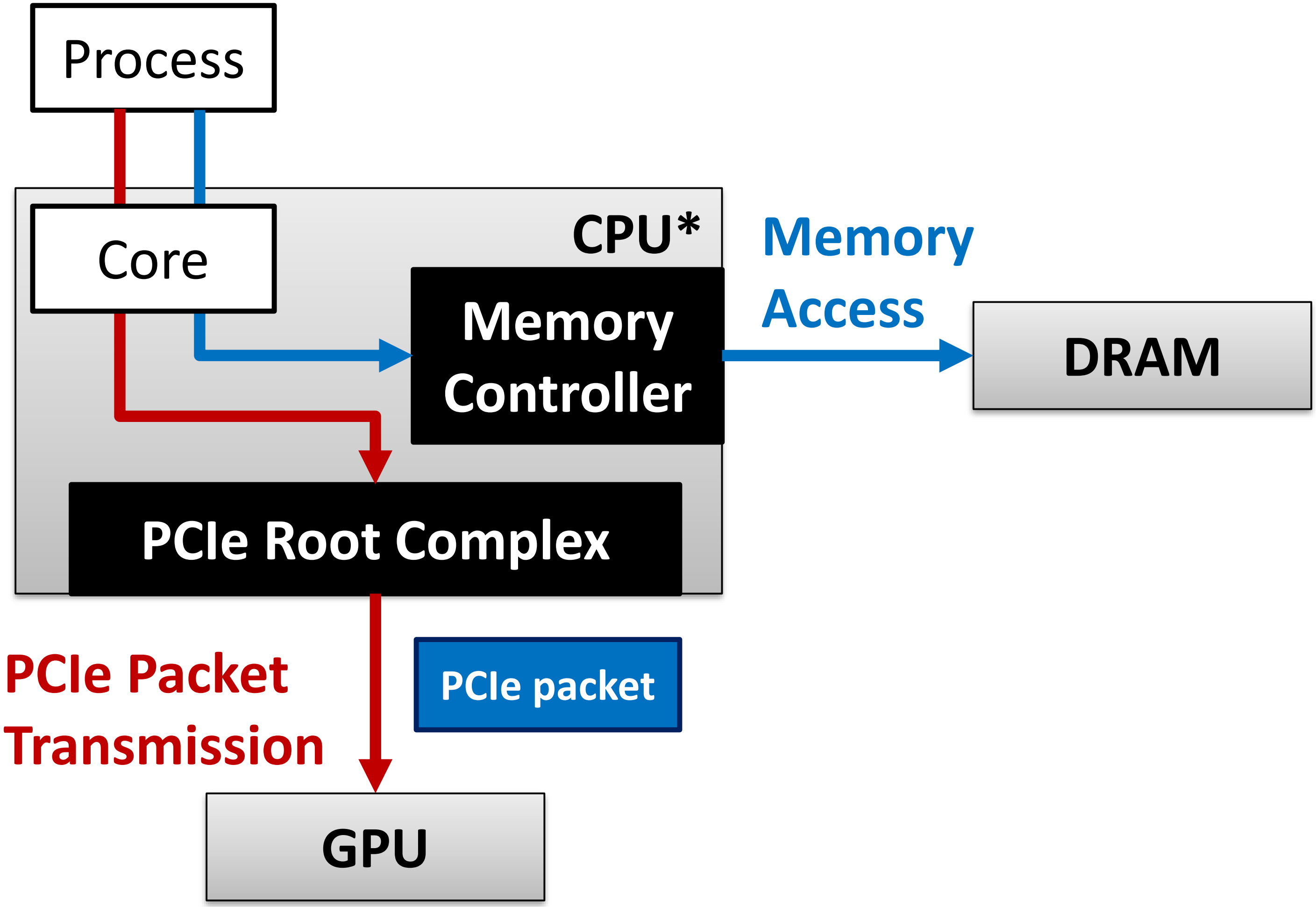


  - **Graviton** **[Volos, OSDI'18]**: use **a modified GPU** with a root of trust



Our Apporach: Securing I/O Path

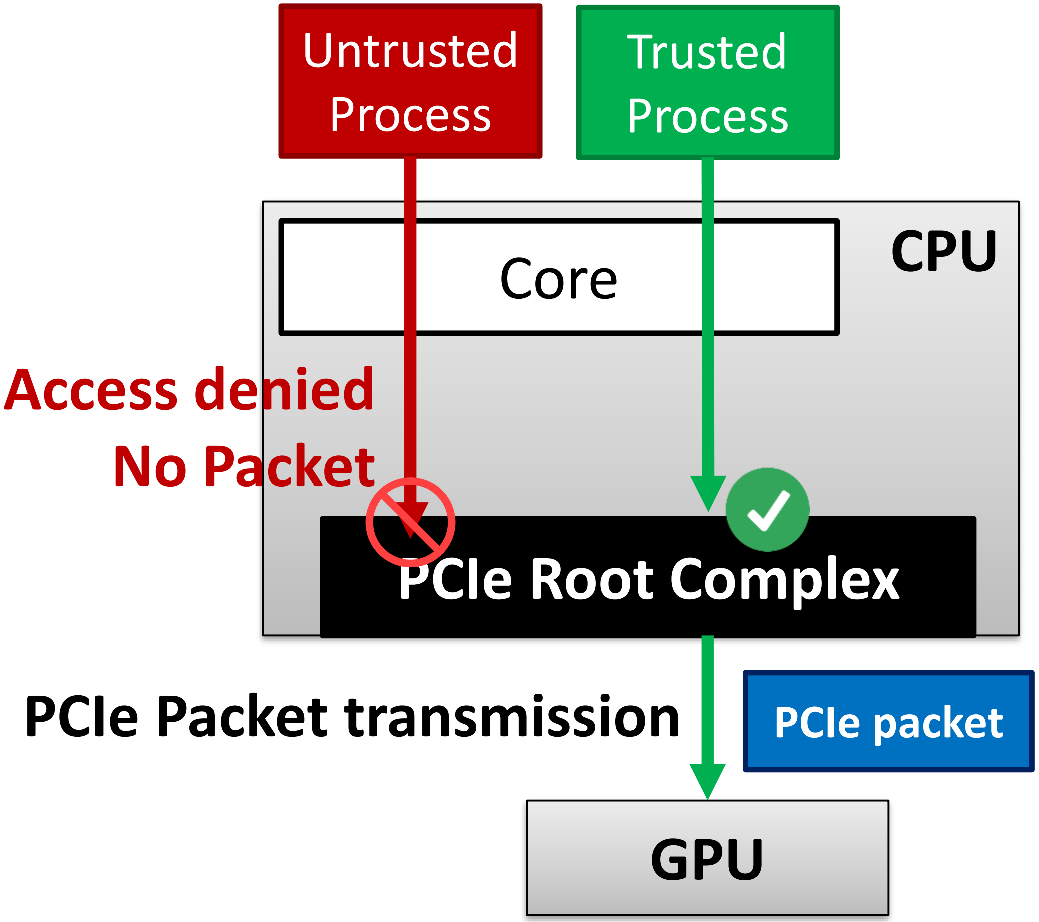
- All device I/O accesses from software are **handled by CPU**



** x86 architecture based*

Our Apporach: Securing I/O Path

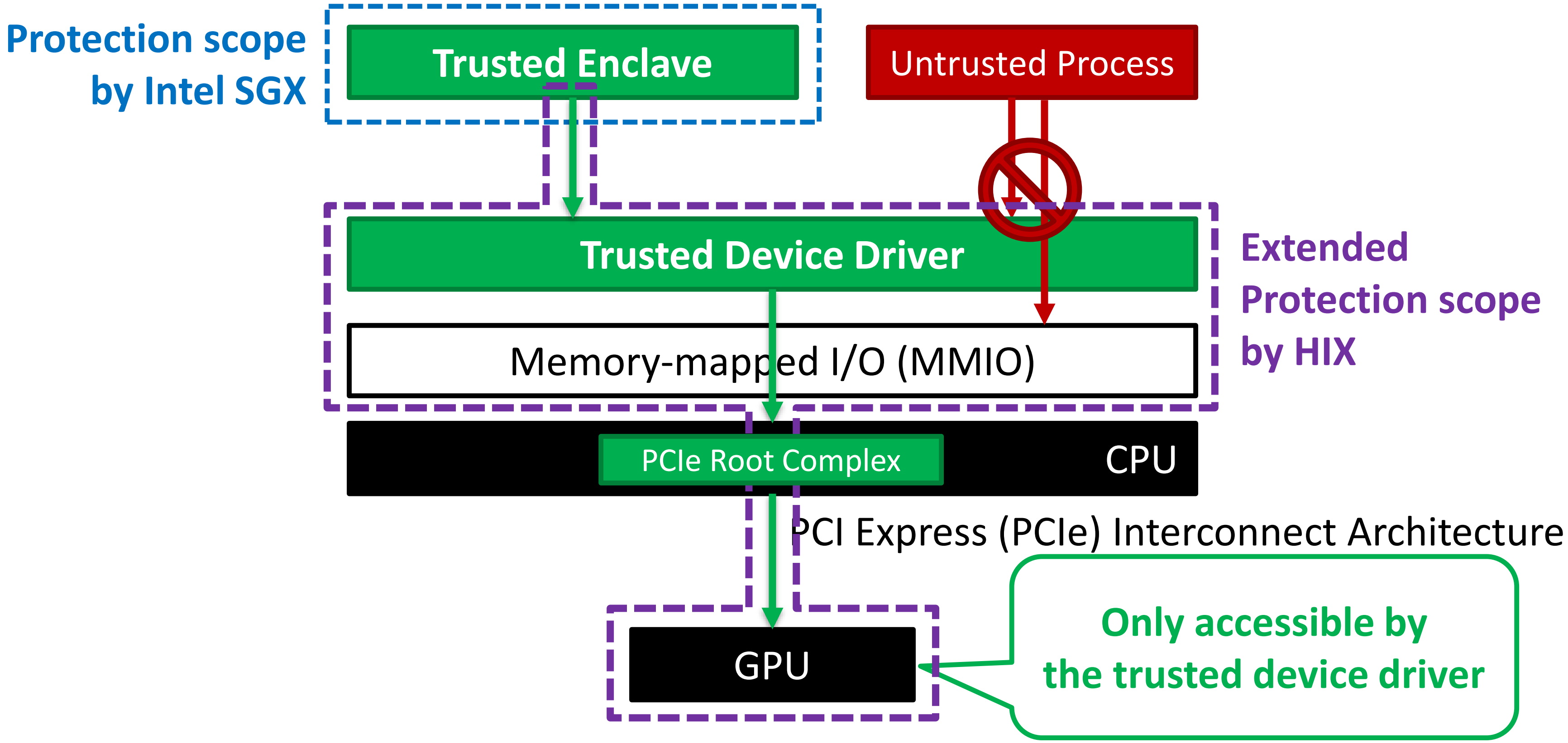
- All device I/O accesses from software are **handled by CPU**



Idea: Prevent I/O from Attackers by Securing I/O Path!

