ASPLOS 2023 – Vancouver, Canada – 29th March

Exit-Less, Isolated, and Shared Access for Virtual Machines

Kenichi Yasukata, Hajime Tazaki, Pierre-Louis Aublin

IIJ

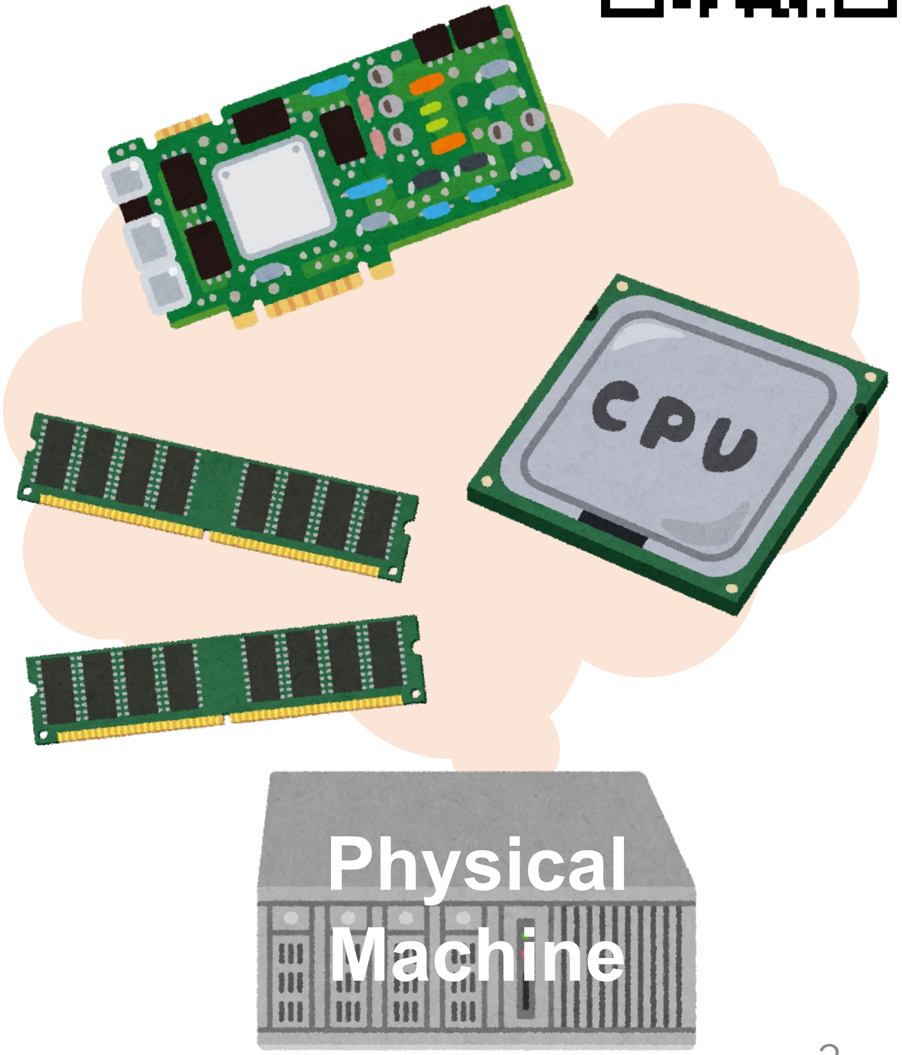
Internet Initiative Japan

About this Work

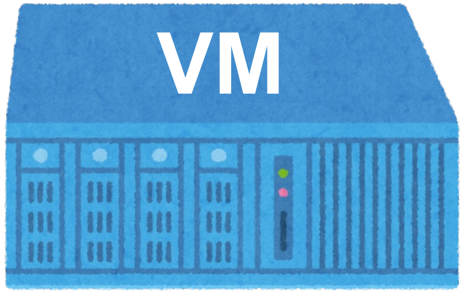
- ELISA: Exit-Less, Isolated, and Shared Access
  - A novel in-memory object sharing scheme for VMs
- Project web page : https://github.com/yasukata/ELISA
  - Paper
  - Slides
  - Source code
  - Commentary

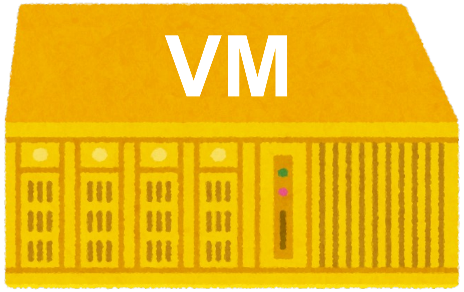
The QR code stays there during the presentation

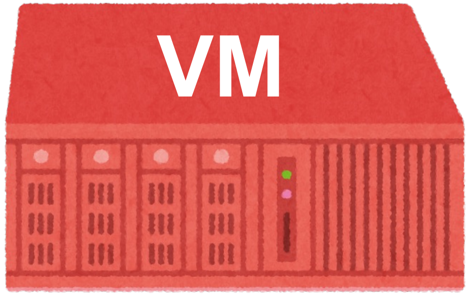
Virtual Machines (VMs)

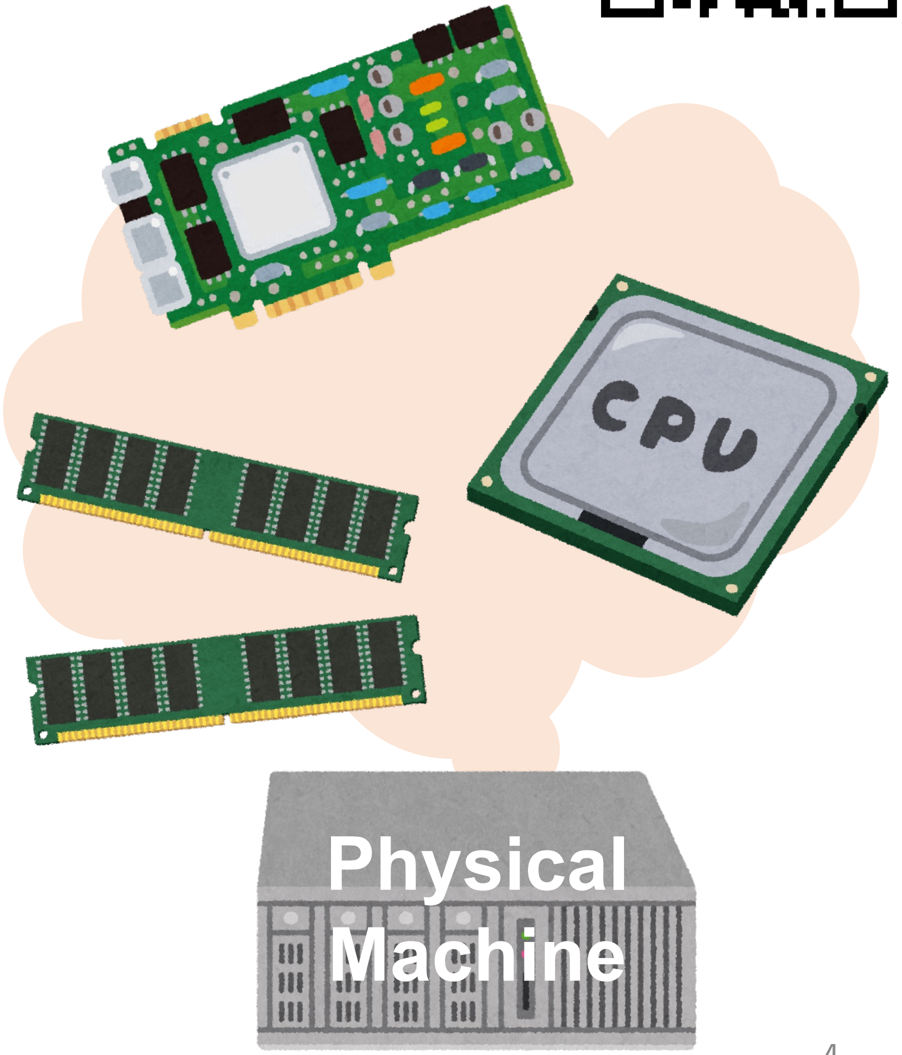


Virtual Machines (VMs)

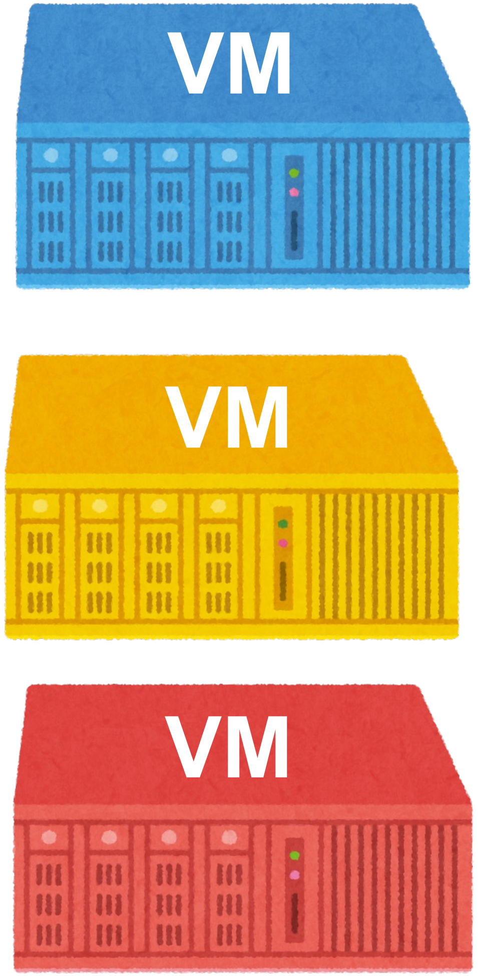


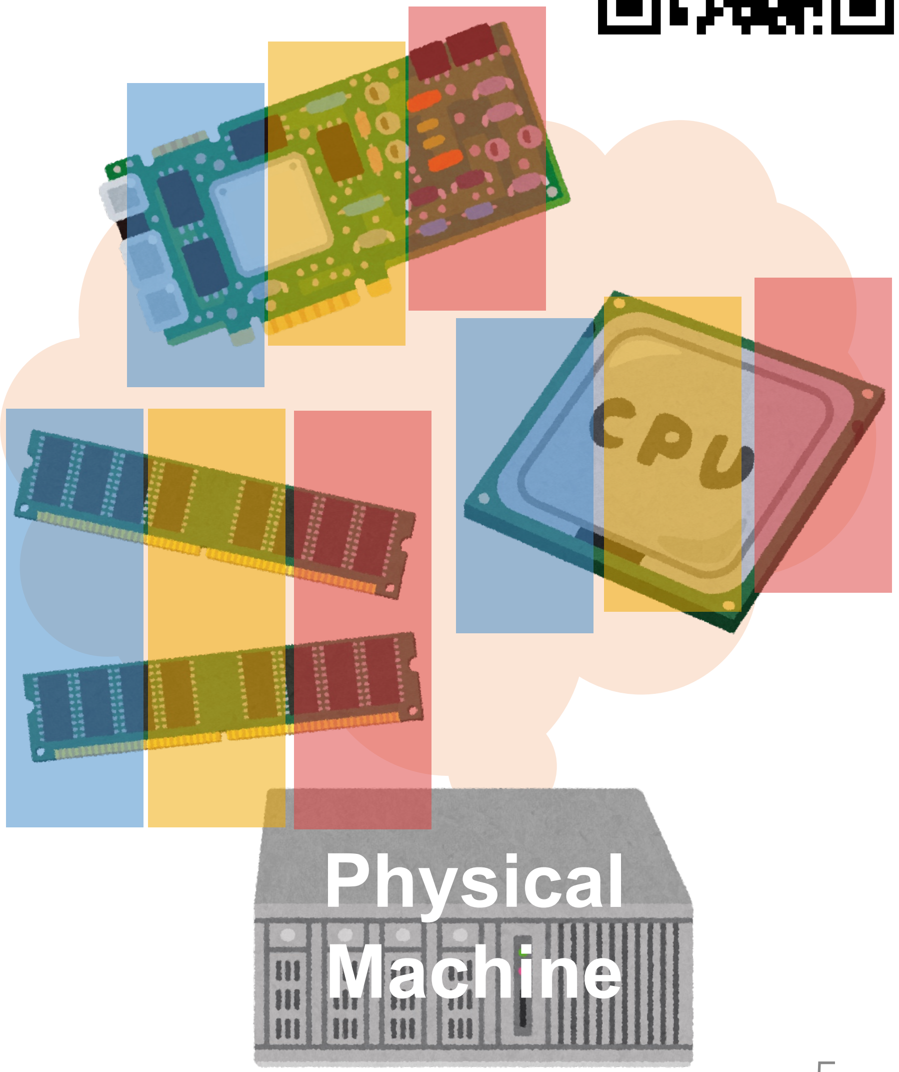




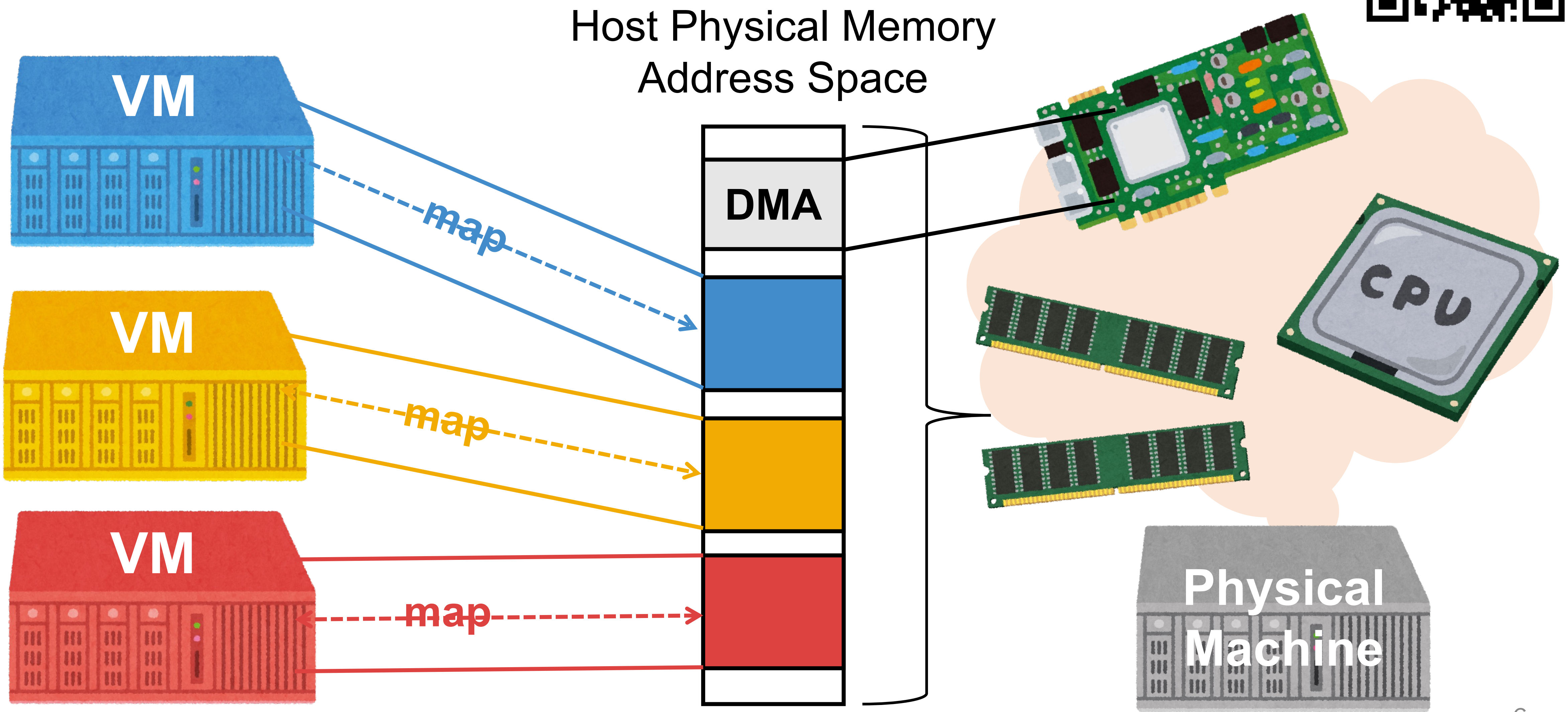


Virtual Machines (VMs)

