Detect Kernel-Mode Rootkits via Real Time Logging & Controlling Memory Access

Igor Korkin, PhD
Independent Researcher
Moscow, Russia

Satoshi Tanda
CrowdStrike, Inc
Vancouver, Canada

2017 CDFSL

The slides are here – www.bit.ly/MemoryMonRWX

We Protect the Computer Memory

Igor Korkin, PhD
Independent Researcher
Moscow, Russia

Satoshi Tanda
CrowdStrike, Inc
Vancouver, Canada

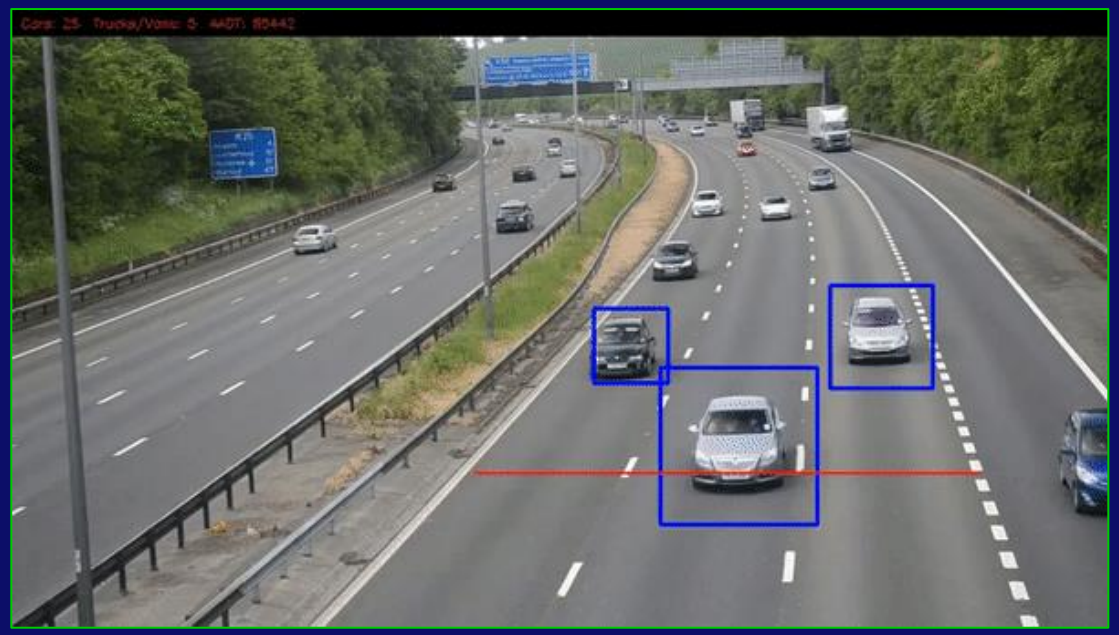
2017 CDFSL

The slides are here – www.bit.ly/MemoryMonRWX

- Igor Korkin, Ph.D.
- His 5 recent papers are double blind peer reviewed
- He has spoken at the ADFSL conferences since 2014

- Satoshi Tanda
- He has 7 years of experience in reverse engineering & Windows internals
- He spoke at the BlueHat v16, REcon 2011 and 2016

Memory accesses look like driving without rules



It is needed to control the memory accesses

Agenda

- Malware avoids detection: trends & experts' views
- Intercepting memory access attempts: methods & projects
- The new memory interceptor MemoryMonRWX: idea & prototype
- Demos
- Future plans with IoT & Digital Security

"... malware, or more specifically, a kernel rootkit, can often tamper with kernel memory data, putting the trustworthiness of memory analysis under question"[1]

1. Prakash, A., Venkataramani, E., Yin, H., & Lin, Z. (2015, October 31). On the Trustworthiness of Memory Analysis - An Empirical Study from the Perspective of Binary Execution, IEEE Transactions on Dependable and Secure Computing (TDSC), 12(5), 1545-5971, http://dx.doi.org/10.1109/TDSC.2014.2366464

What do we have now?

Windows security features	What do we have now?
Driver Signature Enforcement	
PatchGuard (Kernel Patch Protection)	

1. McAfee. (2016, September). Threats Report. McAfee Labs. Retrieved from http://www.mcafee.com/us/resources/reports/rp-quarterly-threats-sep-2016.pdf
2. Singh, A. (2015, April 8). Dissecting Turla Rootkit Malware Using Dynamic Analysis. Retrieved from https://www.lastline.com/labsblog/dissecting-turla-rootkit-malware-using-dynamic-analysis

What do we have now?

Windows security features	What do we have now?
Driver Signature Enforcement	3 million of signed malicious binaries[1] 
PatchGuard (Kernel Patch Protection)	New malware is able to bypass PatchGuard: • 'Turla' rootkit[2] • TDL4/TDSS

1. McAfee. (2016, September). Threats Report. McAfee Labs. Retrieved from http://www.mcafee.com/us/resources/reports/rp-quarterly-threats-sep-2016.pdf
2. Singh, A. (2015, April 8). Dissecting Turla Rootkit Malware Using Dynamic Analysis. Retrieved from https://www.lastline.com/labsblog/dissecting-turla-rootkit-malware-using-dynamic-analysis

Defeat and Protect PatchGuard

№	Pre-emptive Actions	Malware actions	Results & Comments
1		Rootkit is hiding the process	
2		Exploit is disabling PatchGuard Rootkit is hiding the process	
3	Memory protector limits memory access	Exploit is disabling PatchGuard Rootkit is hiding the process	

Demo 1

The online version is here –
https://www.youtube.com/embed/vi9TzLrO_pE?vq=hd1440

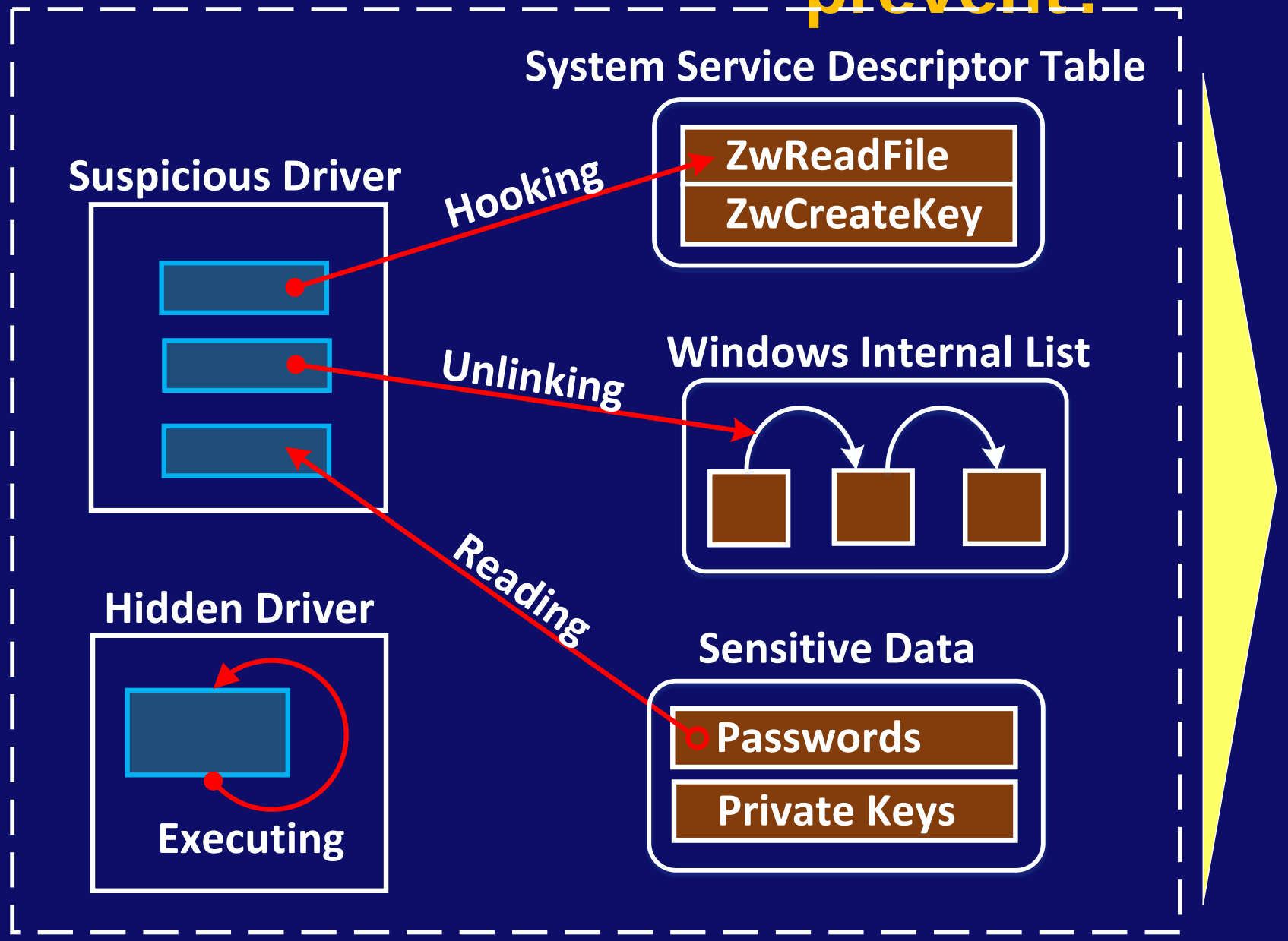
Demo 2

The online version is here –
https://youtu.be/vi9TzLrO_pE?t=70

Defeat and Protect PatchGuard

№	Pre-emptive Actions	Malware actions	Results & Comments
1		Rootkit is hiding the process	**OS has crashed** ✗ (PatchGuard has generated 0x109 BSOD)
2		Exploit is disabling PatchGuard Rootkit is hiding the process	**OS has been infected** ✗ (PatchGuard has been disabled, no BSOD)
3	Memory protector limits memory access	Exploit is disabling PatchGuard Rootkit is hiding the process	**OS has been protected** ✓ (Exploit has failed)

What malware attacks do we want to monitor & prevent?



Control memory accesses:

- Reading
- Writing
- Executing

Memory Interceptor Requirements

1) All types of memory accesses: read, write, execute

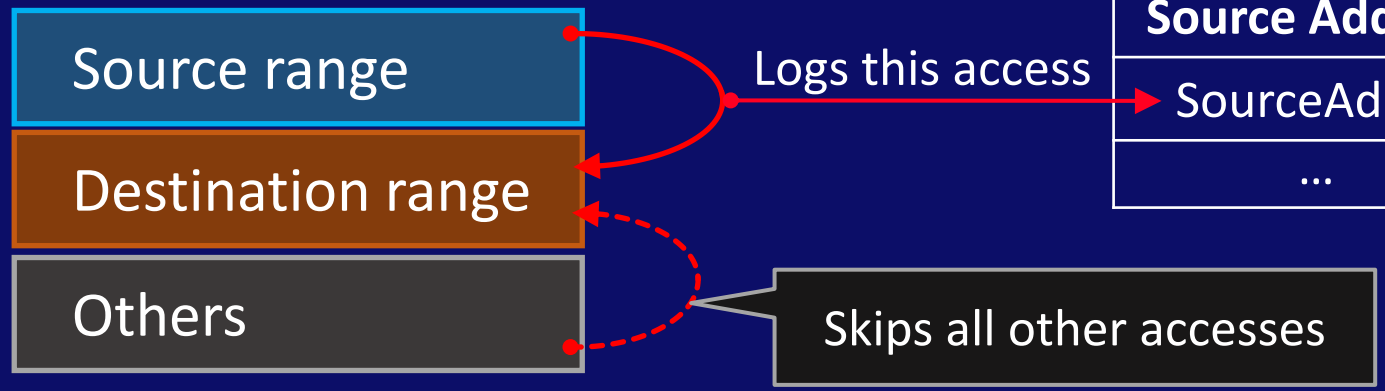
2) Triples for each access:

Memory Interceptor Requirements

1) All types of memory accesses: read, write, execute

2) Triples for each access:

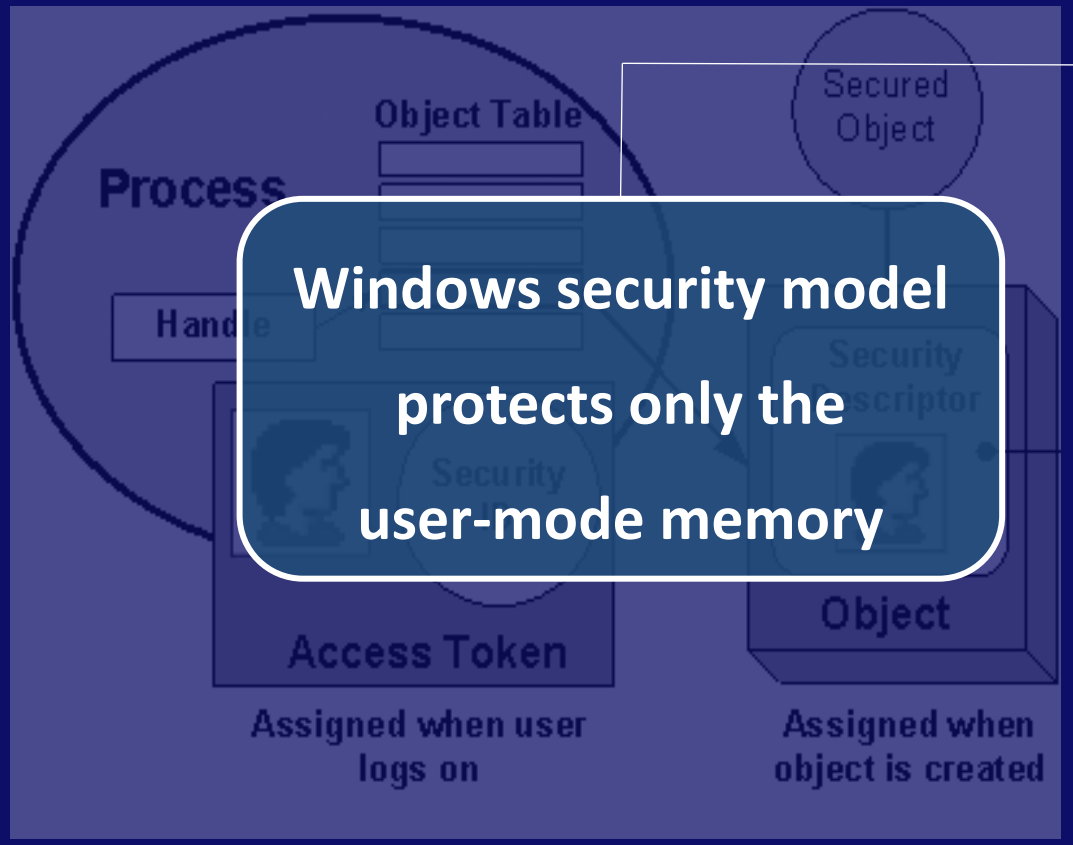
3) Access only from Source range → Destination range:



Source Address	Access Type	Destination Address
SourceAddr1	read	DestinationAddr1
…	…	…

4) A kernel-mode driver, which supports Windows 10 x64 and multi-core CPUs

What can we use as a basis for the memory interceptor?