HIX: Heterogeneous Isolated Execution

- Implementation based on Intel SGX (basic TEE necessary)
- **Extend TEE to I/O path (from SGX enclave to the device)**


Protection scope by Intel SGX
Trusted Enclave
Untrusted Process
Trusted Device Driver
Extended Protection scope by HIX
Memory-mapped I/O (MMIO)
PCIe Root Complex
CPU
PCI Express (PCIe) Interconnect Architecture
GPU
Only accessible by the trusted device driver

Contributions and Threat Model

- Provide **confidentiality and integrity** to user data in GPU
- **No GPU modifications** are required
  - Provide GPU TEE by securing I/O path
  - No protection against physical attacks; software based attacks prevented
- **Threat Model**
  - Attackers have all privileged permission on software level
  - Not consider physical attacks on any hardware
  - **Protect the system from privileged software attacks**

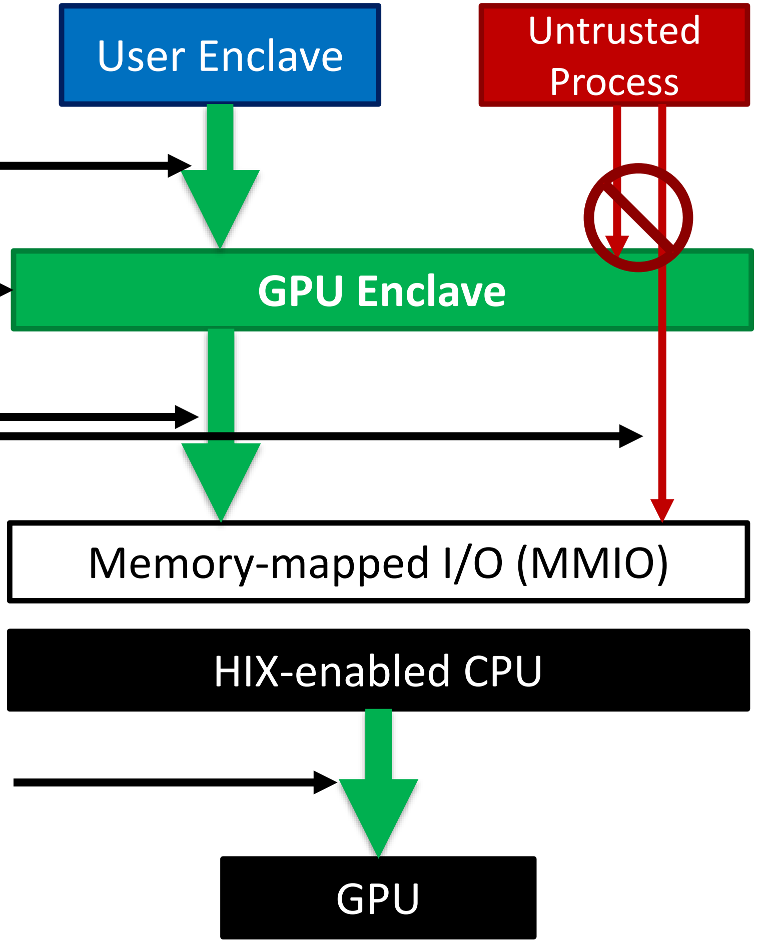
HIX Architecture

- Trusted GPU Device Driver: GPU Enclave
- MMIO Protection
- Inter-Enclave Communication

HIX: Architecture Overview

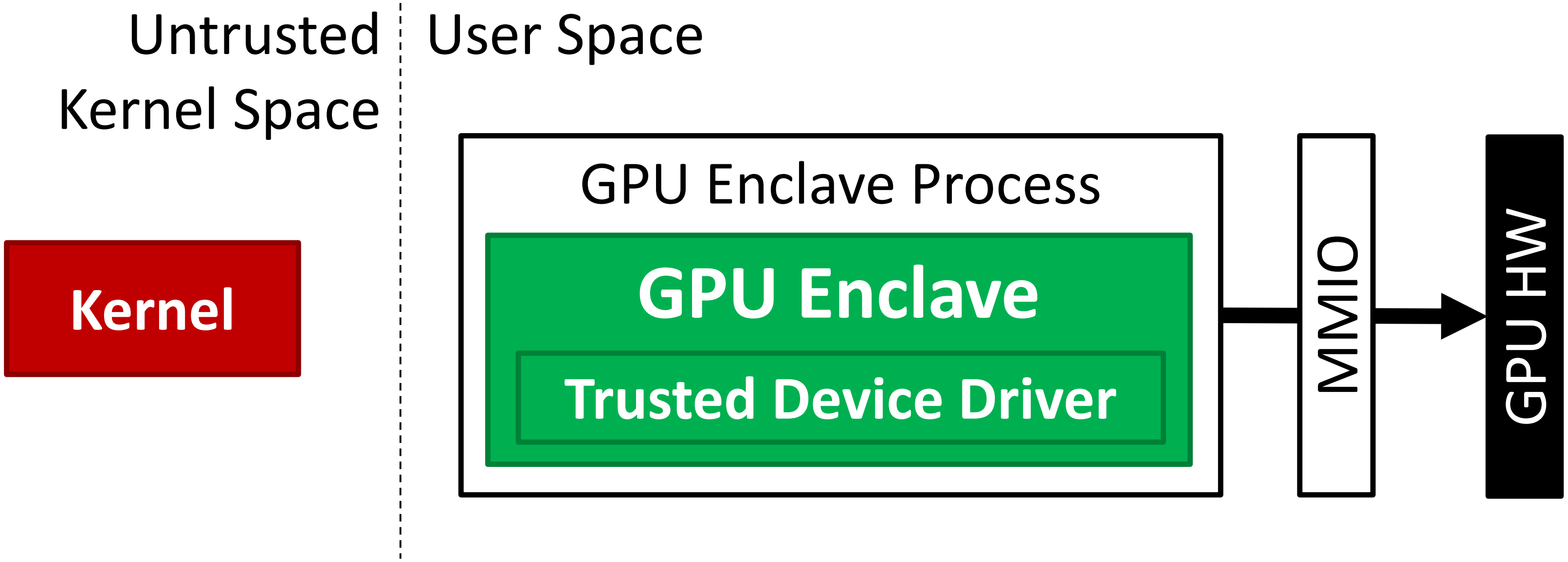
🟩➡ : Three communication paths to be protected

- Inter-enclave communication
- Trusted GPU device driver
- MMIO access validation
- Guaranteed PCIe packet routing



GPU Enclave: Trusted Device Driver

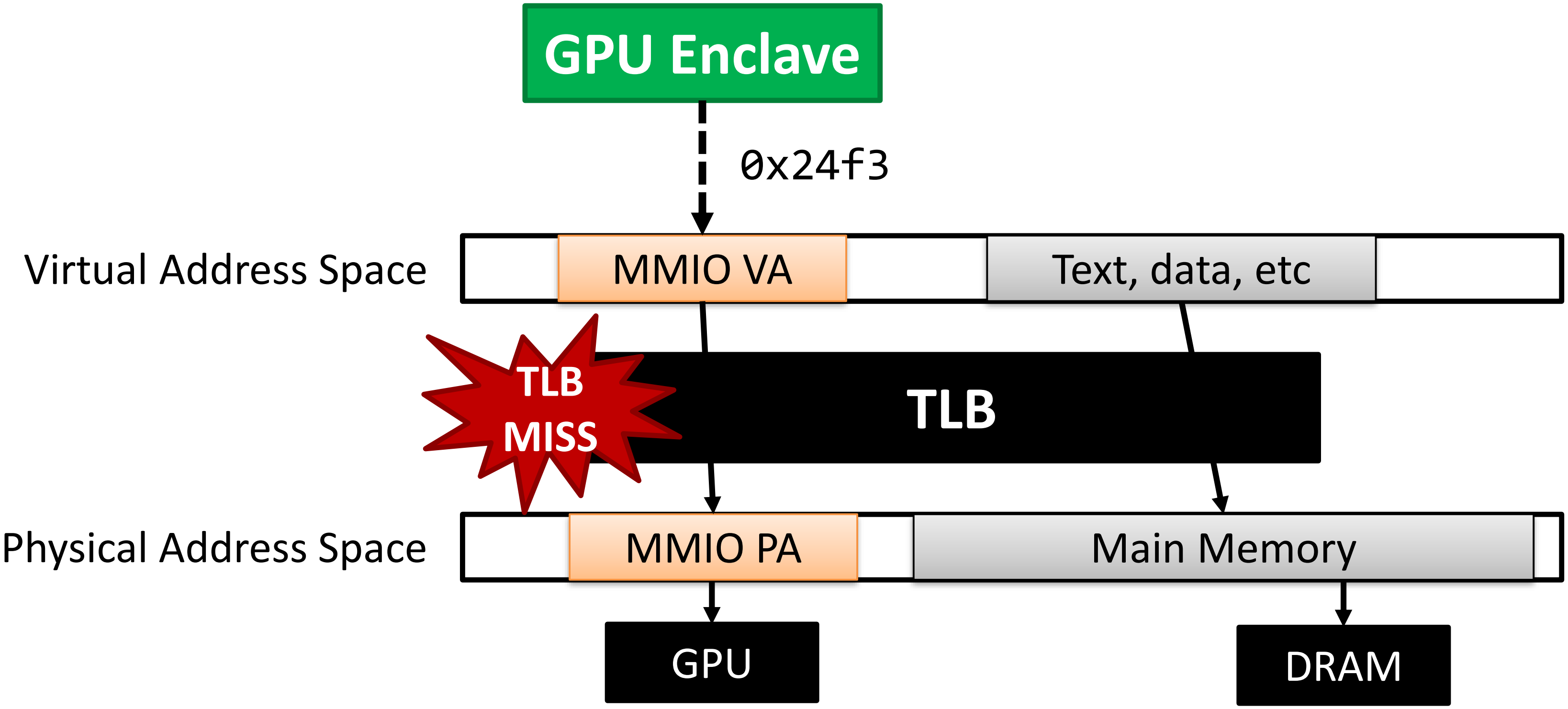
- Move device driver from untrusted kernel space to trusted enclave
- Extended SGX enclave that owns and controls GPU in TEE



