black hat® EUROPE 2018

DECEMBER 3-6, 2018

EXCEL LONDON / UNITED KINGDOM

DIVIDE ET IMPERA: MEMORY RANGER RUNS DRIVERS IN ISOLATED KERNEL SPACES

Igor Korkin, Ph.D

#BHEU / @BLACK HAT EVENTS

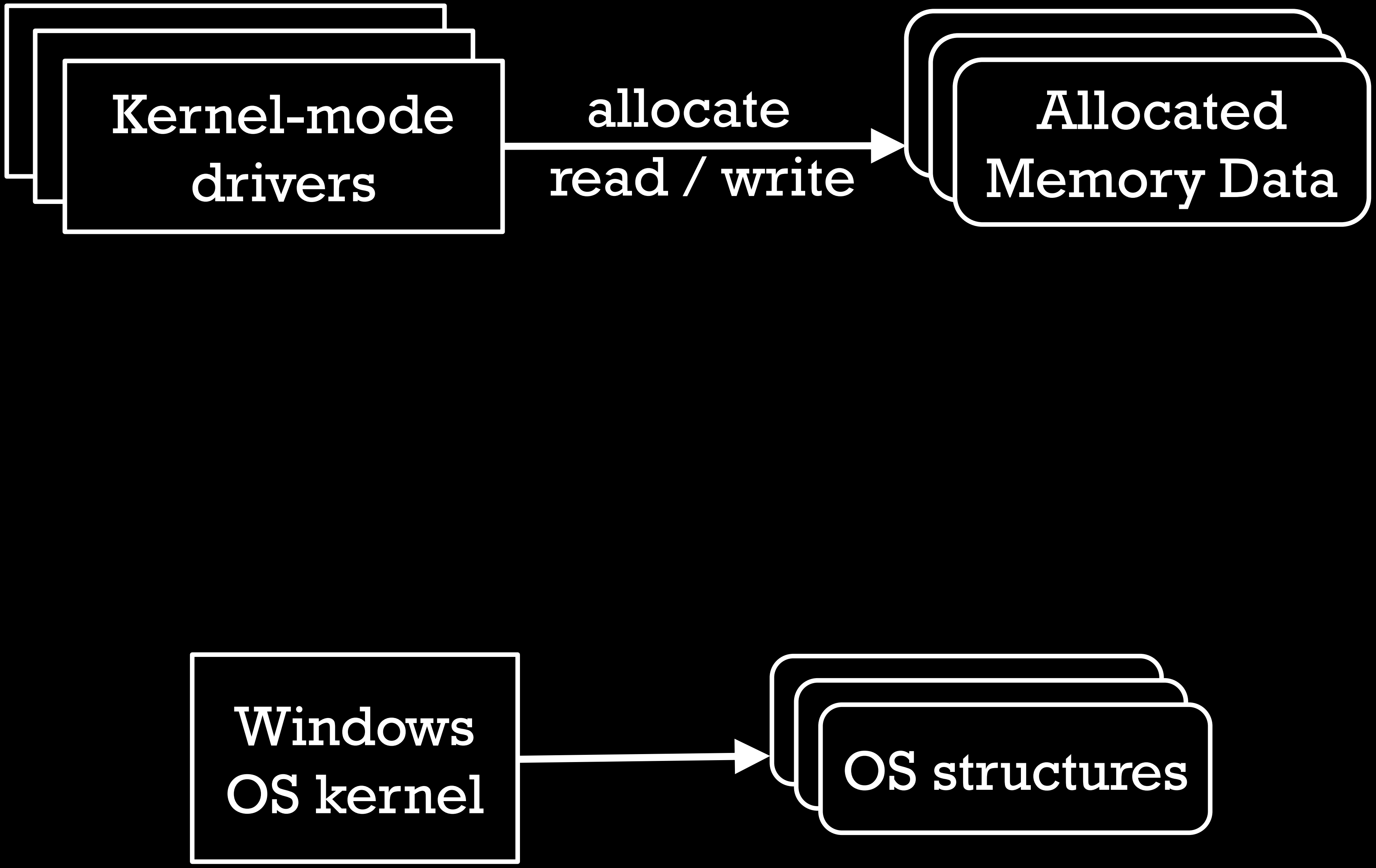
WHOAMI

- MEPhI Alumni, PhD in Cyber Security, published 23 papers
- Area of interest is Windows Kernel security:
  - Memory Forensics
  - Rootkits Detection
  - Bare-Metal Hypervisors
- Fan of academic cross-disciplinary research - igorkorkin.blogspot.com
- Love traveling and powerlifting - igor.korkin

AGENDA

- Attacking the kernel-mode memory
- Existing protection: Windows built-in security and research projects
- MemoryRanger hypervisor: idea, details, demos

ATTACKS ON KERNEL MODE MEMORY



Kernel-mode drivers

allocate

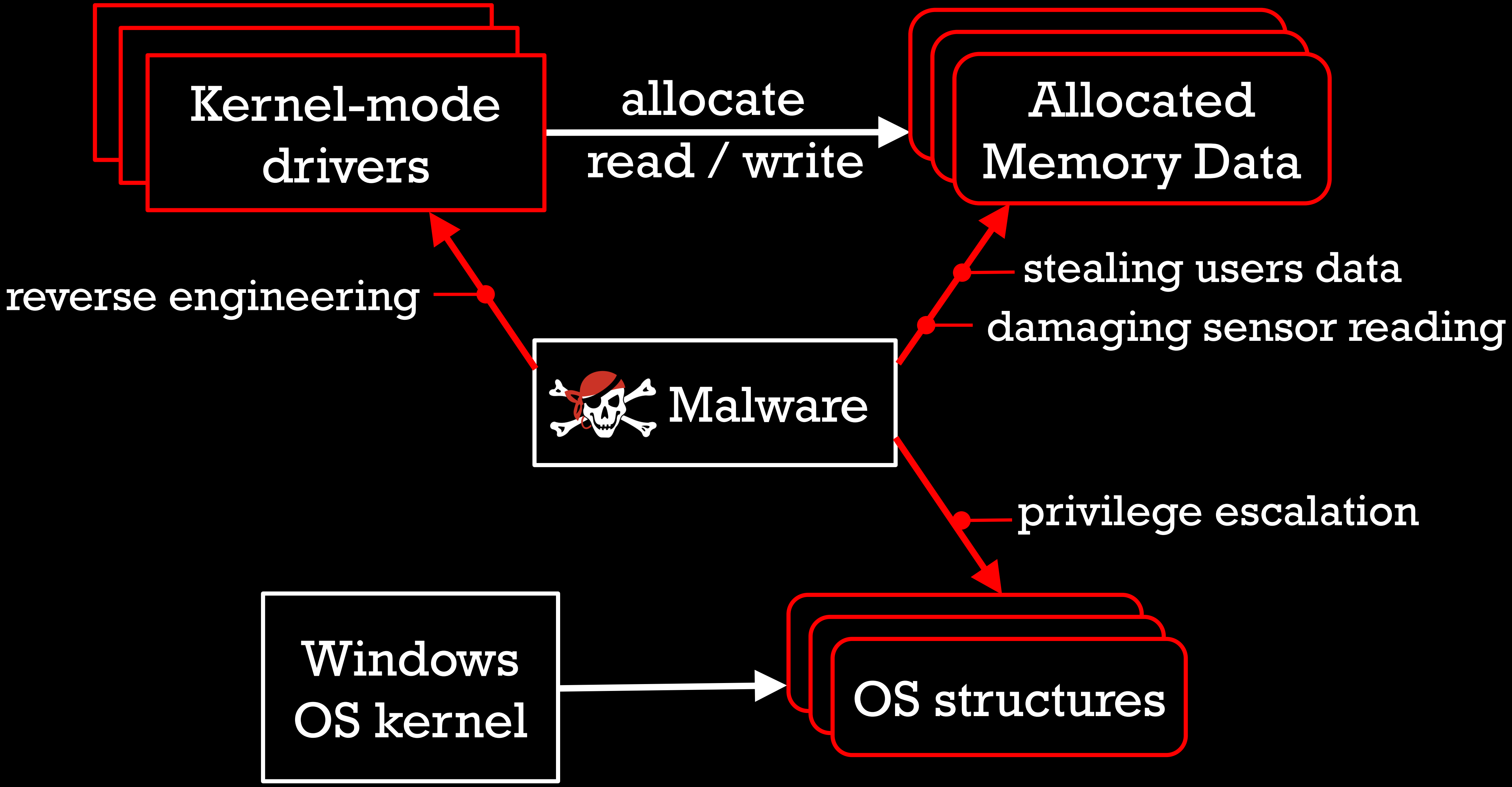
read / write

Allocated Memory Data

Windows OS kernel

OS structures

ATTACKS ON KERNEL MODE MEMORY



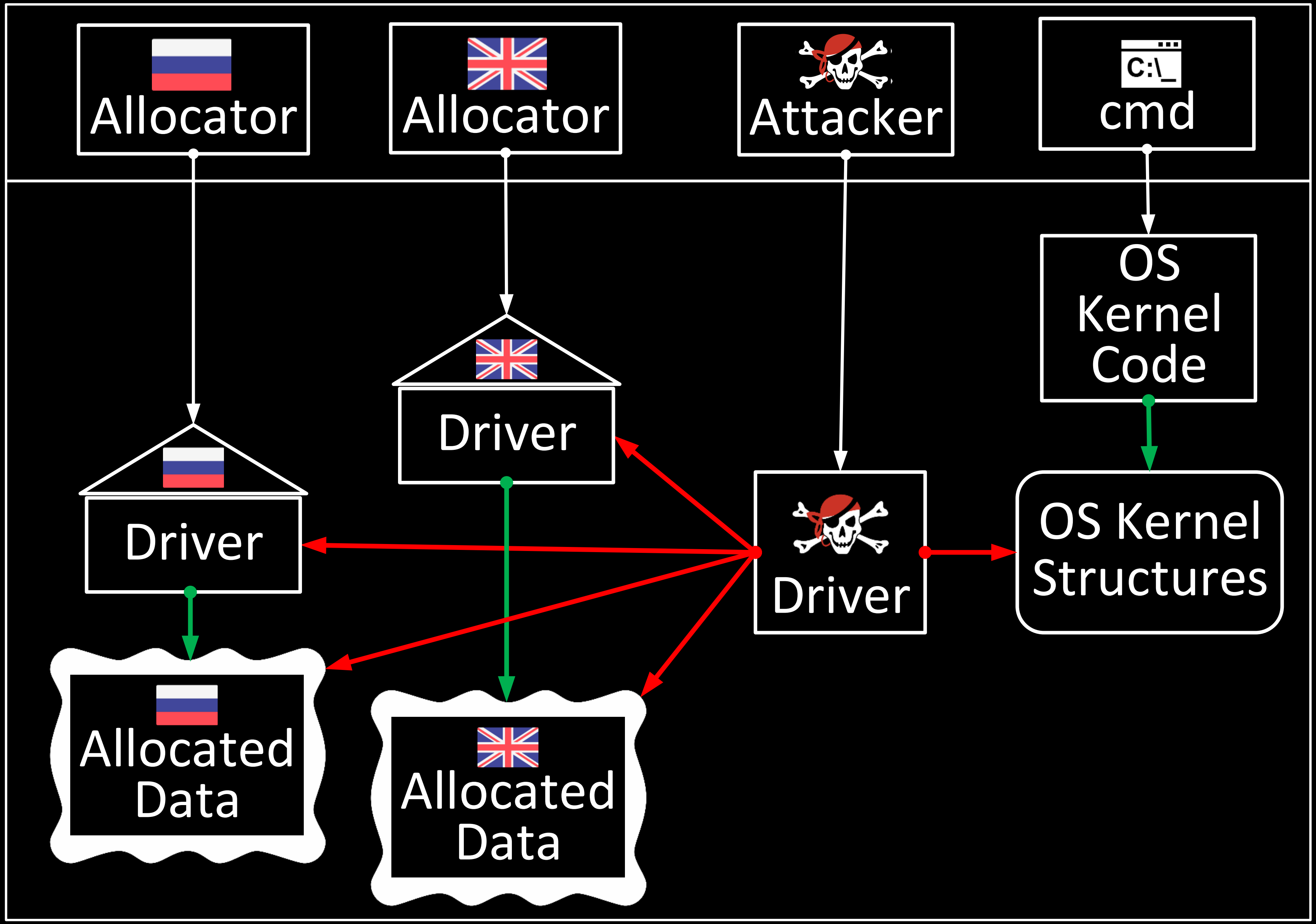
TWO HOUSES WITH PRIVATE ART COLLECTIONS

Stars by Wassily Kandinsky

Girl with Balloon by Banksy

Hi all! I am a hacker-attacker! I will inspect these houses to steal and damage painting!