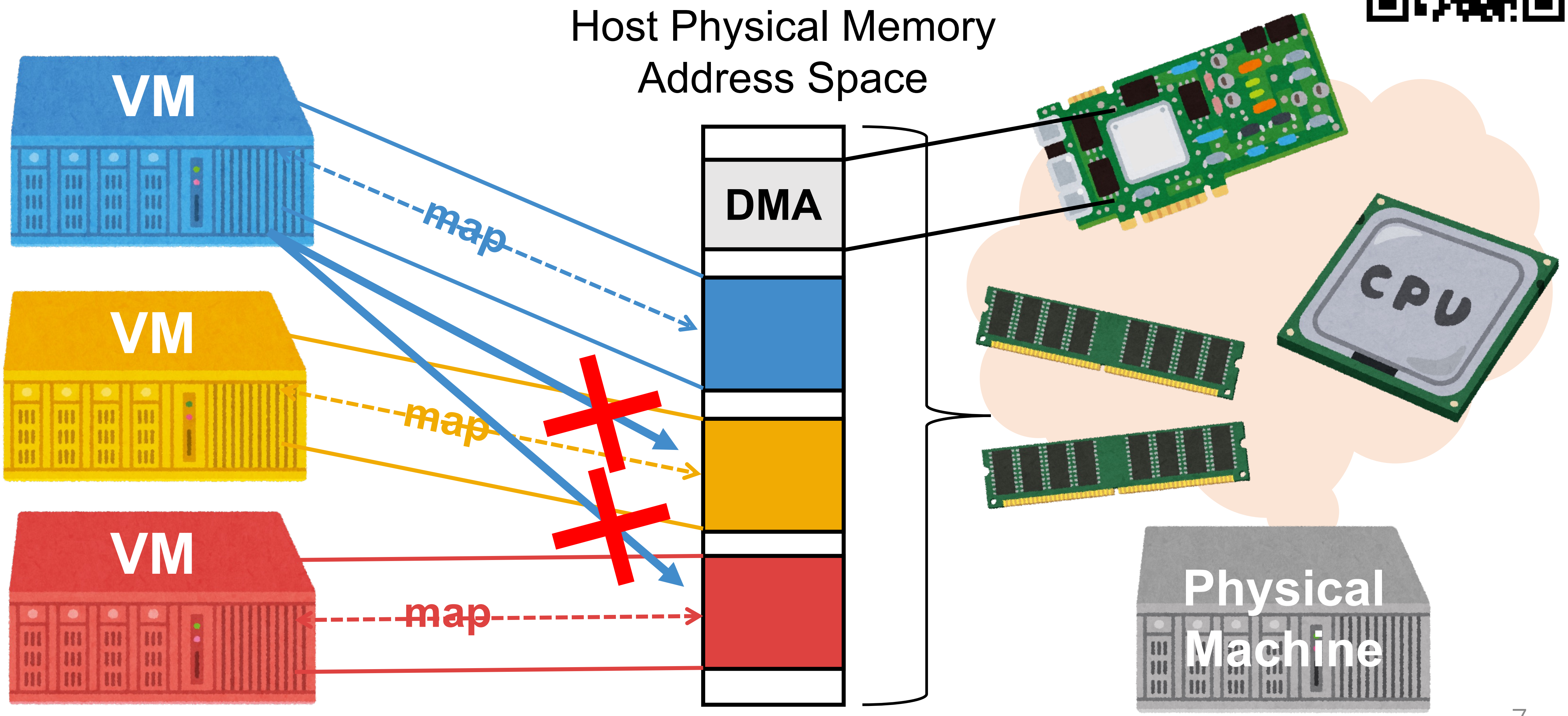




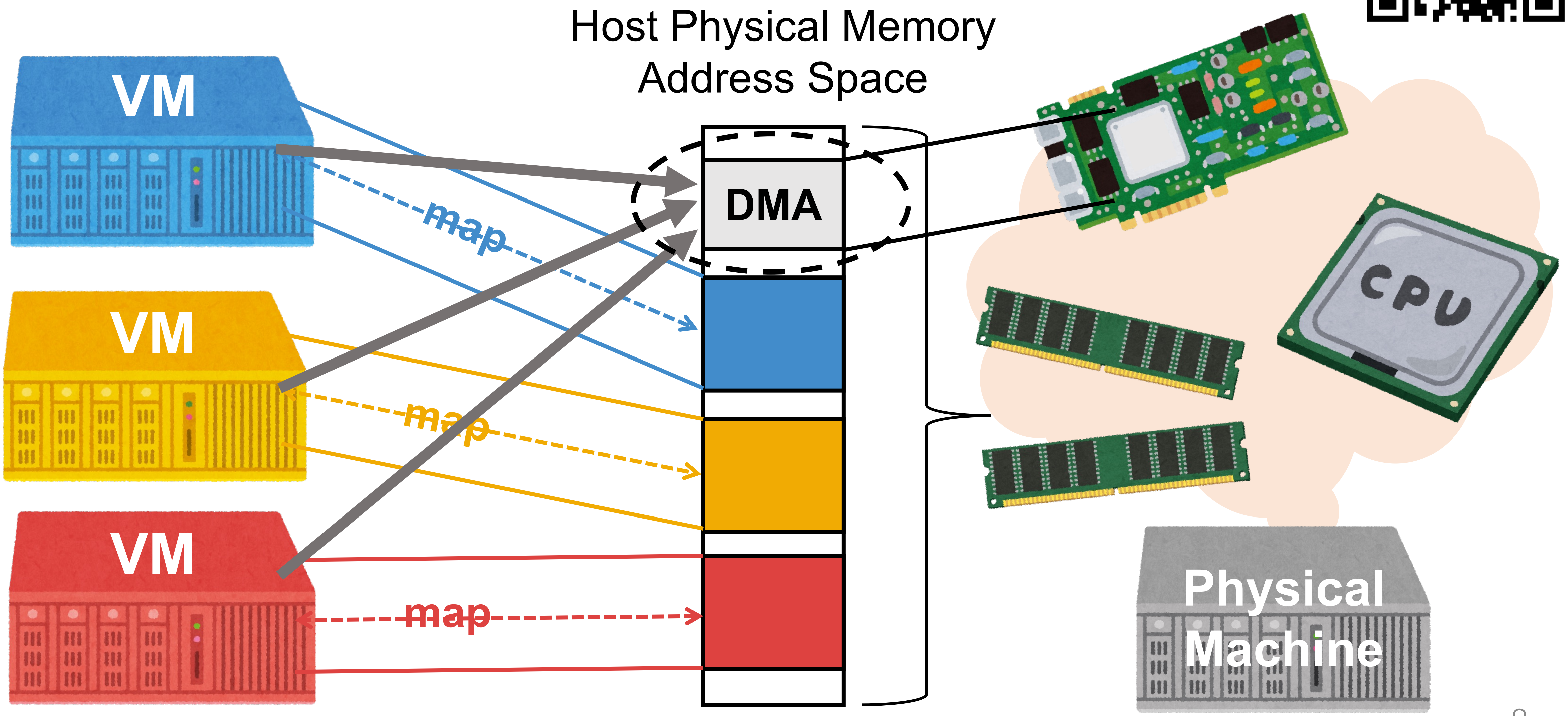
Memory Isolation



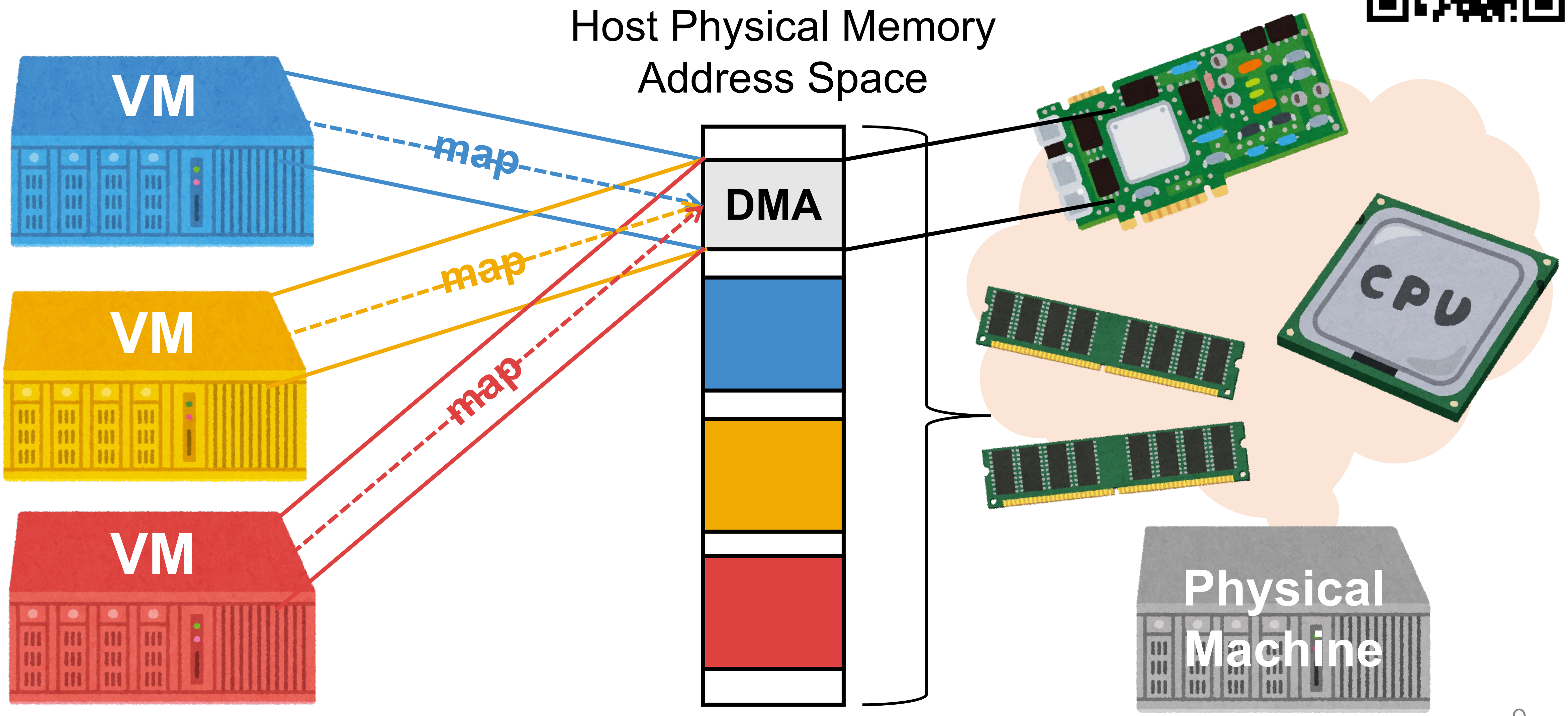
Memory Isolation



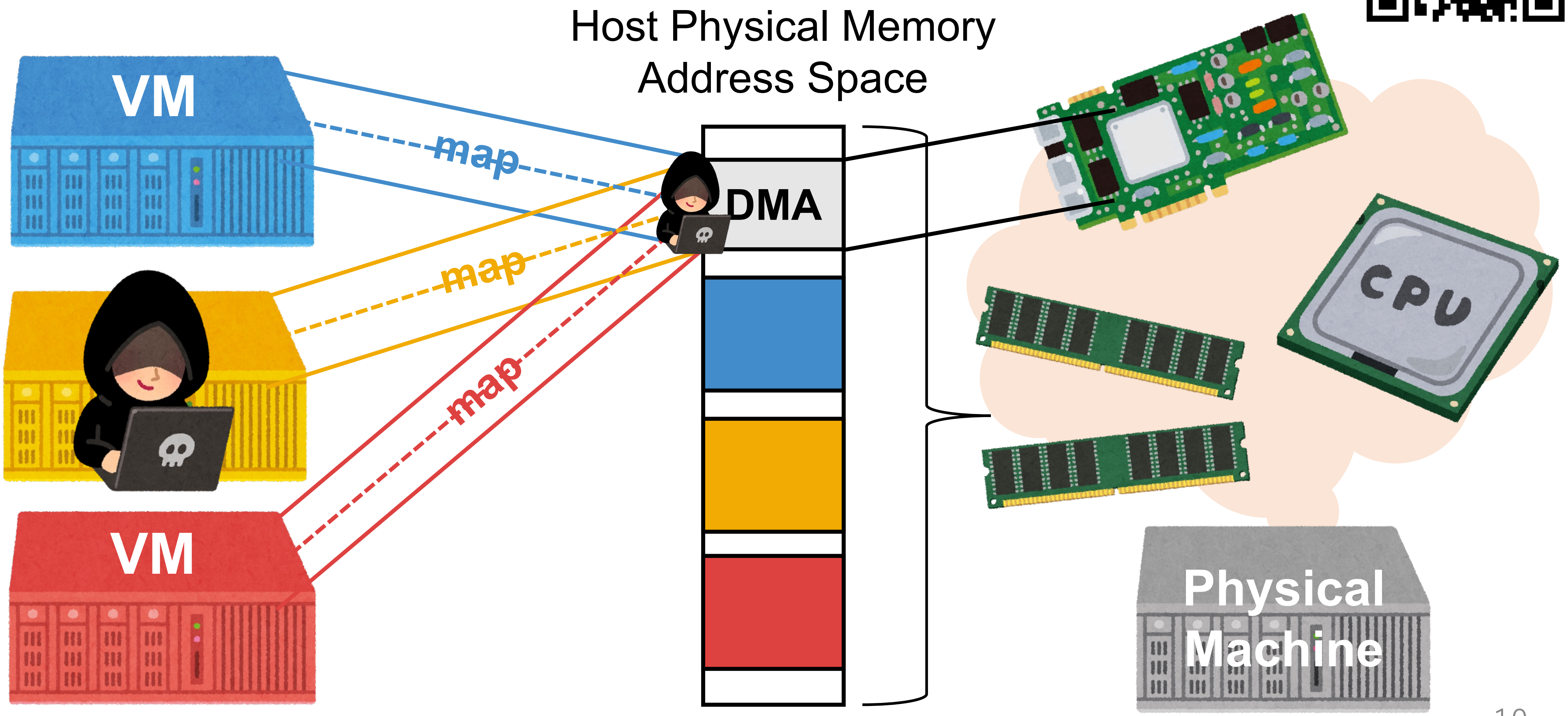
Memory Isolation



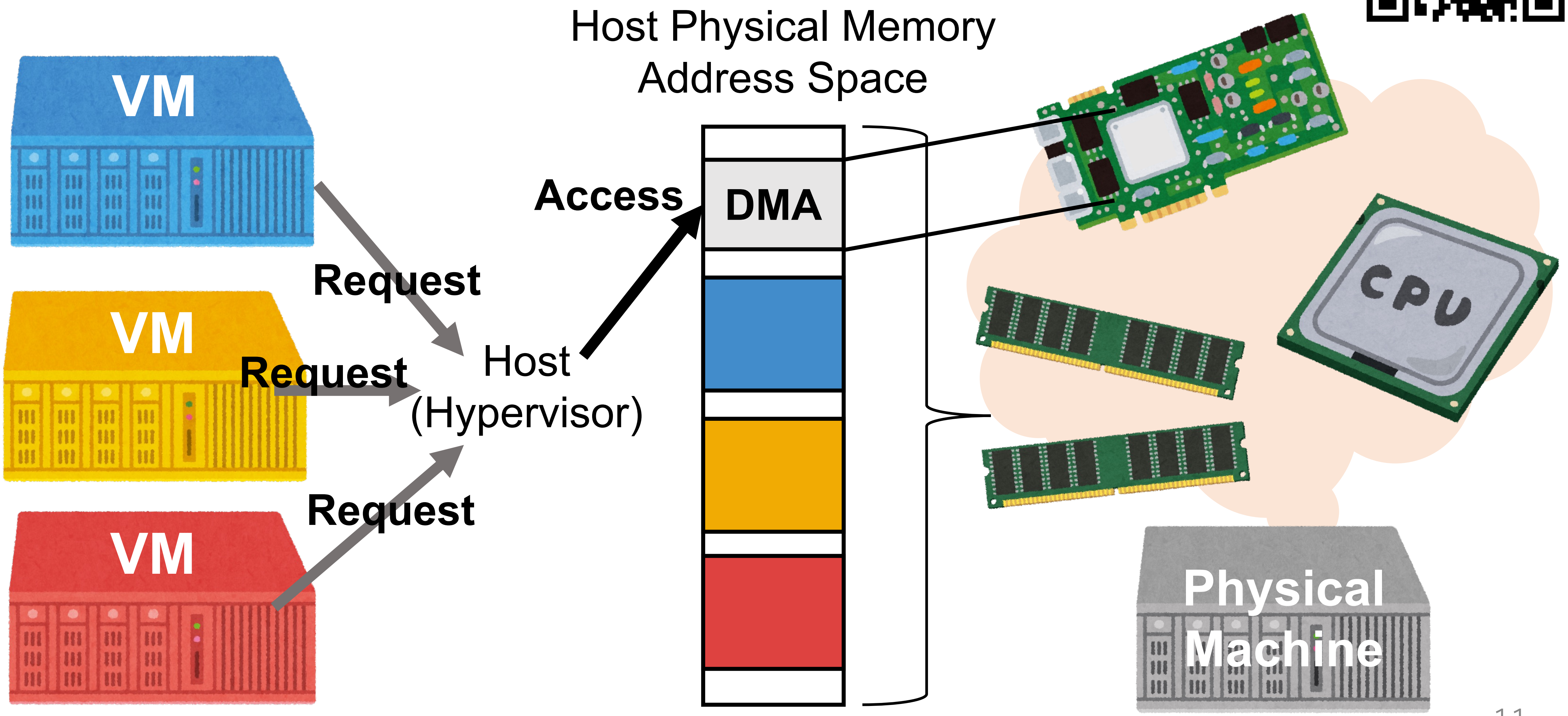
Sharing Scheme 1 : Direct-mapping



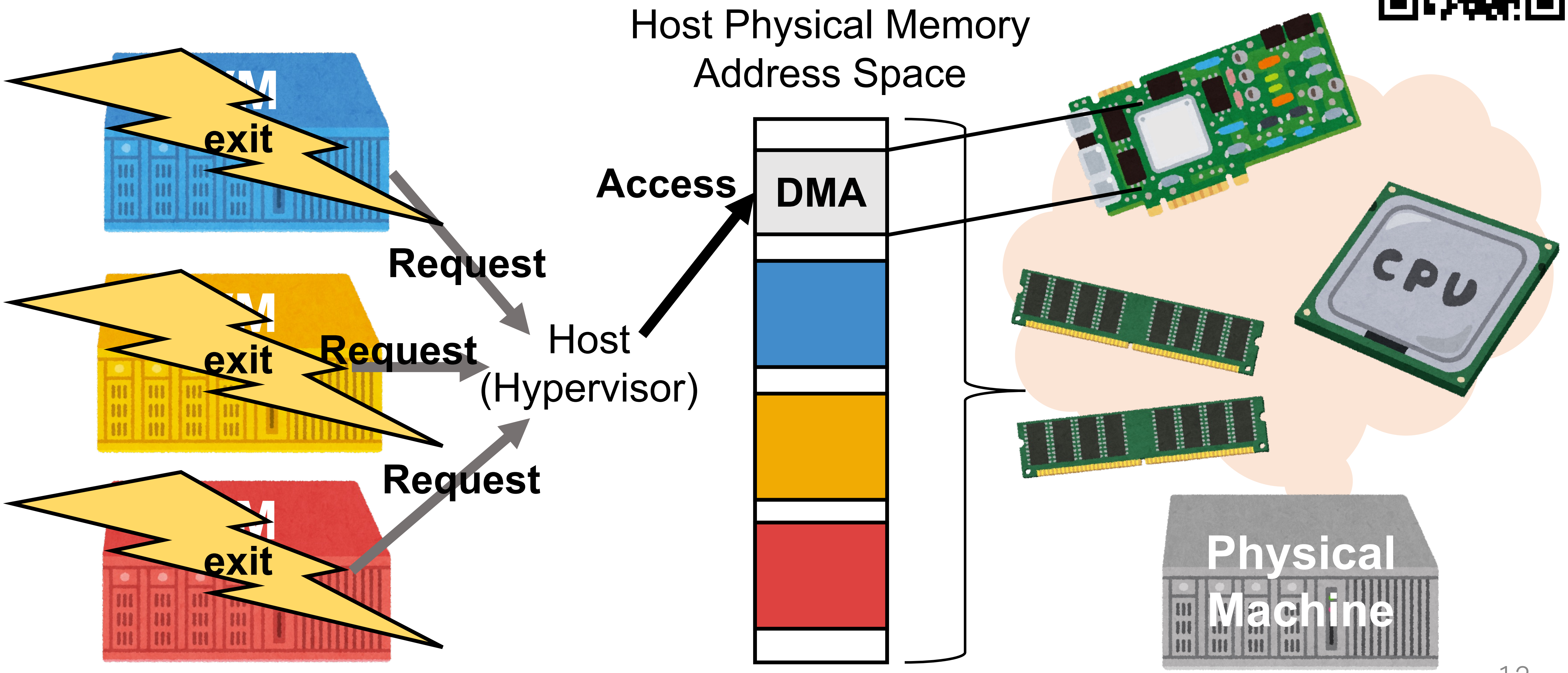
Sharing Scheme 1 : Direct-mapping



Sharing Scheme 2 : Host-interposition



Sharing Scheme 2 : Host-interposition



Host Physical Memory
Address Space

exit

exit

exit

Request

Request

Request

Host
(Hypervisor)

Access

DMA

Physical
Machine

Problem Statement

- The two existing in-memory object sharing schemes cannot offer isolation and low overhead at once

Description	Shared access	Isolation overhead
Direct-mapping	No isolation 😭	None
Host-interposition	Isolated	High 😭

This Work

- The two existing in-memory object sharing schemes cannot offer isolation and low overhead at once
- We explore a new in-memory object sharing scheme which achieves isolation at a low overhead

Description	Shared access	Isolation overhead
Direct-mapping	No isolation 😭	None