- **Exclusively** access to GPU in the system through MMIO

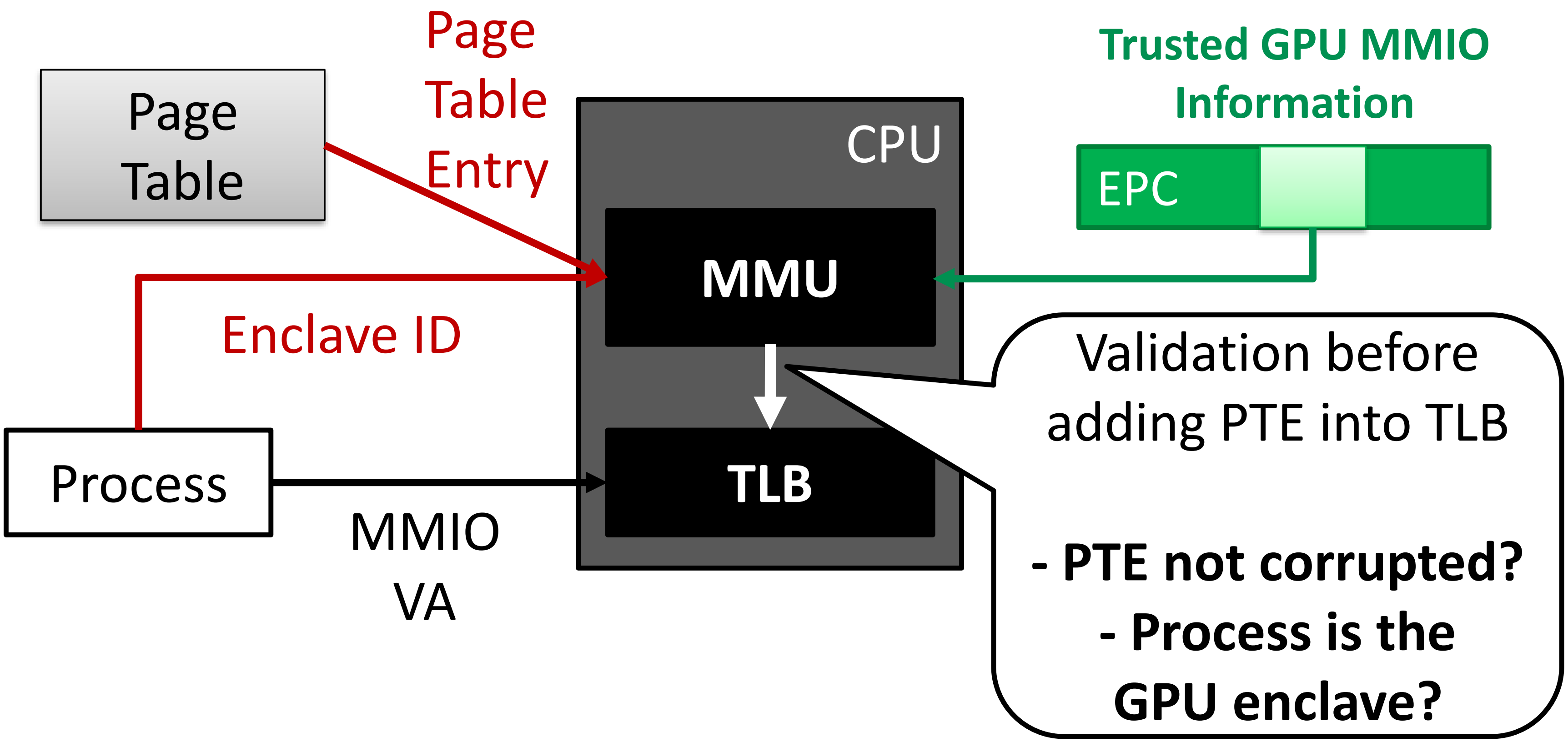
MMIO Access Validation

- **Exclusively** access to GPU in the system through MMIO : **How?**
- Extend SGX EPC access validation mechanism for MMIO
  - Validate **address translation information** during **TLB misses**



MMIO Access Validation

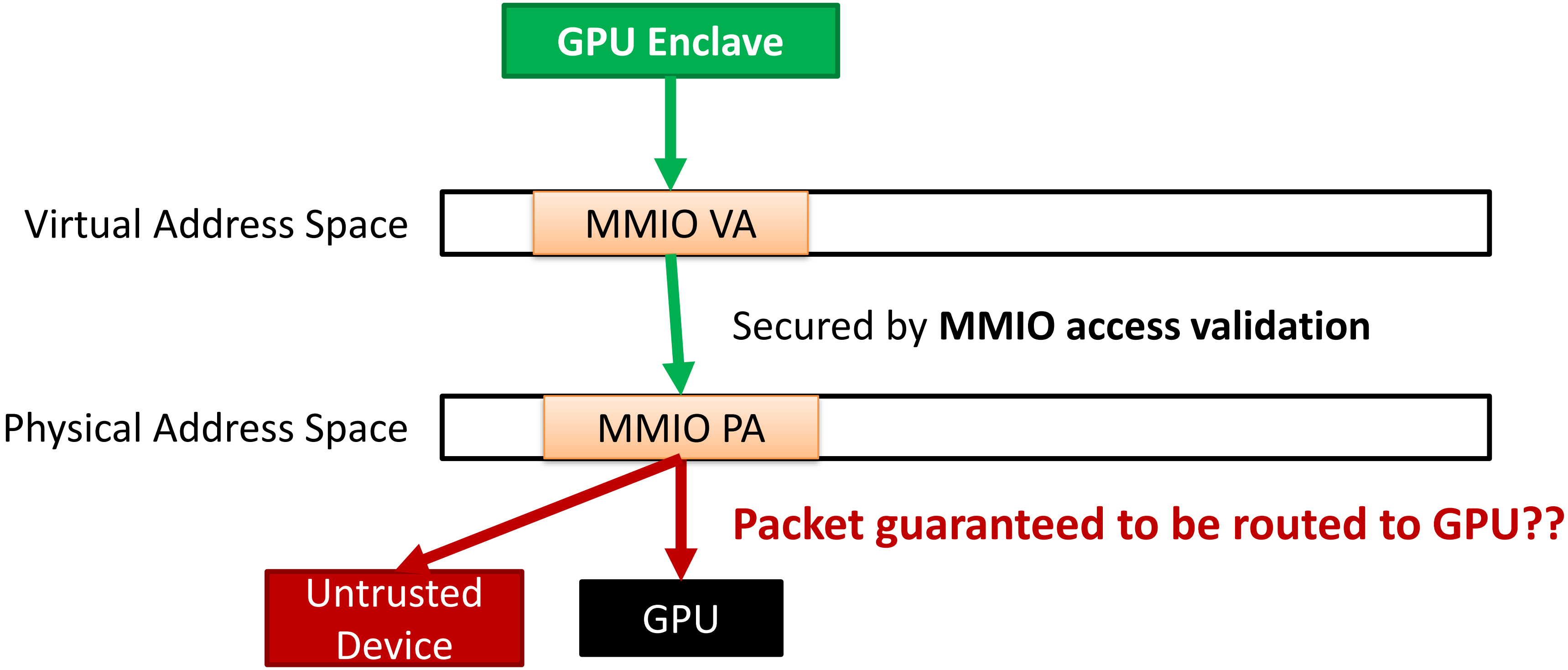
- **Exclusively** access to GPU in the system through MMIO : **How?**
- Extend SGX EPC access validation mechanism for MMIO
  - Validate **address translation information** during **TLB misses**



Validation before adding PTE into TLB

- PTE not corrupted?
- Process is the GPU enclave?