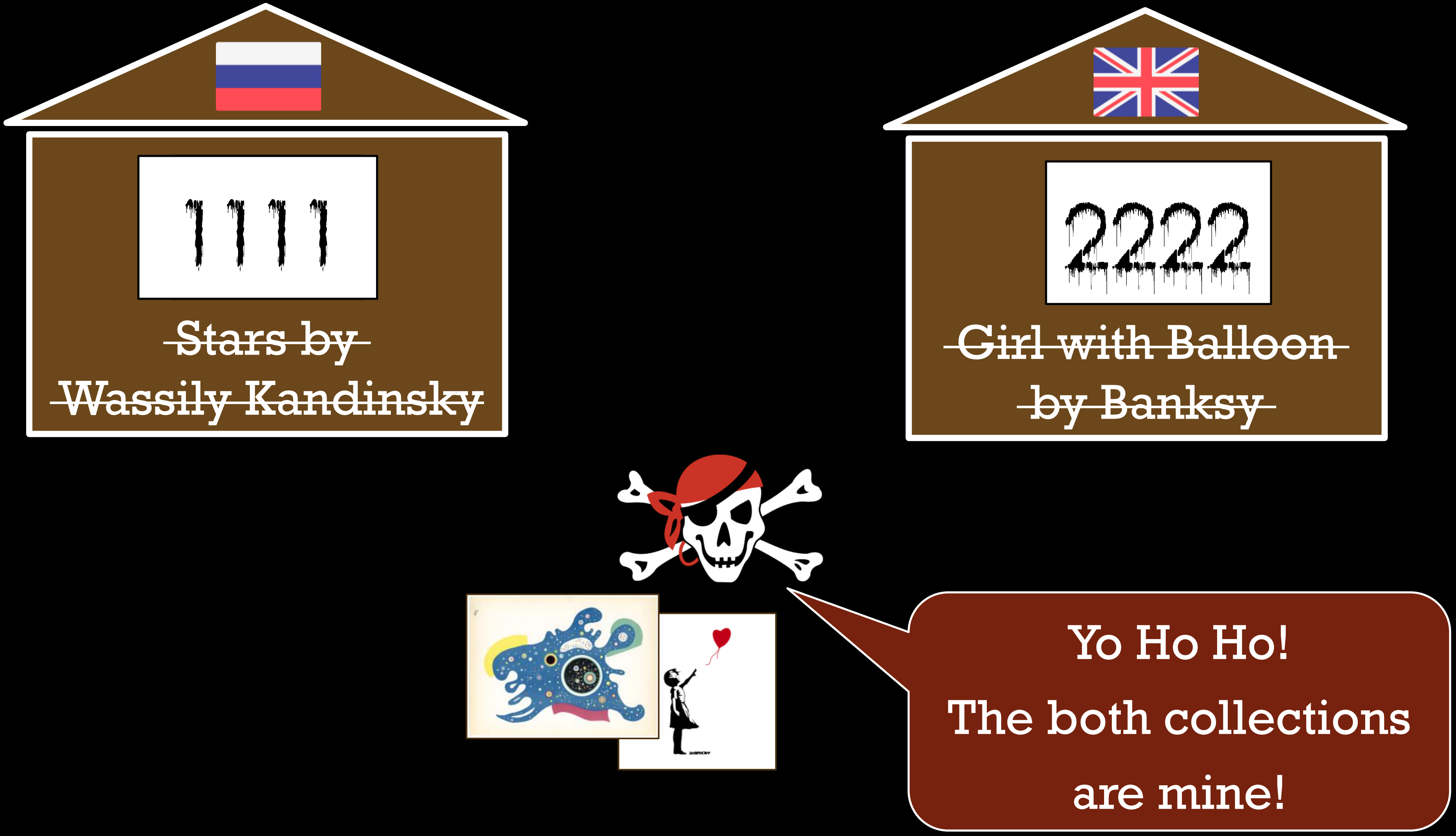
DEMO: THE ATTACK



DEMO: THE ATTACK

The online version is here – https://www.youtube.com/embed/HNxc-tjy3QA?vq=hd1080

TWO HOUSES WITH PRIVATE ART COLLECTIONS



1111

~~Stars by Wassily Kandinsky~~

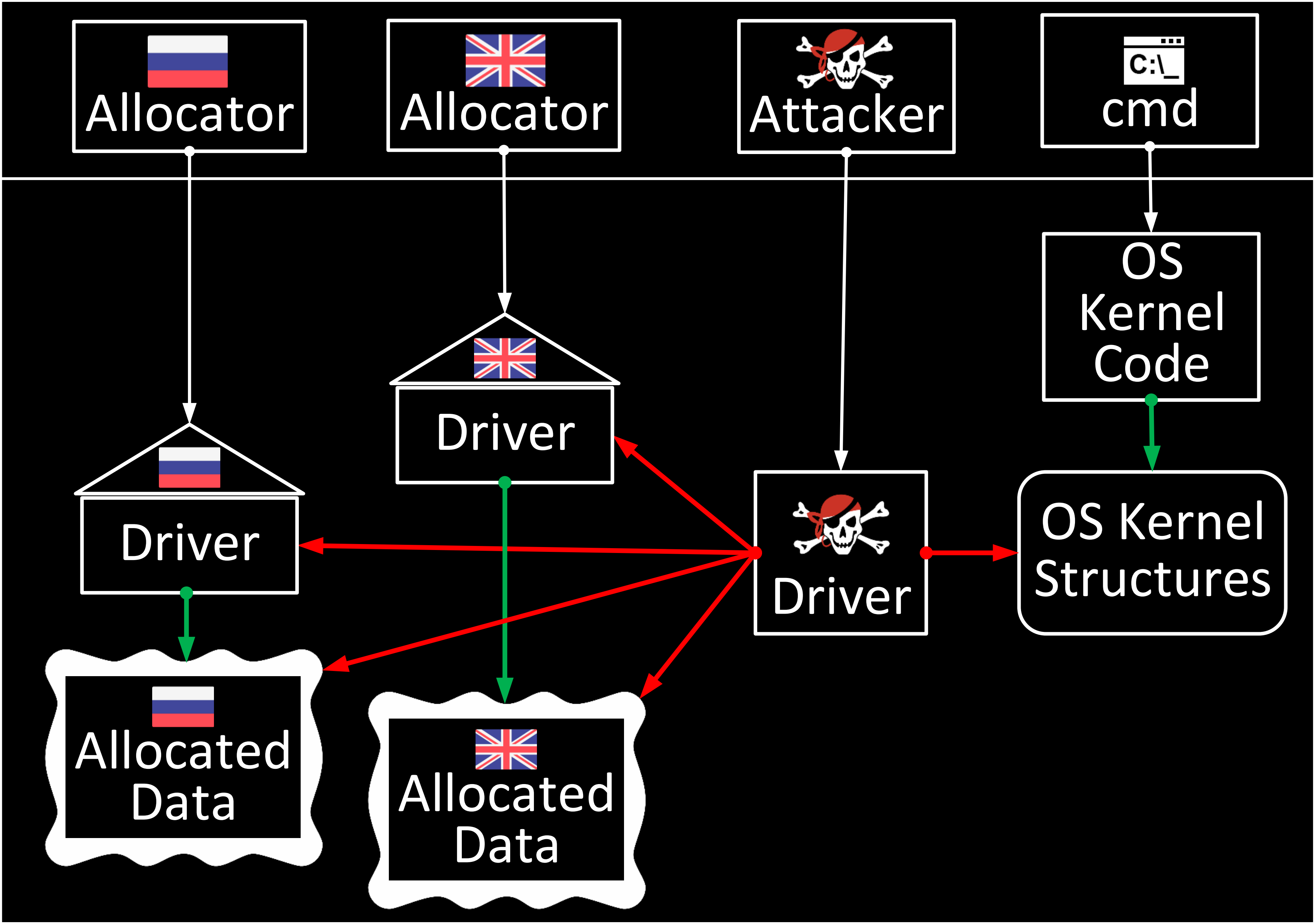
2222

~~Girl with Balloon by Banksy~~

Yo Ho Ho!

The both collections are mine!

THE ATTACK HAS NOT BEEN PREVENTED

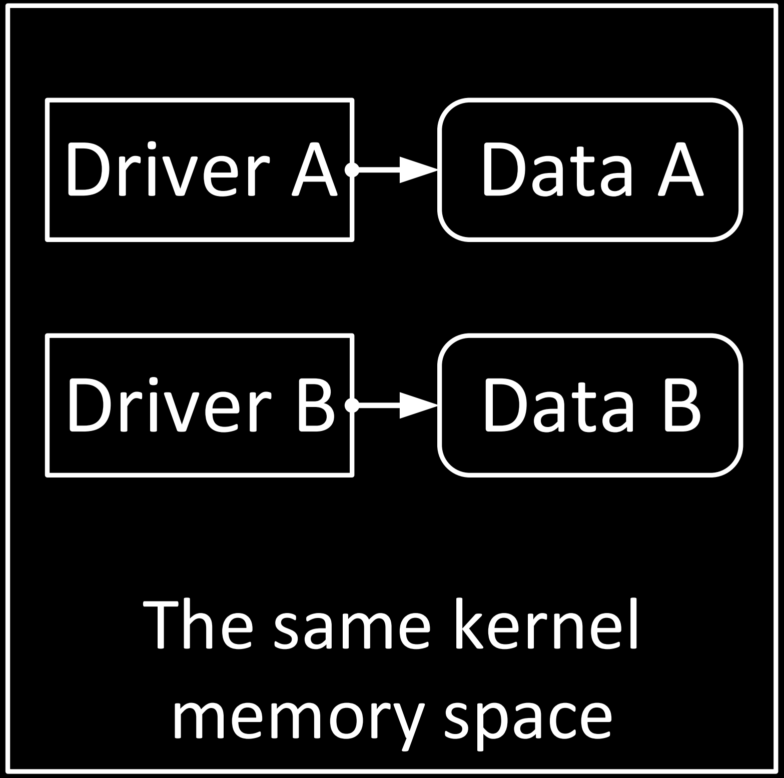


BACKGROUND ANALYSIS

Memory protection projects	Malware attacks on					
	Code of OS & third-party drivers		OS data: internal structures		Data of third-party drivers	
	Read	Write	Read	Write	Read	Write
Windows Security	-	BSOD 0xBE by Device Guard	-	BSOD 0x109 by PatchGuard	-	-
PrivGuard	-	-	-	+	-	-
LAKEED	+	+	+	+	-	-
LKMG	-	+	+	+	+	+
rR^X	+	+	-	-	-	-
AllMemPro	-	-	+	+	+	+
Memory Ranger	+	+	+	+	+	+

IDEA OF DRIVERS EXECUTION ISOLATION

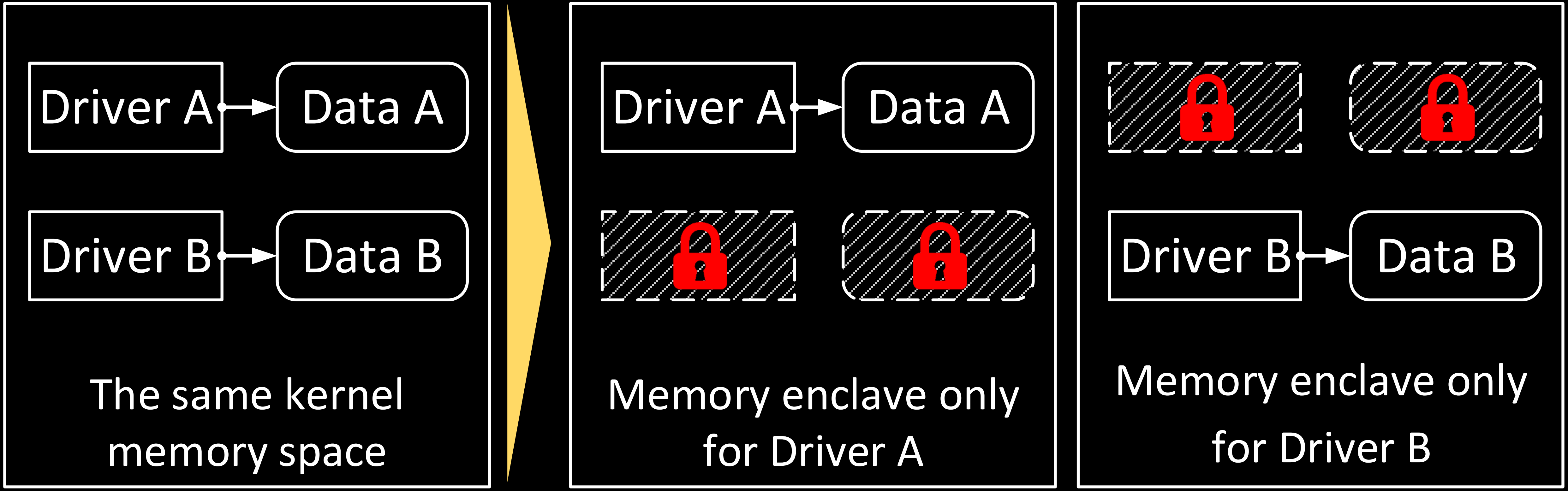
Now all drivers share the same memory space