Host-interposition	Isolated	High 😭
ELISA (this work)	Isolated	Low

ELISA: Exit-Less, Isolated, and Shared Access

ELISA: Exit-Less, Isolated, and Shared Access

- ELISA employs Extended Page Table (EPT) separation to isolate shared in-memory objects

ELISA: Exit-Less, Isolated, and Shared Access

- ELISA employs Extended Page Table (EPT) separation to isolate shared in-memory objects

- In ELISA, a VM leverages EPT Pointer (EPTP) switching feature of VMFUNC to access the shared in-memory objects

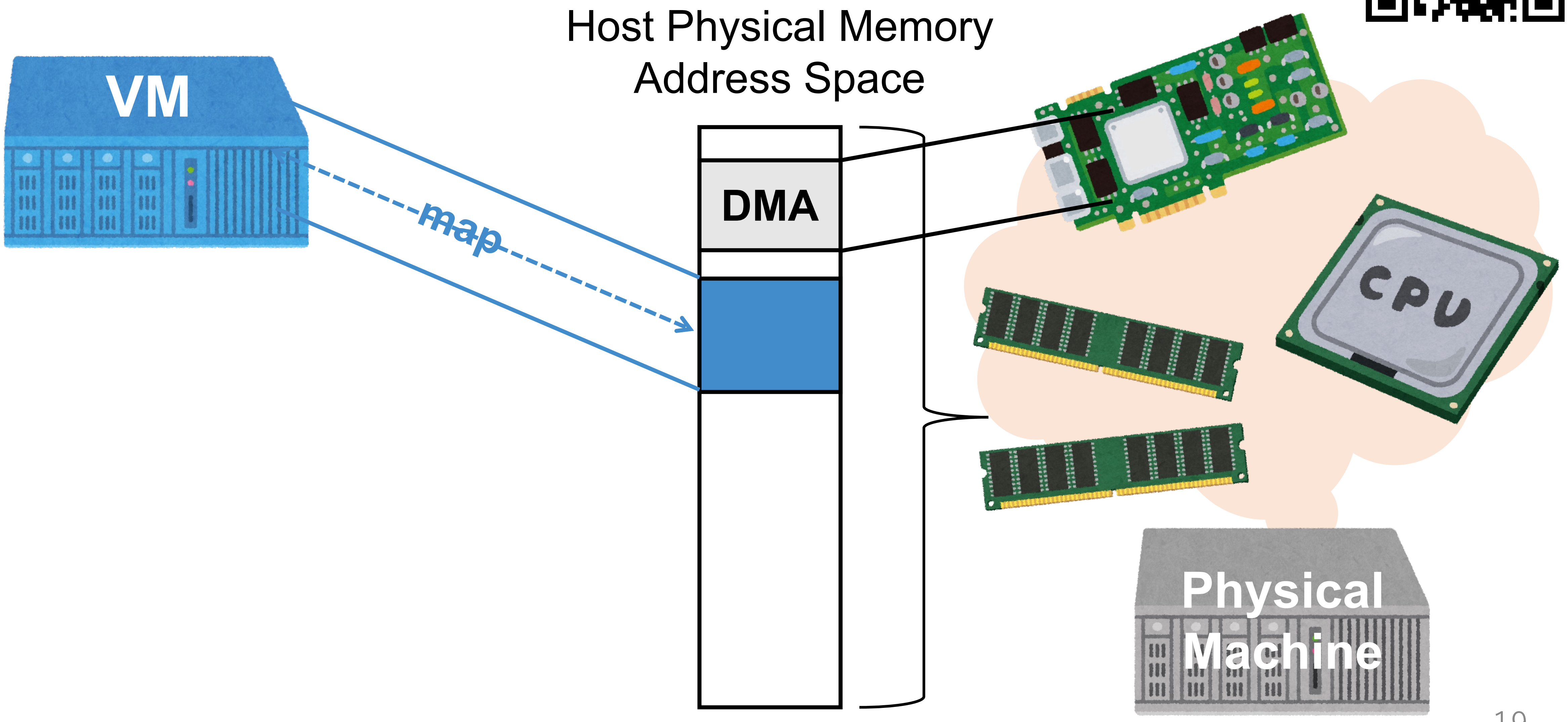
ELISA: Exit-Less, Isolated, and Shared Access

- ELISA employs Extended Page Table (EPT) separation to isolate shared in-memory objects

- In ELISA, a VM leverages EPT Pointer (EPTP) switching feature of VMFUNC to access the shared in-memory objects

- VMFUNC is fast, thus, ELISA offers isolation at a low overhead

Extended Page Table (EPT)



VM

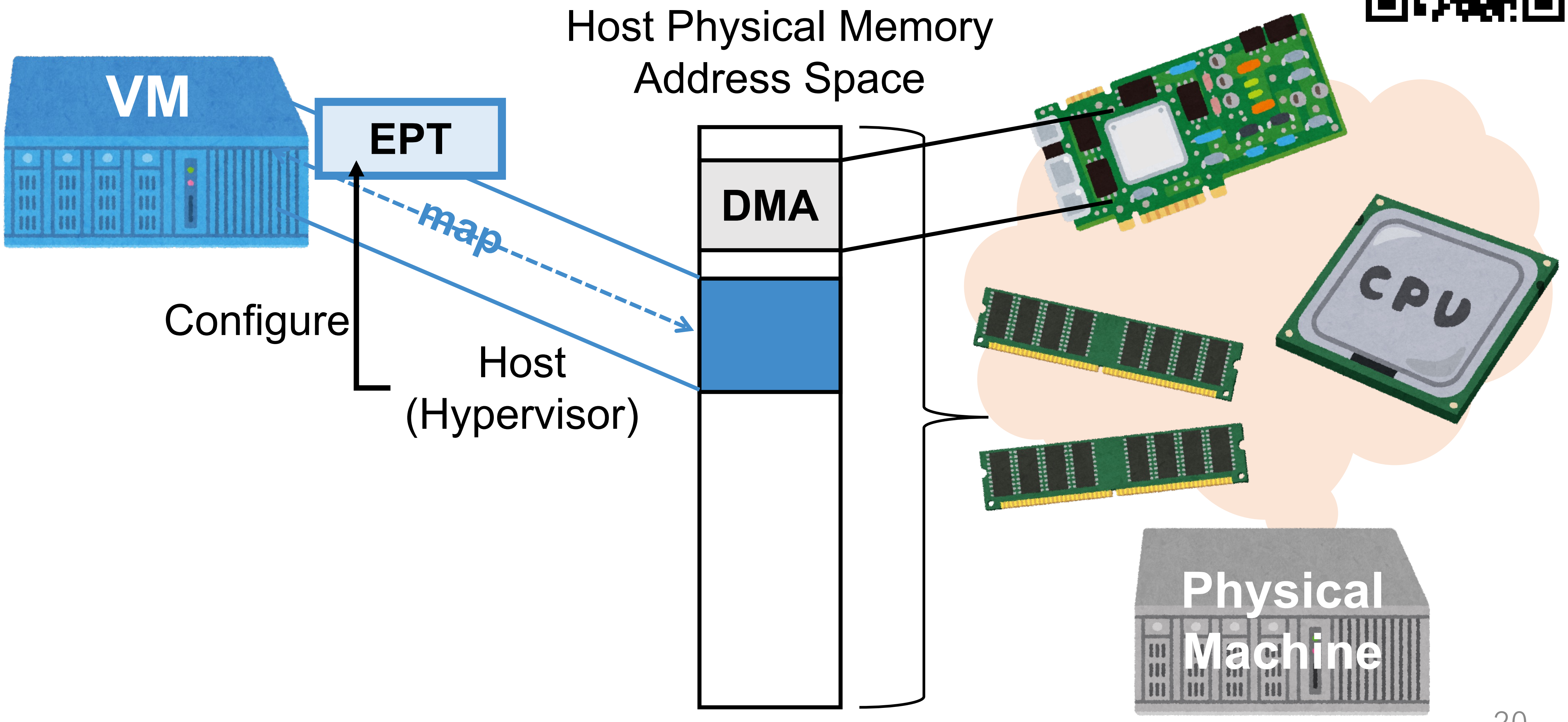
map

Host Physical Memory
Address Space

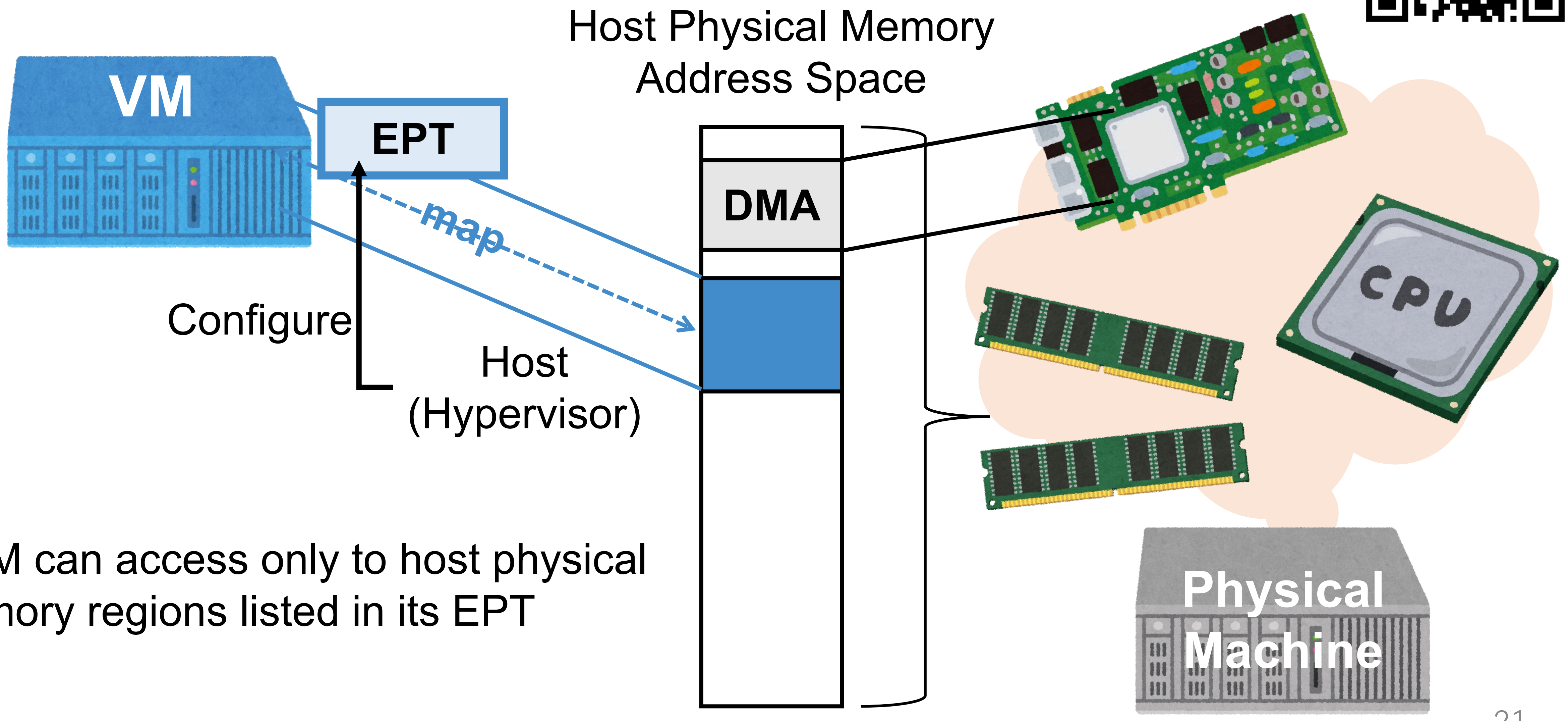
DMA

Physical
Machine

Extended Page Table (EPT)

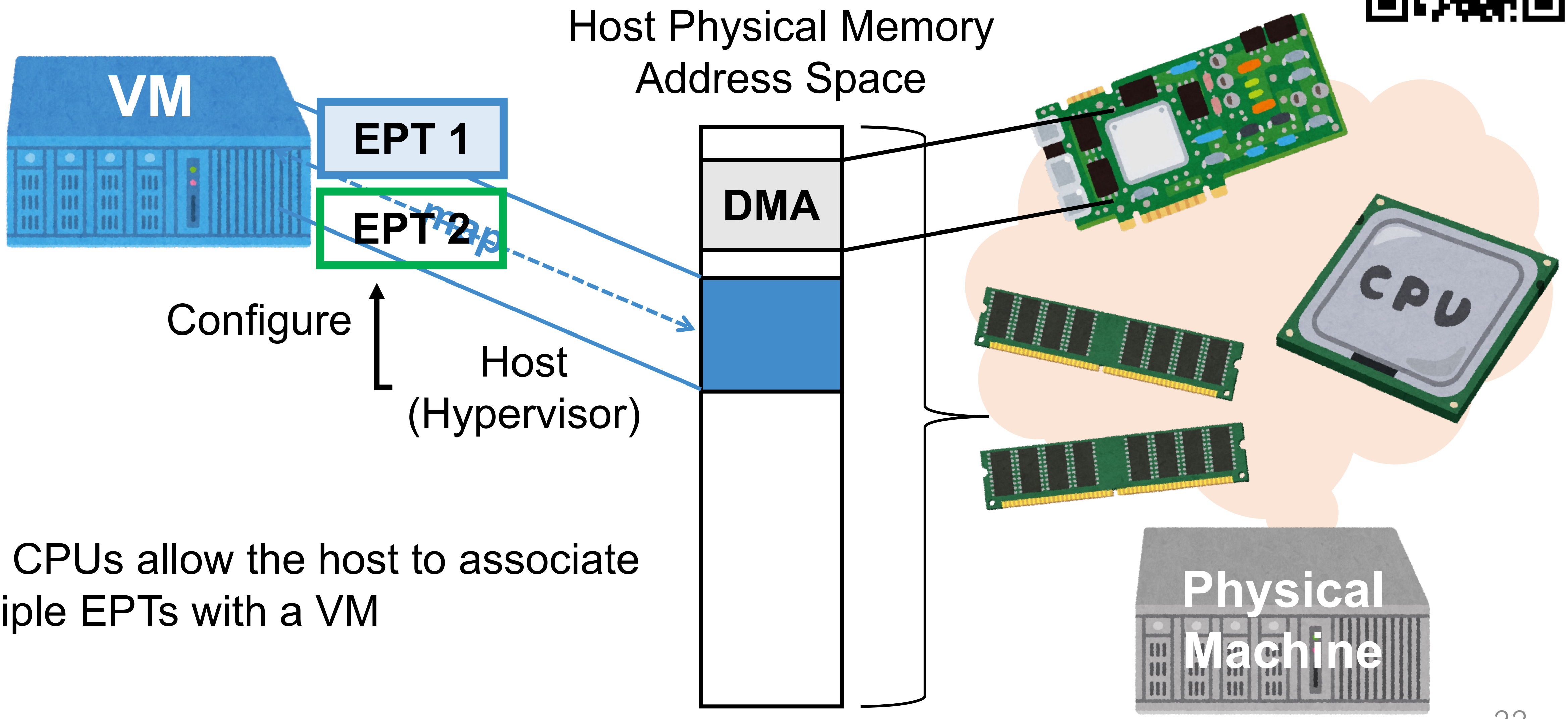


Extended Page Table (EPT)