PCIe Packet Routing



GPU Enclave

Virtual Address Space

MMIO VA

Secured by **MMIO access validation**

Physical Address Space

MMIO PA

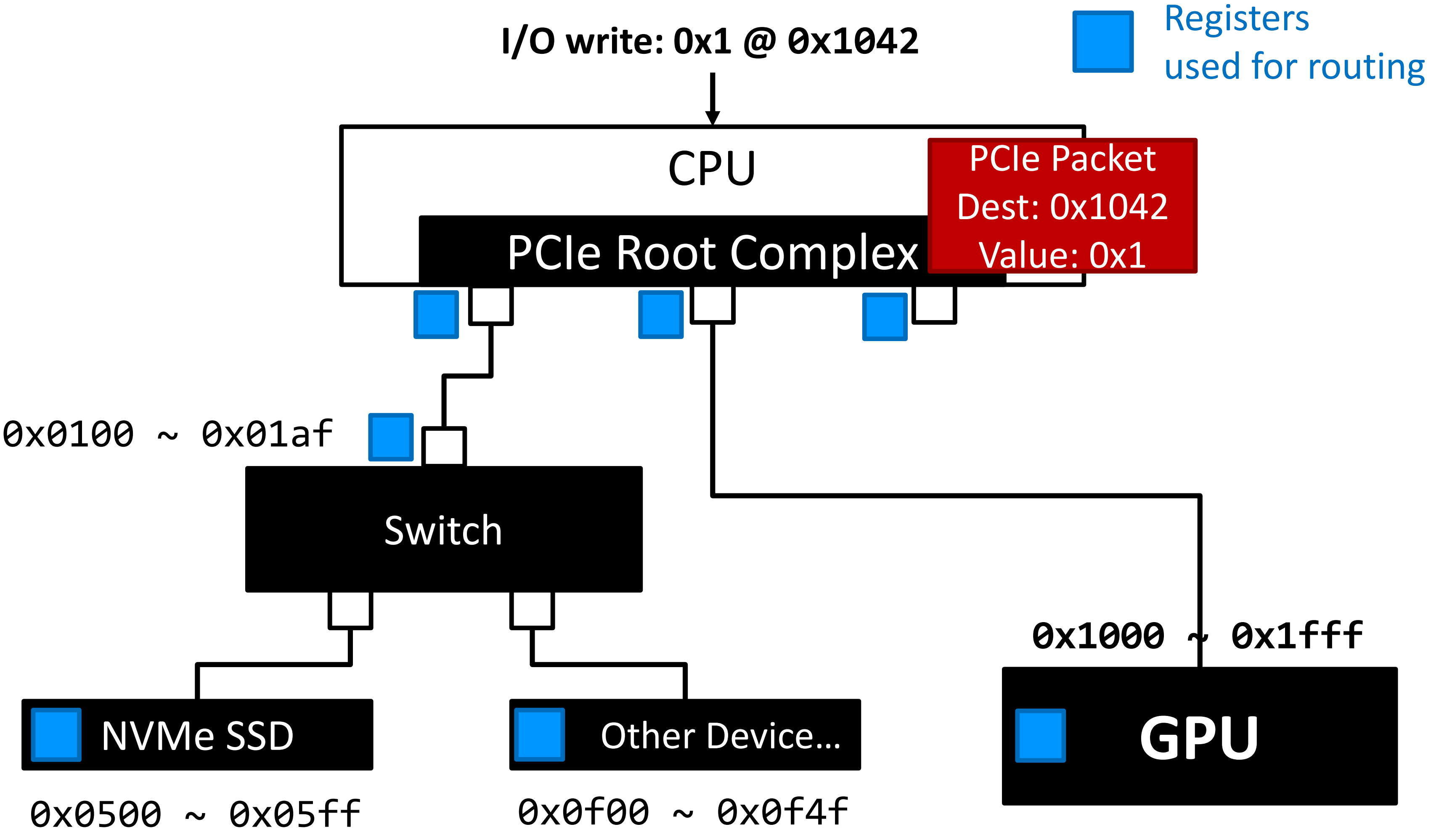
Packet guaranteed to be routed to GPU??

Untrusted Device

GPU

PCIe Packet Routing: Introduction

- Use PCIe hardware registers for packet routing (e.g. BARs*)