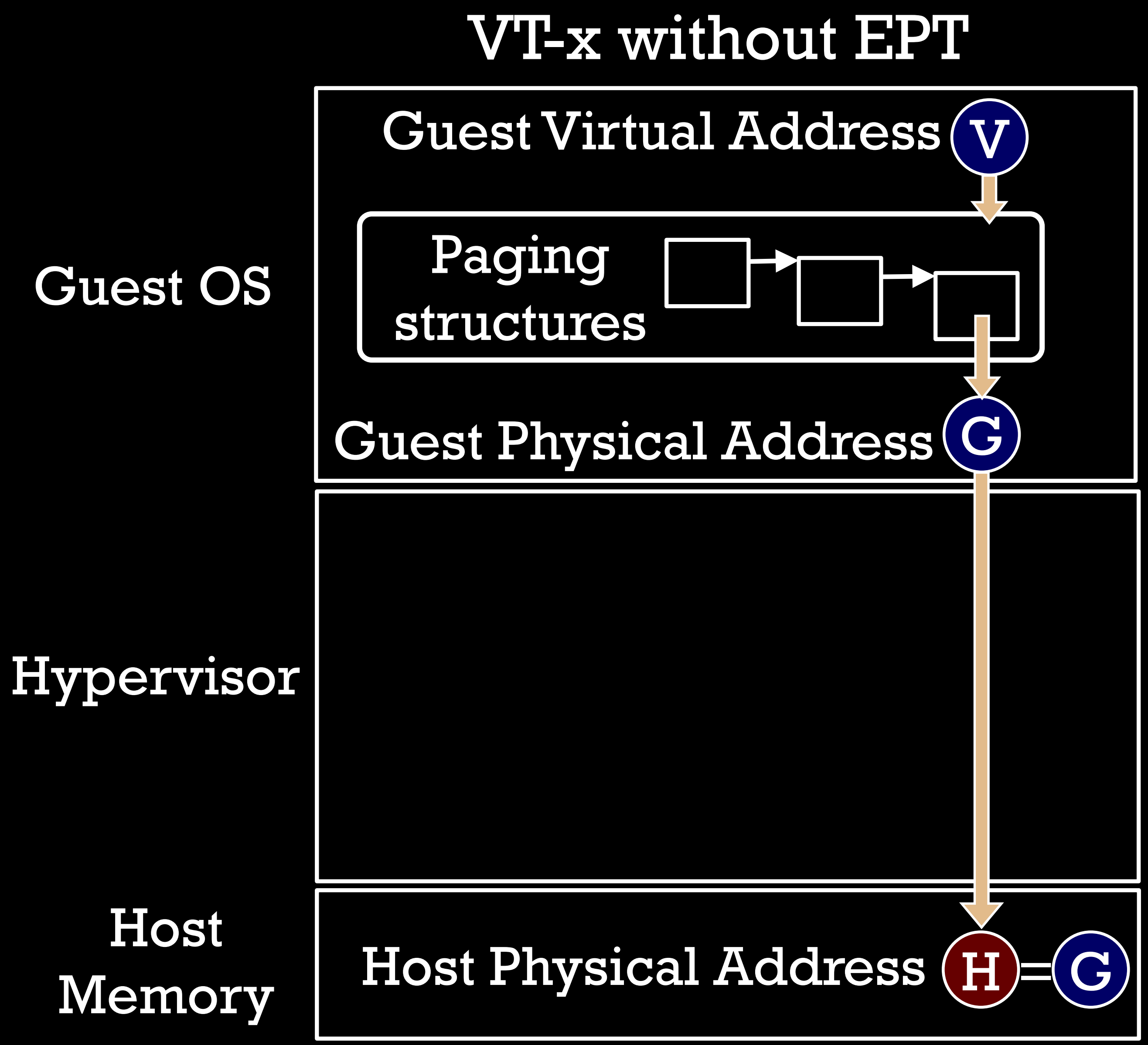
IDEA OF DRIVERS EXECUTION ISOLATION

Now all drivers share the same memory space

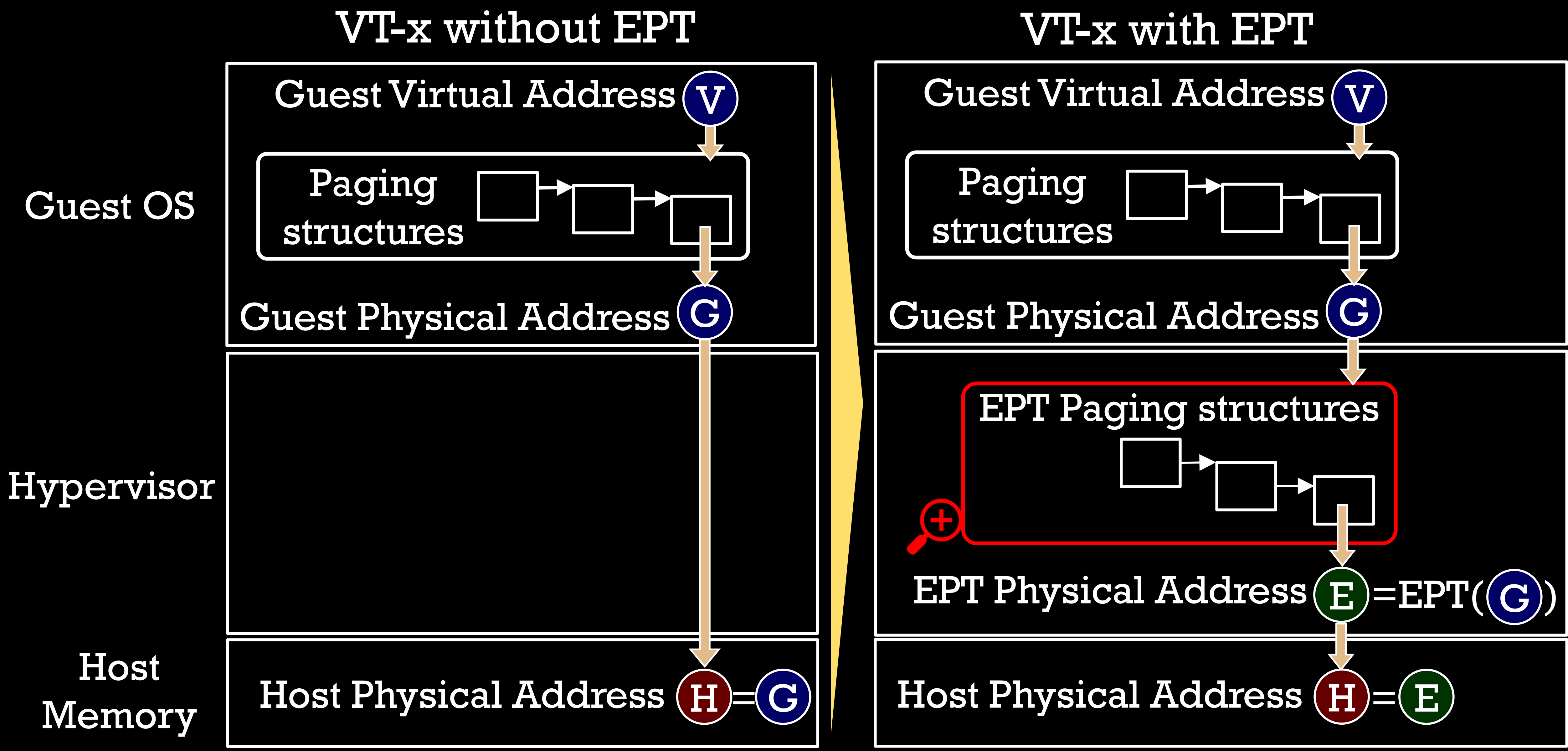
Let’s execute these two drivers into separate memory enclosures


Driver A
Data A
Driver B
Data B
The same kernel memory space
Driver A
Data A
Memory enclave only for Driver A
Driver B
Data B
Memory enclave only for Driver B