A VM can access only to host physical memory regions listed in its EPT

Extended Page Table (EPT)



Host Physical Memory Address Space

VM

EPT 1

EPT 2

map

Configure

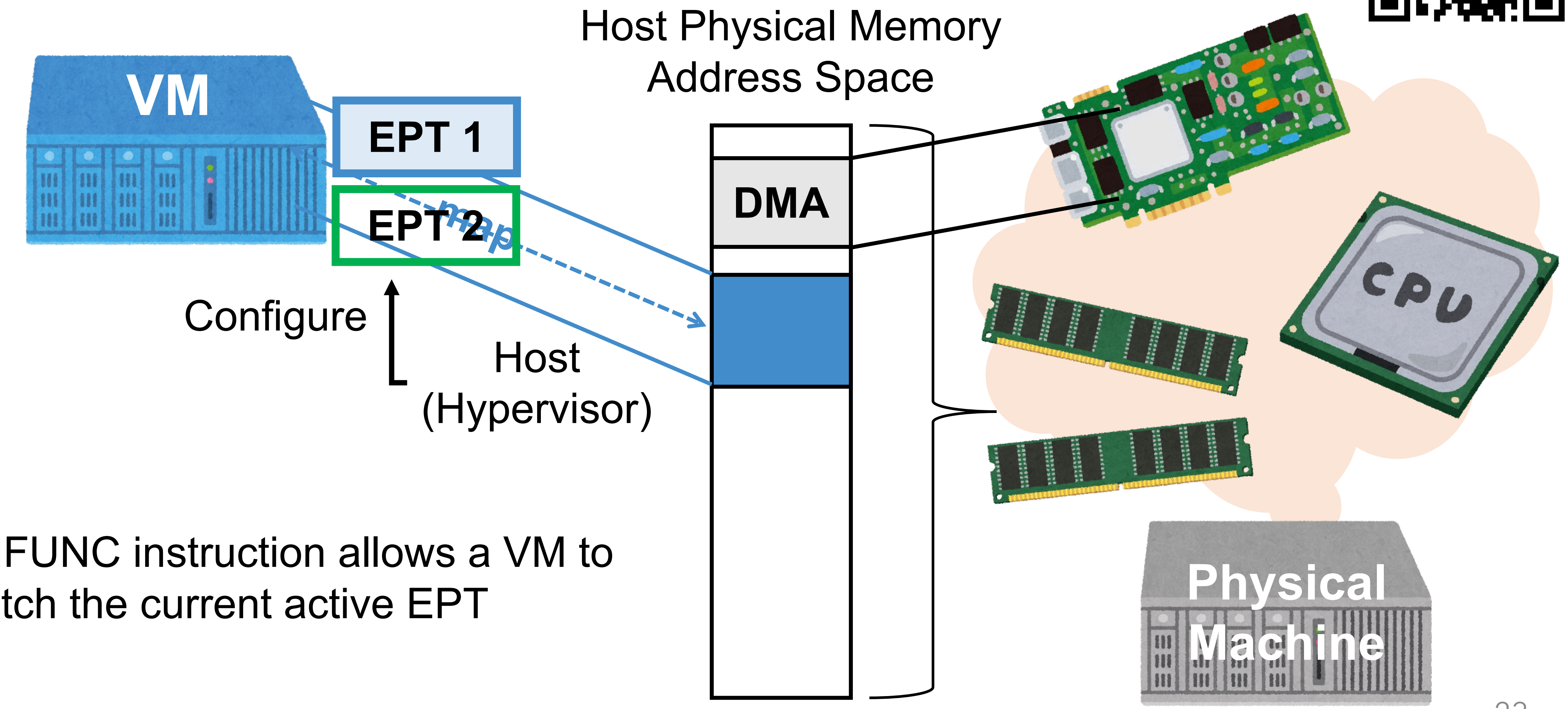
Host (Hypervisor)

DMA

Physical Machine

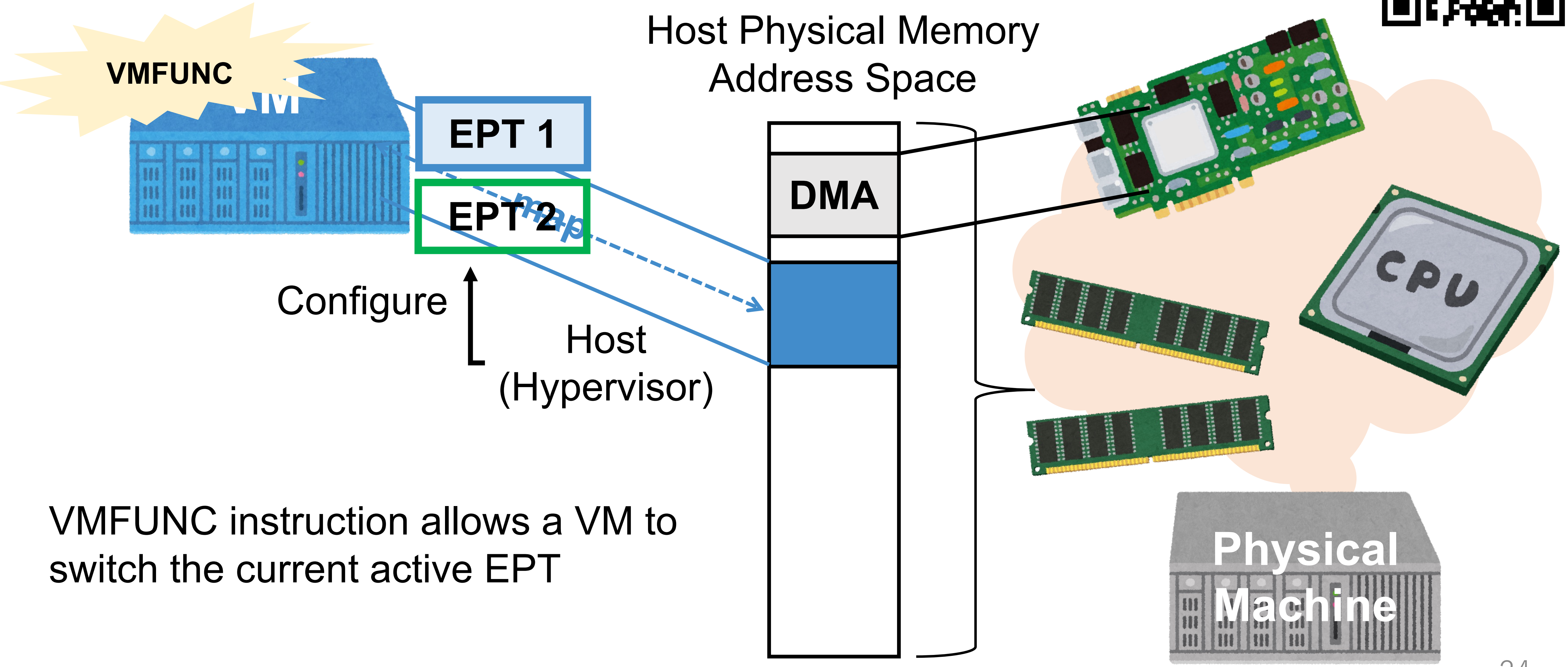
Intel CPUs allow the host to associate multiple EPTs with a VM

EPTP Switching by VMFUNC



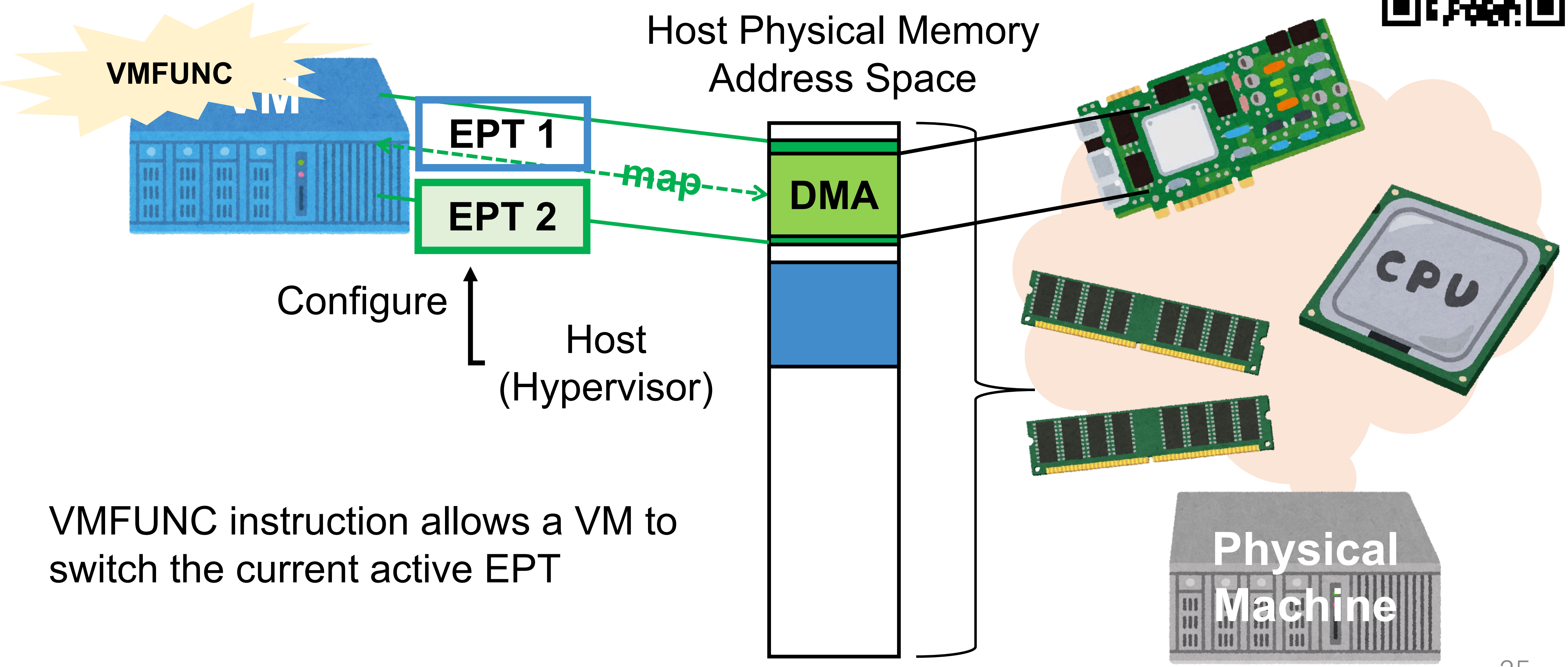
VMFUNC instruction allows a VM to switch the current active EPT

EPTP Switching by VMFUNC



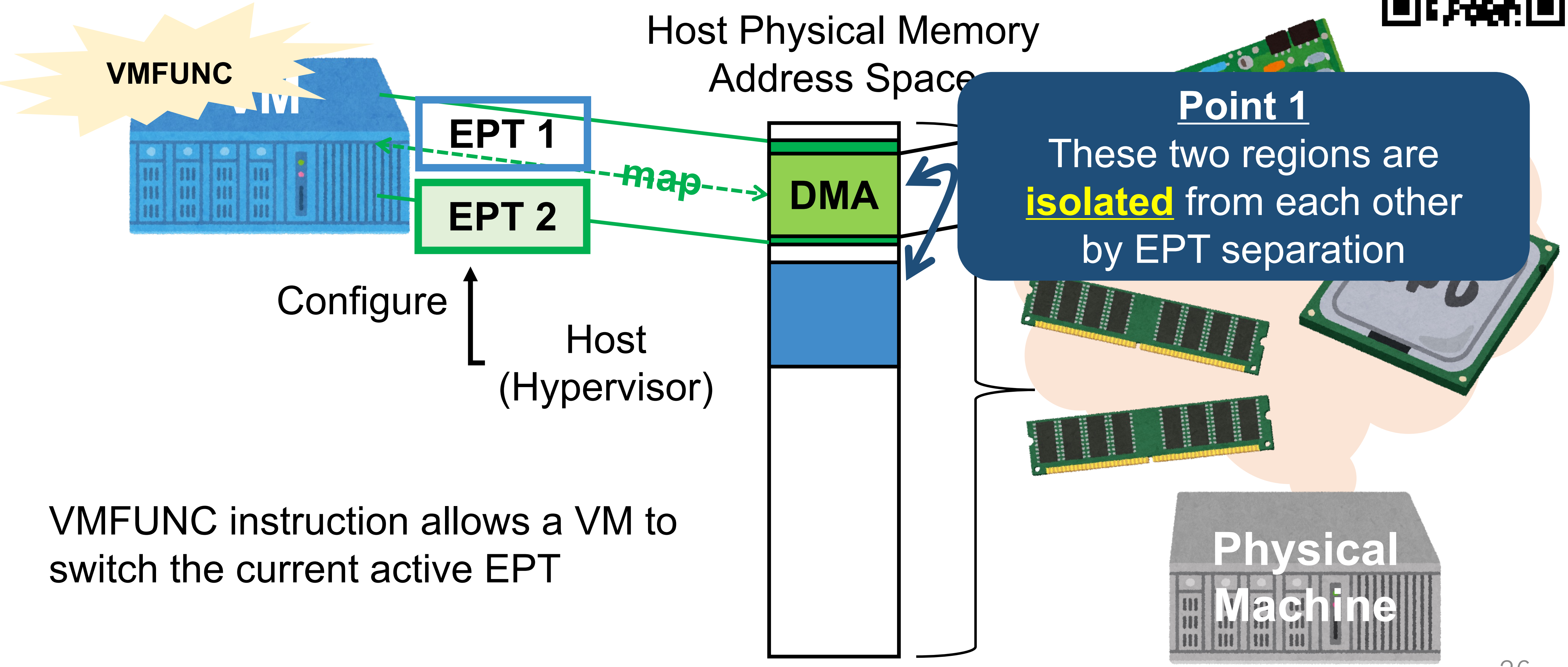
VMFUNC instruction allows a VM to switch the current active EPT