Windows security model protects only the user-mode memory

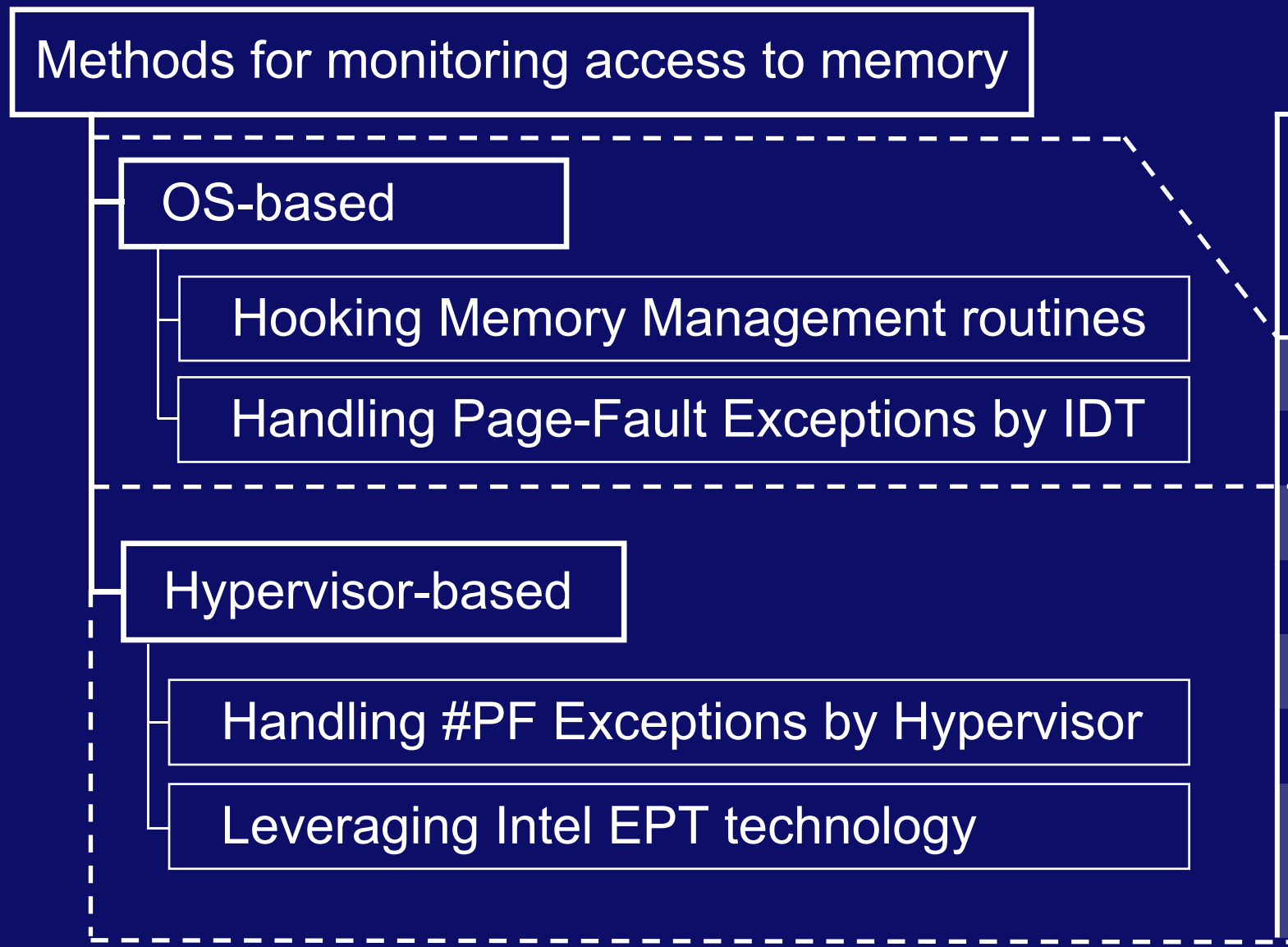
Memory monitoring methods based on OS & hypervisor facilities

➔ There is no build-in tools for controlling kernel mode memory

Intercepting memory access: methods & projects

- Methods for monitoring access to memory
  - OS-based
    - Hooking Memory Management routines
    - Handling Page-Fault Exceptions by IDT
  - Hypervisor-based
    - Handling #PF Exceptions by Hypervisor
    - Leveraging Intel EPT technology

Intercepting memory access: methods & projects



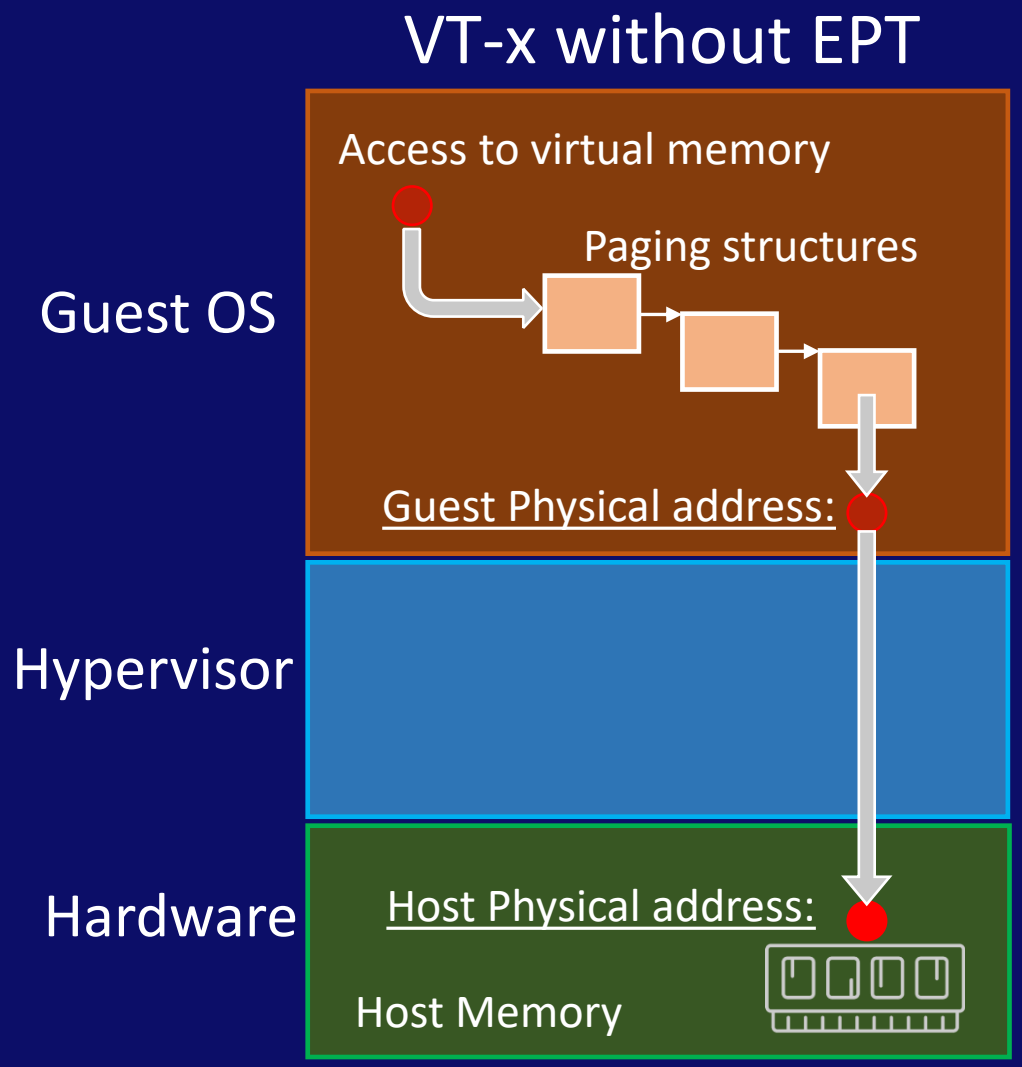
Project title, year	Read/ Write/ Execute
SPIDER, 2013	+/+/–
SecVisor, 2007	–/+/+
HyperSleuth, 2010	+/–/–
CXPInspector, 2013	–/–/+
HyperTap, 2014	–/+/+
DRAKVUF, 2014	–/–/+
MemoryMonRWX, 2017 (The proposed system)	+ / + / +

New Advanced Technology: Intel VT-x with Extended Page Tables (EPT)

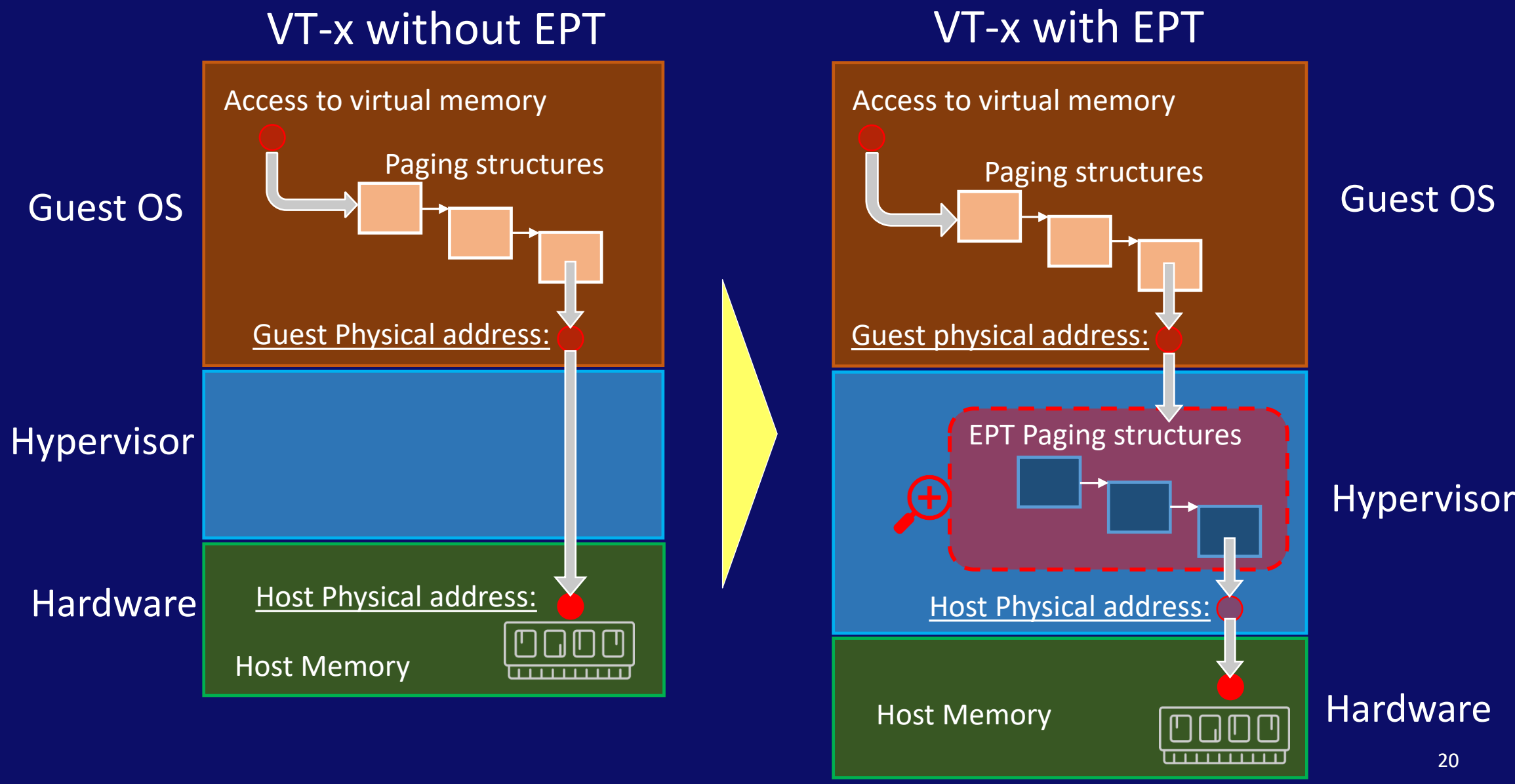
- EPT Overview
- EPT paging structures
- Applying EPT to monitor & limit memory access