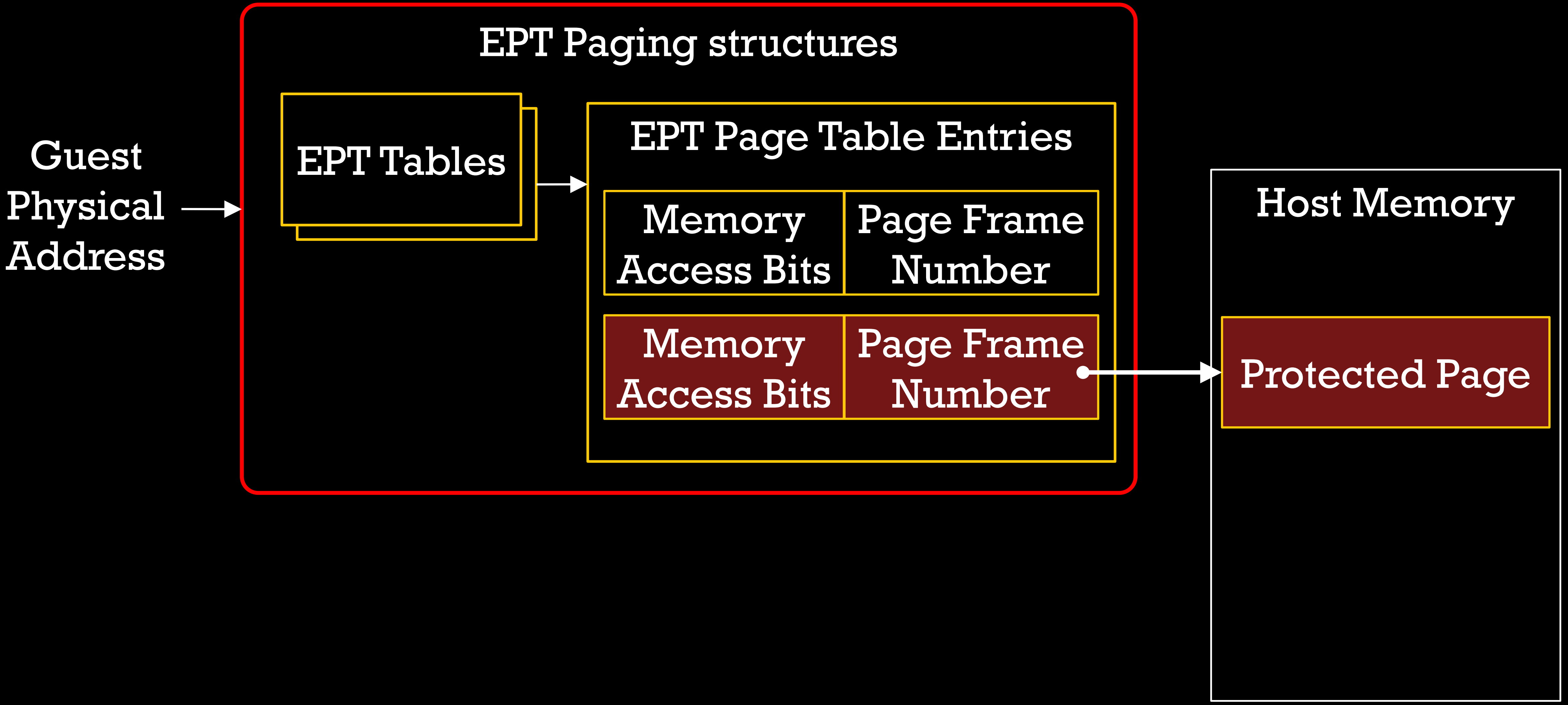
PROCESSING MEMORY ACCESS: EPT FEATURE



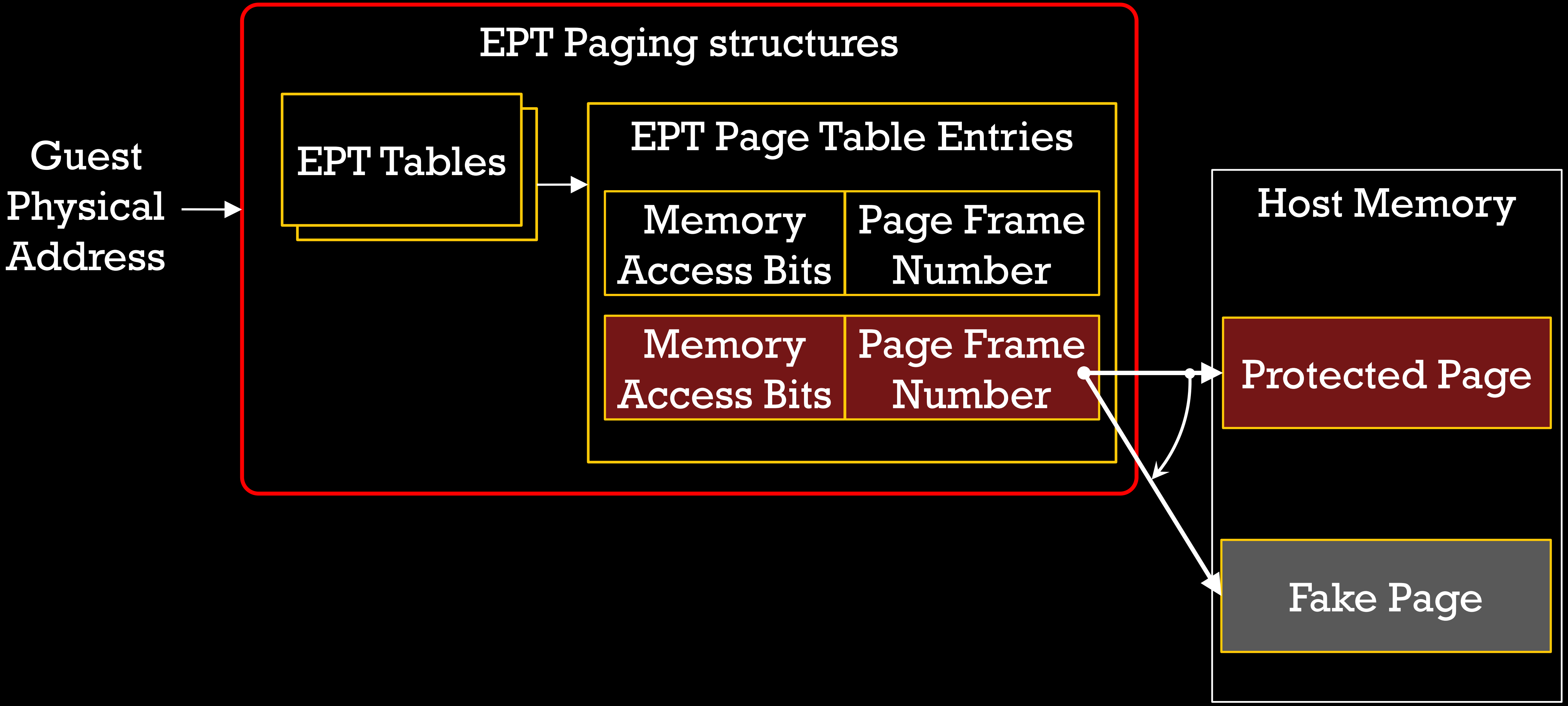
PROCESSING MEMORY ACCESS: EPT FEATURE



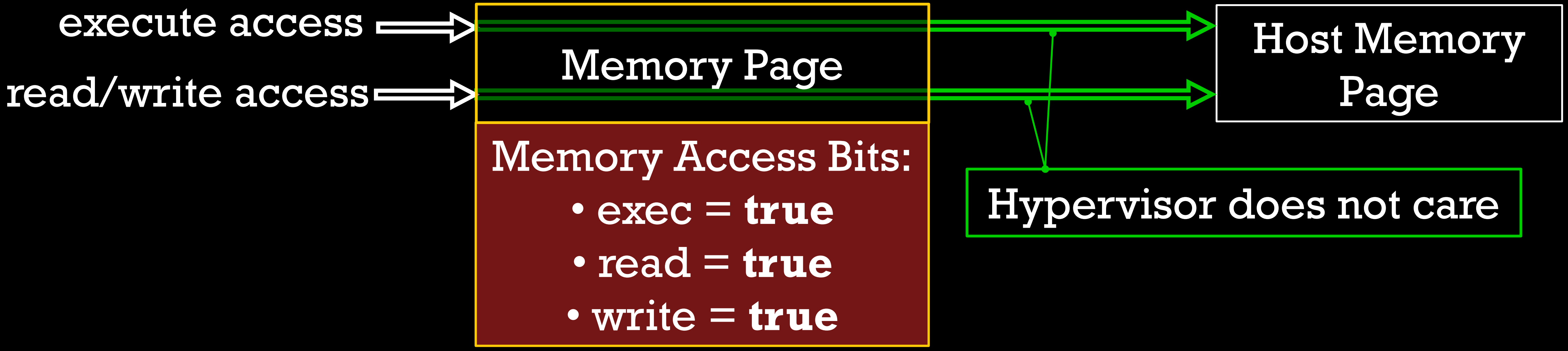
INSIDE EPT PAGING STRUCTURES. EPT PFN

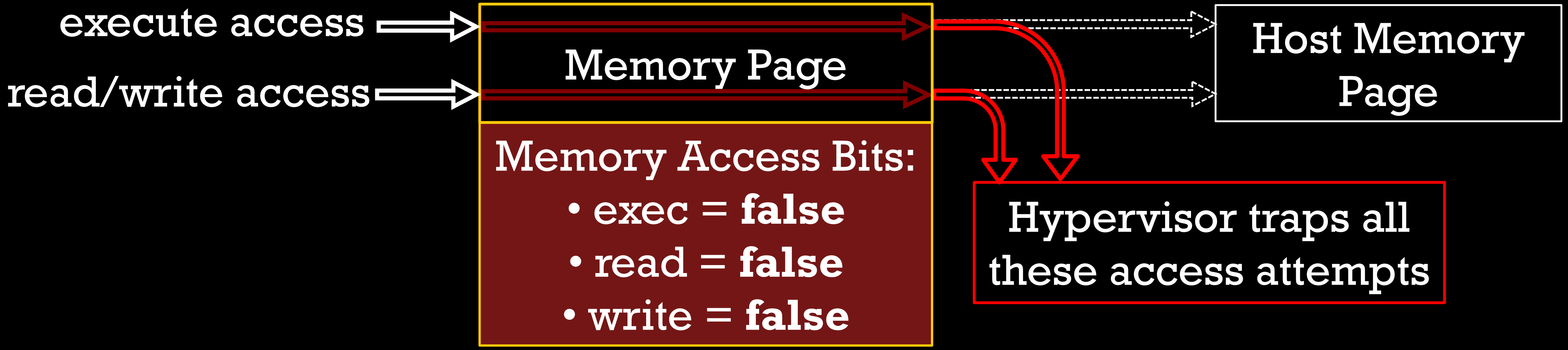


INSIDE EPT PAGING STRUCTURES. EPT PFN

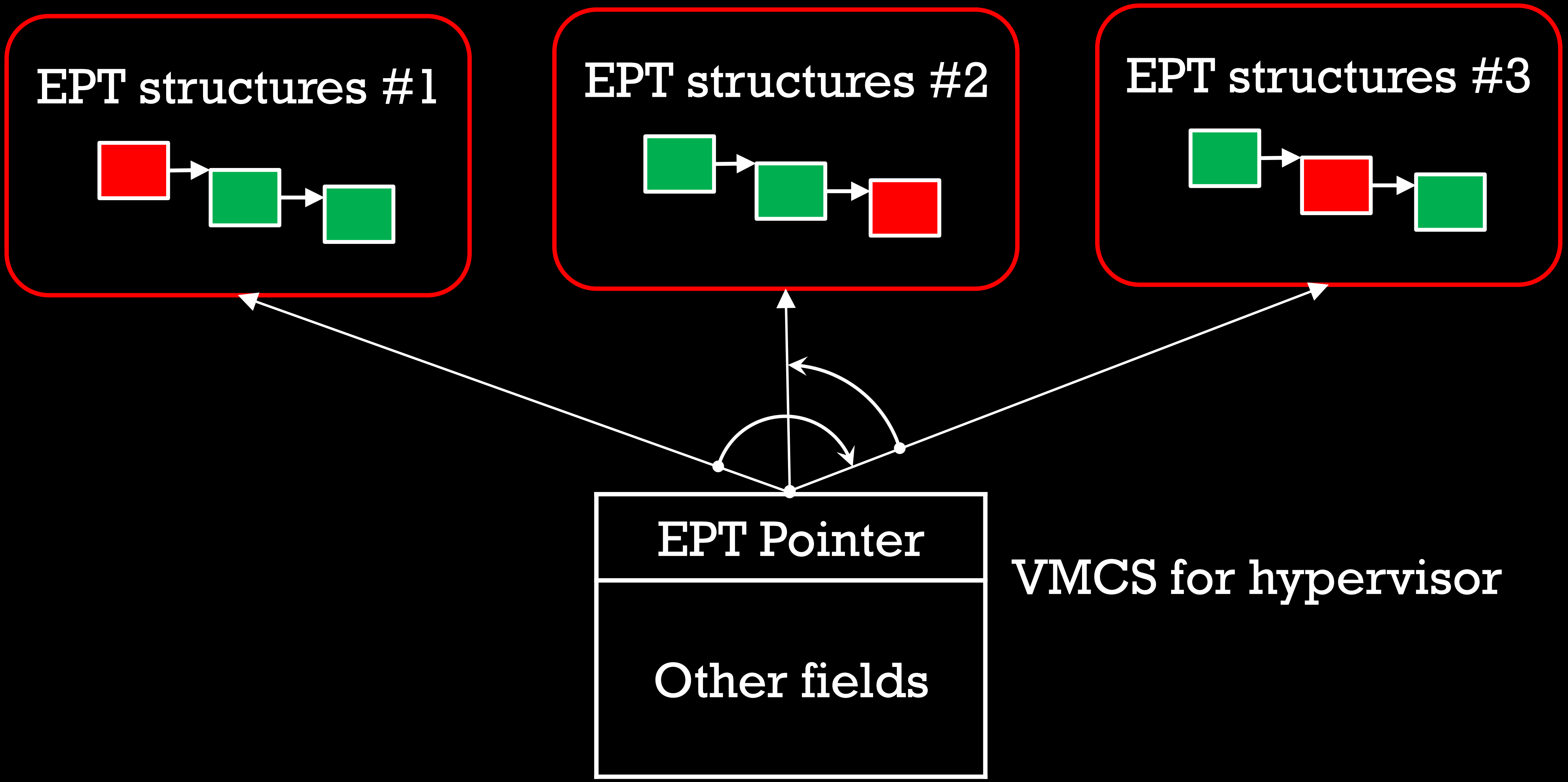


INSIDE EPT PAGING STRUCTURES. EPT BITS

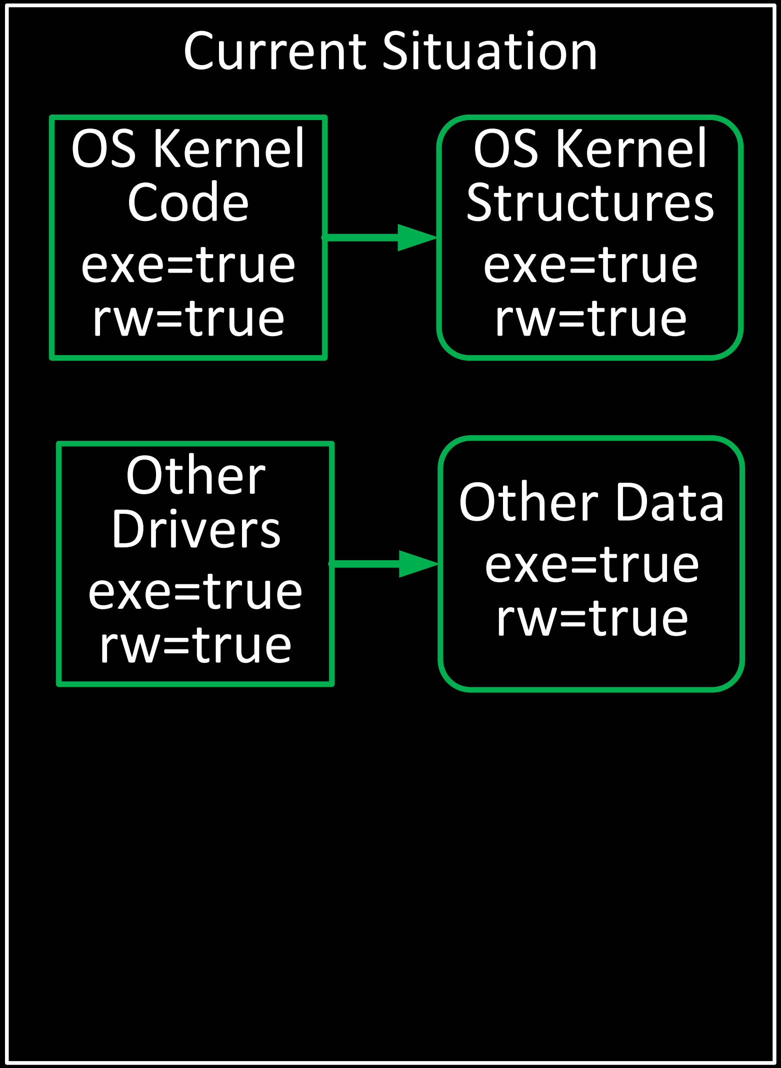




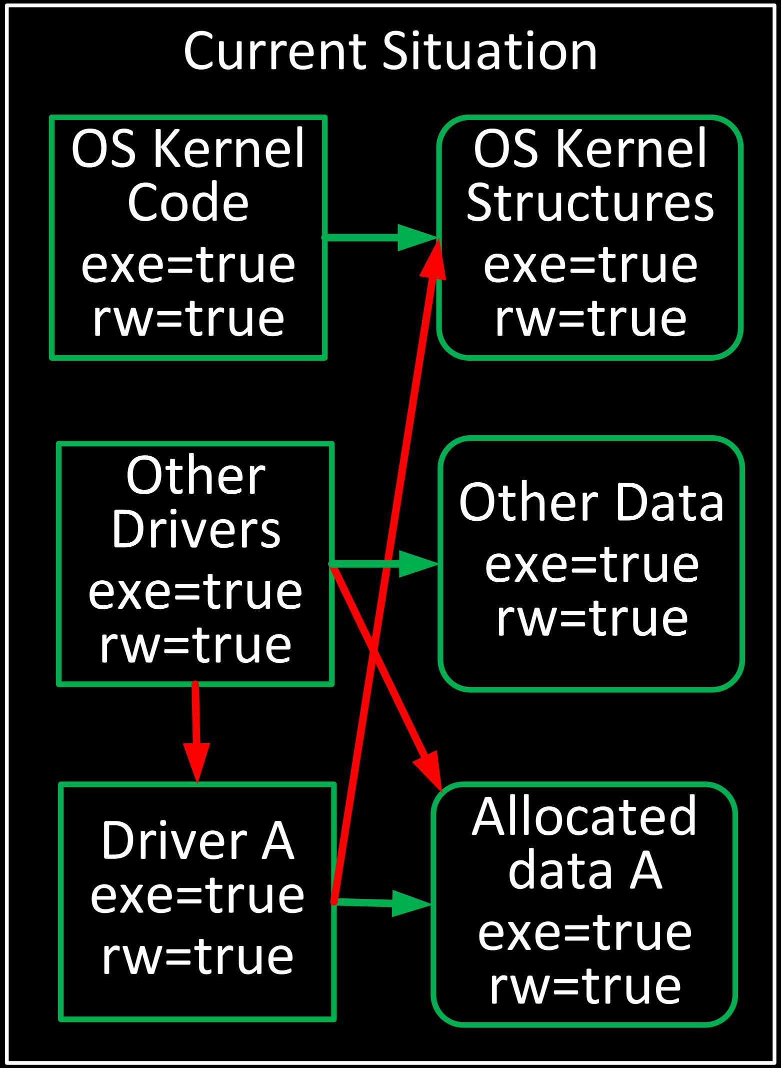
INSIDE EPT PAGING STRUCTURES



APPLYING EPT FOR DRIVERS ISOLATION



APPLYING EPT FOR DRIVERS ISOLATION



Current Situation

OS Kernel Code
exe=true
rw=true

OS Kernel Structures
exe=true
rw=true

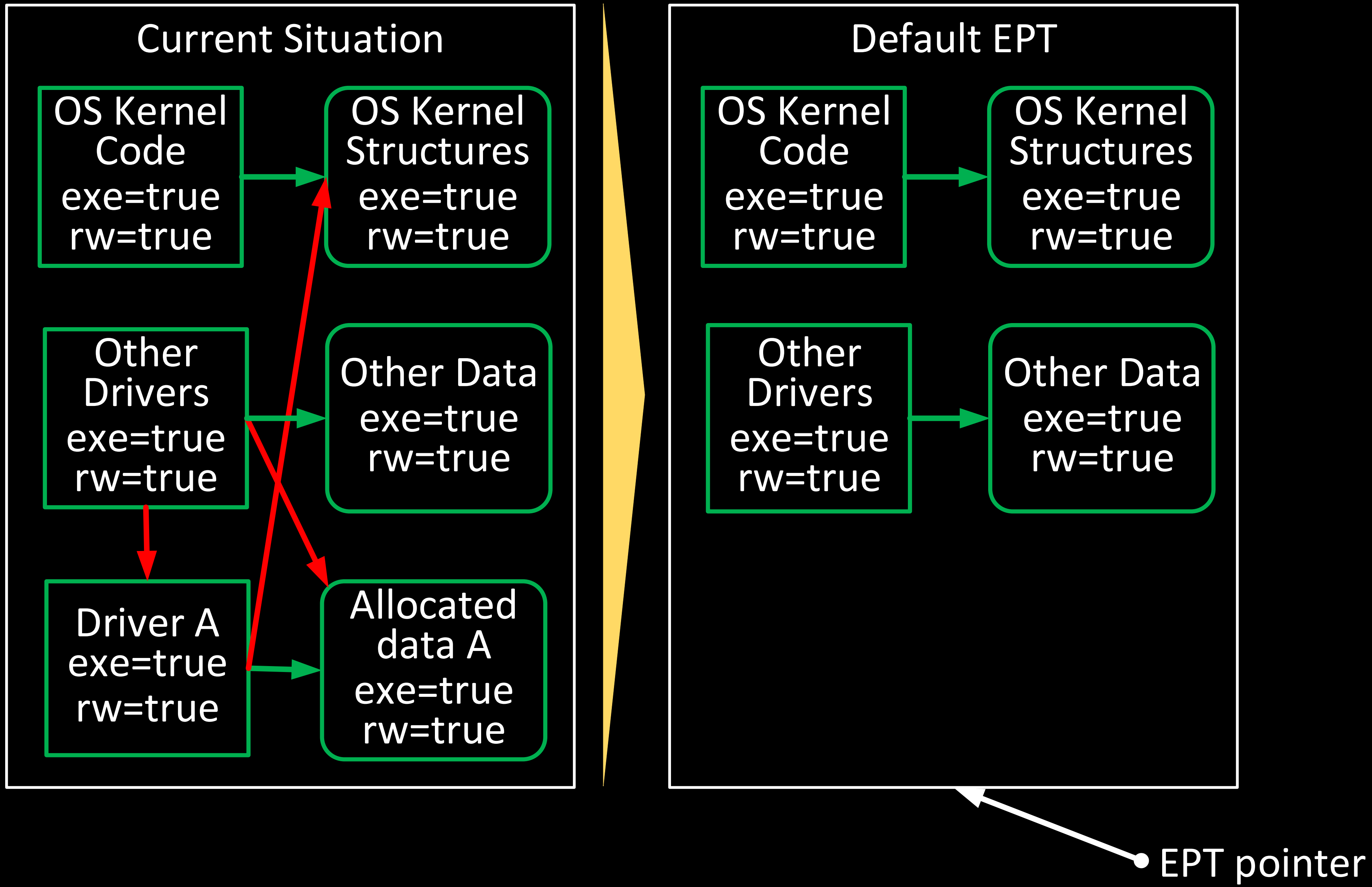
Other Drivers
exe=true
rw=true

Other Data
exe=true
rw=true

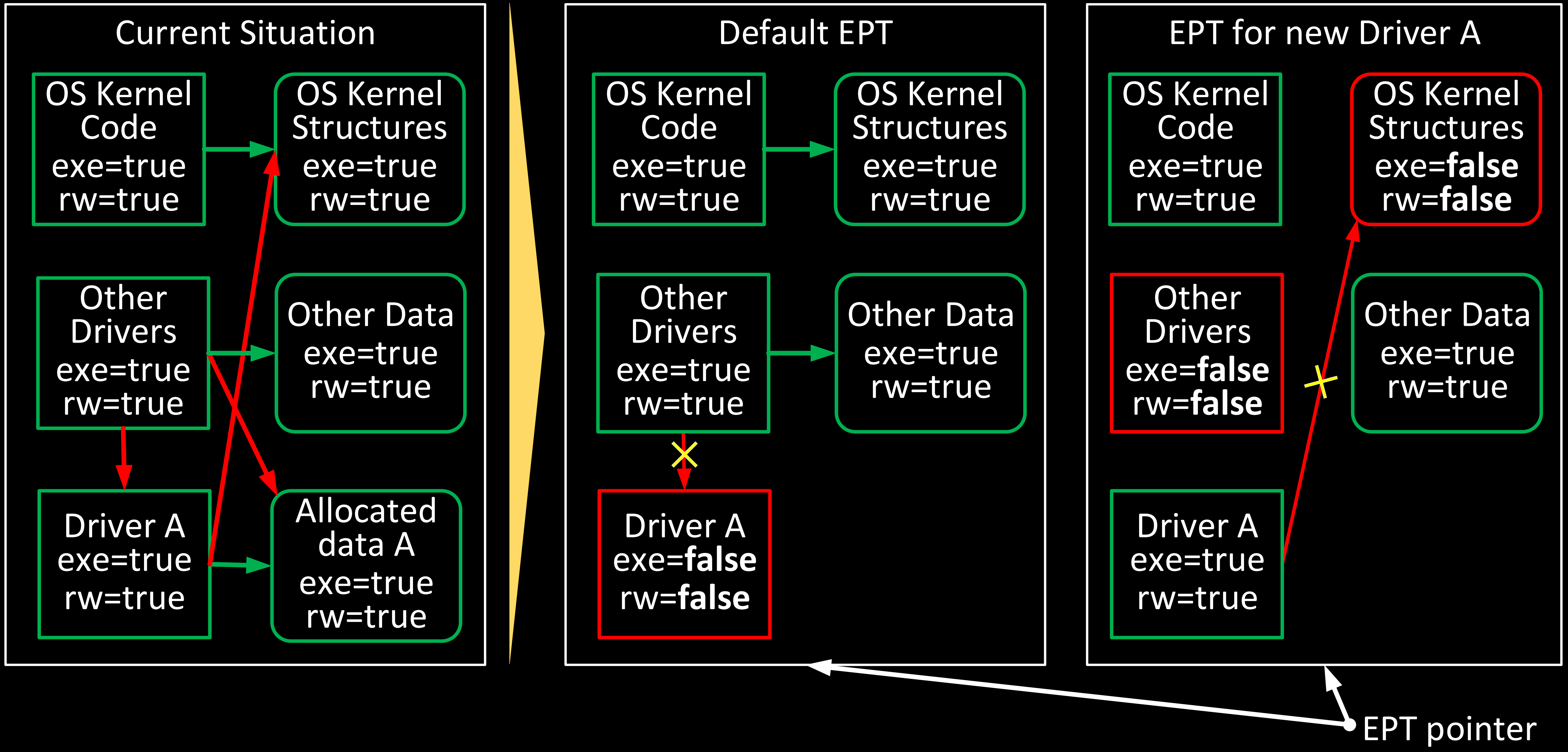
Driver A
exe=true
rw=true

Allocated data A
exe=true
rw=true

APPLYING EPT FOR DRIVERS ISOLATION


Current Situation