EPTP Switching by VMFUNC



VMFUNC instruction allows a VM to switch the current active EPT

EPTP Switching by VMFUNC

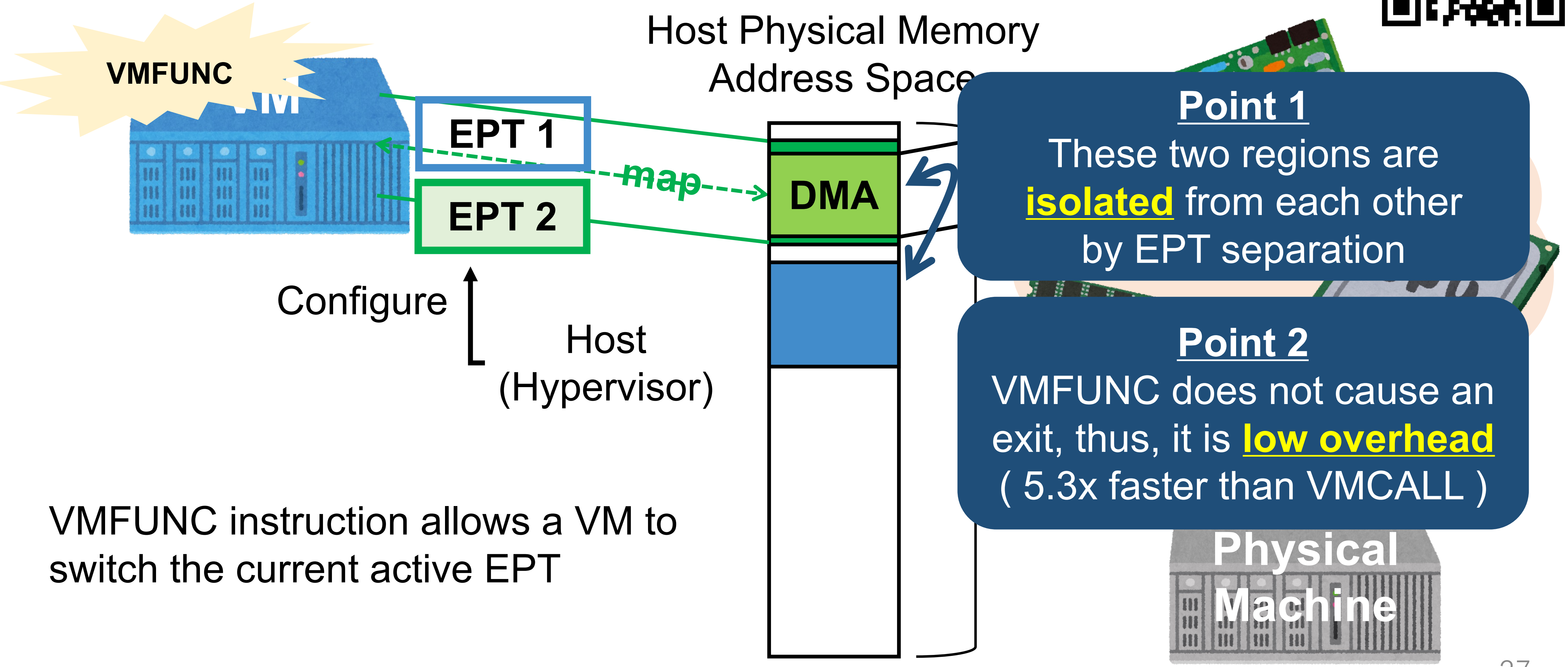


Point 1

These two regions are **isolated** from each other by EPT separation

VMFUNC instruction allows a VM to switch the current active EPT

EPTP Switching by VMFUNC



Point 1

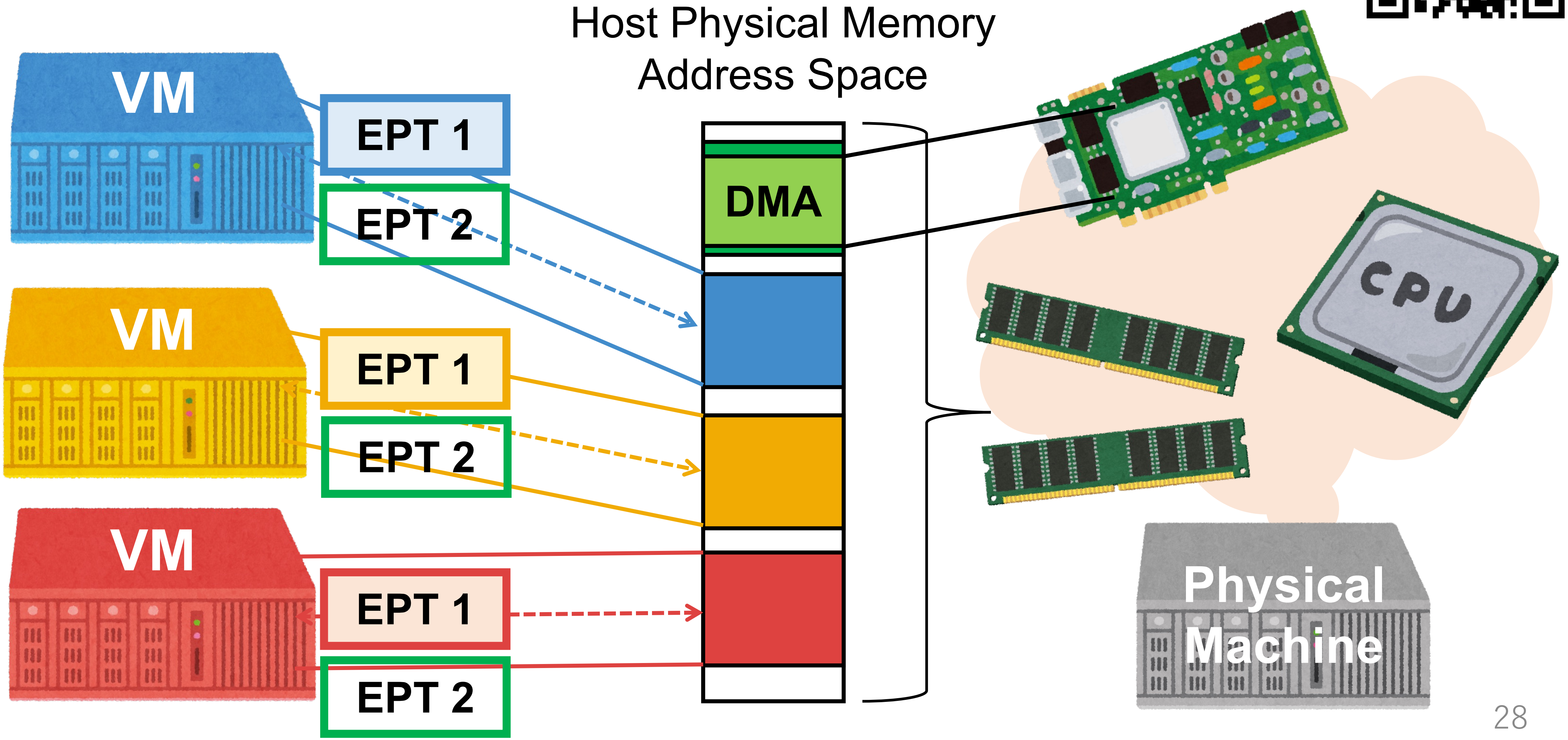
These two regions are **isolated** from each other by EPT separation

Point 2

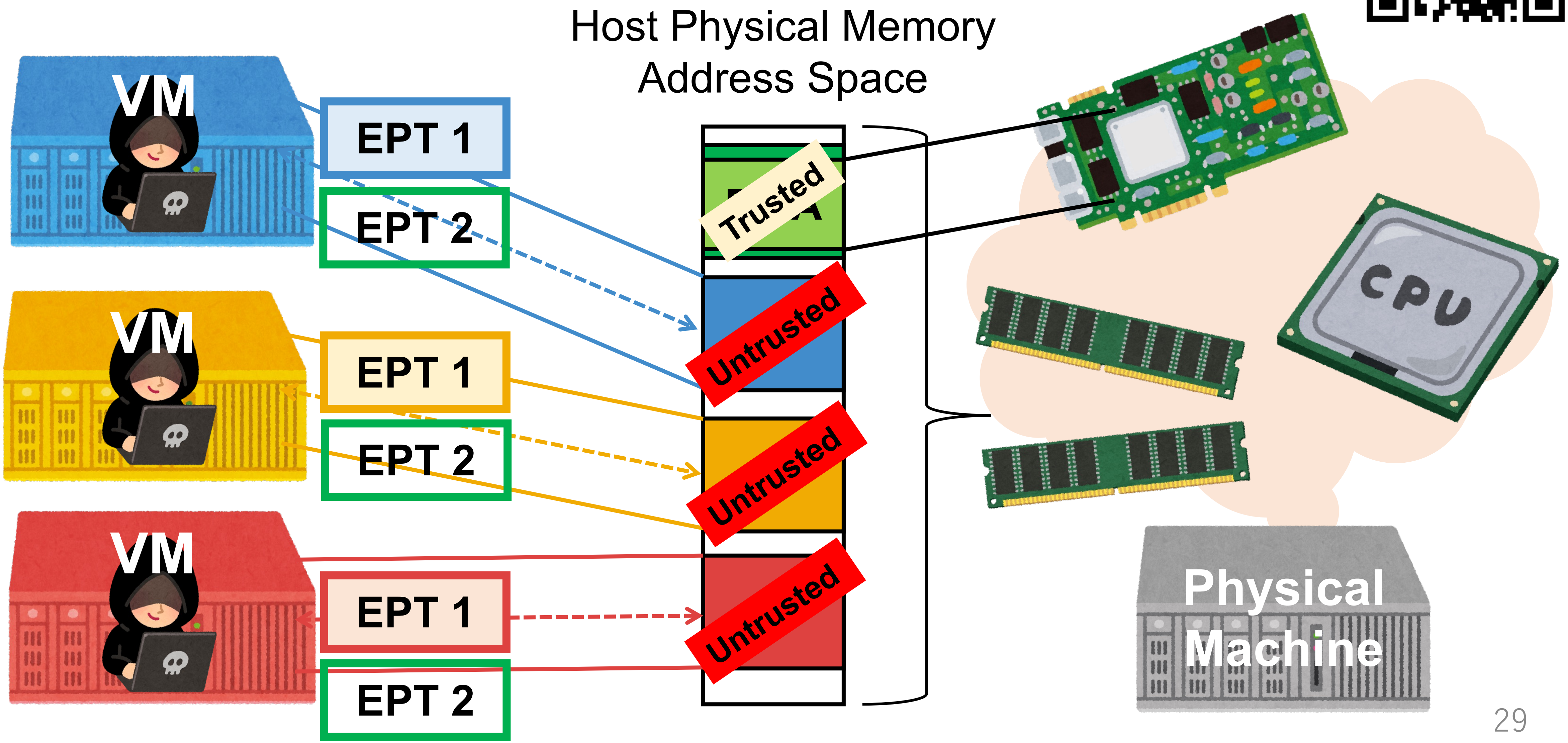
VMFUNC does not cause an exit, thus, it is **low overhead** (5.3x faster than VMCALL)

VMFUNC instruction allows a VM to switch the current active EPT

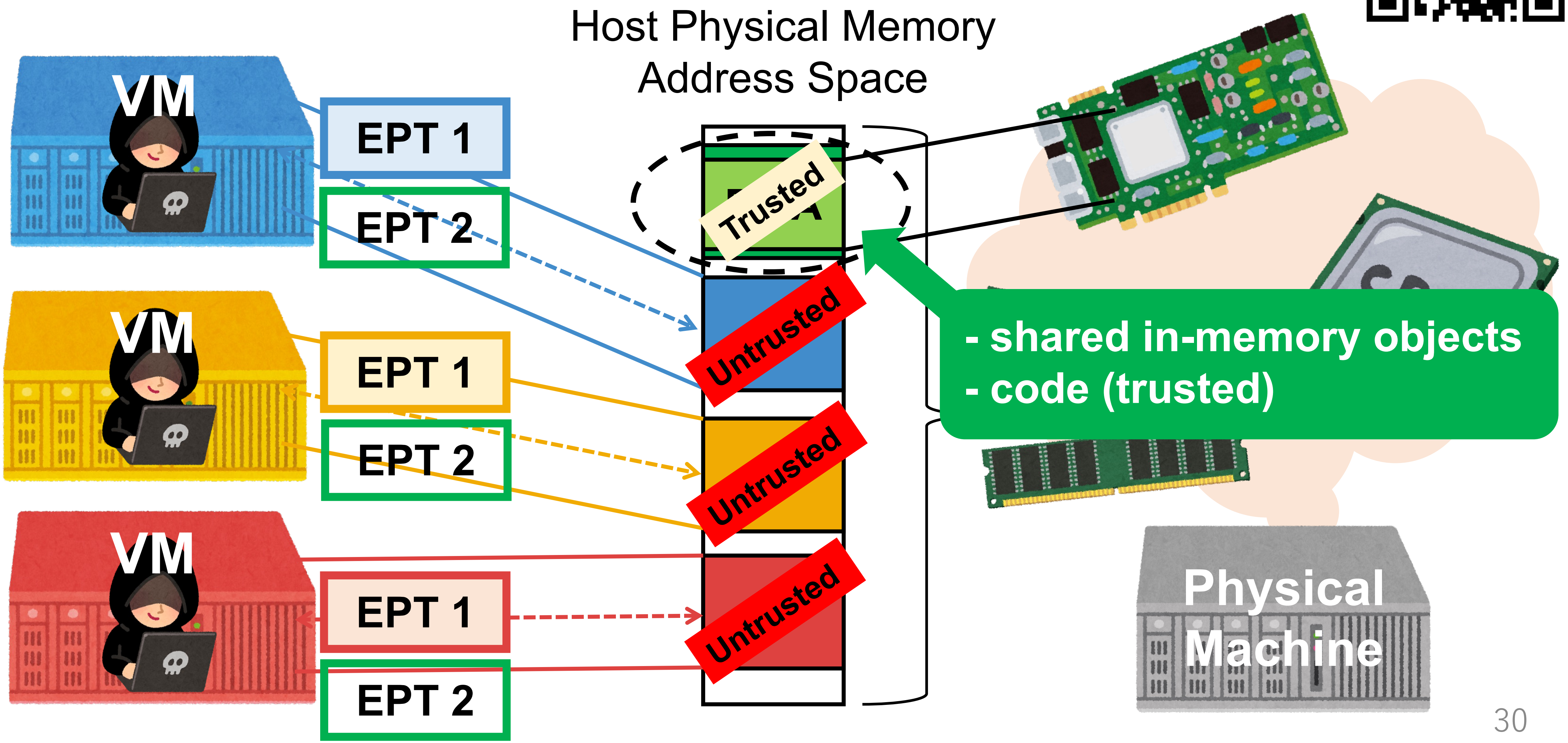
ELISA: Exit-Less, Isolated, and Shared Access