EPT plays the role of traffic lights for memory accesses

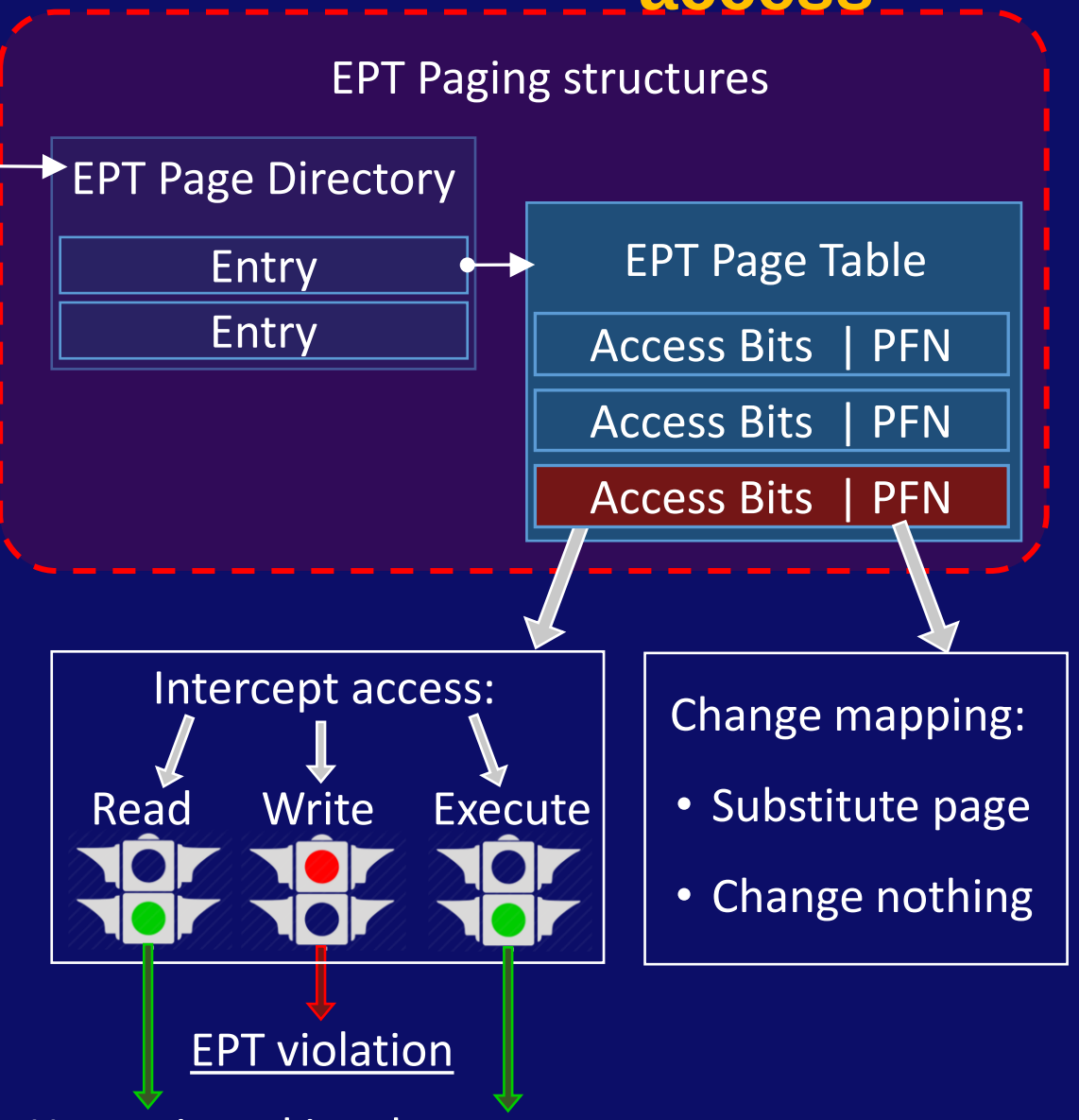
Processing memory access: VT-x vs. VT-x with EPT



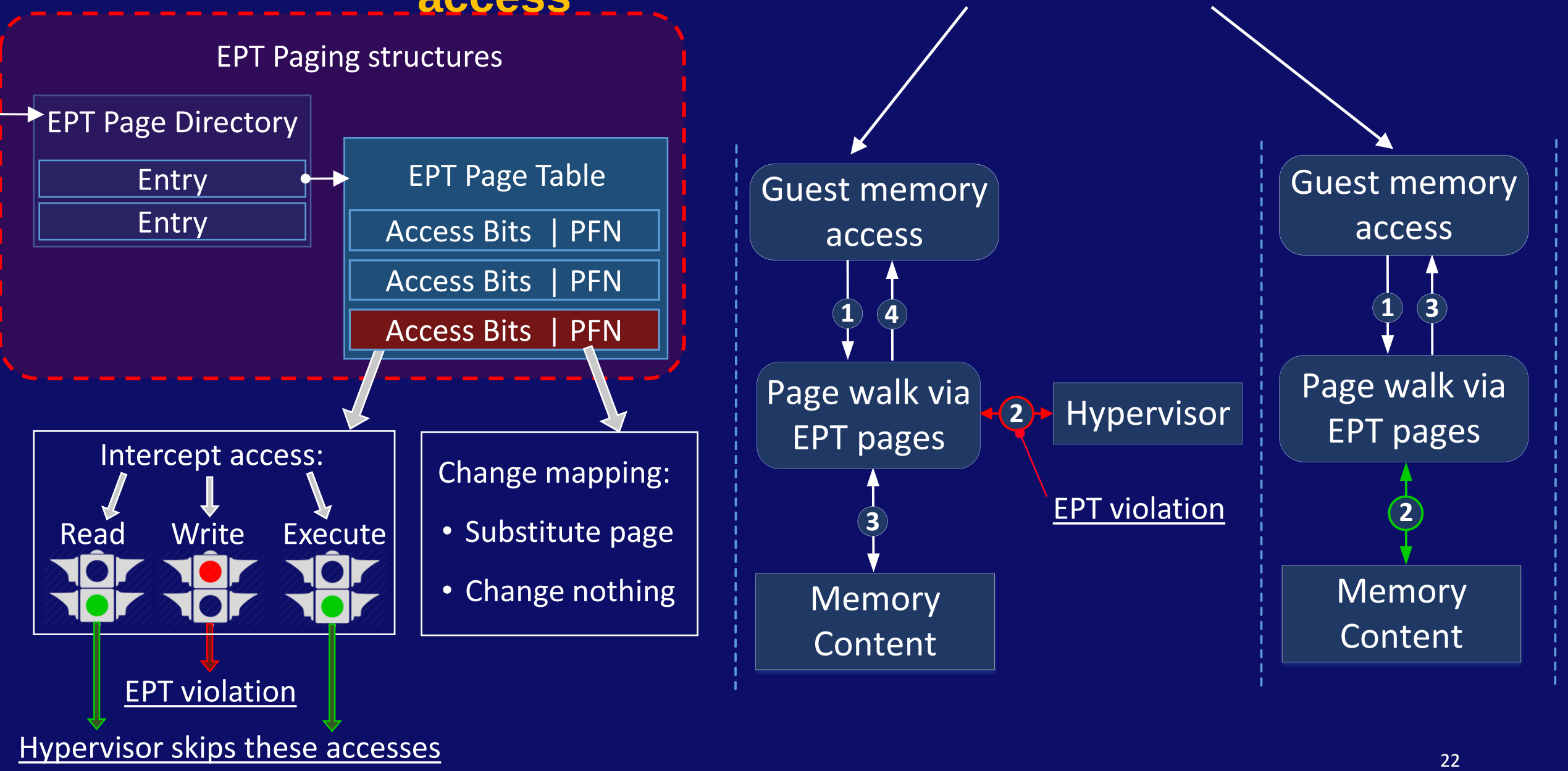
Processing memory access: VT-x vs. VT-x with EPT



Applying EPT features to trap and skip memory access

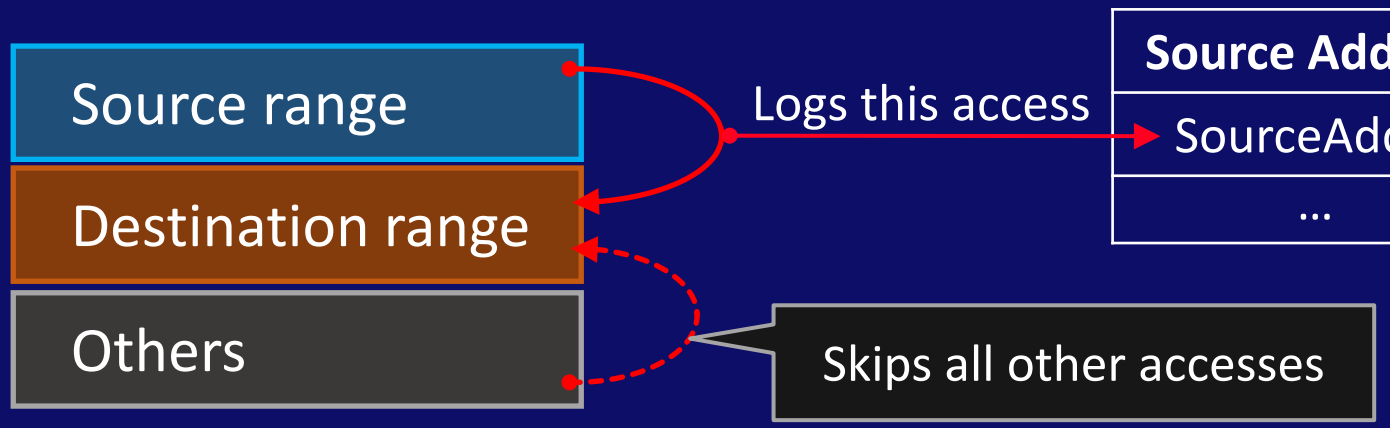


Applying EPT features to trap and skip memory access



How to apply EPT to monitor access only

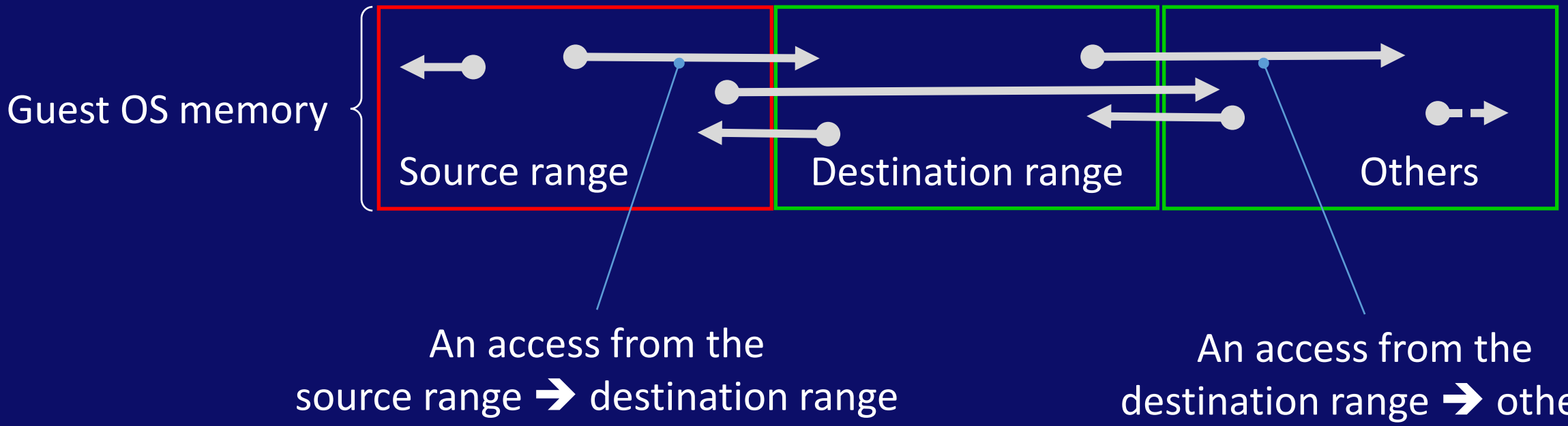
- from Source range to Destination range
- and skip all the rest?