OS Kernel Code exe=true rw=true
OS Kernel Structures exe=true rw=true
Other Drivers exe=true rw=true
Other Data exe=true rw=true
Driver A exe=true rw=true
Allocated data A exe=true rw=true
Default EPT
OS Kernel Code exe=true rw=true
OS Kernel Structures exe=true rw=true
Other Drivers exe=true rw=true
Other Data exe=true rw=true
EPT pointer

APPLYING EPT FOR DRIVERS ISOLATION



Current Situation

OS Kernel Code exe=true rw=true

OS Kernel Structures exe=true rw=true

Other Drivers exe=true rw=true

Other Data exe=true rw=true

Driver A exe=true rw=true

Allocated data A exe=true rw=true

Default EPT

OS Kernel Code exe=true rw=true

OS Kernel Structures exe=true rw=true

Other Drivers exe=true rw=true

Other Data exe=true rw=true

Driver A exe=**false** rw=**false**

EPT for new Driver A

OS Kernel Code exe=true rw=true

OS Kernel Structures exe=**false** rw=**false**

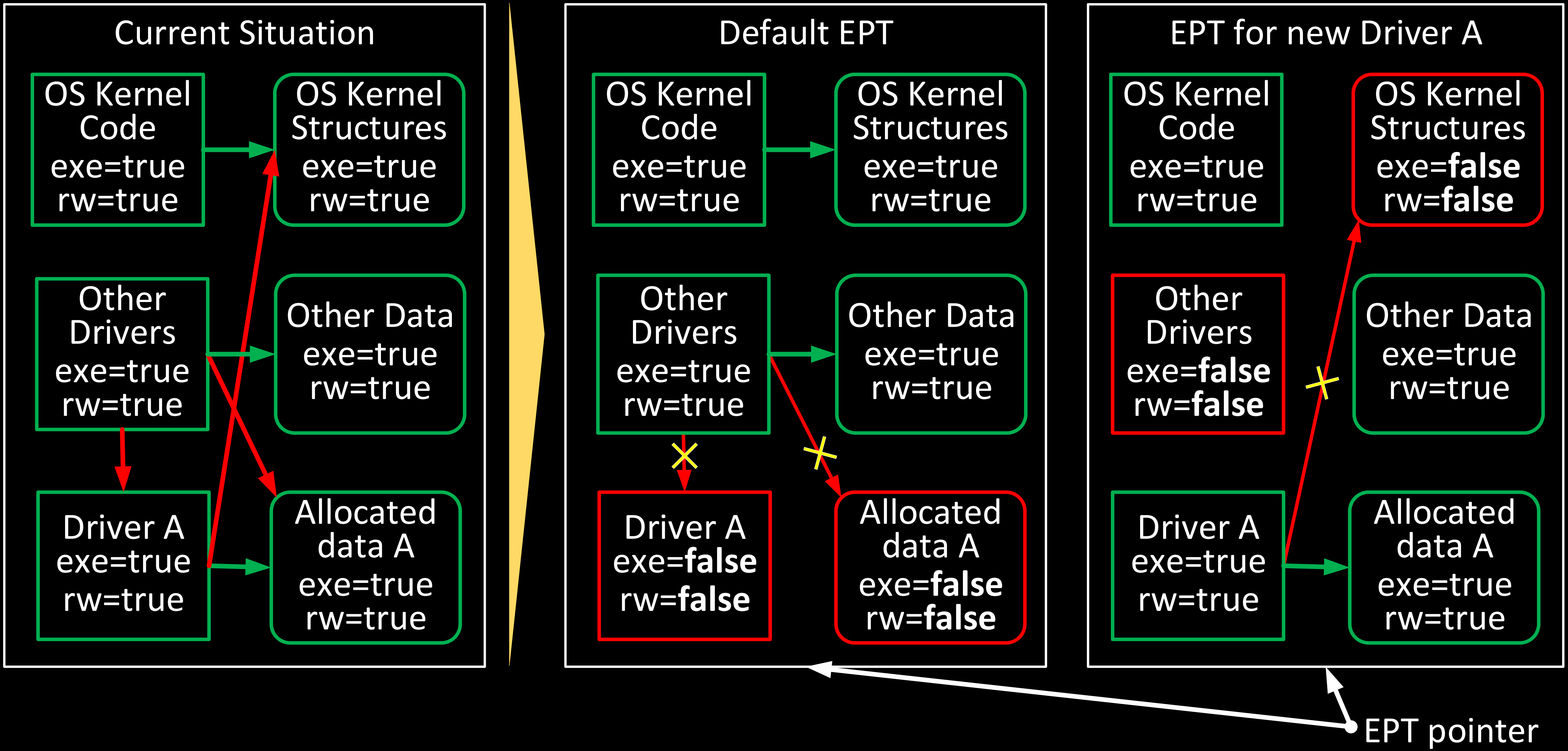
Other Drivers exe=**false** rw=**false**