I/O write: 0x1 @ 0x1042

🟦 Registers used for routing

CPU

PCIe Root Complex

PCIe Packet
Dest: 0x1042
Value: 0x1

0x0100 ~ 0x01af

Switch

0x1000 ~ 0x1fff

NVMe SSD

Other Device…

GPU

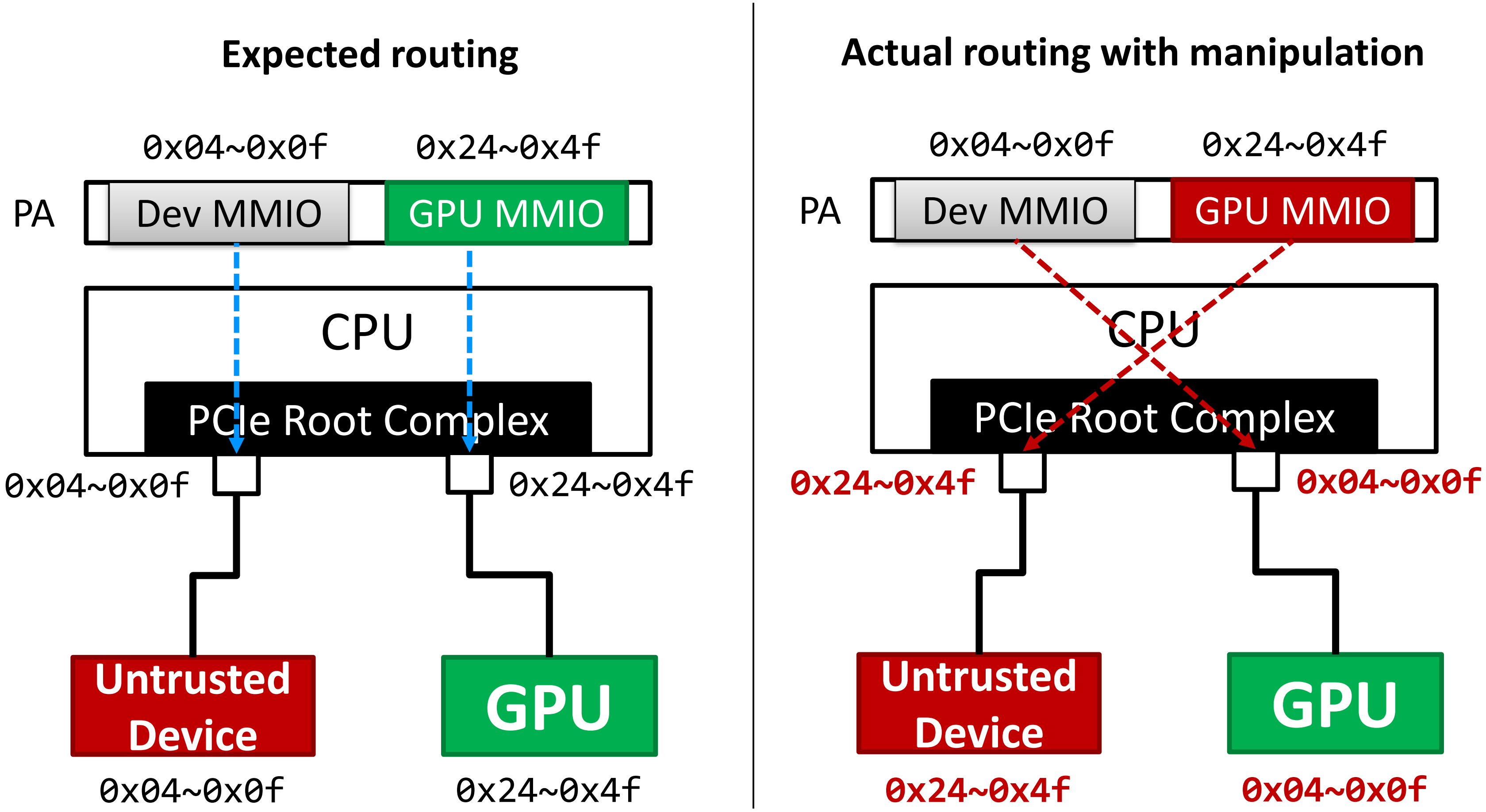
0x0500 ~ 0x05ff

0x0f00 ~ 0x0f4f

** BAR: Base Address Register*

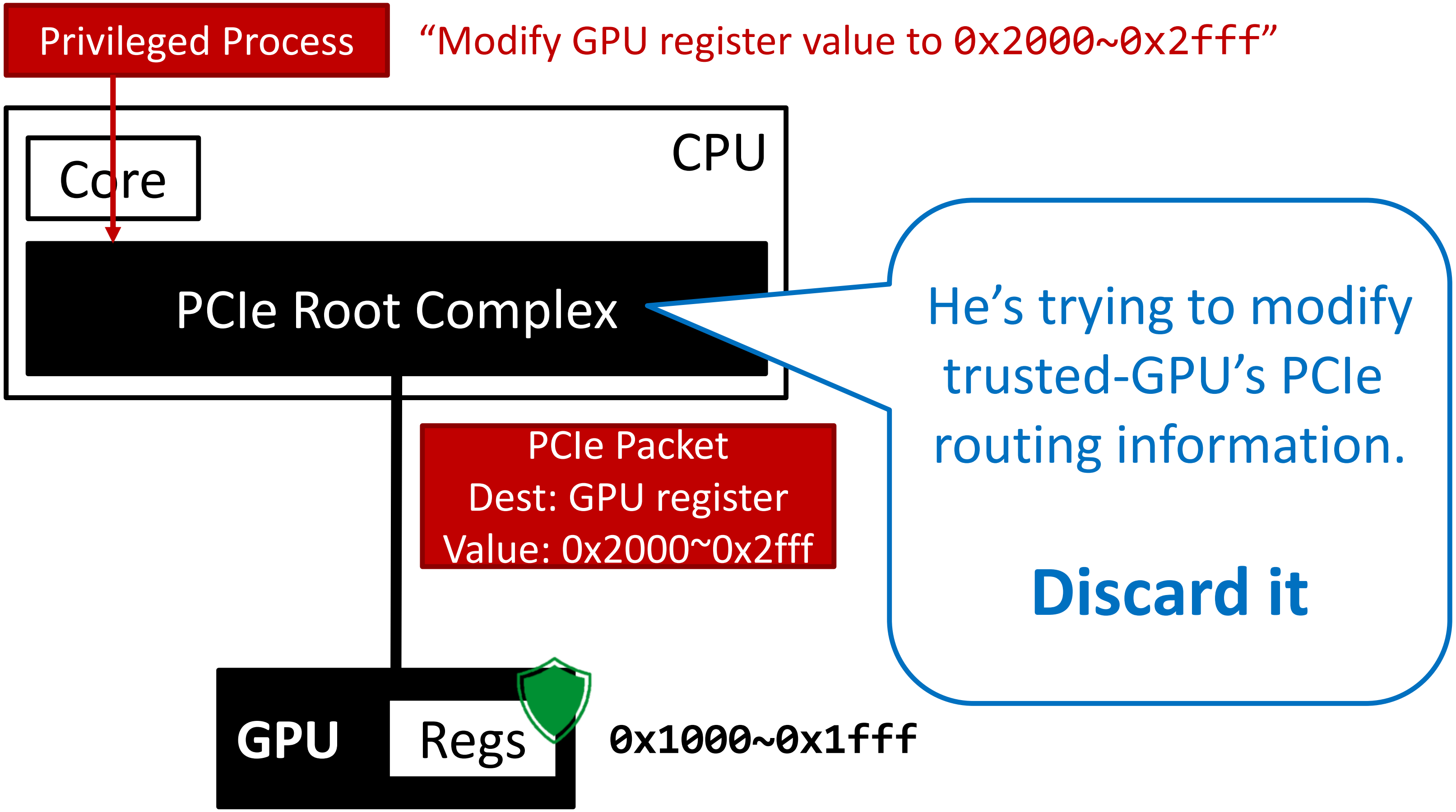
PCIe Packet Routing: Challenge

- PCIe hardware registers **can be manipulated** by software



MMIO Lockdown

- PCIe hardware registers **can be manipulated** by software
- **Solution:** freeze MMIO routing information (MMIO lockdown)

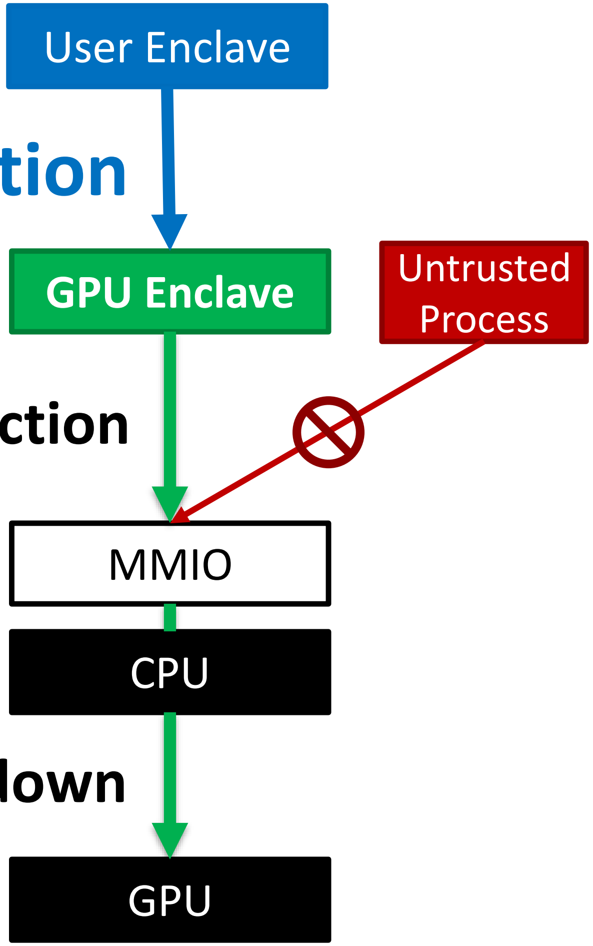


Architecture Review

Next: Inter-Enclave Communication

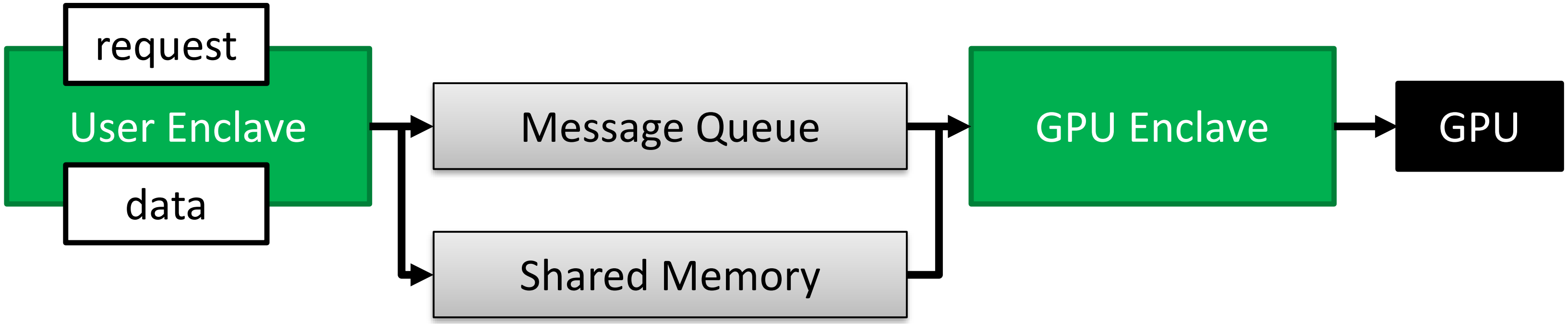
Protected by MMIO Access Restriction

Protected by MMIO Lockdown



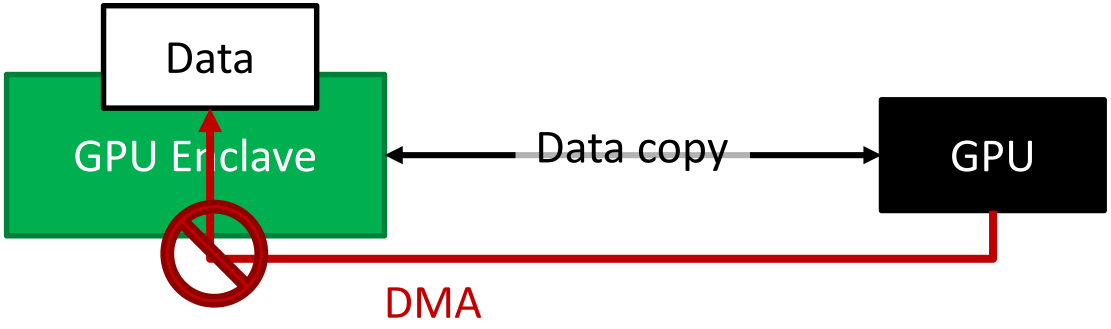
Inter-Enclave Communication

- Inter-process communication: message queue & shared memory
- Confidentiality & integrity provided by **authenticated encryption**



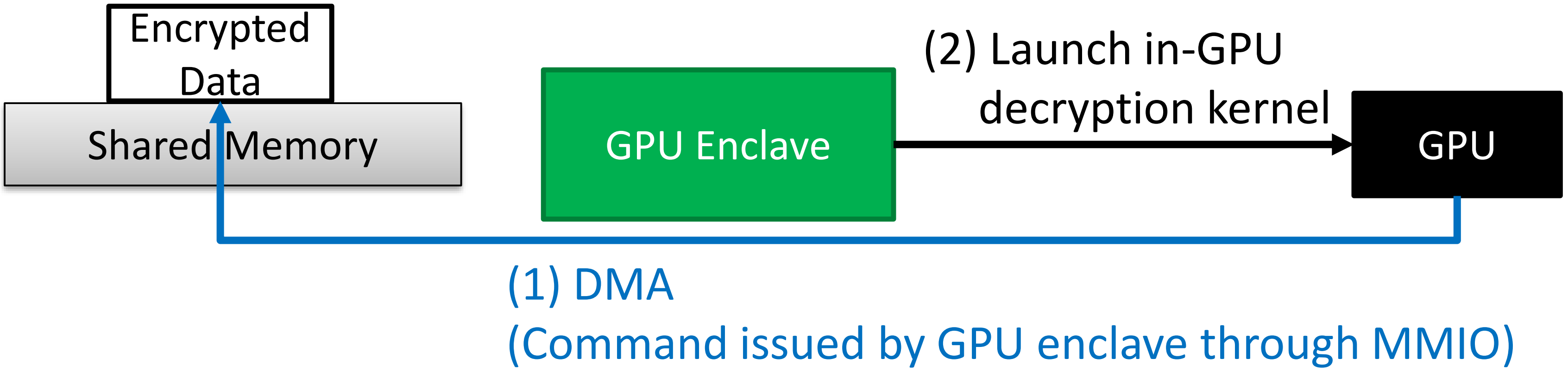
Communication Challenge: DMA

- **Challenge**
  - – DMA from device to enclaves not allowed by SGX
  - – Data copy can only be done through (slow) MMIO



Trusted DMA Support

- GPU DMAs encrypted data from shared memory to GPU
- GPU enclave launches in-GPU decryption kernel