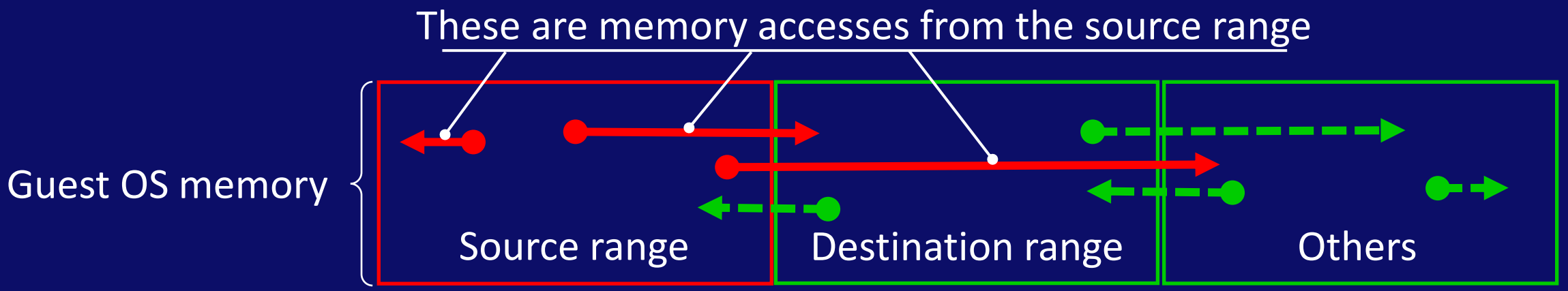
Source Address	Access Type	Destination Address
SourceAddr1	read	DestinationAddr1
…	…	…

We propose the following 5 steps

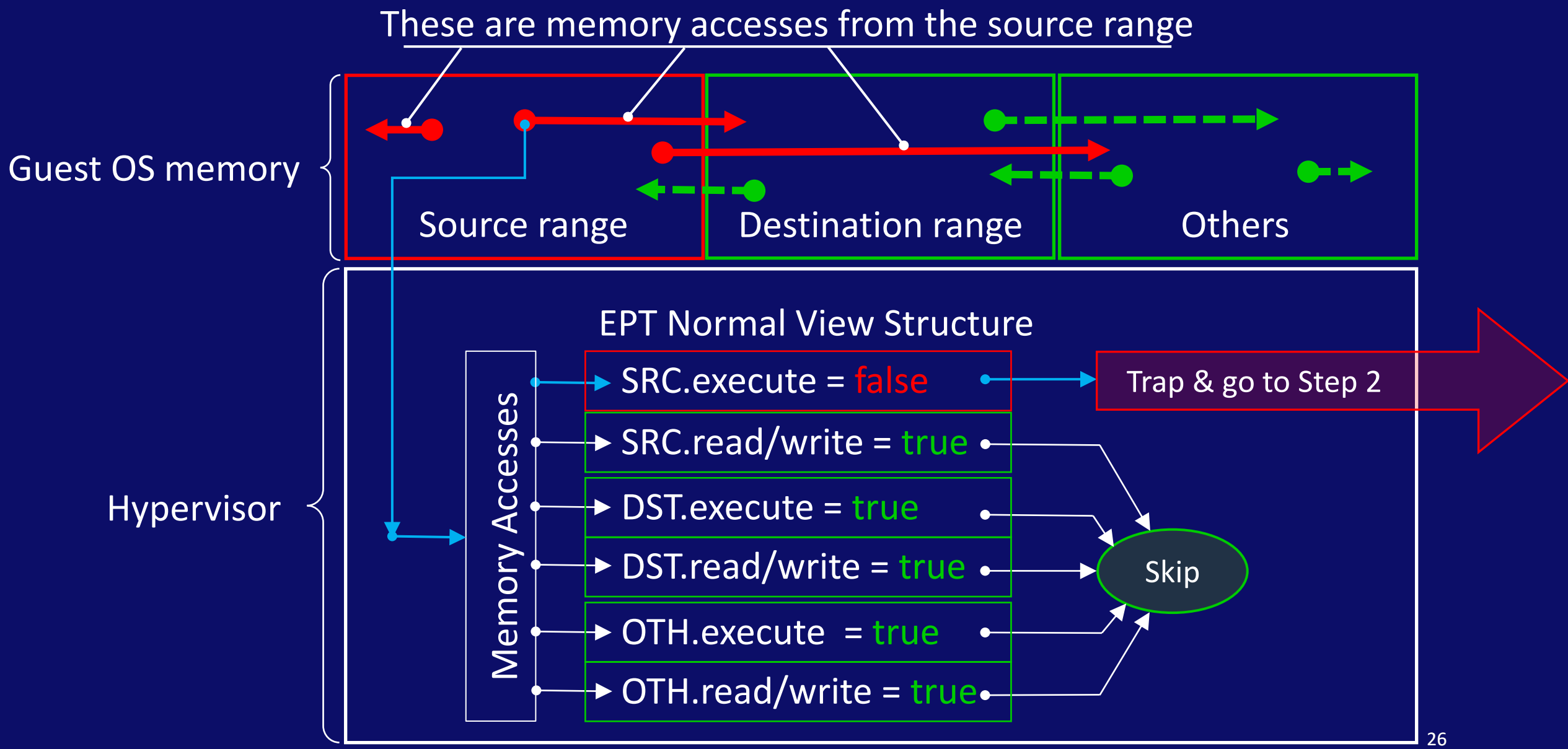
Step 1. Trapping execution on Source range



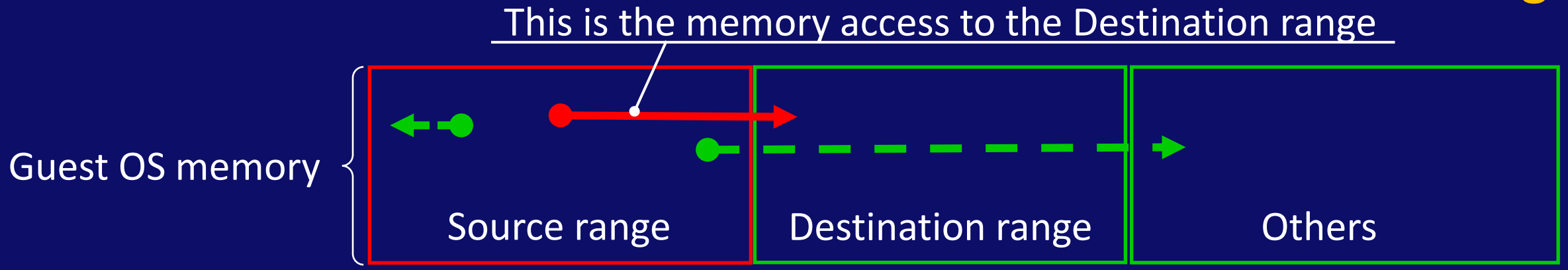
Step 1. Trapping execution on Source range



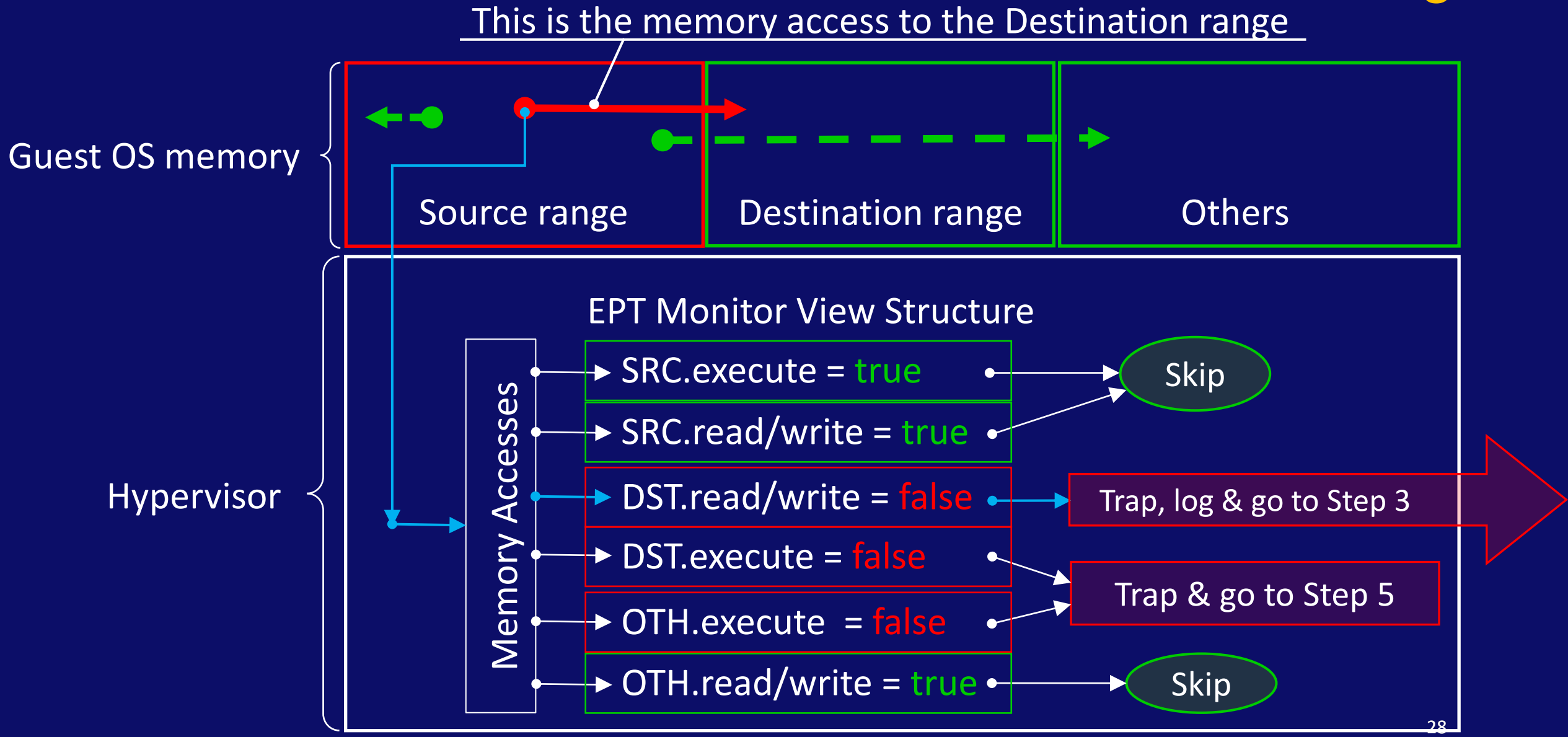
Step 1. Trapping execution on Source range