Other Data exe=true rw=true

Driver A exe=true rw=true

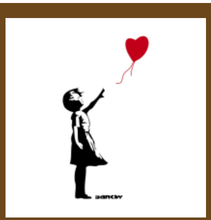
EPT pointer

APPLYING EPT FOR DRIVERS ISOLATION


Current Situation
OS Kernel Code exe=true rw=true
OS Kernel Structures exe=true rw=true
Other Drivers exe=true rw=true
Other Data exe=true rw=true
Driver A exe=true rw=true
Allocated data A exe=true rw=true
Default EPT
OS Kernel Code exe=true rw=true
OS Kernel Structures exe=true rw=true
Other Drivers exe=true rw=true
Other Data exe=true rw=true
Driver A exe=false rw=false
Allocated data A exe=false rw=false
EPT for new Driver A
OS Kernel Code exe=true rw=true
OS Kernel Structures exe=false rw=false
Other Drivers exe=false rw=false
Other Data exe=true rw=true
Driver A exe=true rw=true
Allocated data A exe=true rw=true
EPT pointer

THREE HOUSES WITH PRIVATE ART COLLECTIONS

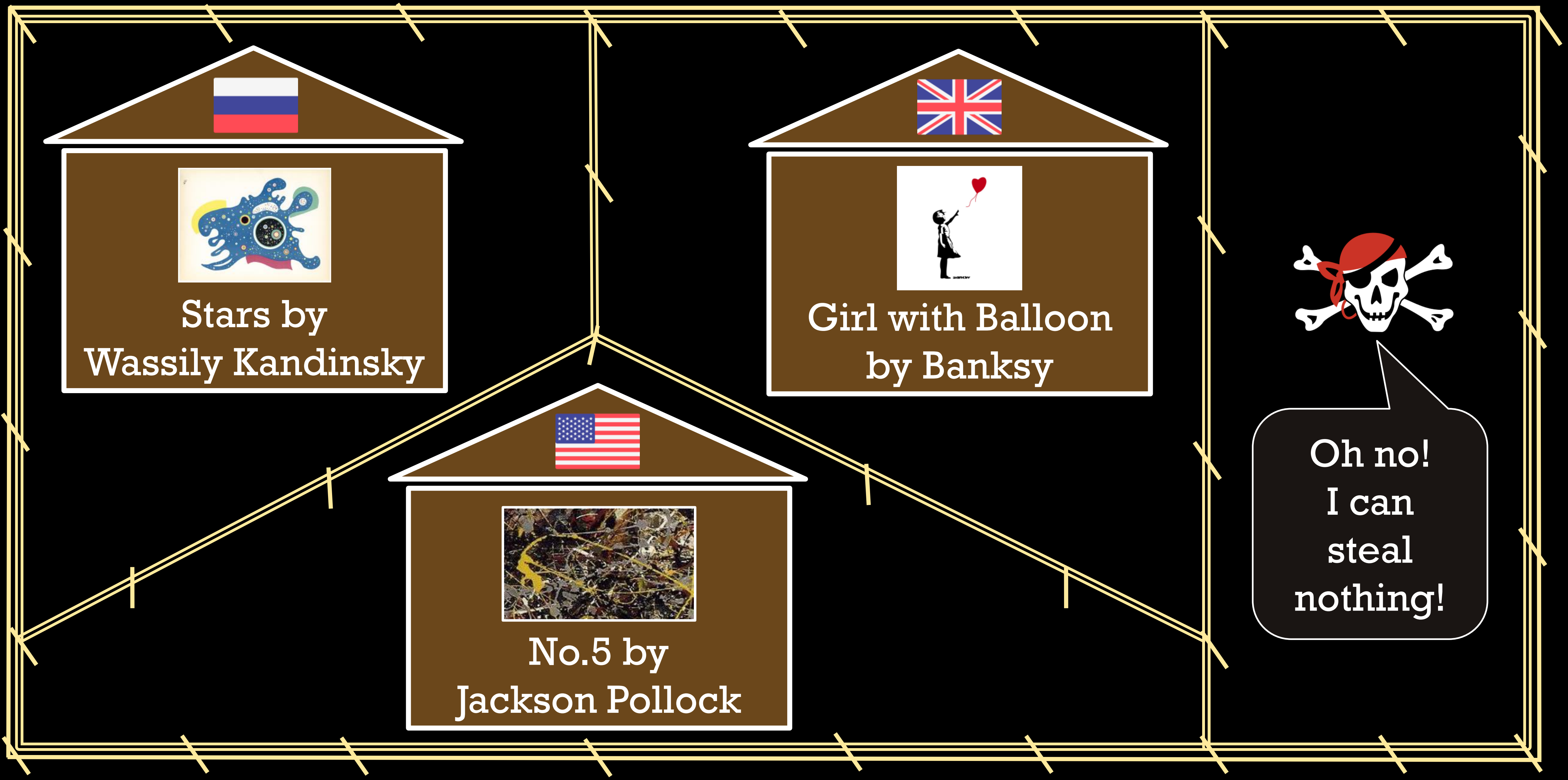
Stars by
Wassily Kandinsky



Girl with Balloon
by Banksy

No.5 by
Jackson Pollock

THREE HOUSES WITH PRIVATE ART COLLECTIONS



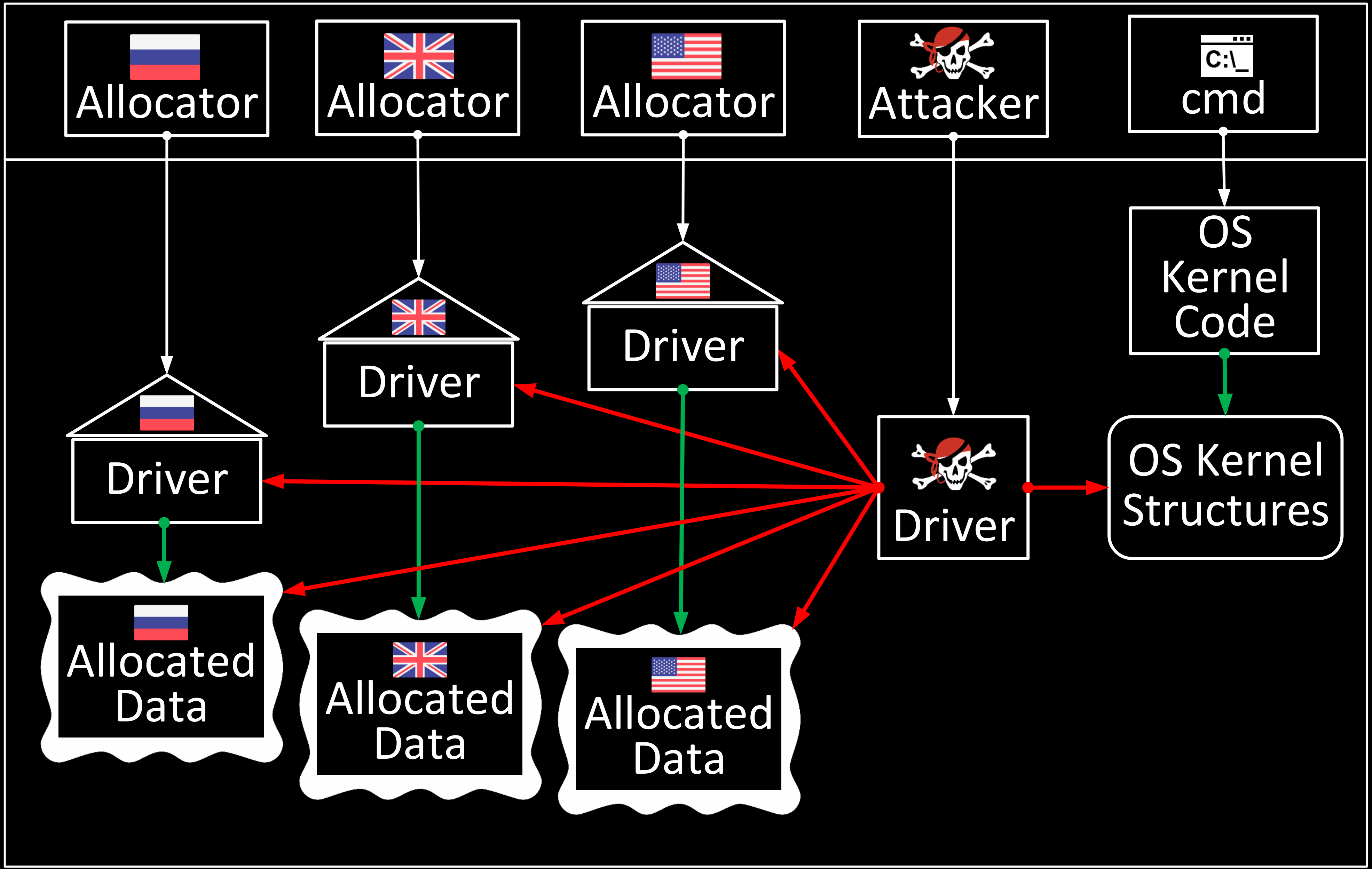
Stars by
Wassily Kandinsky

Girl with Balloon
by Banksy

No.5 by
Jackson Pollock

Oh no!
I can
steal
nothing!

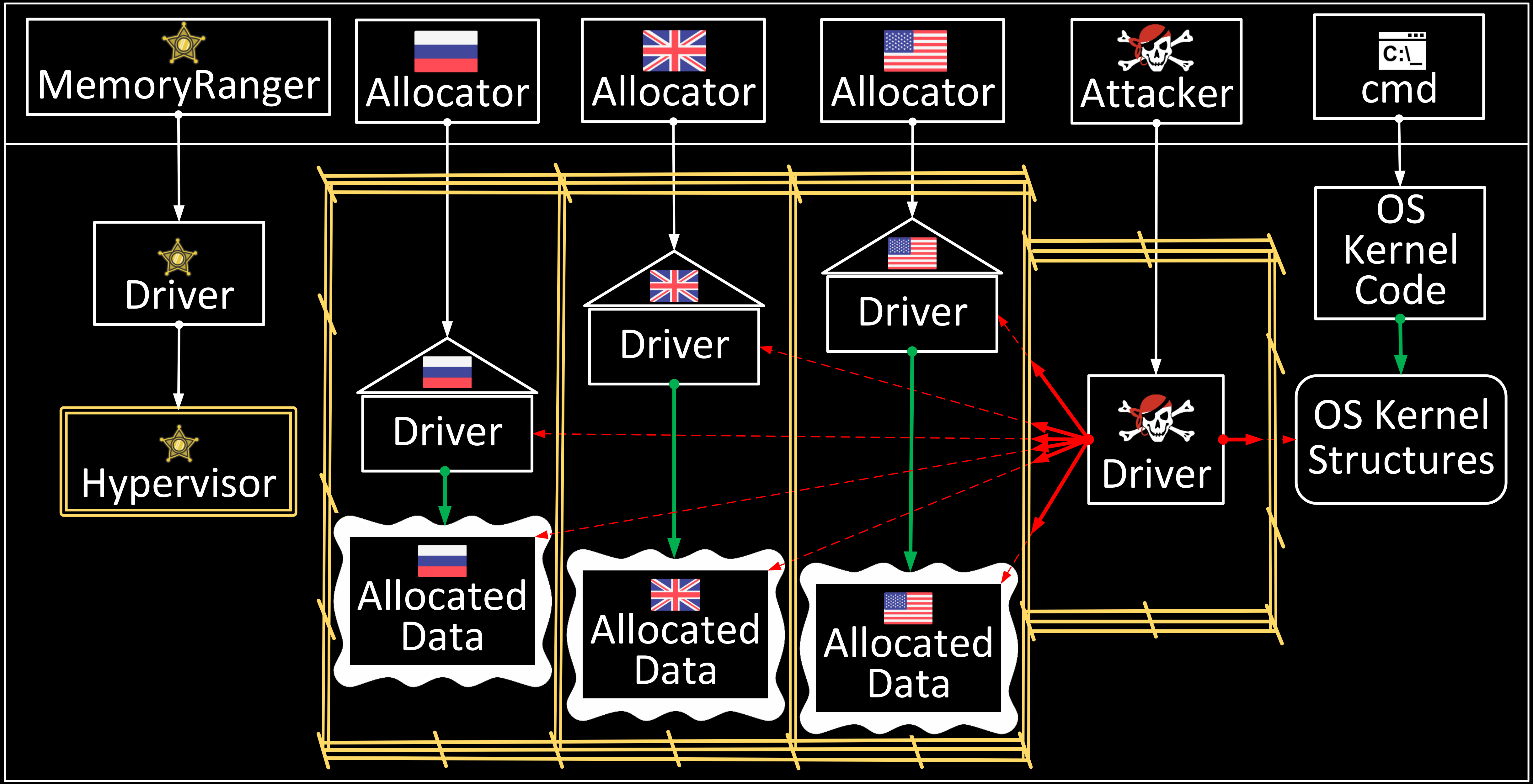
DEMO: THE ATTACK PREVENTION



DEMO: THE ATTACK PREVENTION

The online version is here – https://www.youtube.com/embed/vrm9cgn5DsU?vq=hd1080

DEMO: THE ATTACK PREVENTION



MEMORY RANGER: PRINCIPLE OF LEAST PRIVILEGE

Kernel-mode drivers	Drivers Code			
	🇷🇺	🇬🇧	🇺🇸	🏴‍☠️
🇷🇺	✓			
🇬🇧		✓		
🇺🇸			✓	
🏴‍☠️				✓
OS kernel	✓	✓	✓	✓