Encrypted Data
Shared Memory
GPU Enclave
(2) Launch in-GPU decryption kernel
GPU
(1) DMA
(Command issued by GPU enclave through MMIO)

Evaluation

Evaluation

- Prototype Implementation
  - Hardware changes are emulated in a KVM/QEMU virtual machine
  - GPU enclave implementation is based on Gdev [Kato, ATC'12]

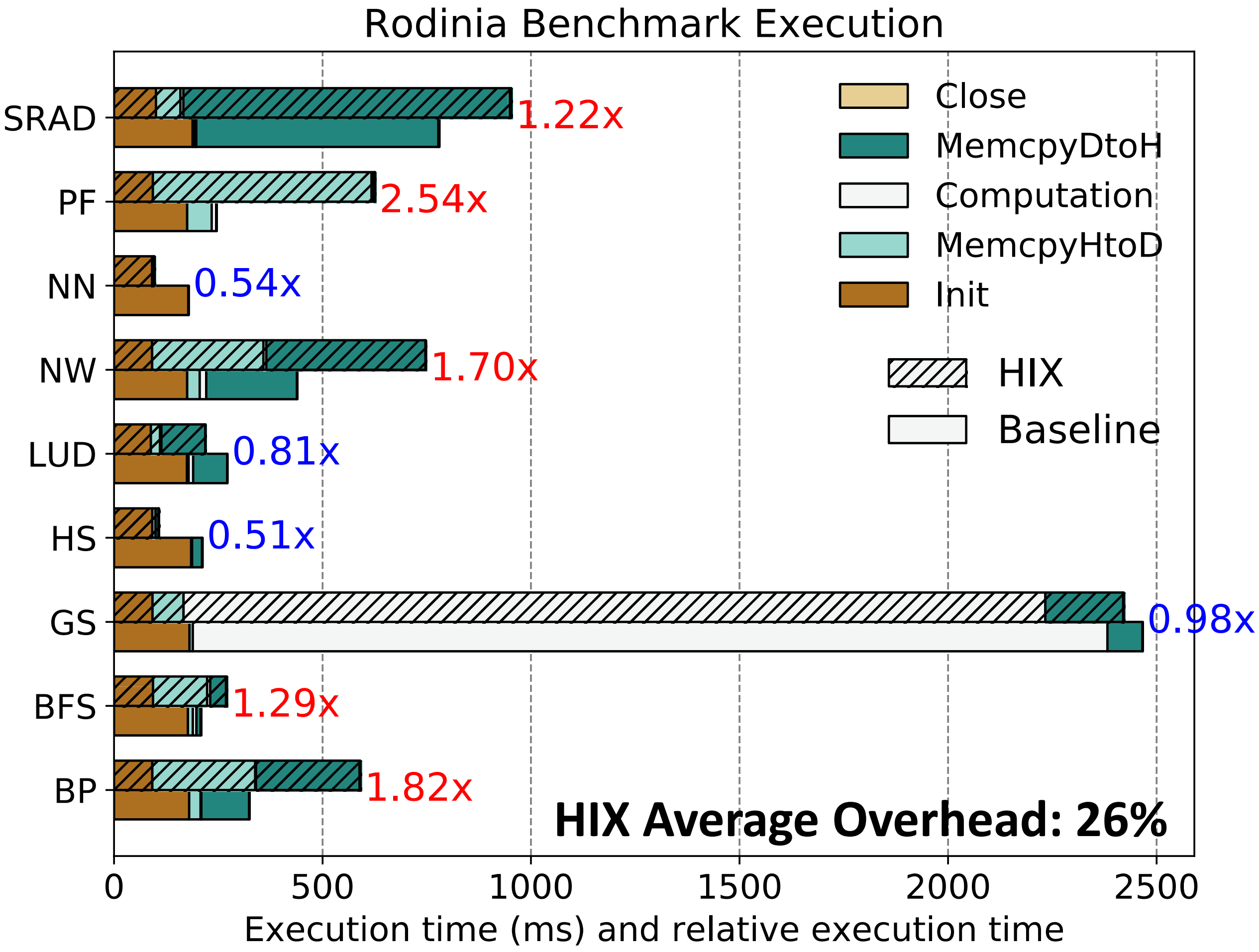
- Performance analysis: Rodinia GPU microbenchmark
  - Measure overheads due to cryptography, etc.
  - Baseline: unmodified Gdev NVIDIA GPU driver

	Baseline	HIX
Trusted Execution	No	Yes
Encryption	N/A	AES-OCB [Rogaway '14]
GPU	NVIDIA Geforce GTX 580*	

** Newer devices are not supported by Gdev*

Performance Result: Rodinia

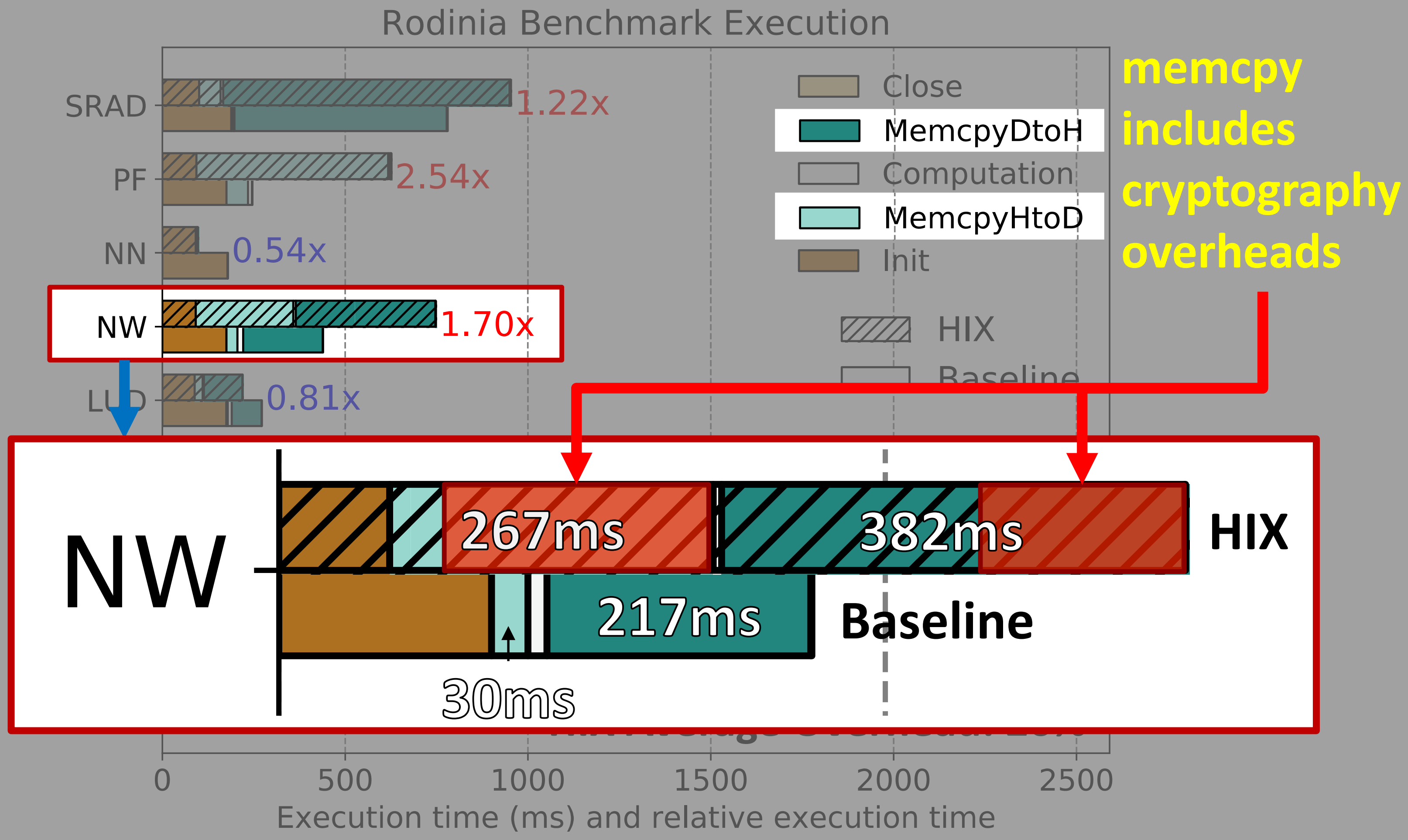
App Name	SRAD	PF	NN	NW	LUD	HS	GS	BFS	BP
Memcpy	48.4MB	256.0MB	501.2KB	192.1MB	32.0MB	12.0MB	64.0MB	46.9MB	159.8MB