Step 2. Process VM-Exit to separate access to the Destination range



Step 2. Process VM-Exit to separate access to the Destination range

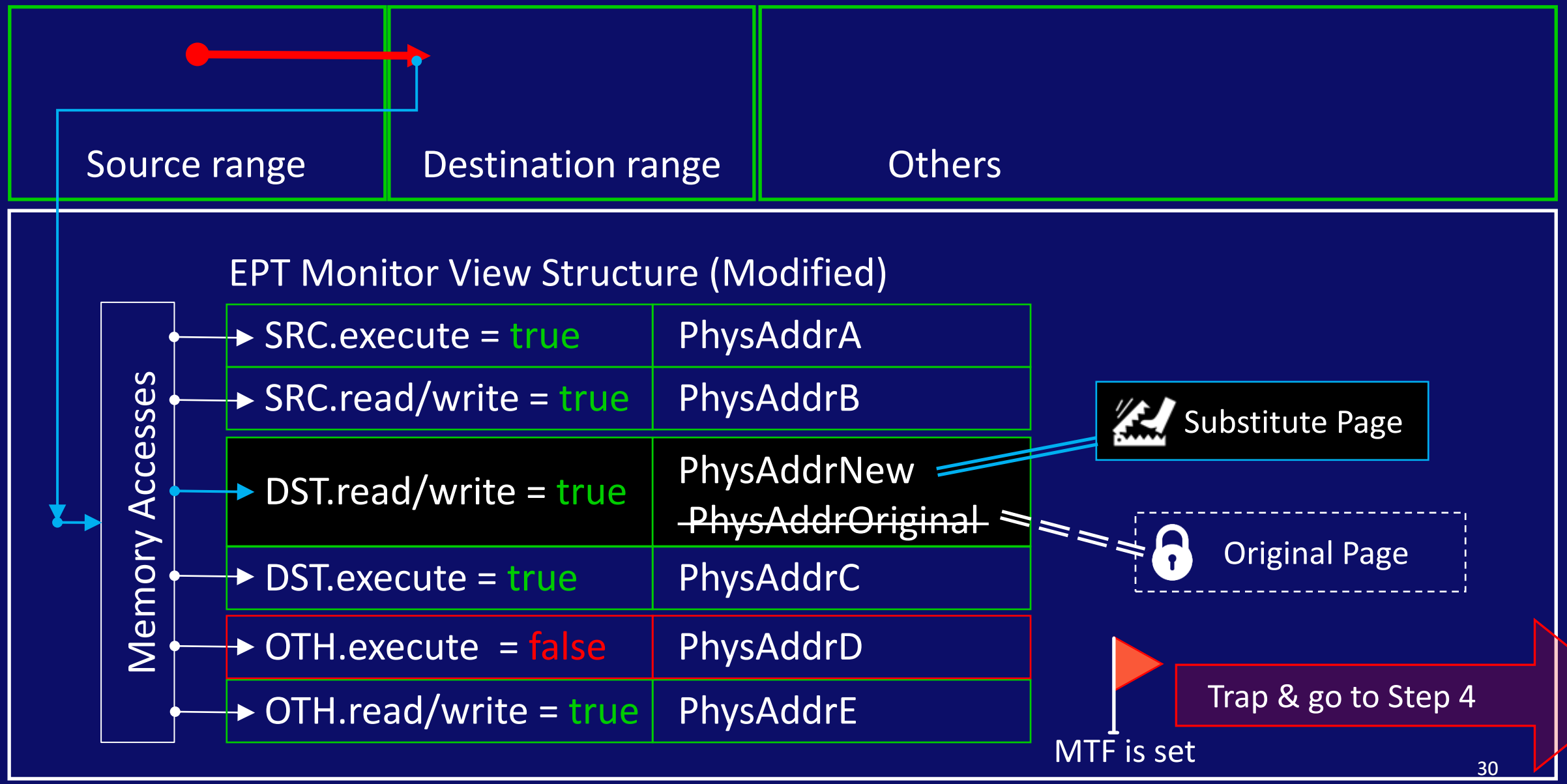


Step 3. Process VM-Exit, because of access on Destination range

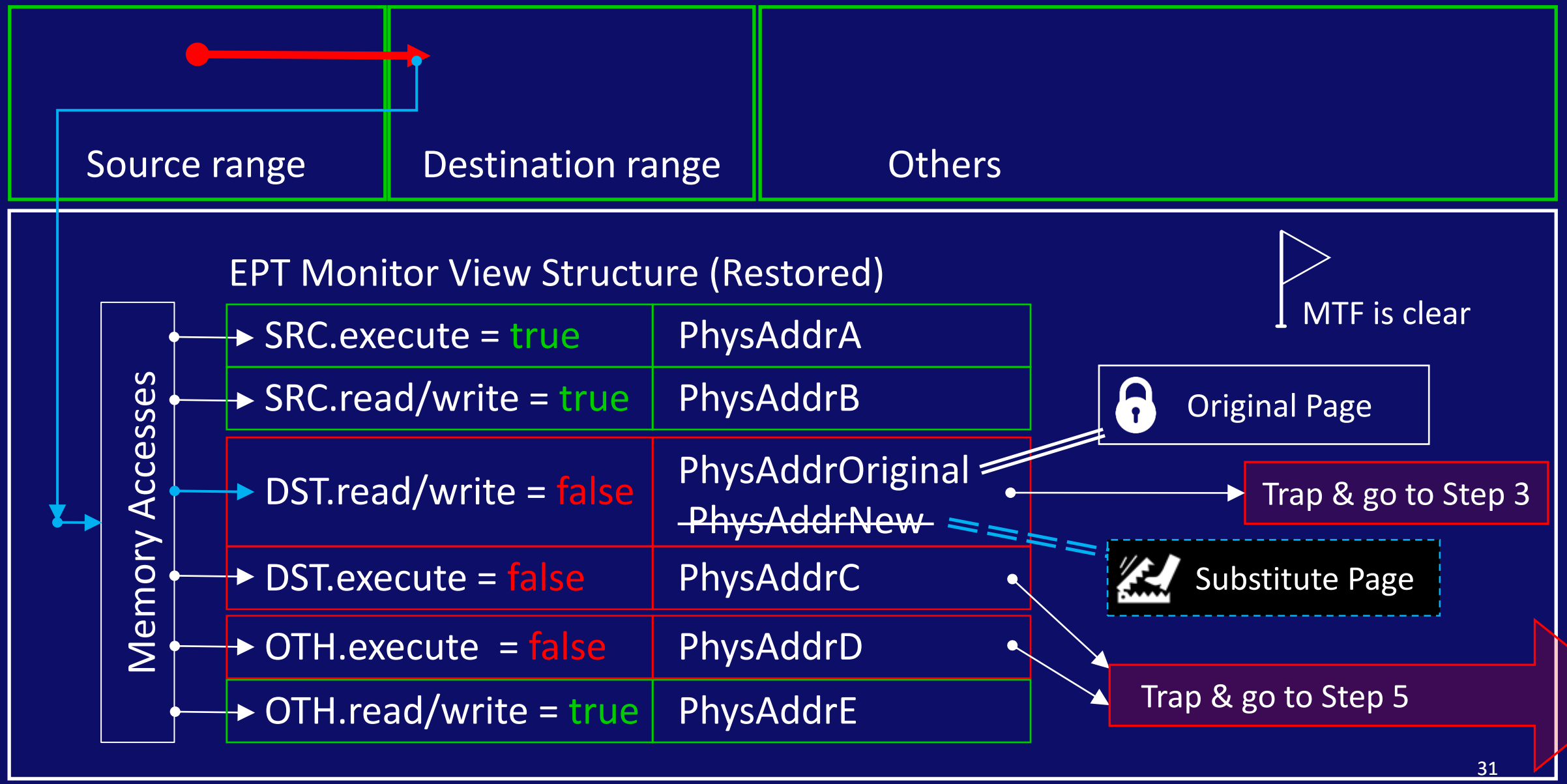
Logs this access

Source Address	Access Type	Destination Address
SourceAddr1	read	DestinationAddr1
…	…	…

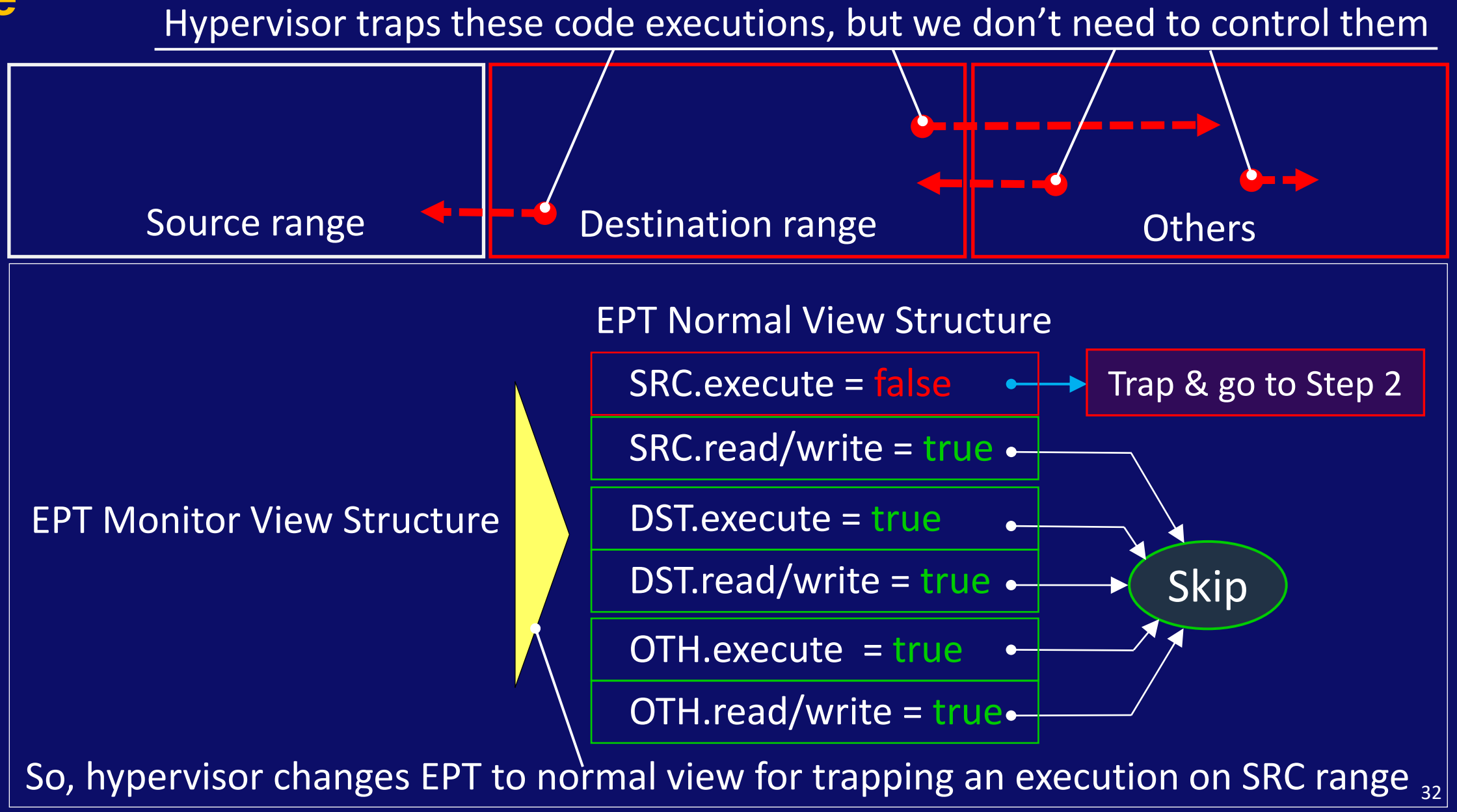
Step 3. Process VM-Exit, because of access on Destination range



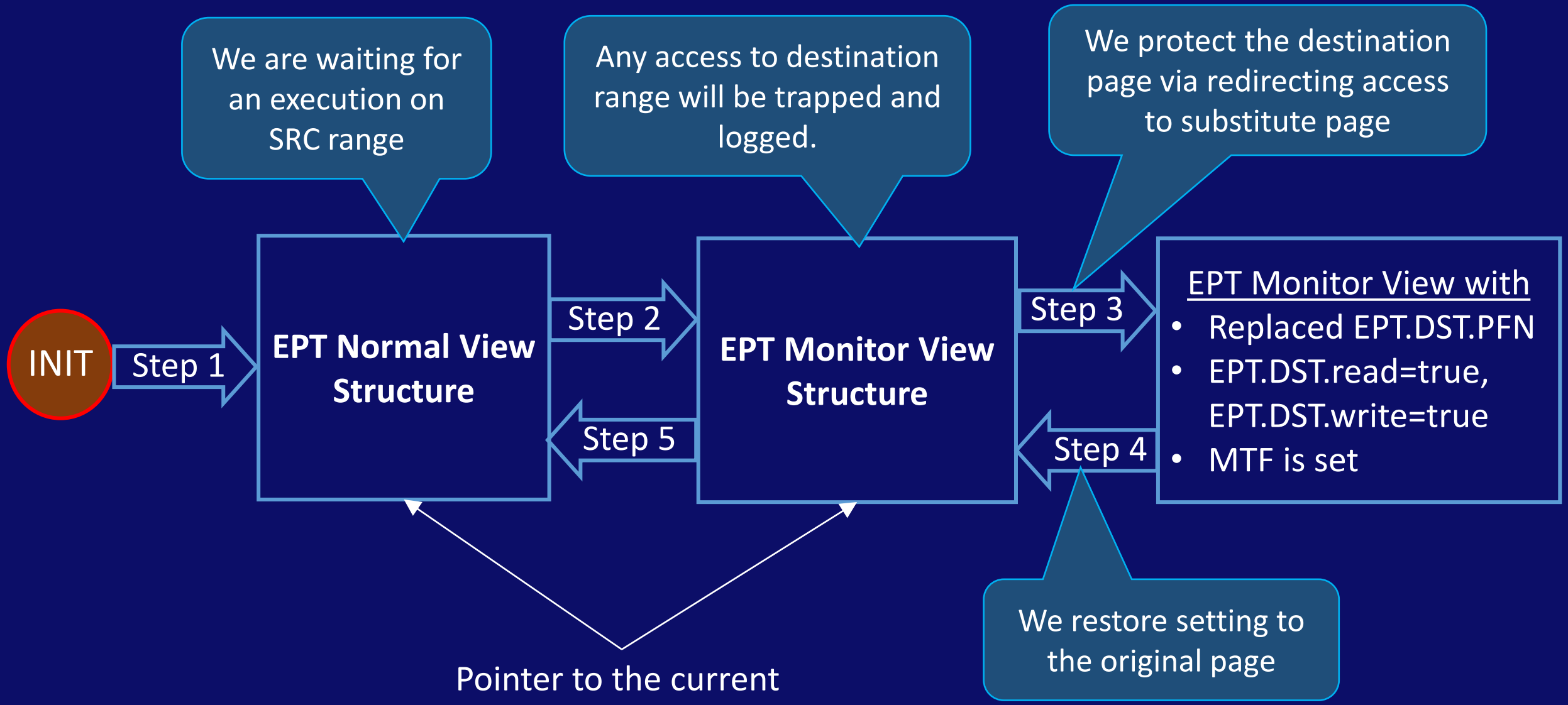
Step 4 (Restore setting). Process VM-Exit, because of MTF