Threat Model



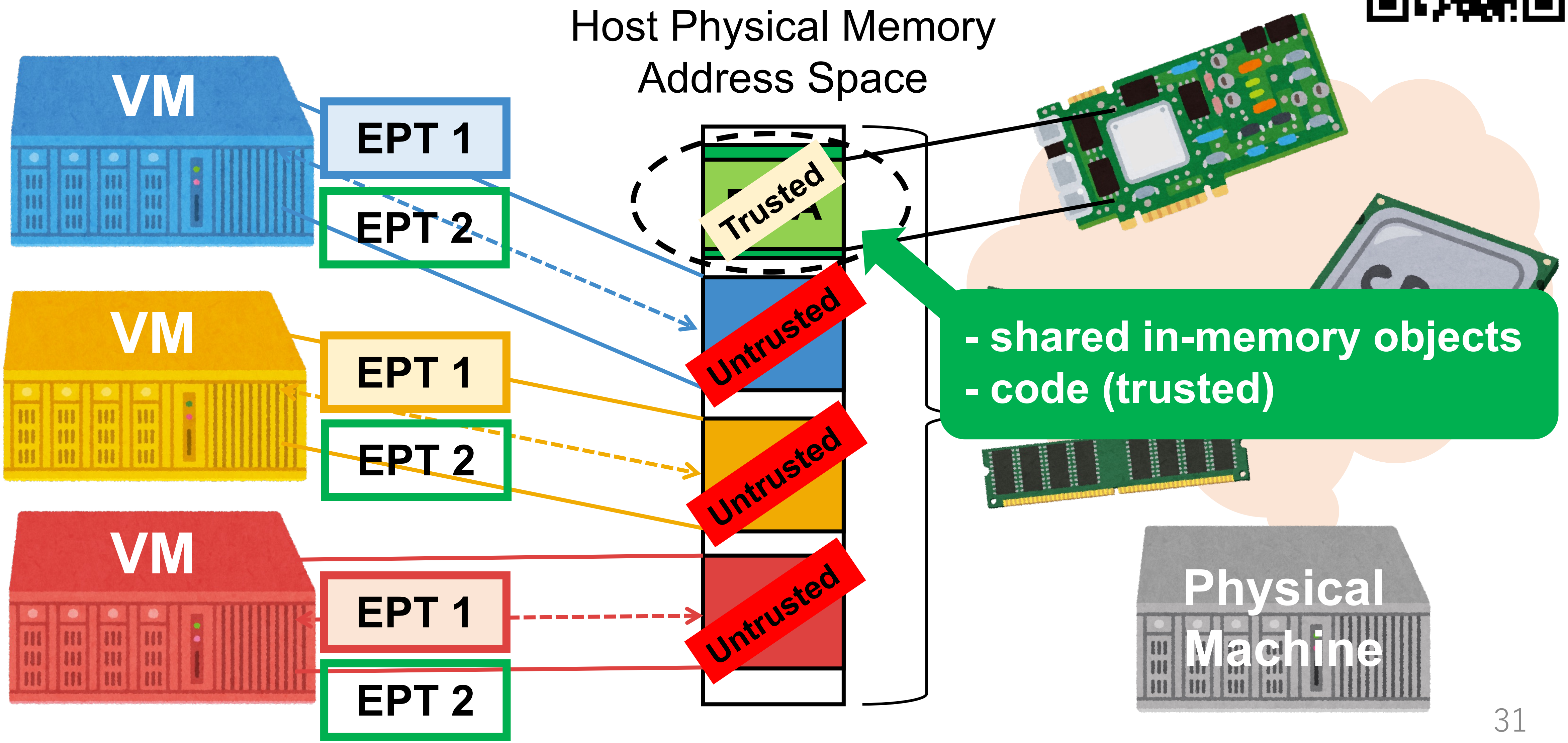
Threat Model



Host Physical Memory Address Space

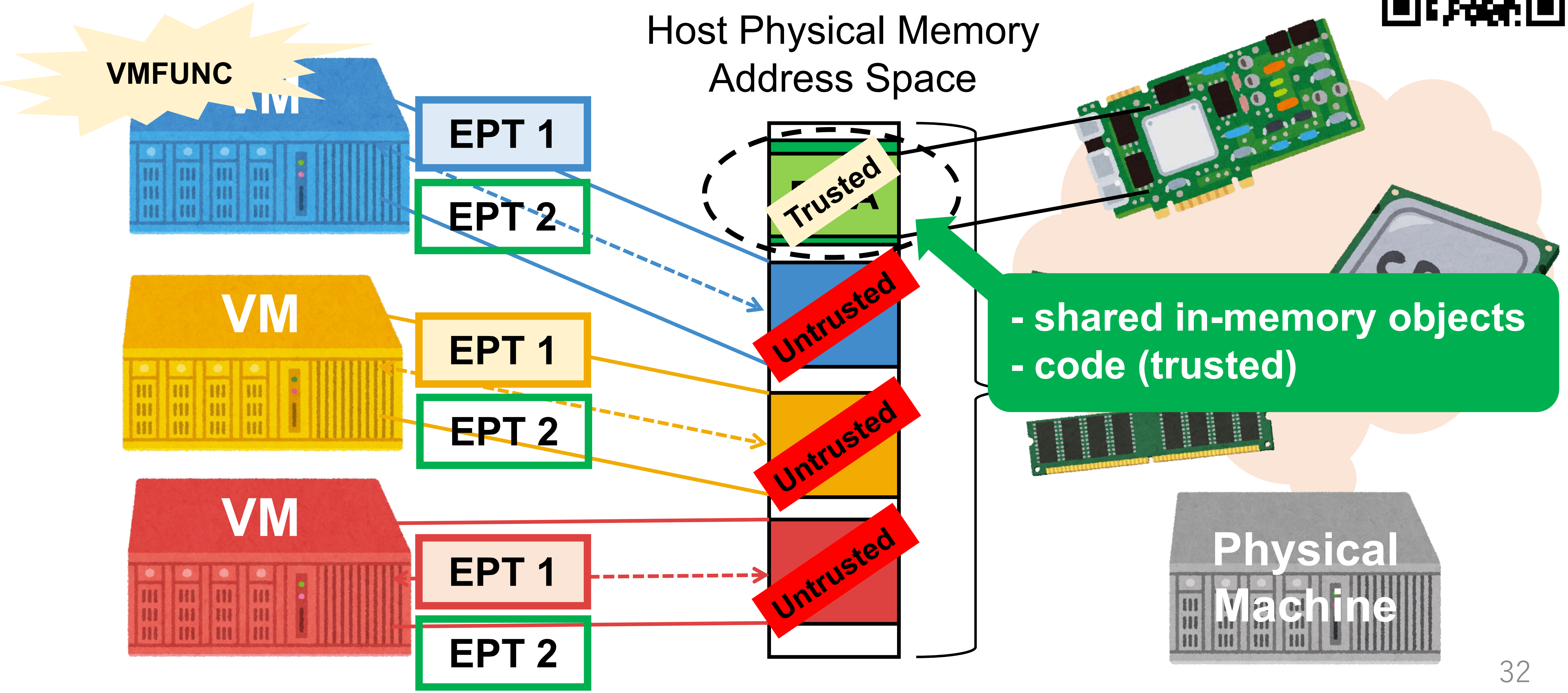
- shared in-memory objects
- code (trusted)

ELISA: Access to a Shared Object



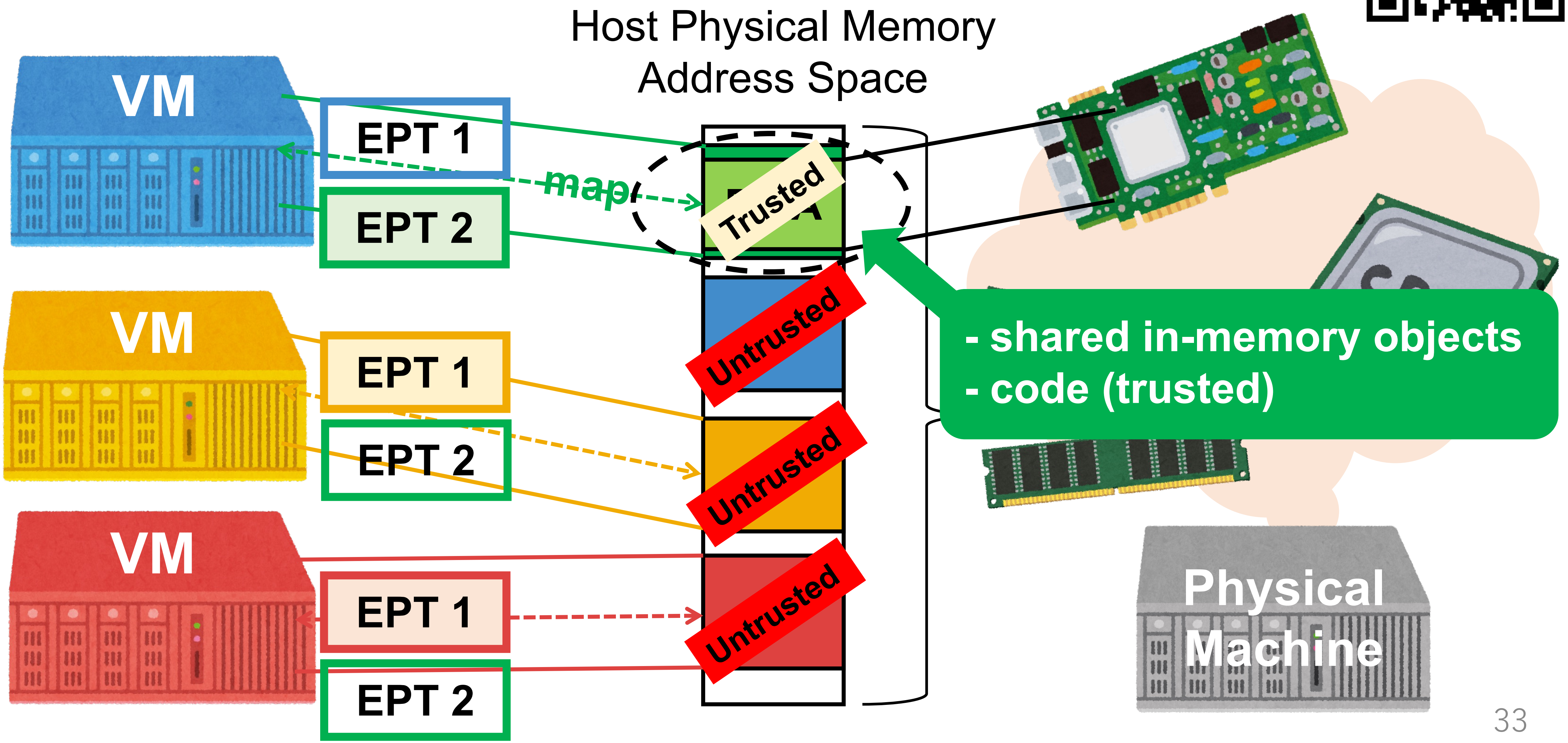
- shared in-memory objects
- code (trusted)

ELISA: Access to a Shared Object



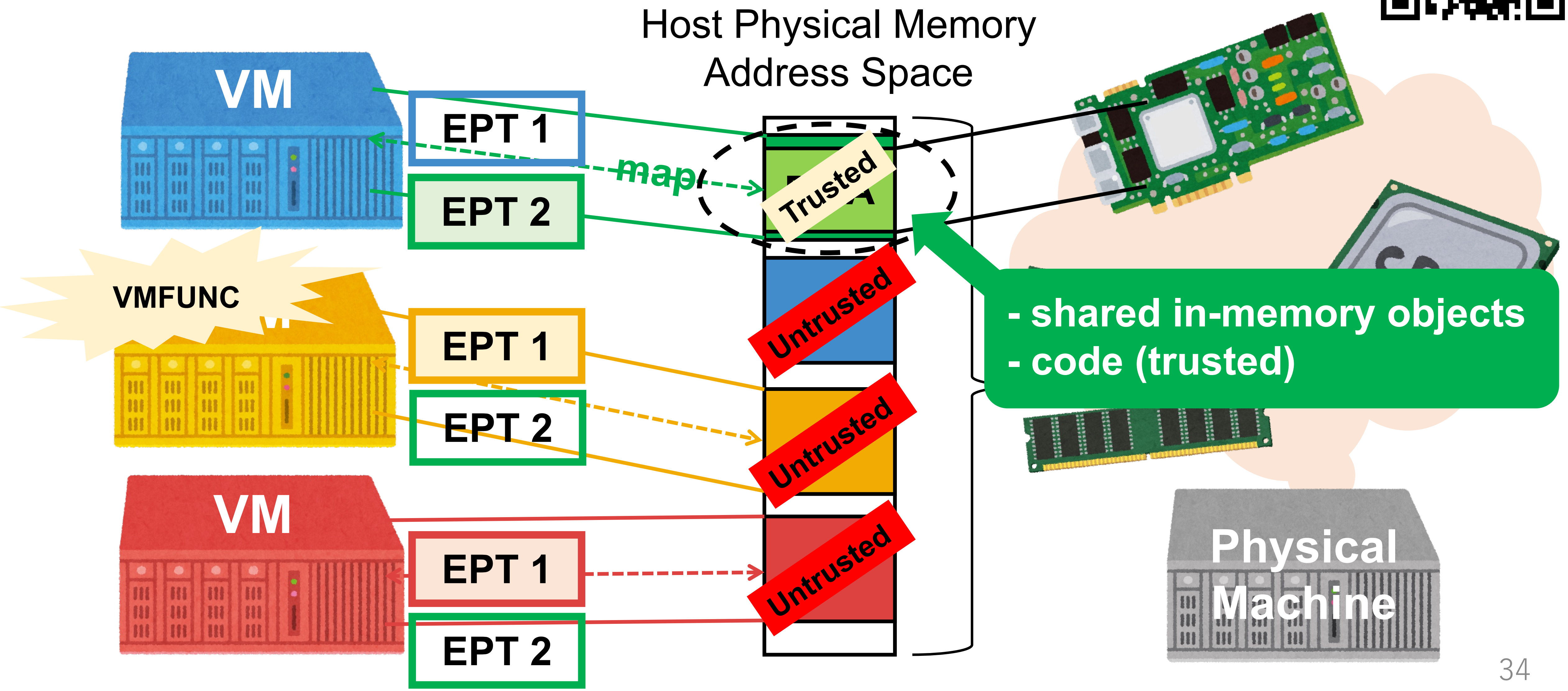
- shared in-memory objects
- code (trusted)

ELISA: Access to a Shared Object



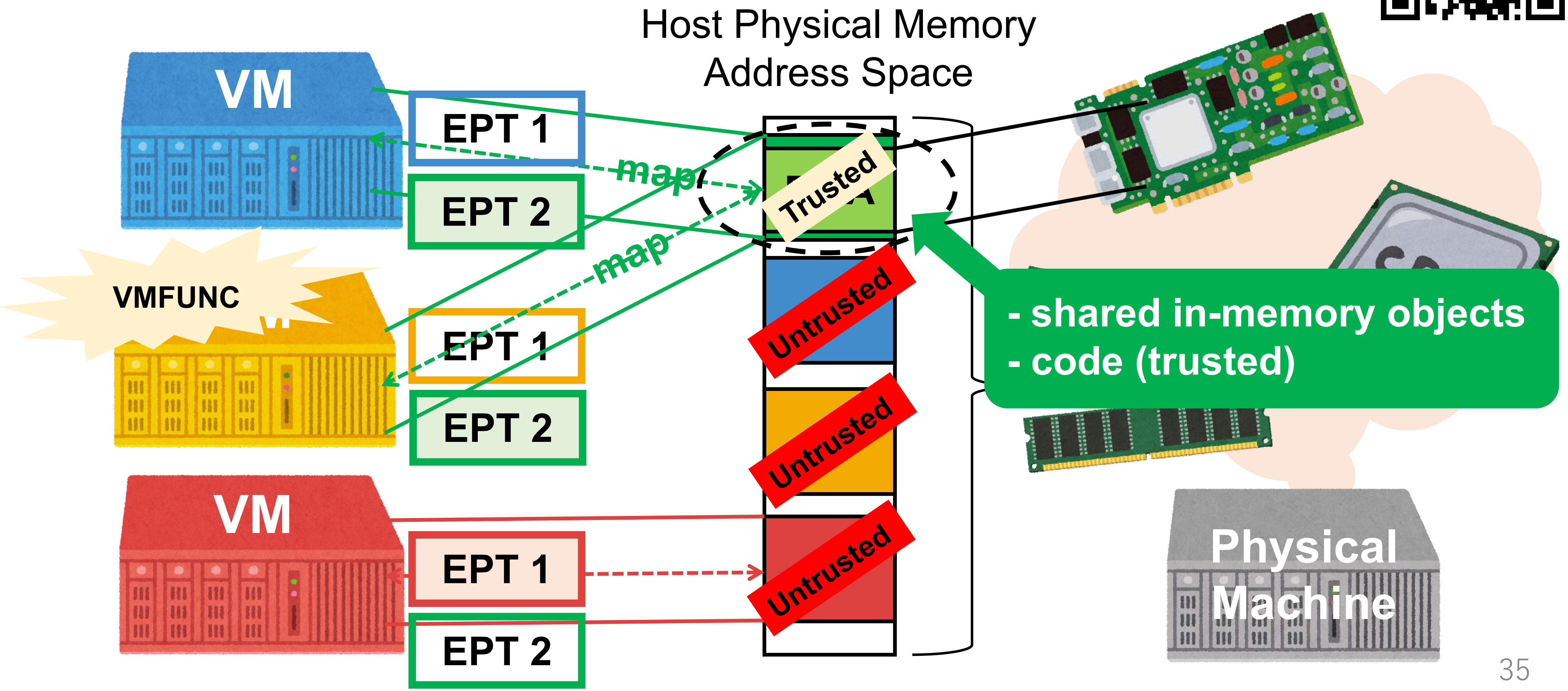
- shared in-memory objects
- code (trusted)

ELISA: Access to a Shared Object



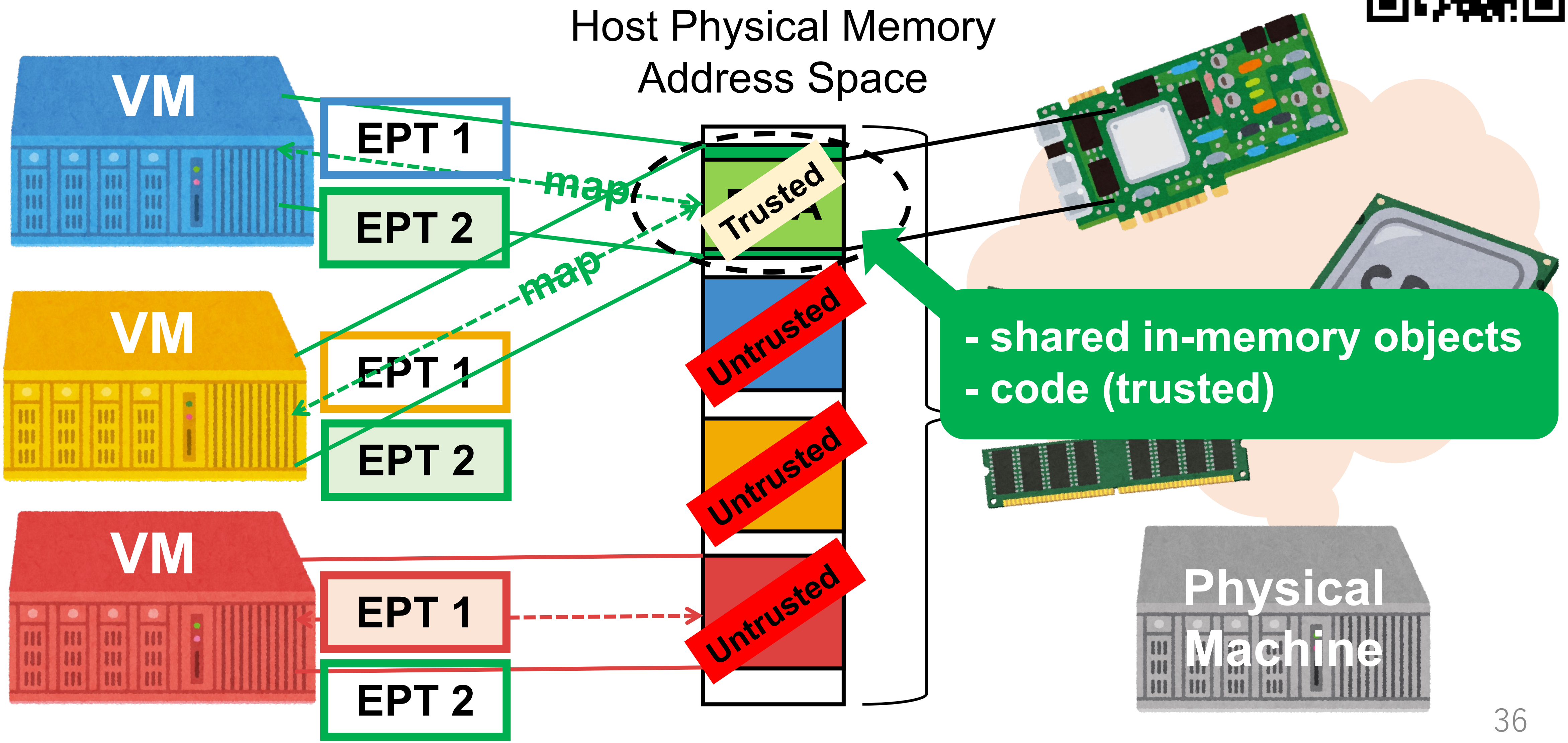
- shared in-memory objects
- code (trusted)