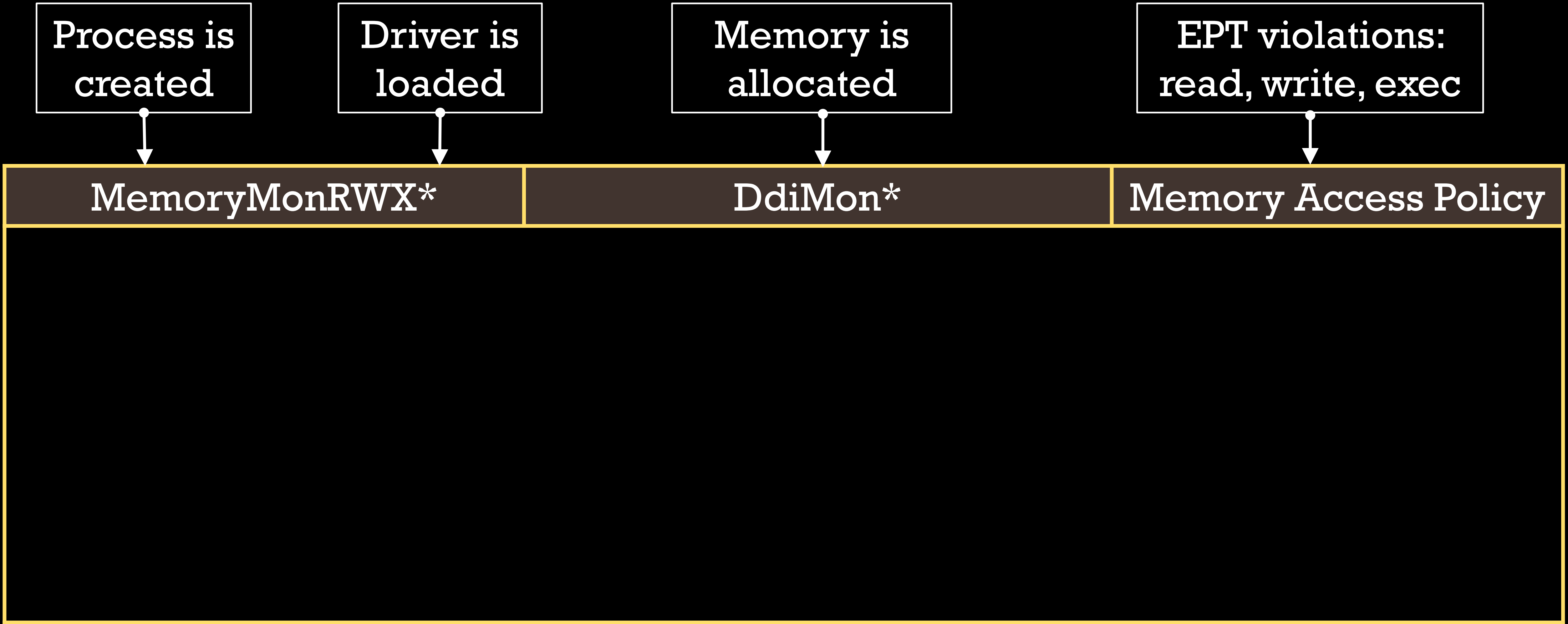
MEMORY RANGER: PRINCIPLE OF LEAST PRIVILEGE

Kernel-mode drivers	Drivers Code				Allocated Memory Data			
	🇷🇺	🇬🇧	🇺🇸	☠	🇷🇺	🇬🇧	🇺🇸	EPROCESS structures
🇷🇺	✓				✓			
🇬🇧		✓				✓		
🇺🇸			✓				✓	
☠				✓				
OS kernel	✓	✓	✓	✓	✓	✓	✓	✓

MEMORY RANGER: PRINCIPLE OF LEAST PRIVILEGE

Kernel-mode drivers	Drivers Code				Allocated Memory Data			
	🇷🇺	🇬🇧	🇺🇸	🏴‍☠️	🇷🇺	🇬🇧	🇺🇸	EPROCESS structures
🇷🇺	✓	✗	✗	✗	✓	✗	✗	✗
🇬🇧	✗	✓	✗	✗	✗	✓	✗	✗
🇺🇸	✗	✗	✓	✗	✗	✗	✓	✗
🏴‍☠️	✗	✗	✗	✓	✗	✗	✗	✗
OS kernel	✓	✓	✓	✓	✓	✓	✓	✓