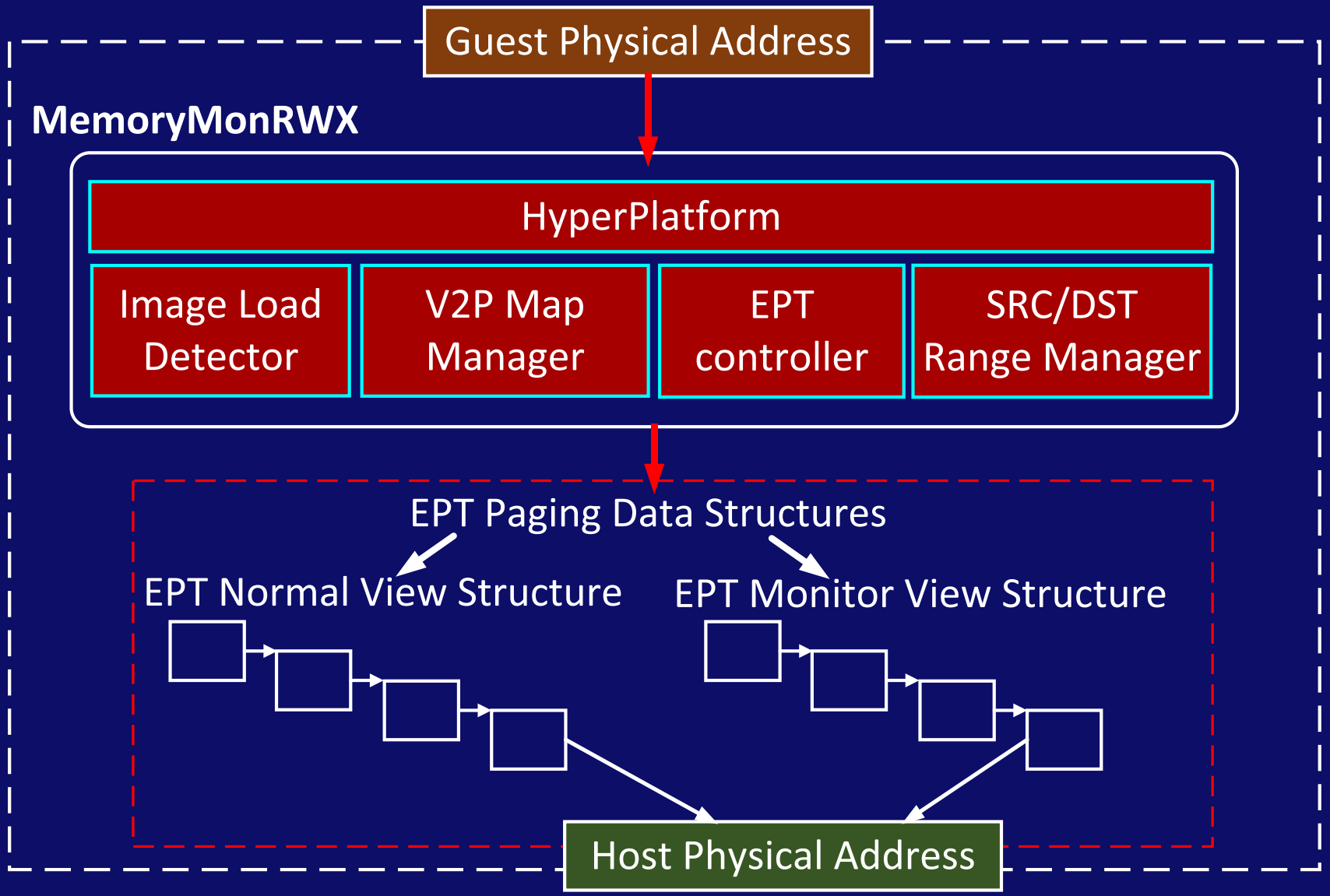
Step 5. Process VM-Exit, because of execution on Destination range



Five steps together

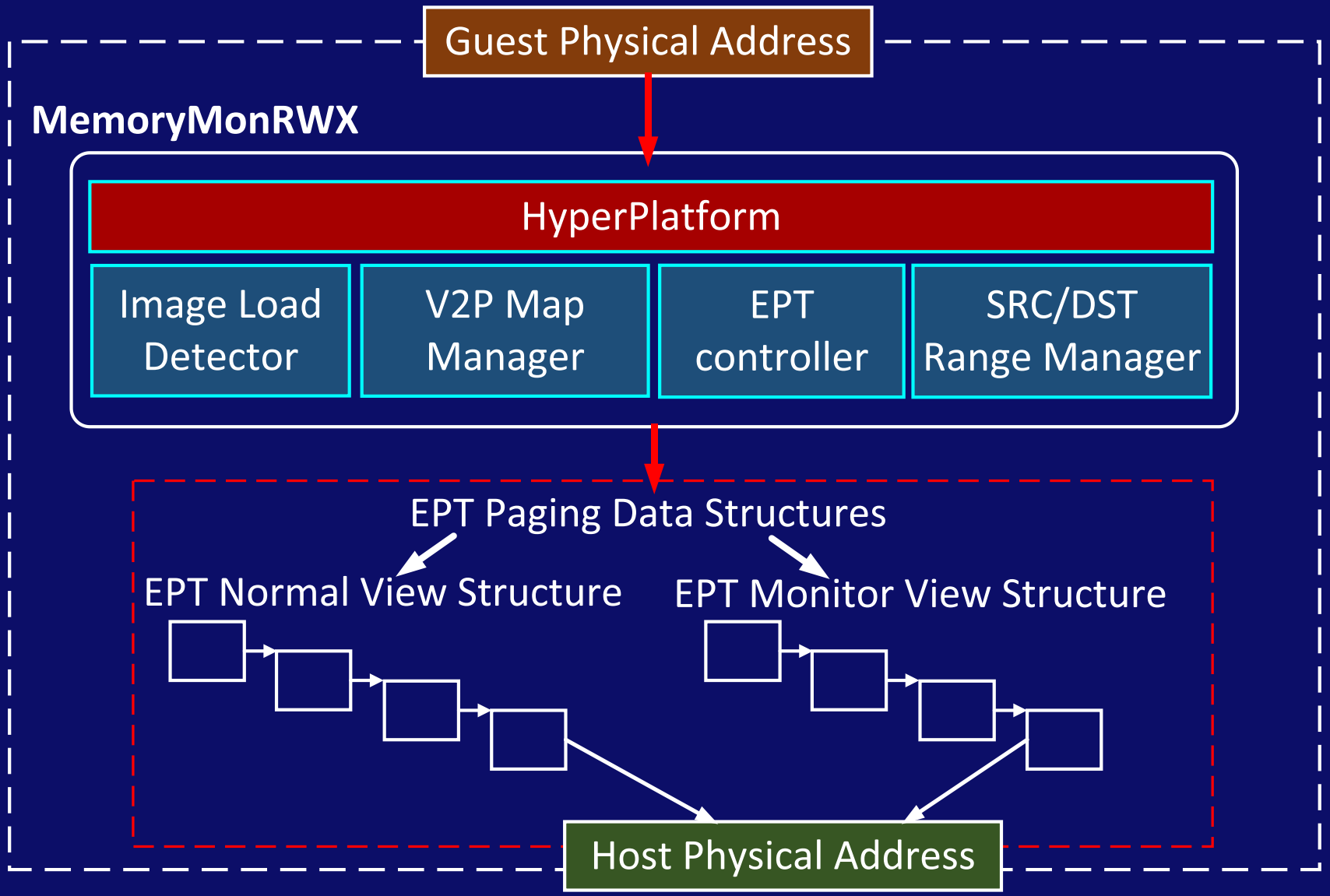


MemoryMonRWX architecture



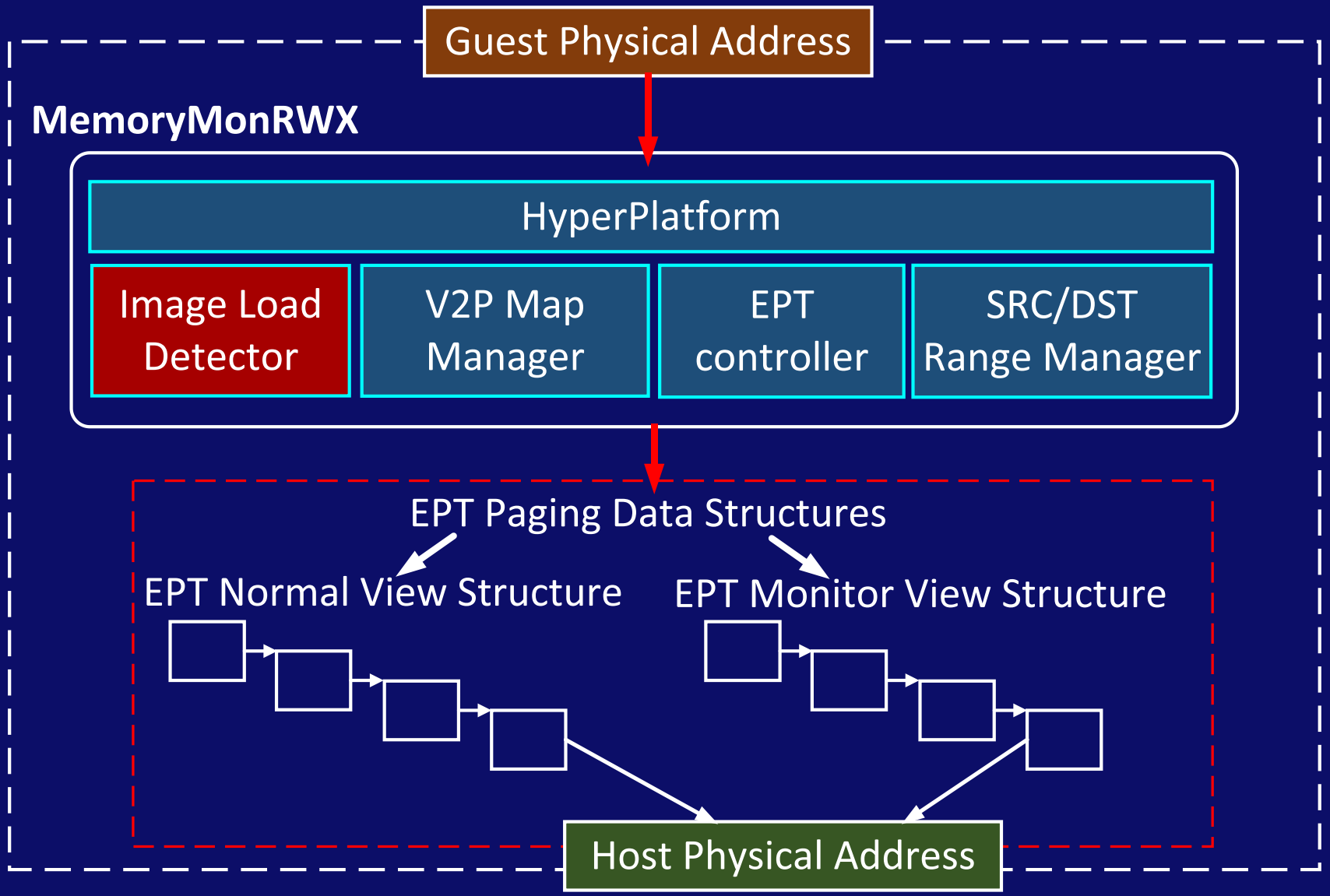
The source code is here - http://bit.ly/MemoryMonRWX

MemoryMonRWX architecture



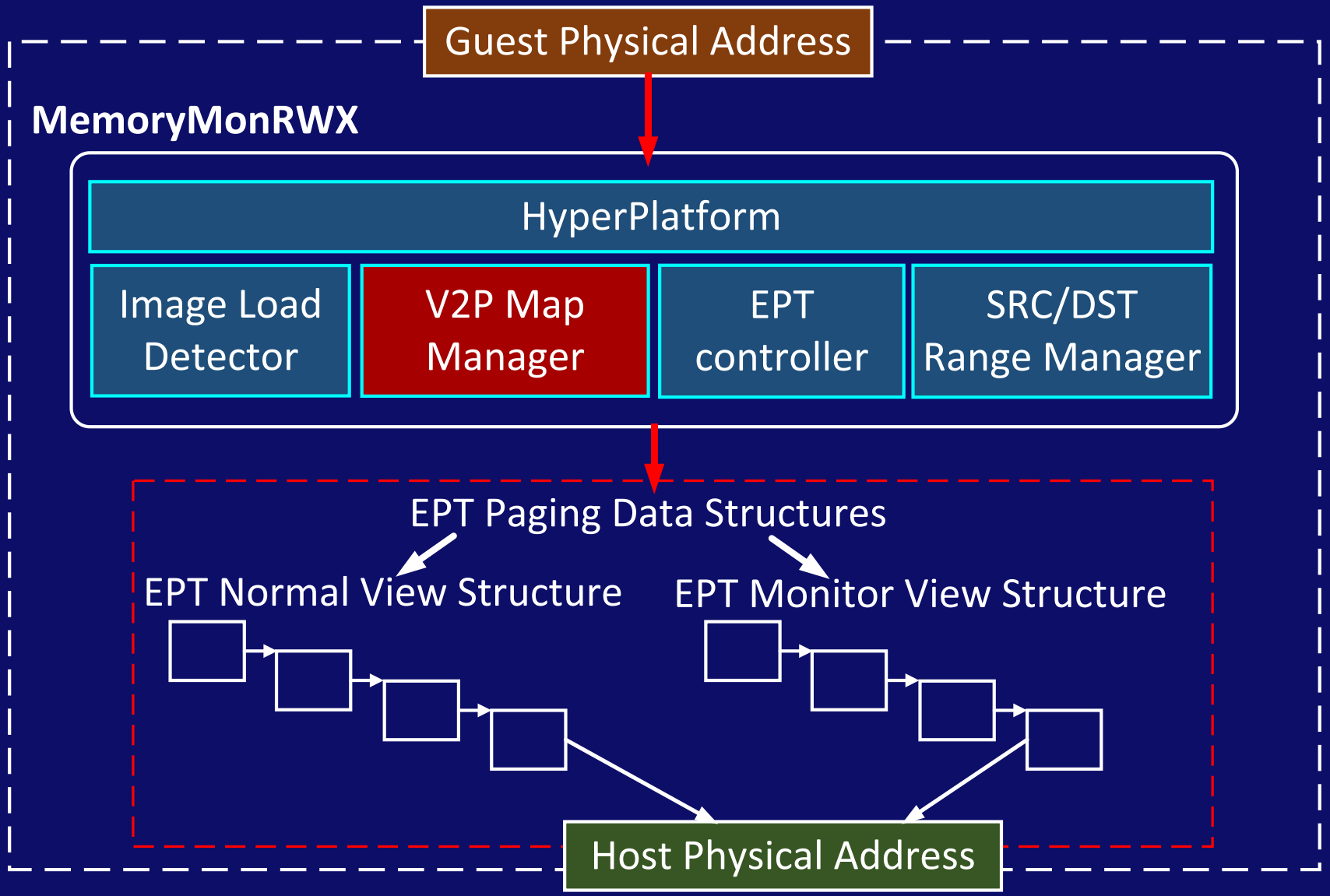
The source code is here - http://bit.ly/MemoryMonRWX