HIX Average Overhead: 26%

Performance Result: Rodinia

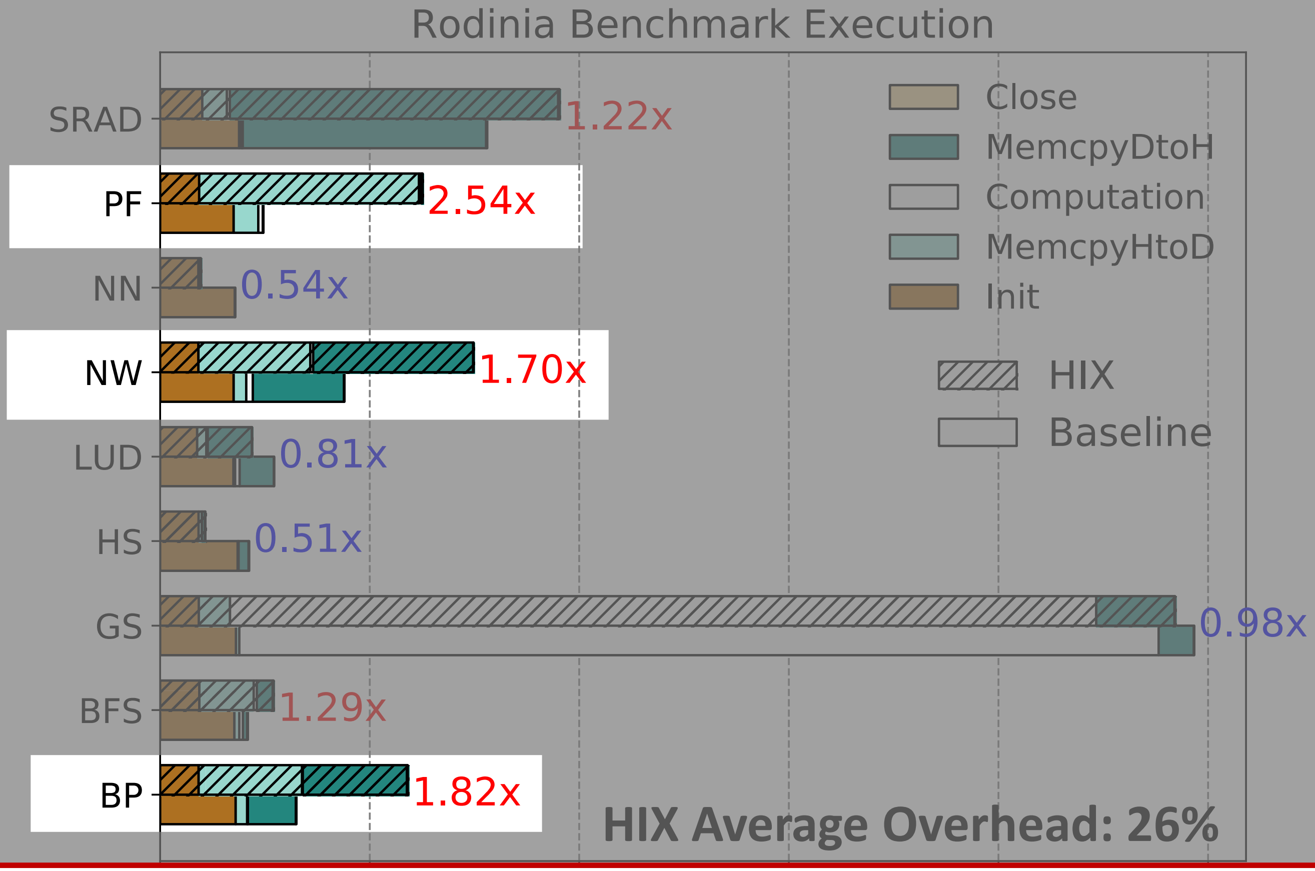
App Name	SRAD	PF	NN	NW	LUD	HS	GS	BFS	BP
Memcpy	48.4MB	256.0MB	501.2KB	192.1MB	32.0MB	12.0MB	64.0MB	46.9MB	159.8MB



memcpy includes cryptography overheads

Performance Result: Rodinia

App Name	SRAD	PF	NN	NW	LUD	HS	GS	BFS	BP
Memcpy	48.4MB	256.0MB	501.2KB	192.1MB	32.0MB	12.0MB	64.0MB	46.9MB	159.8MB

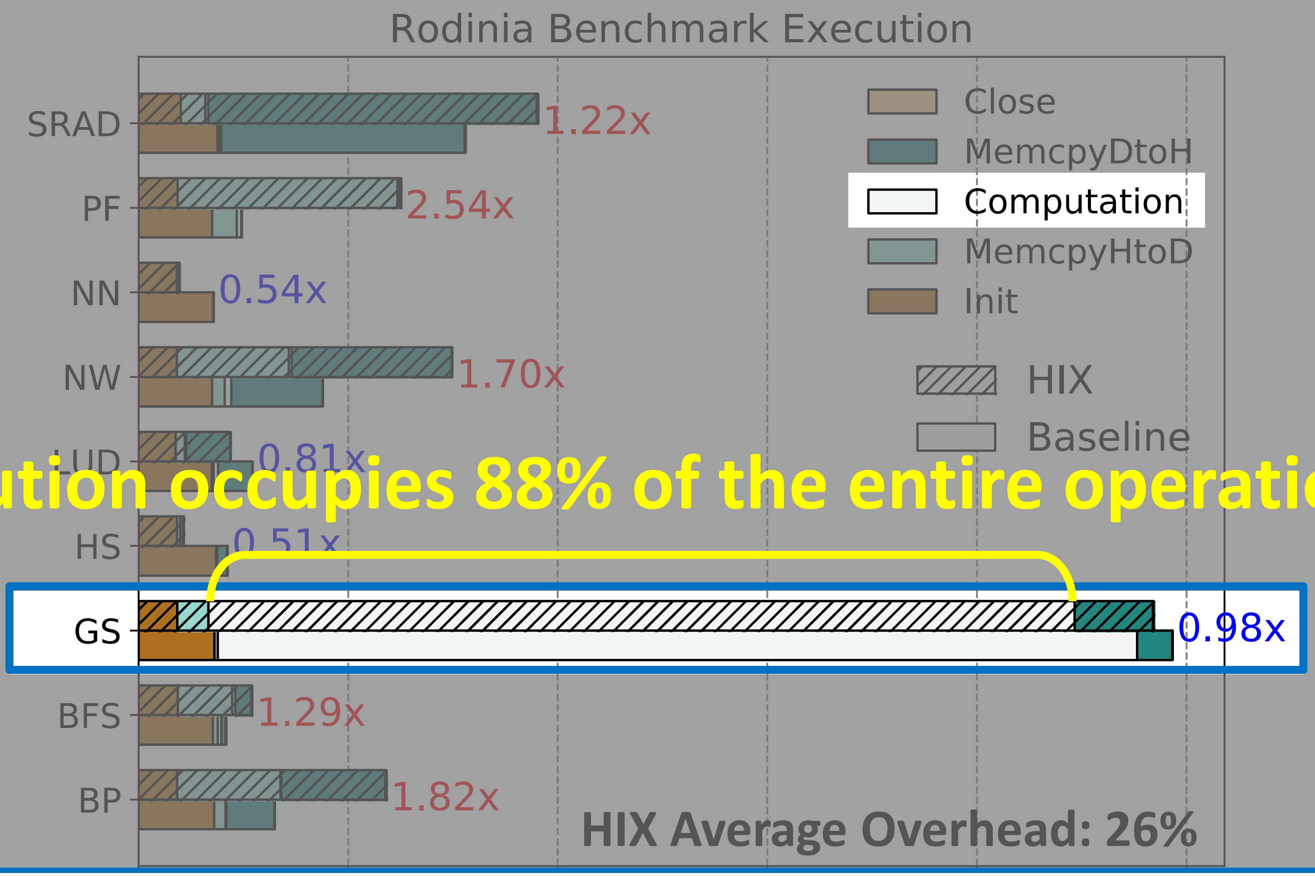


HIX Average Overhead: 26%

Large Amount of Data → High Cryptography Overheads

Performance Result: Rodinia

App Name	SRAD	PF	NN	NW	LUD	HS	GS	BFS	BP
Memcpy	48.4MB	256.0MB	501.2KB	192.1MB	32.0MB	12.0MB	64.0MB	46.9MB	159.8MB



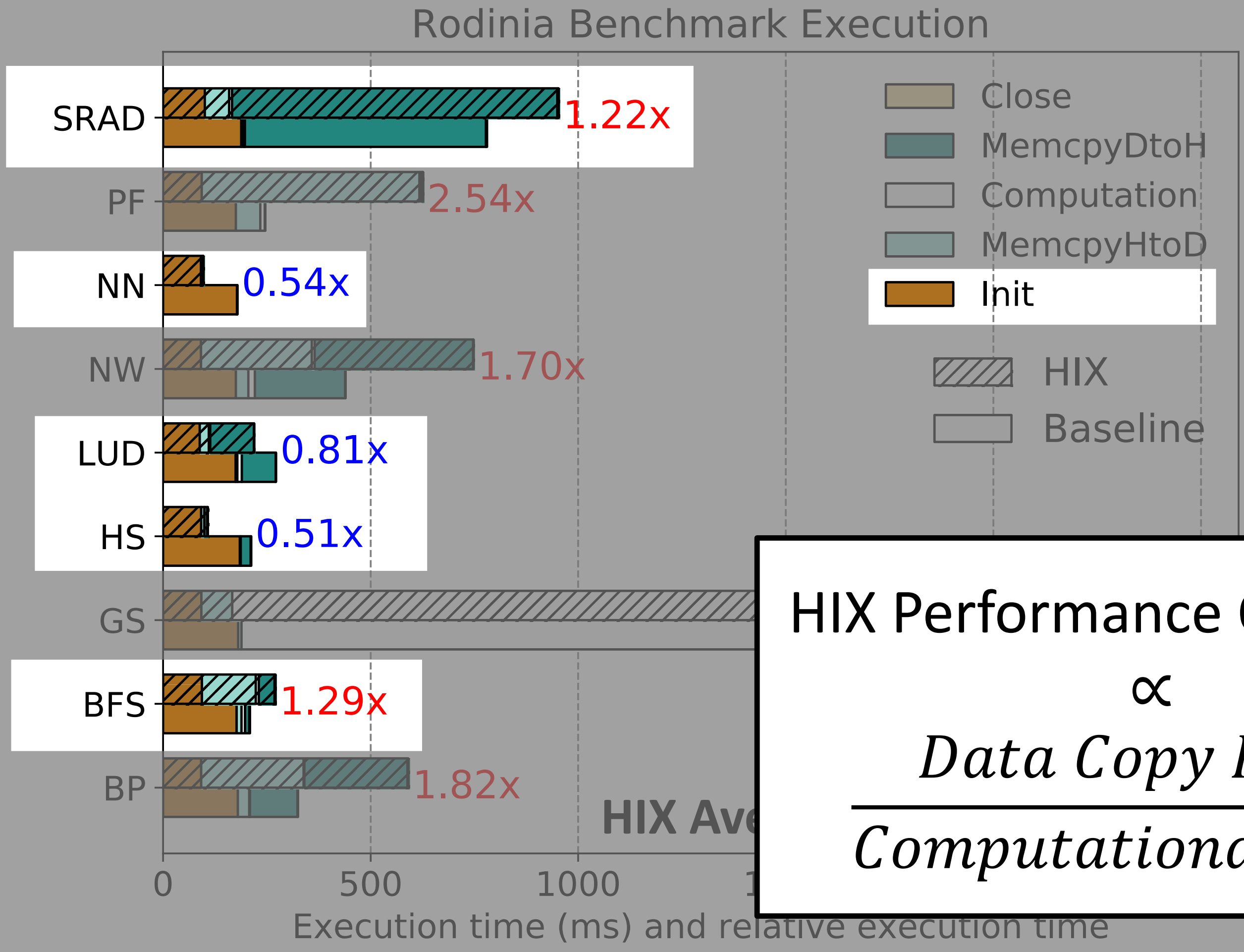
Execution occupies 88% of the entire operation time