MEMORY RANGER ARCHITECTURE: THREE PARTS



Process is created

Driver is loaded

Memory is allocated

EPT violations: read, write, exec

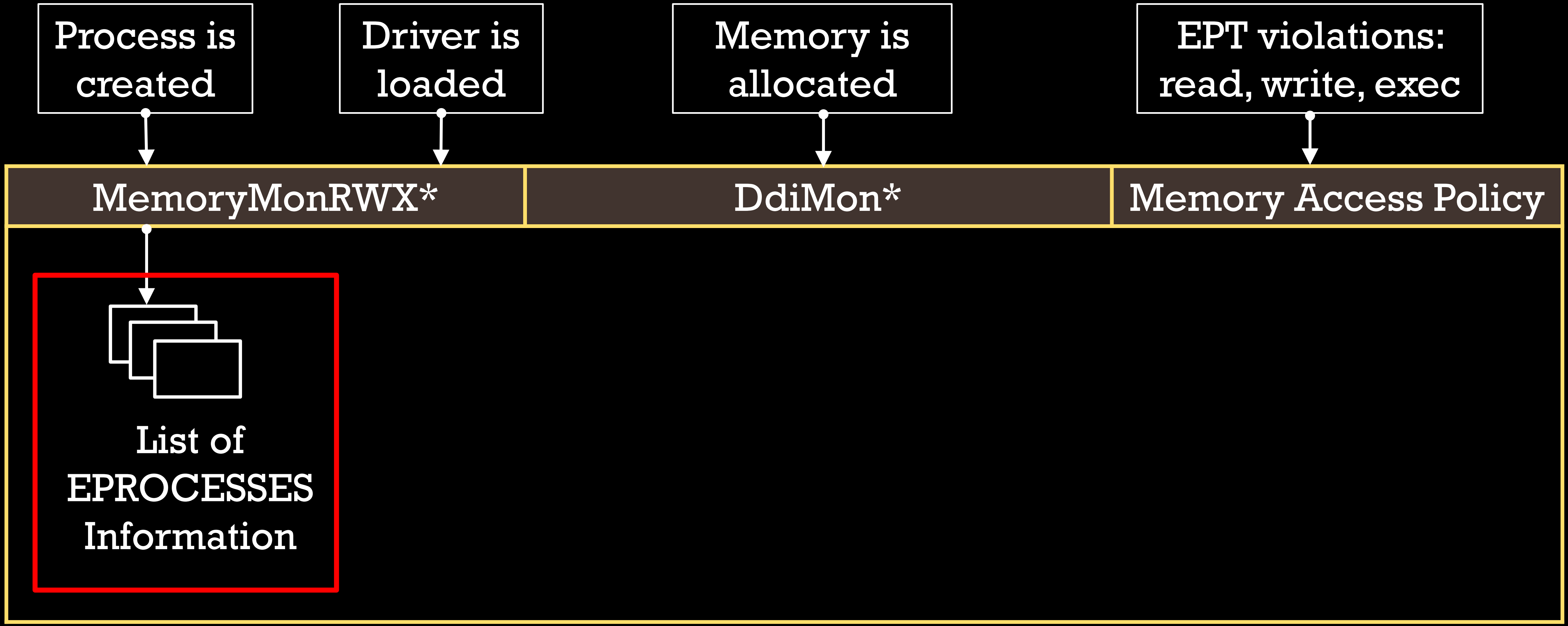
MemoryMonRWX*

DdiMon*

Memory Access Policy

* by Satoshi Tanda, @standa_t, https://github.com/tandasat

MEMORY RANGER ARCHITECTURE: THREE PARTS



Process is created

Driver is loaded

Memory is allocated

EPT violations: read, write, exec

MemoryMonRWX*

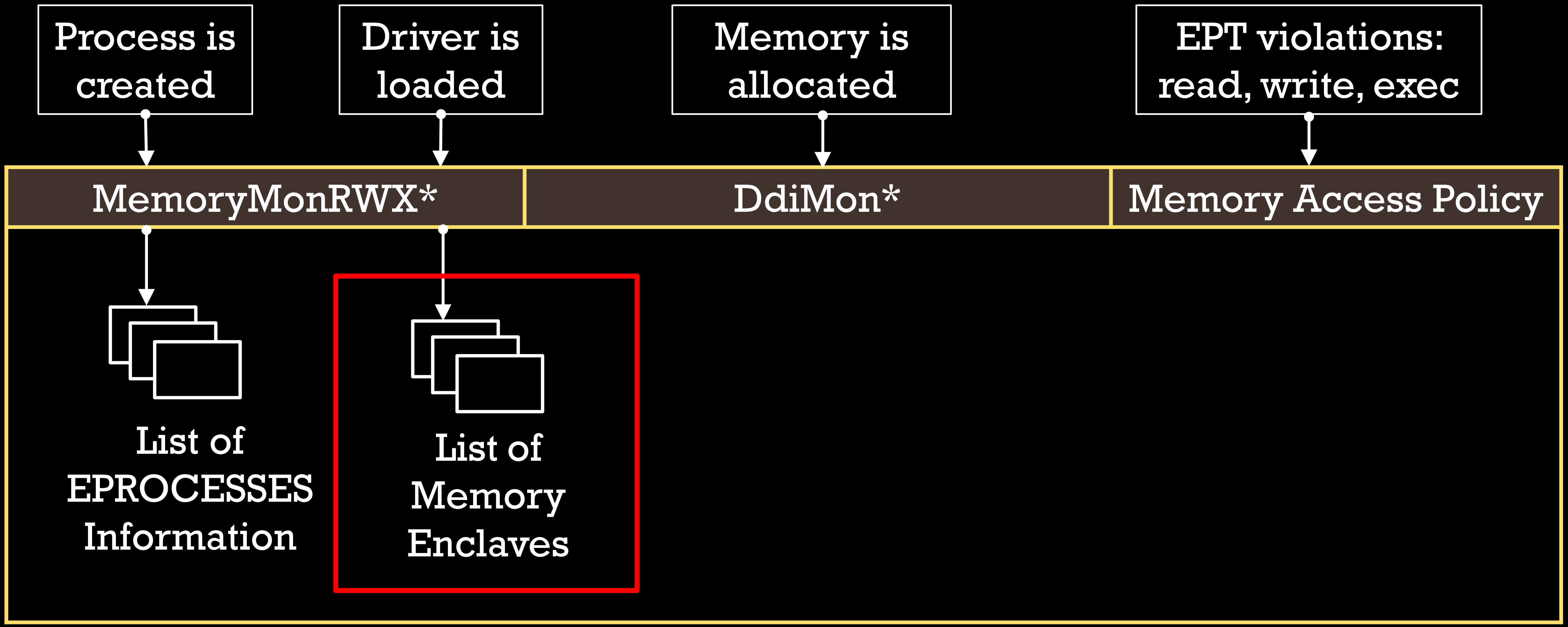
DdiMon*

Memory Access Policy

List of EPROCESSES Information

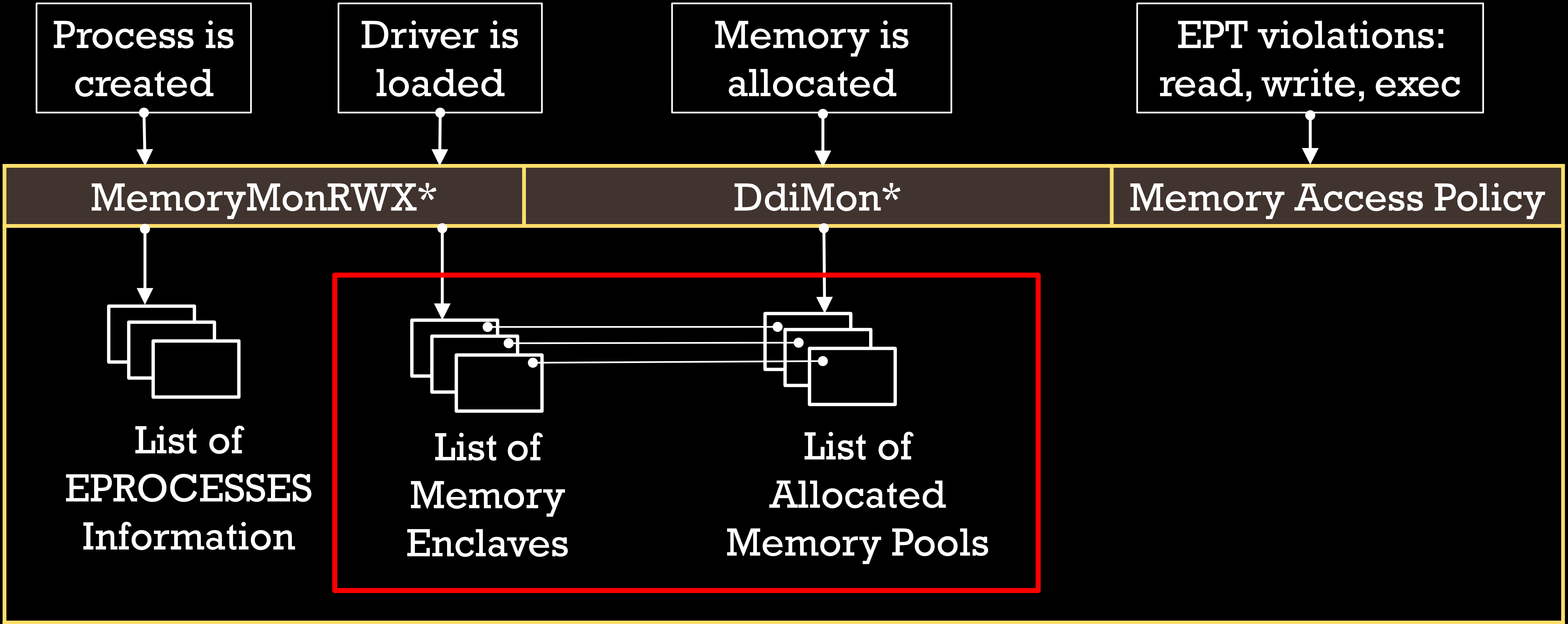
* by Satoshi Tanda, @standa_t, https://github.com/tandasat

MEMORY RANGER ARCHITECTURE: THREE PARTS



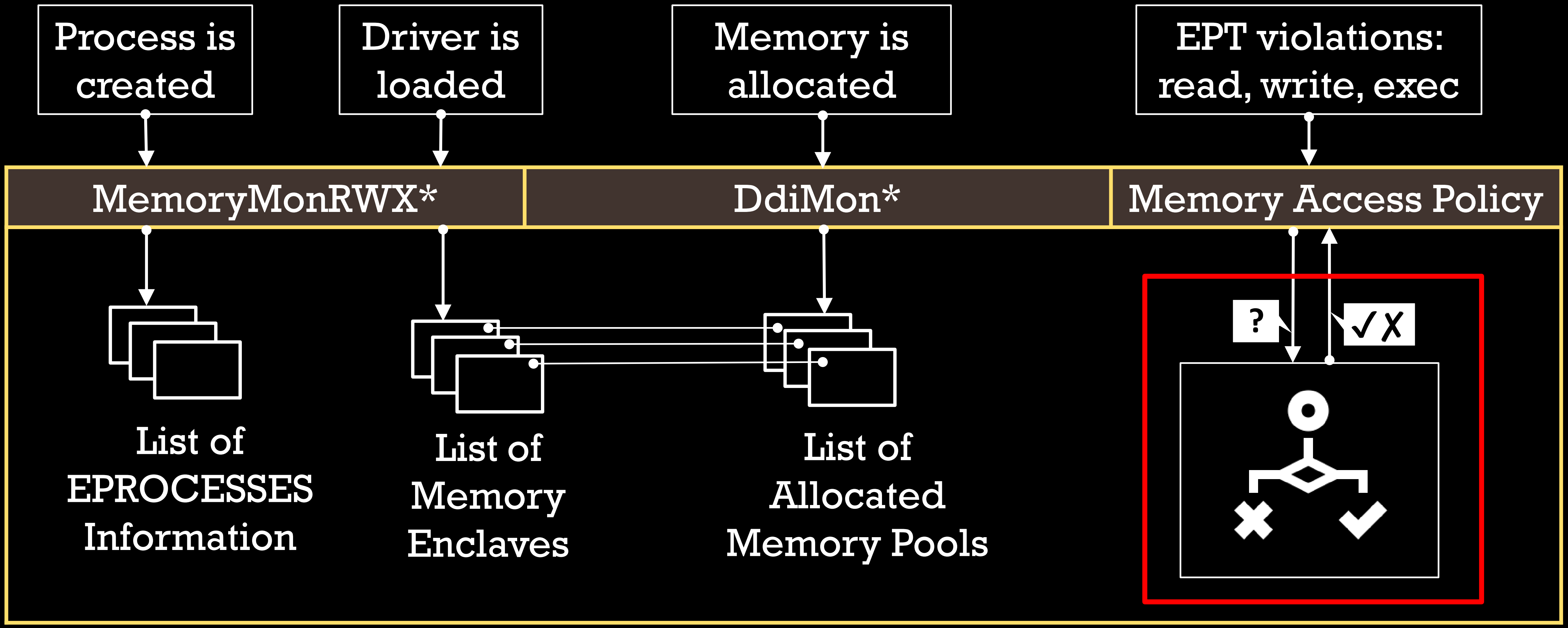
* by Satoshi Tanda, @standa_t, https://github.com/tandasat

MEMORY RANGER ARCHITECTURE: THREE PARTS



* by Satoshi Tanda, @standa_t, https://github.com/tandasat

MEMORY RANGER ARCHITECTURE: THREE PARTS



Process is created

Driver is loaded

Memory is allocated

EPT violations: read, write, exec

MemoryMonRWX*

DdiMon*

Memory Access Policy

List of EPROCESSES Information

List of Memory Enclaves

List of Allocated Memory Pools

* by Satoshi Tanda, @standa_t, https://github.com/tandasat

MEMORY RANGER DISPATCHER (SIMPLIFIED)

```
switch (exit_reason){
        case (execute_violation):
                change_ept();
                break;
        case (read_violation| | write_violation):
                if (access_legal()==false){
                        set_pte(pfn, read|write, fake_page);
                        set_monitor_trap_flag();
                        break;
                }
        case (monitor_trap_flag):
                set_pte(pfn, no_access, original_page);
                clear_monitor_trap_flag();
                break;
}
```

MEMORY RANGER DISPATCHER (SIMPLIFIED)

```
switch (exit_reason){
    case (execute_violation):
        change_ept();
        break;
    case (read_violation|| write_violation):
        if (access_legal()==false){
            set_pte(pfn, read|write, fake_page);
            set_monitor_trap_flag();
            break;
        }
    case (monitor_trap_flag):
        set_pte(pfn, no_access, original_page);
        clear_monitor_trap_flag();
        break;
}
```

MEMORY RANGER DISPATCHER (SIMPLIFIED)

```
switch (exit_reason){
    case (execute_violation):
        change_ept();
        break;
    case (read_violation|| write_violation):
        if (access_legal()==false){
            set_pte(pfn, read|write, fake_page);
            set_monitor_trap_flag();
            break;
        }
    case (monitor_trap_flag):
        set_pte(pfn, no_access, original_page);
        clear_monitor_trap_flag();
        break;
}
```

MEMORY RANGER DISPATCHER (SIMPLIFIED)

```
switch (exit_reason){
        case (execute_violation):
                change_ept();
                break;
        case (read_violation|| write_violation):
                if (access_legal()==false){
                        set_pte(pfn, read|write, fake_page);
                        set_monitor_trap_flag();
                        break;
                }
        case (monitor_trap_flag):
                set_pte(pfn, no_access, original_page);
                clear_monitor_trap_flag();
                break;
}
```

HOW TO PROTECT YOUR DATA IN MEMORY?

1. Callback - creating a list of protected objects
   - Add objects' addresses & sizes to the list
   - Restrict memory access for objects memory via EPT

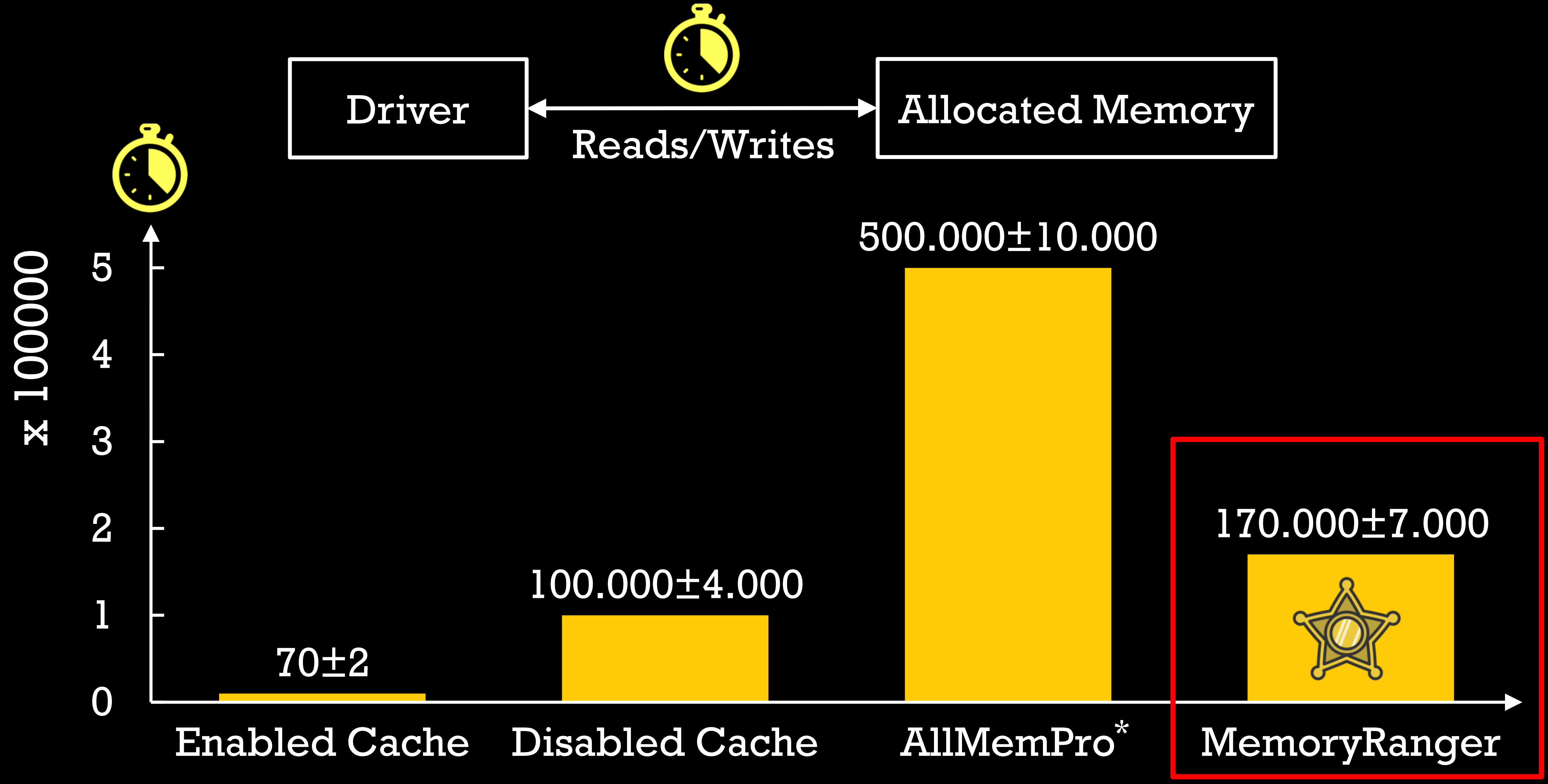
2. EPT dispatcher – processing EPT violations for this data
   - type_of_access – read or write
   - guest_ip is the 'source address'
   - fault_va is the 'destination address'
   - Temporary allow access to the data using MTF
   - Redirect access to the fake data using MTF and EPT.PFN

HOW TO PROTECT YOUR DATA IN MEMORY?

1. Callback - creating a list of protected objects
   - Add objects' addresses & sizes to the list
   - Restrict memory access for objects memory via EPT

2. EPT dispatcher – processing EPT violations for this data
   - type_of_access – read or write
   - guest_ip is the 'source address'
   - fault_va is the 'destination address'
   - Temporary allow access to the data using MTF
   - Redirect access to the fake data using MTF and EPT.PFN

MEMORY RANGER BENCHMARKS: MEMORY ACCESS TIME



Driver

Reads/Writes

Allocated Memory

* AllMemPro details - http://bit.ly/AllMemPro

BLACK HAT SOUND BYTES OR CONCLUSION

- Kernel-mode memory is out of control

- MemoryRanger isolates drivers execution by using a specific EPT structure for each driver

- MemoryRanger seems to prevent Spectre and Meltdown CPU attacks: research is ongoing

Dīvide et Imperā*

from Latin divide and rule

* Cartledge, P. (2013). Sparta and Lakonia: A regional history 1300-362 BC. Routledge.

Thank you!

Igor Korkin igor.korkin@gmail.com

All the details & my CV are here igorkorkin.blogspot.com