MemoryMonRWX architecture



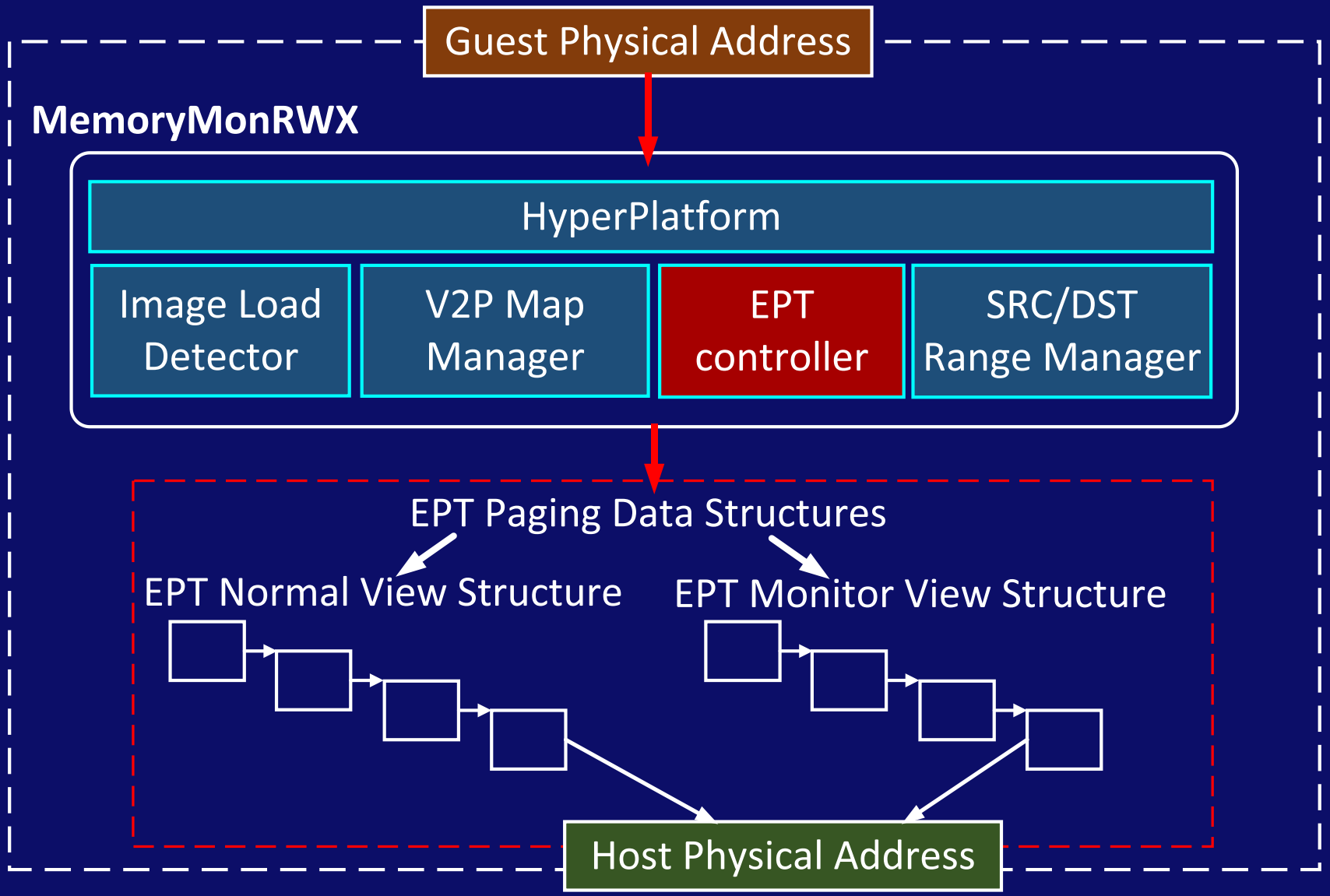
The source code is here - http://bit.ly/MemoryMonRWX

MemoryMonRWX architecture



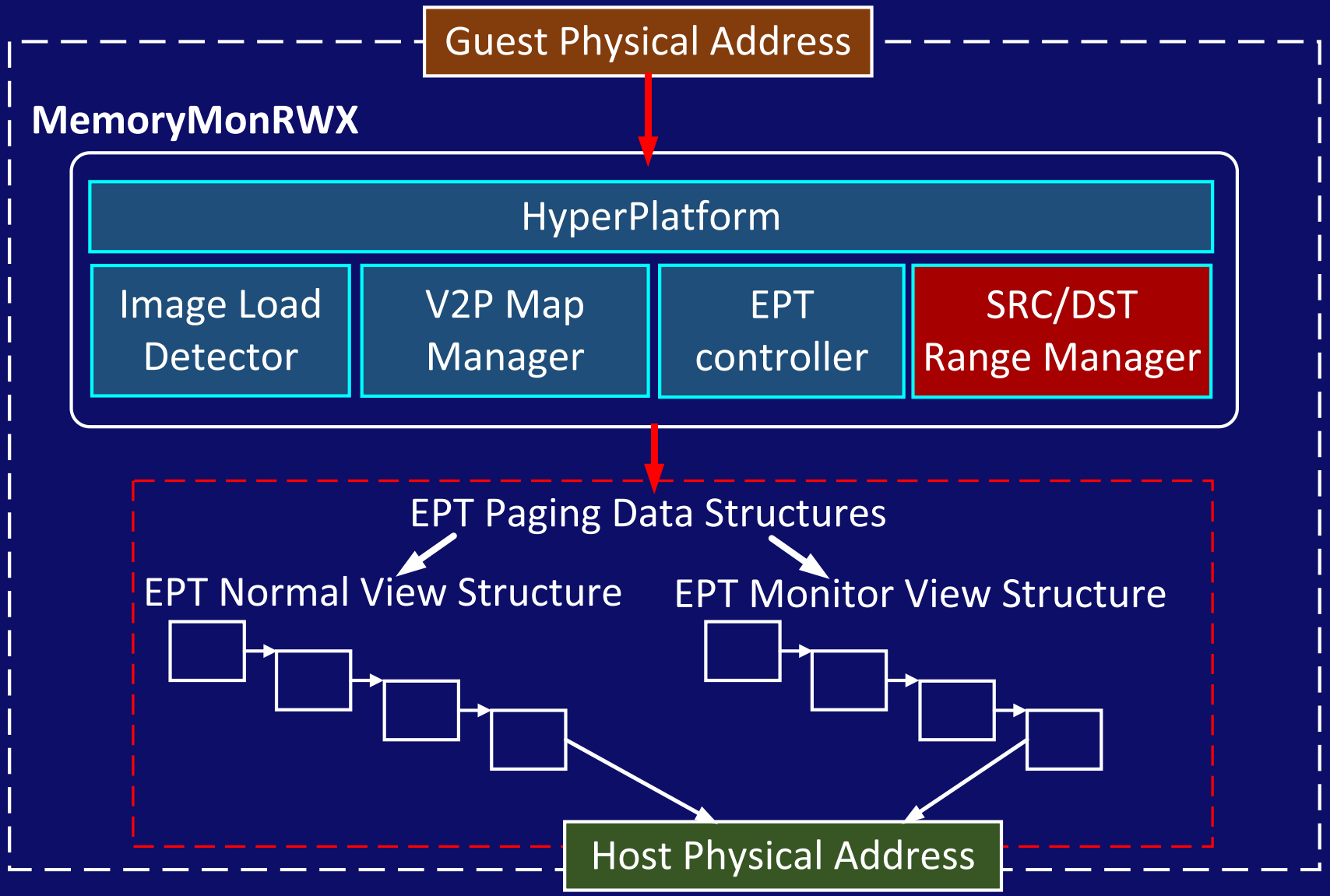
The source code is here - http://bit.ly/MemoryMonRWX

MemoryMonRWX architecture



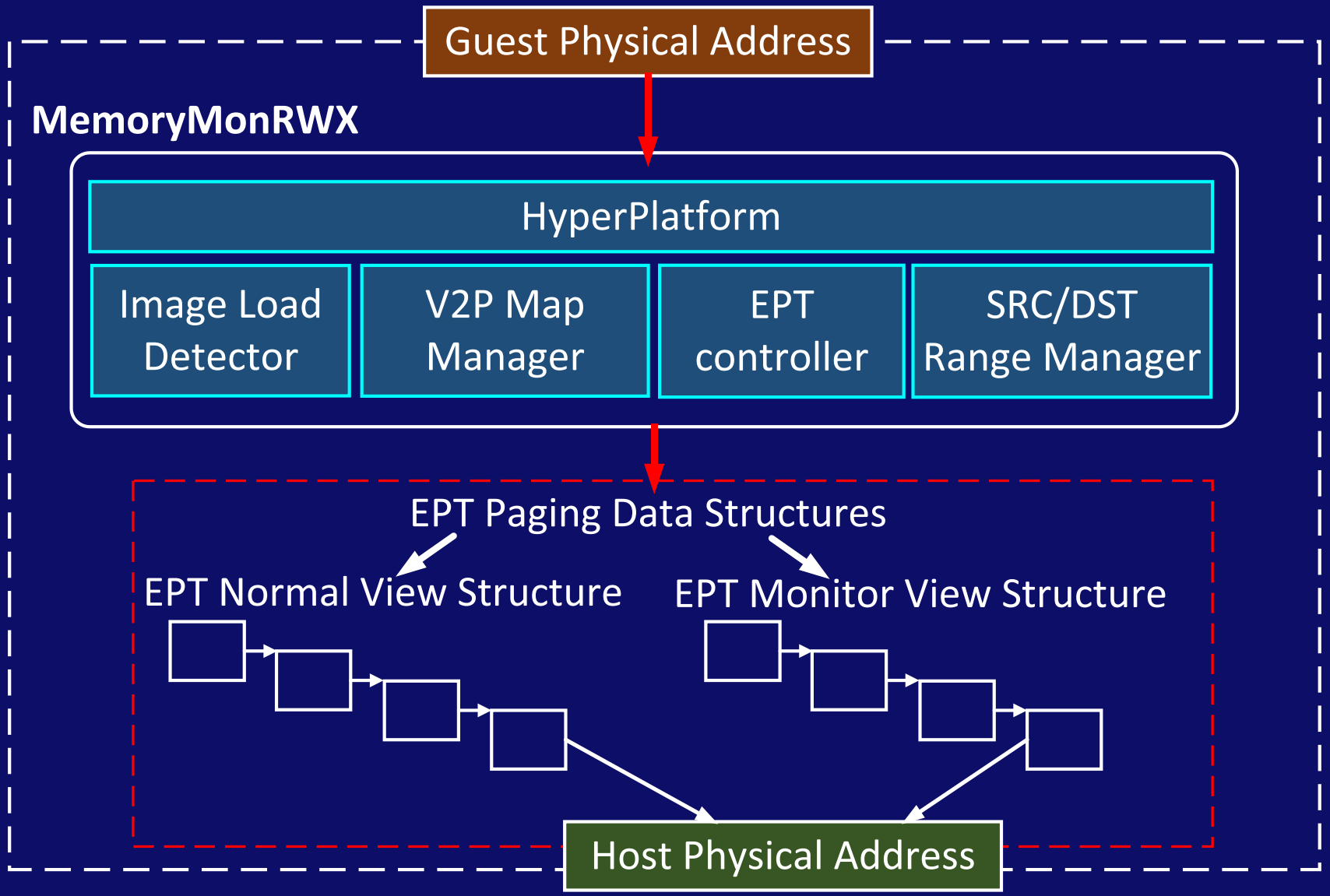
The source code is here - http://bit.ly/MemoryMonRWX