HIX Average Overhead: 26%

High Computational Ratio → Cryptography Overhead Ratio Reduced

Performance Result: Rodinia

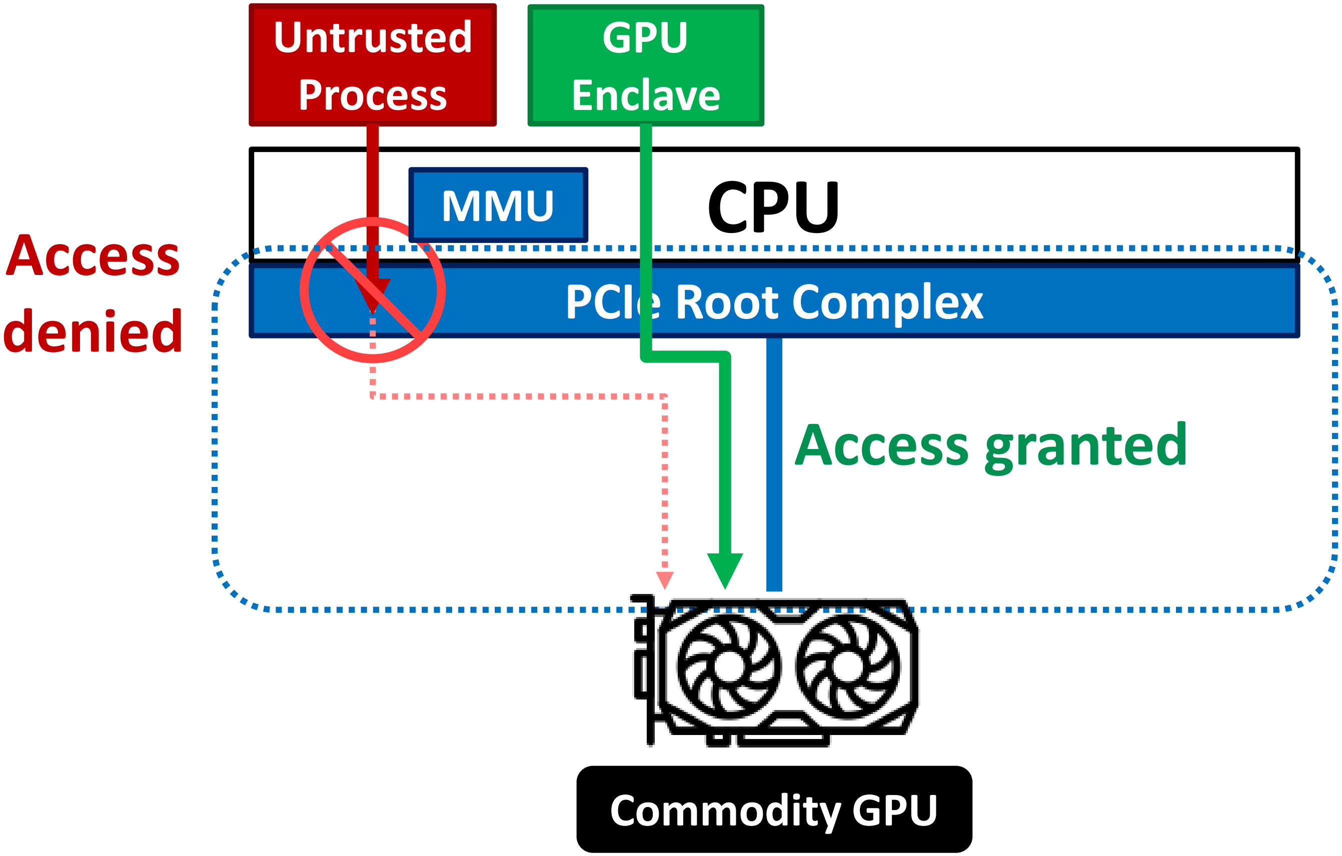
App Name	SRAD	PF	NN	NW	LUD	HS	GS	BFS	BP
Memcpy	48.4MB	256.0MB	501.2KB	192.1MB	32.0MB	12.0MB	64.0MB	46.9MB	159.8MB



$$\text{HIX Performance Overheads} \propto \frac{\textit{Data Copy Ratio}}{\textit{Computational Ratio}}$$

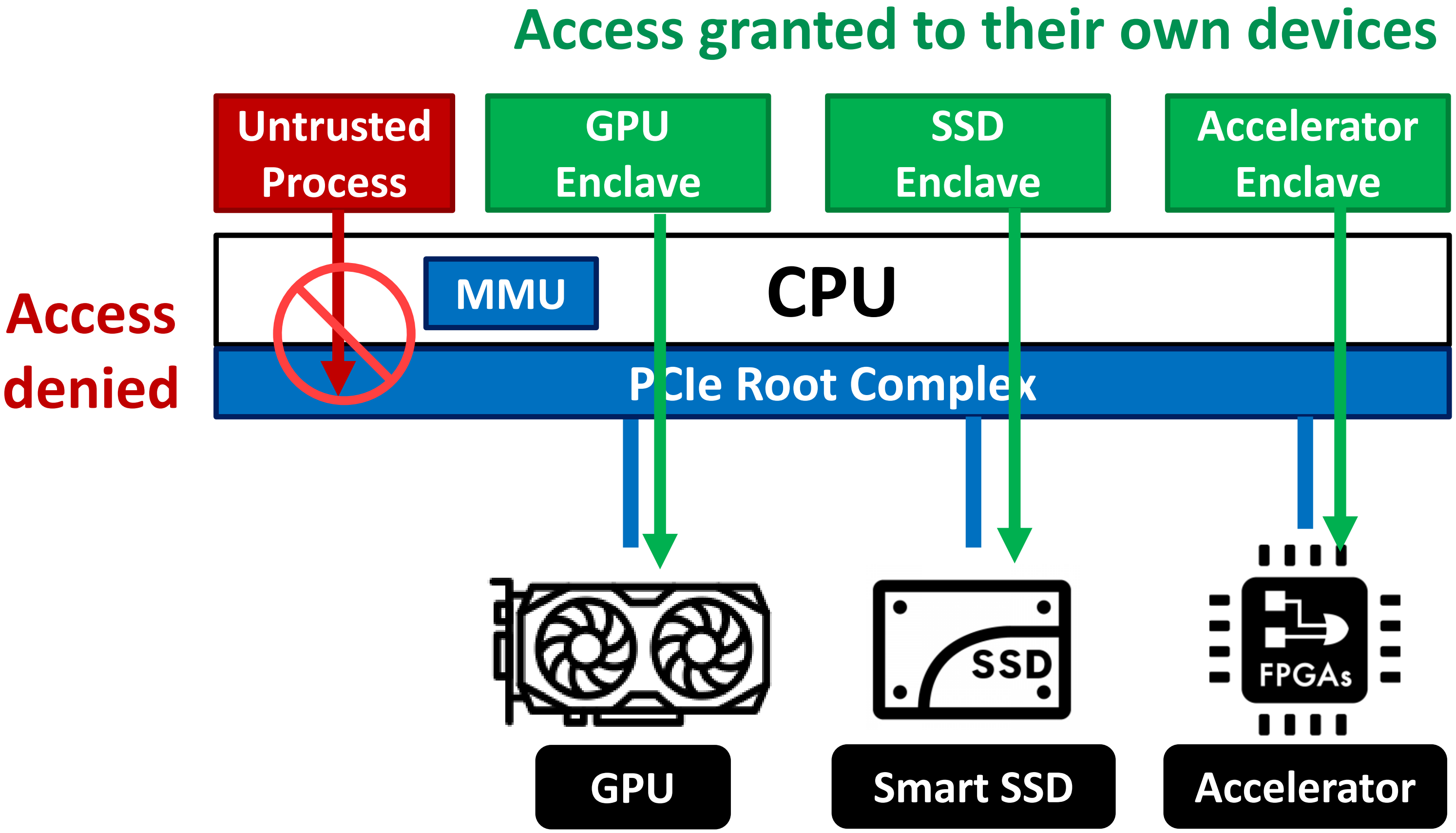
Conclusion

- HIX: Provide trusted execution environment to **commodity GPUs**



Conclusion

- HIX: Provide trusted execution environment to **commodity GPUs**



Expandable Device Protection

Heterogeneous Isolated Execution for Commodity GPUs

Thank you for Listening!

Q&A

KAIST School of Computing

COLUMBIA | ENGINEERING
The Fu Foundation School of Engineering and Applied Science