ELISA: Access to a Shared Object



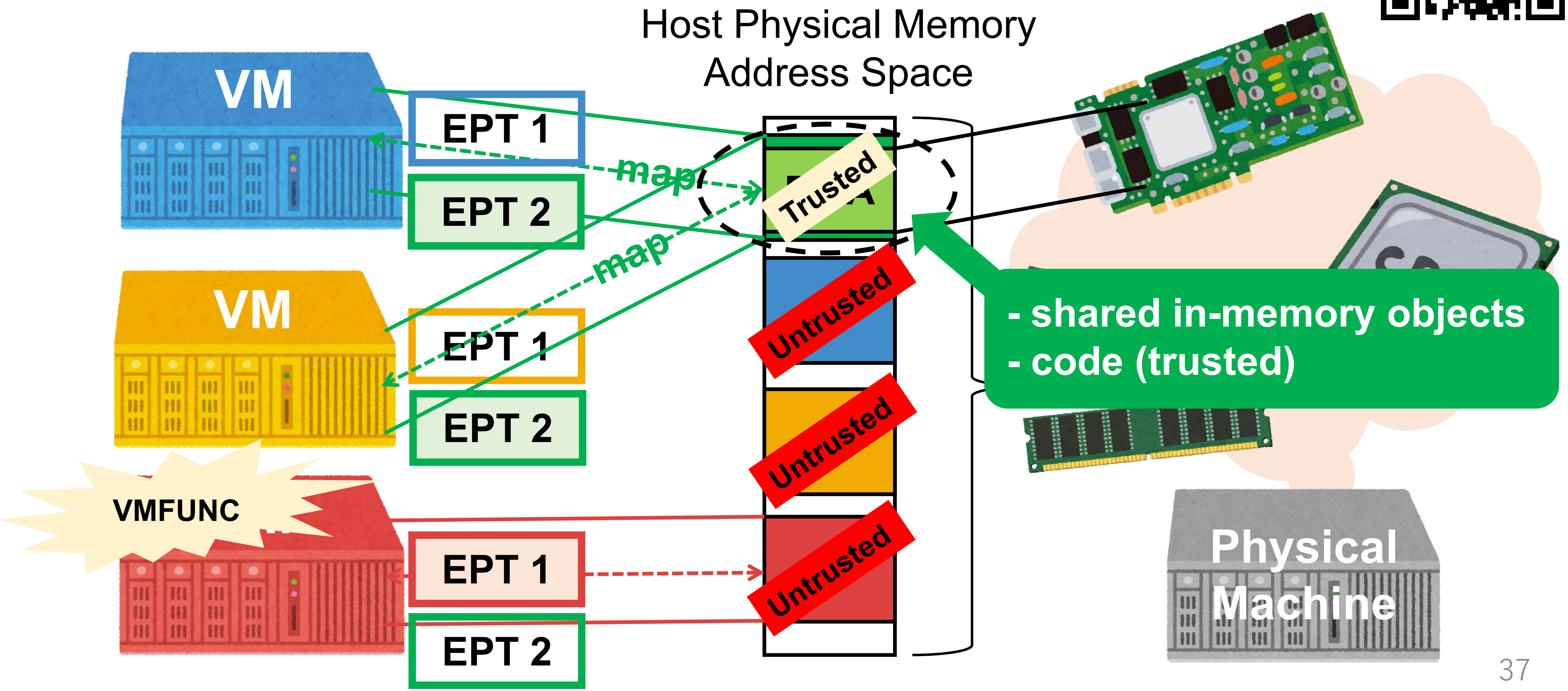
- shared in-memory objects
- code (trusted)

ELISA: Access to a Shared Object



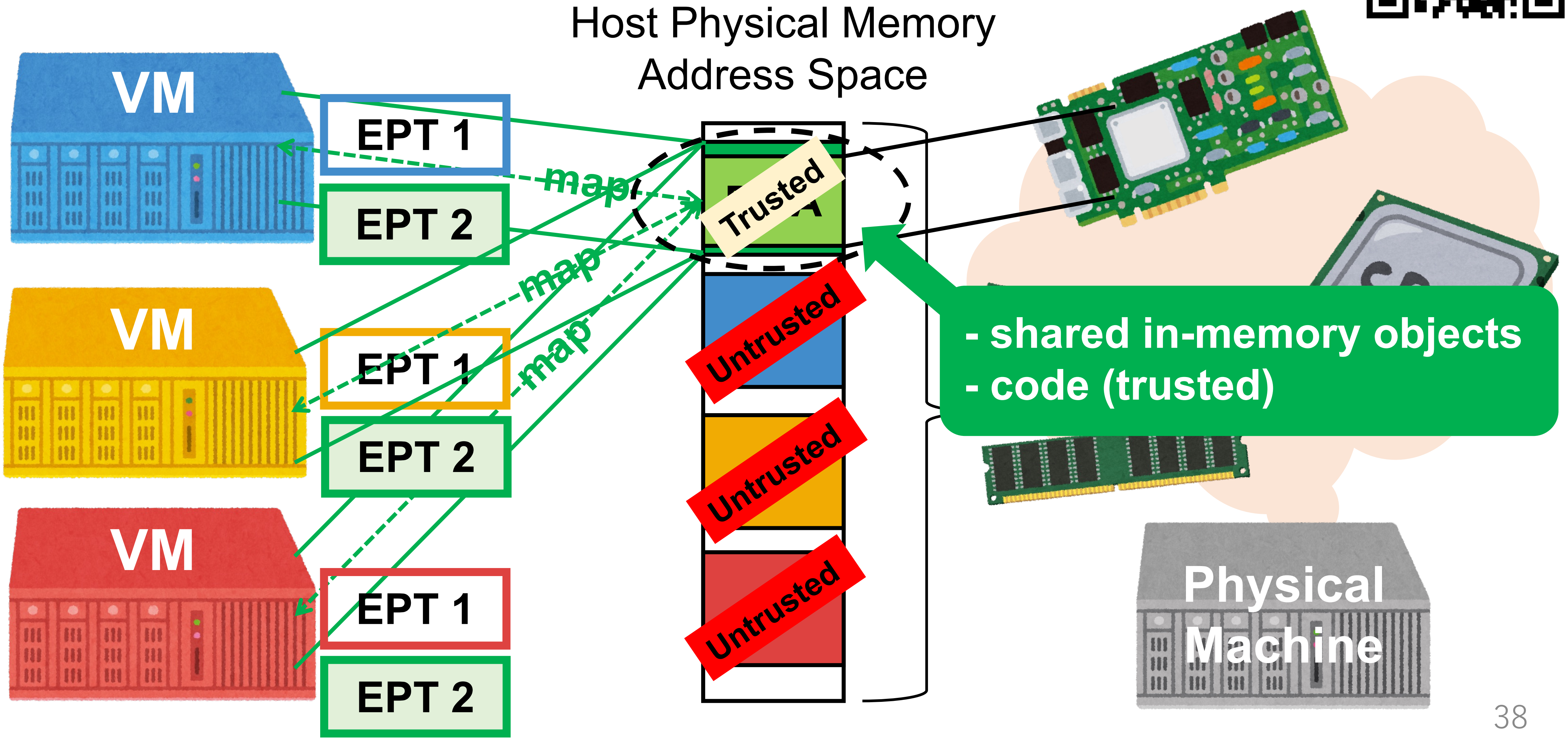
- shared in-memory objects
- code (trusted)

ELISA: Access to a Shared Object



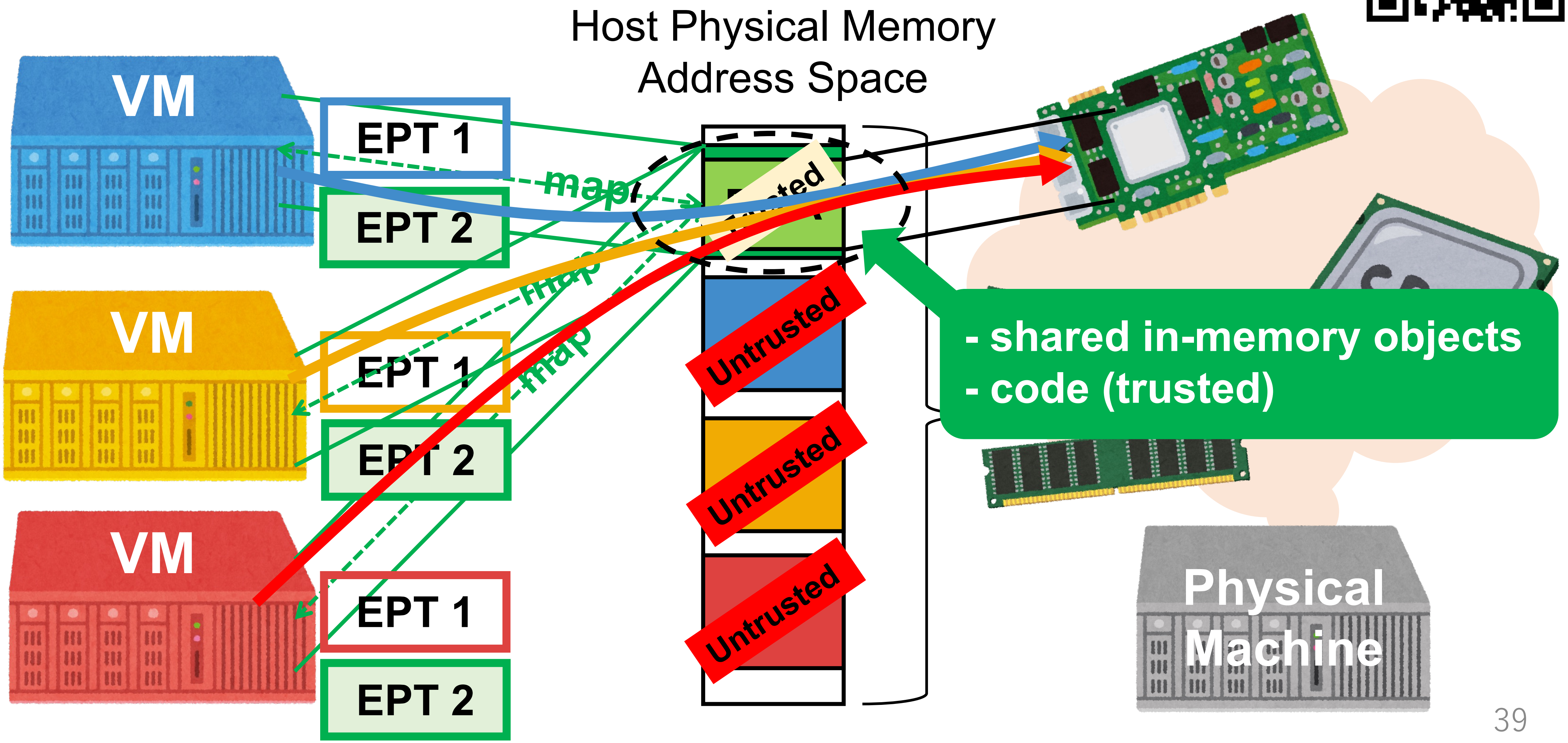
- shared in-memory objects
- code (trusted)

ELISA: Access to a Shared Object



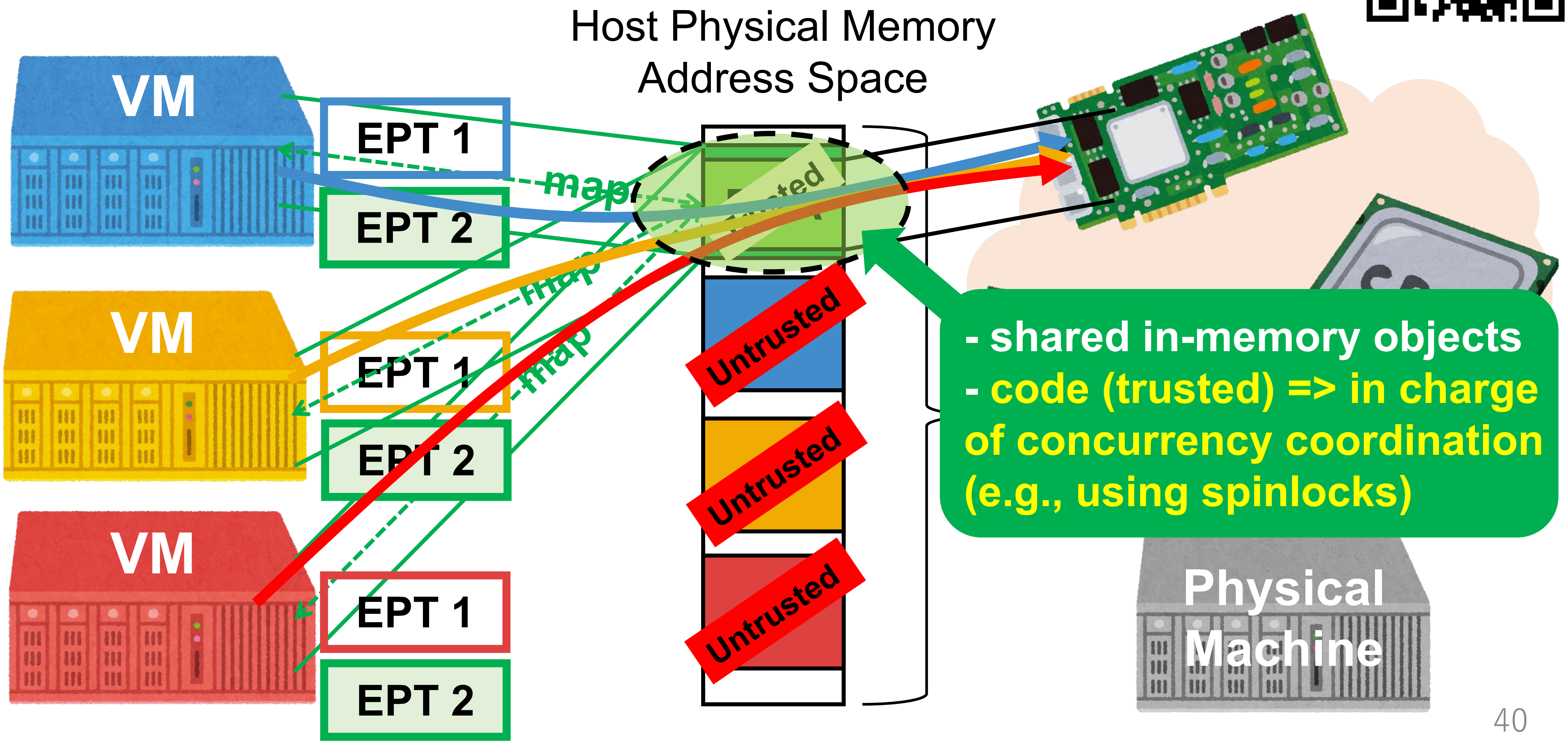
- shared in-memory objects
- code (trusted)

ELISA: Access to a Shared Object



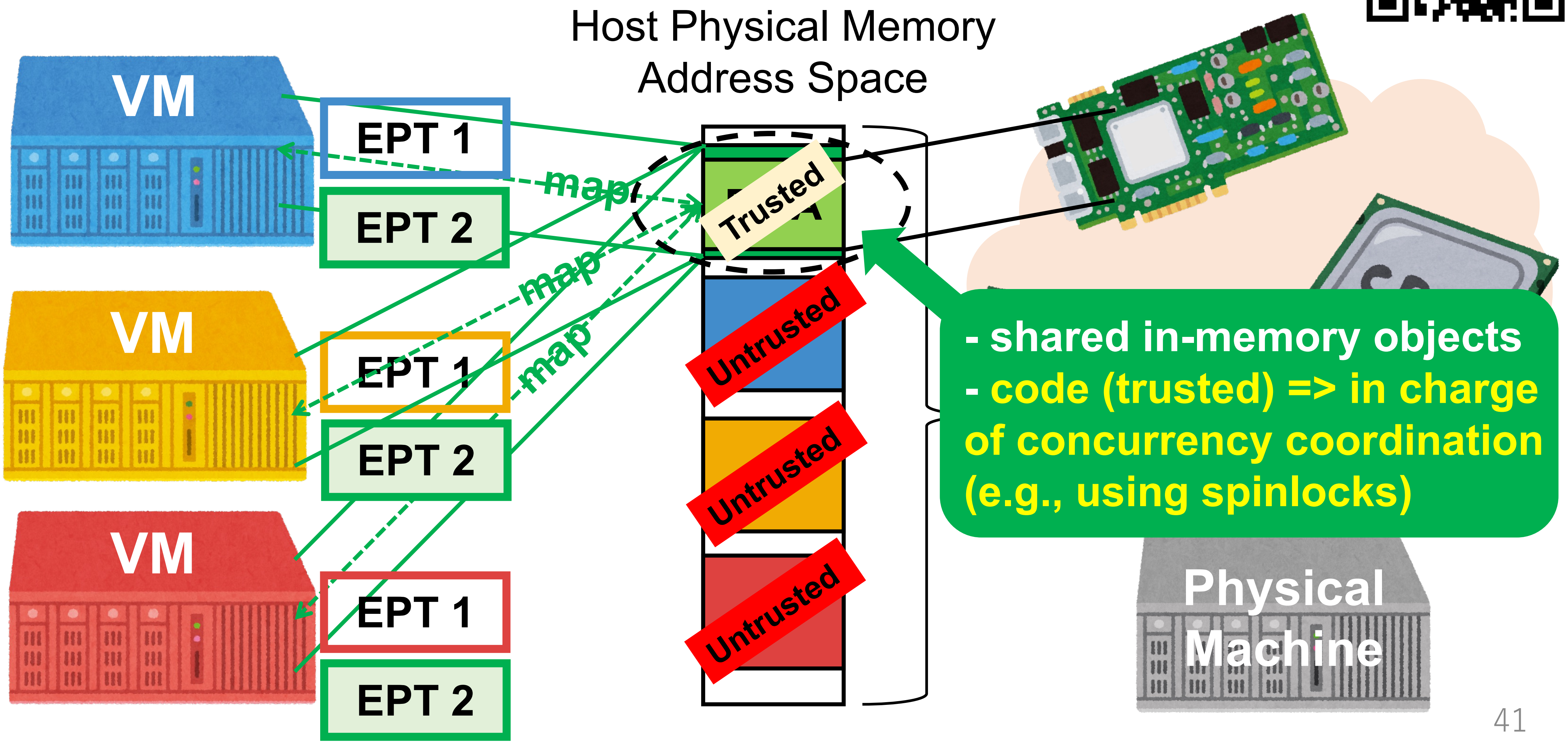
- shared in-memory objects
- code (trusted)