MemoryMonRWX architecture



The source code is here - http://bit.ly/MemoryMonRWX

MemoryMonRWX architecture

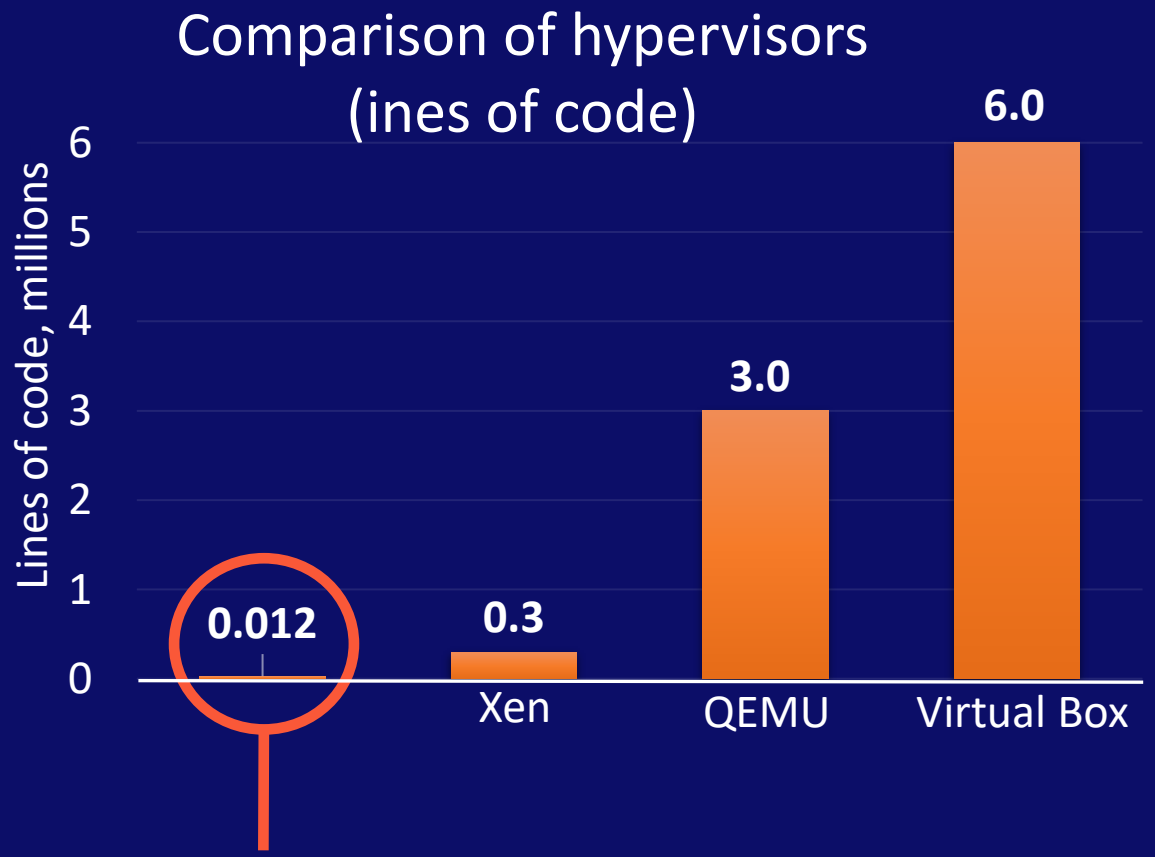


The source code is here - http://bit.ly/MemoryMonRWX

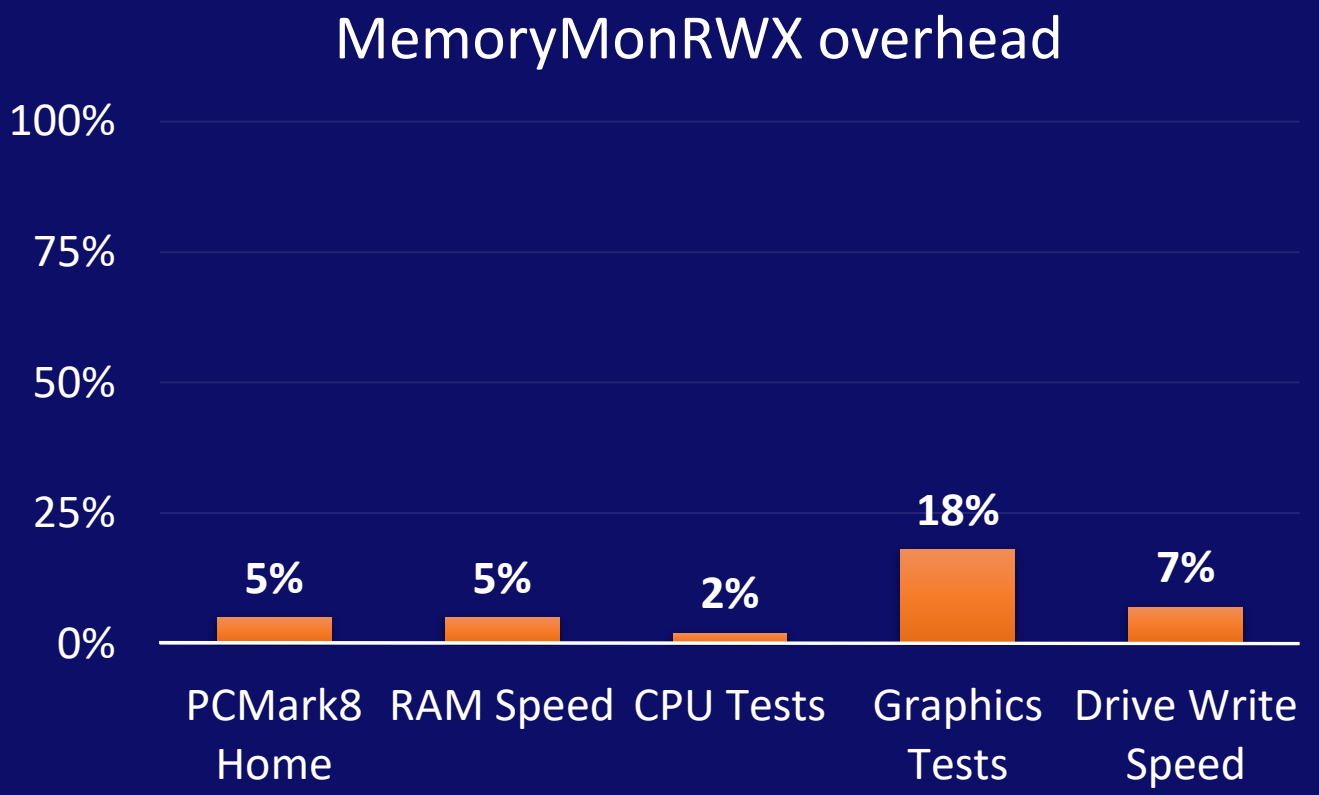
Demo 3

The online version is here –
https://youtu.be/vi9TzLrO_pE?t=157

MemoryMonRWX is small and fast



MemoryMonRWX is made up of less than 12,000 lines of code, which is less than 3% of Xen



0% - the system without hypervisor,

100% – the full system overload

Conclusions

- MemoryMonRWX logs & controls all memory accesses in a real time
- It is a hypervisor, which supports newest Windows 10 x64
- MemoryMonRWX can be used in various tasks:
  - Trace malware activity
  - Protect memory of 3rd party drivers

Acquire Physical Memory & Detect Hidden Software by Raspberry Pi

CaptureGUARD Physical Memory Acquisition Hardware

$7,799.00

"This is an ExpressCard device capable of imaging the physical memory of the computer it's connected to. Creates dump files in the standard WinDD format.."[1]

1. CaptureGUARD Physical Memory Acquisition Hardware – ExpressCard. Windowsscope. http://www.windowsscope.com/product/captureguard-physical-memory-acquisition-hardware-expresscard/
2. Aumaitre, D., and Devine, C. Subverting Windows 7 x64 Kernel with DMA attacks. Sogeti ESEC Lab: http://esec-lab.sogeti.com/dotclear/public/ publications/10-hitbamsterdam-dmaattacks.pdf, July 2010.

Acquire Physical Memory & Detect Hidden Software by Raspberry Pi

CaptureGUARD Physical Memory Acquisition Hardware

$7,799.00

"This is an ExpressCard device capable of imaging the physical memory of the computer it's connected to. Creates dump files in the standard WinDD format.."[1]

CardBus FPGA dev platform
Xilinx Spartan-3 400[2]

$295.00

+

Raspberry Pi 3
Model B

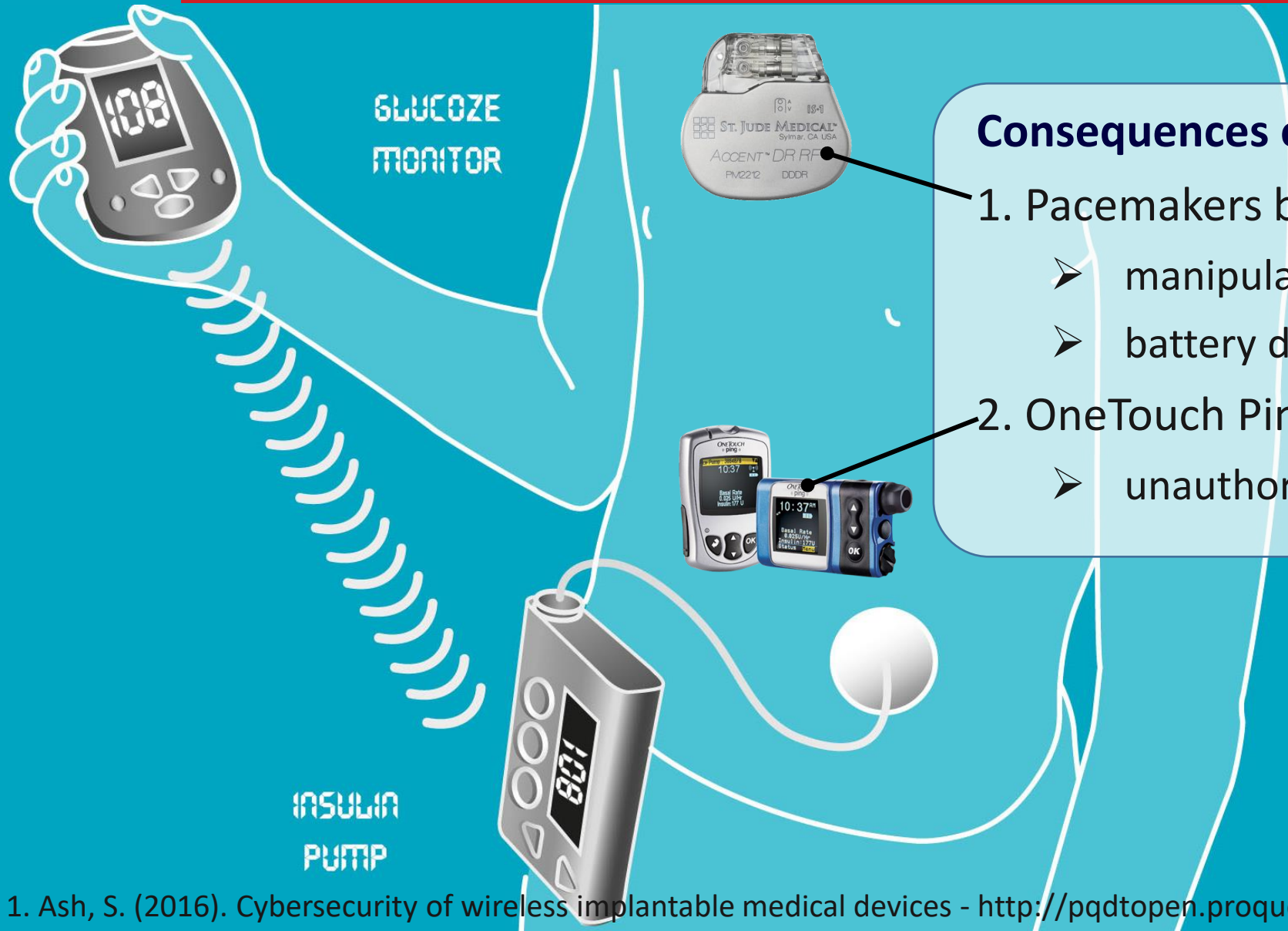
$35.00

= **$330.00**

Lower price with more features

1. CaptureGUARD Physical Memory Acquisition Hardware – ExpressCard. Windowsscope. http://www.windowsscope.com/product/captureguard-physical-memory-acquisition-hardware-expresscard/
2. Aumaitre, D., and Devine, C. Subverting Windows 7 x64 Kernel with DMA attacks. Sogeti ESEC Lab: http://esec-lab.sogeti.com/dotclear/public/ publications/10-hitbamsterdam-dmaattacks.pdf, July 2010.

In the USA in upwards of 2.5 million people depend on wireless implantable medical devices, which all can be hijacked remotely[1]



Consequences of attacks on implants:

1. Pacemakers by St. Jude Medical Inc.(2016)
   - manipulation of beat rates
   - battery drain
2. OneTouch Ping Insulin Pump by J&J (2016)
   - unauthorized insulin injections

1. Ash, S. (2016). Cybersecurity of wireless implantable medical devices - http://pqdtopen.proquest.com/doc/1796055559.html?FMT=ABS

Protection of Wireless Implantable Medical Devices

Our Team:

- Veronika Domova
- Software developer, Sweden
- IoT and Industrial Cyber Security

- Igor Korkin, Ph.D.

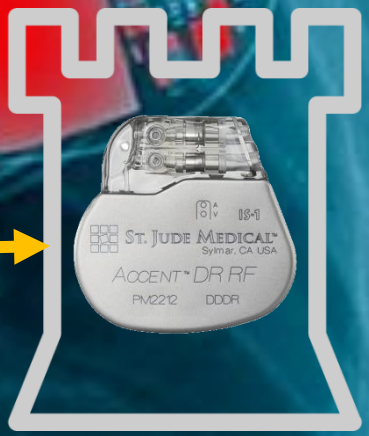
Our Idea:

Vulnerable implant

Input implant's technical specifications

Choose the lightweight crypto cipher

Verify the firmware



Protected implant

Thank you!

Igor Korkin igor.korkin@gmail.com

Satoshi Tanda tanda.sat@gmail.com

The slides, source code and all details are here –
www.bit.ly/MemoryMonRWX