ELISA: Access to a Shared Object



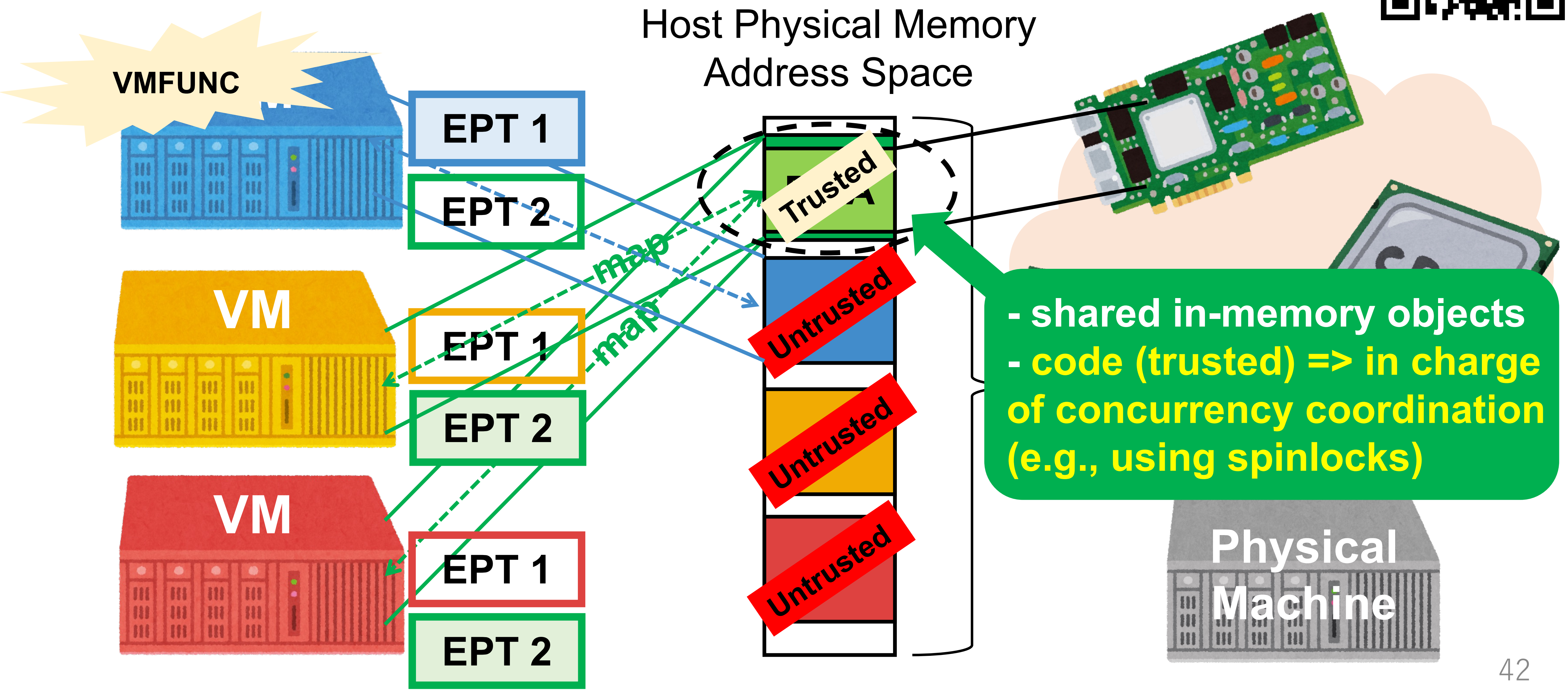
- shared in-memory objects
- code (trusted) => in charge of concurrency coordination (e.g., using spinlocks)

ELISA: Access to a Shared Object



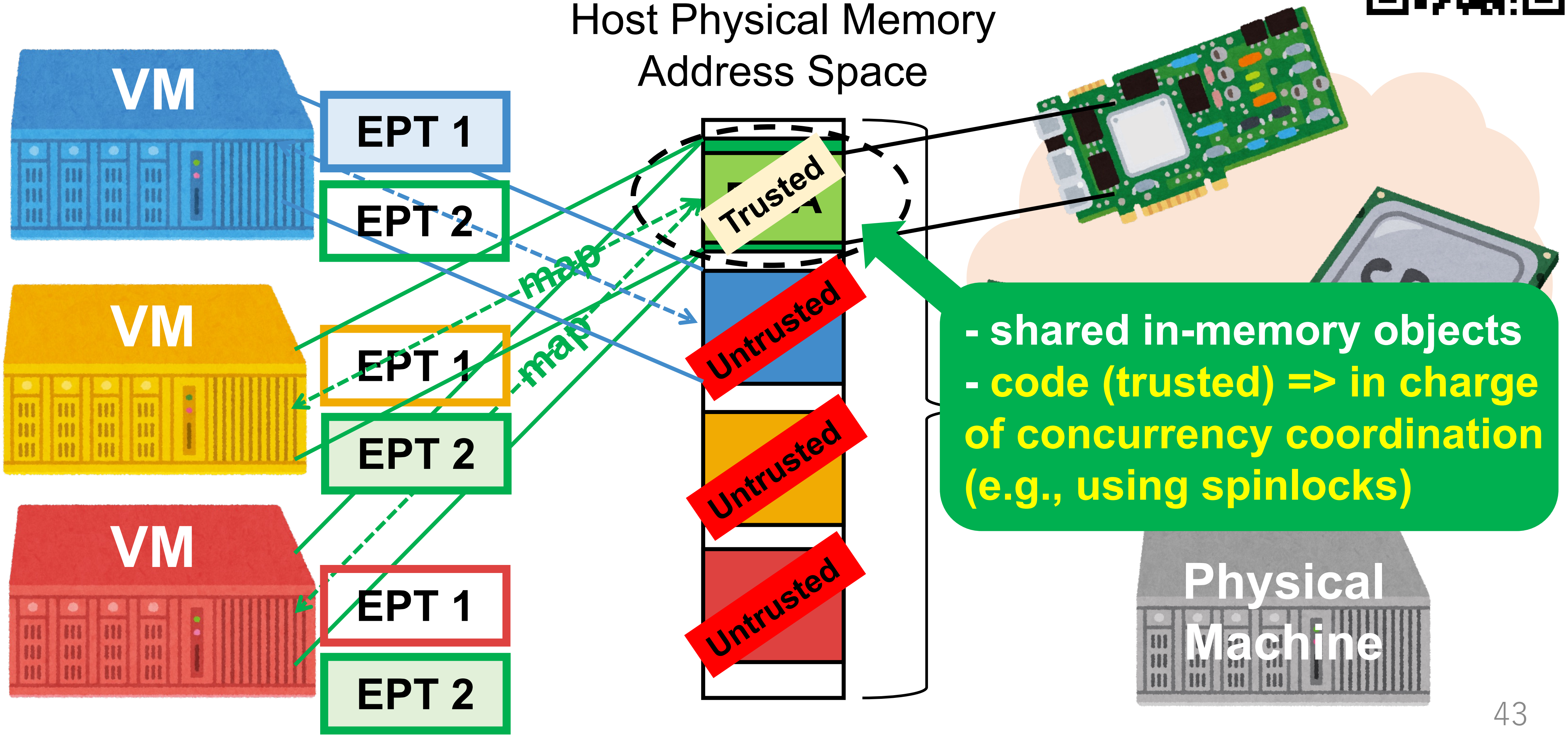
- shared in-memory objects
- code (trusted) => in charge of concurrency coordination (e.g., using spinlocks)

ELISA: Access to a Shared Object



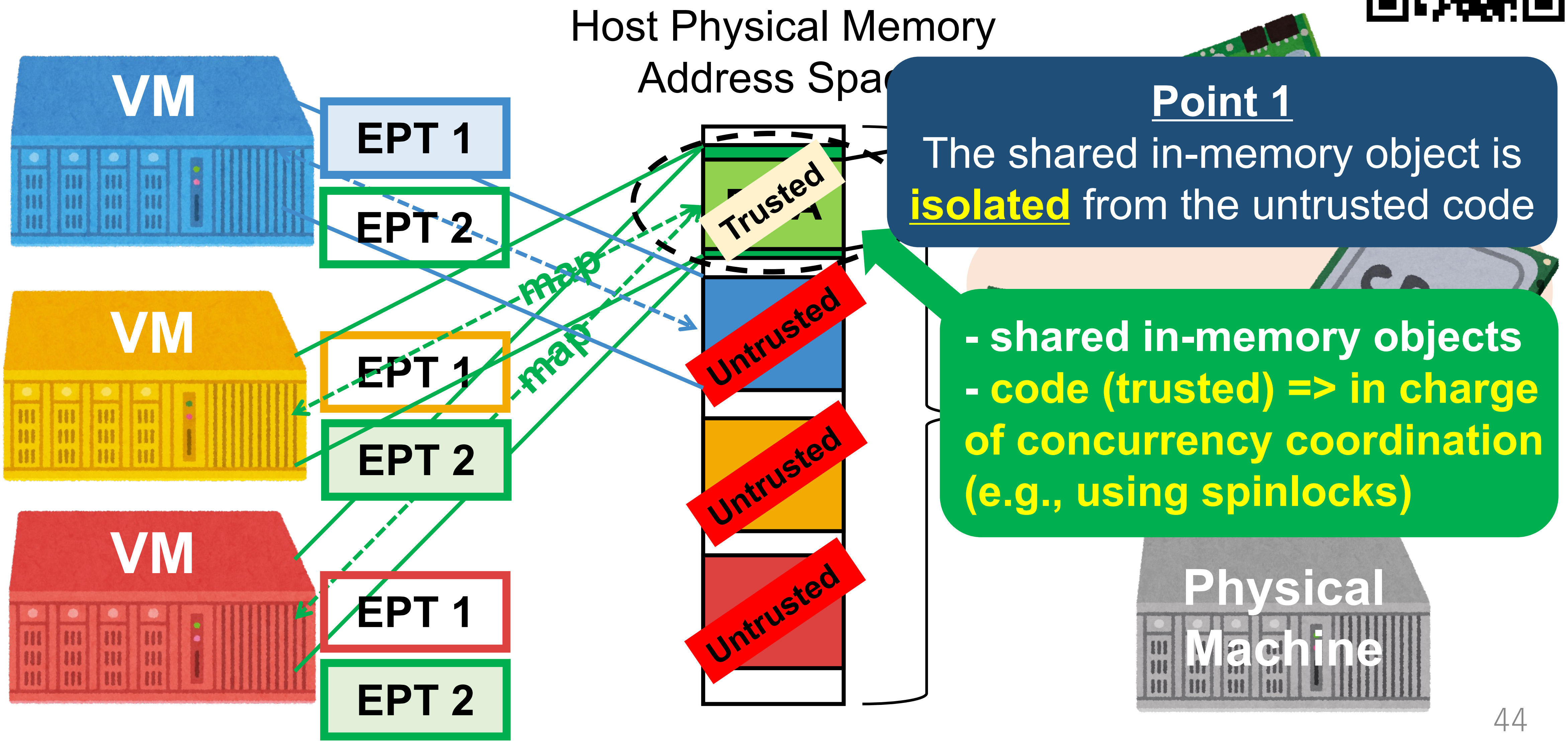
- shared in-memory objects
- code (trusted) => in charge of concurrency coordination (e.g., using spinlocks)

ELISA: Access to a Shared Object



- shared in-memory objects
- code (trusted) => in charge of concurrency coordination (e.g., using spinlocks)

ELISA: Access to a Shared Object



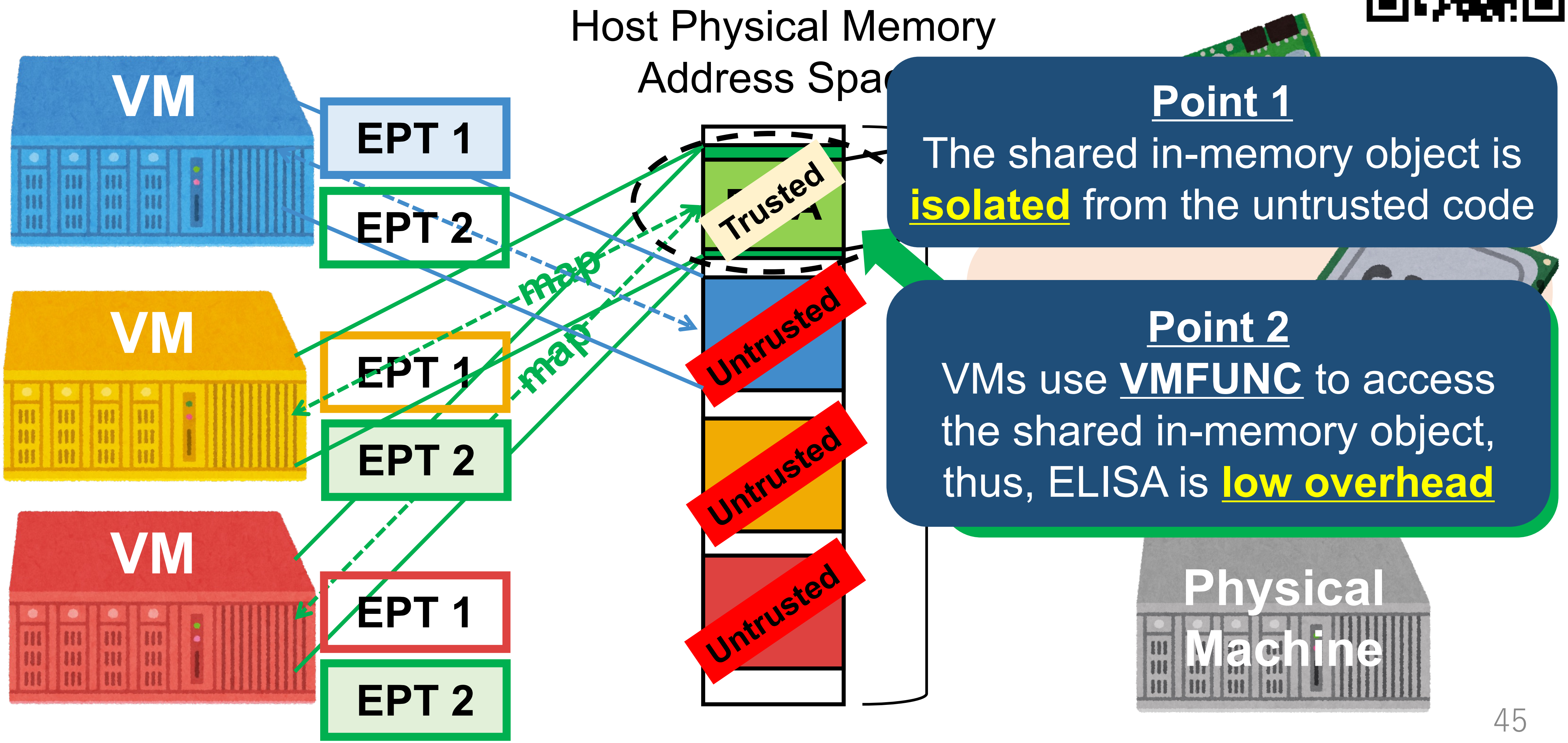
Point 1

The shared in-memory object is **isolated** from the untrusted code

- shared in-memory objects

- code (trusted) => in charge of concurrency coordination (e.g., using spinlocks)

ELISA: Access to a Shared Object



Point 1

The shared in-memory object is **isolated** from the untrusted code

Point 2

VMs use **VMFUNC** to access the shared in-memory object, thus, ELISA is **low overhead**

Contributions of the Paper

Contributions of the Paper

- Challenging issues

Contributions of the Paper

- Challenging issues
  - Issue: page table maintenance overhead to keep page tables at the same Guest Physical Address in default and non-default EPT contexts
    ➔ Solution: Anywhere Page Table (APT)

Contributions of the Paper

- Challenging issues
  - Issue: page table maintenance overhead to keep page tables at the same Guest Physical Address in default and non-default EPT contexts
    ➔ Solution: Anywhere Page Table (APT)
  - Issue: potential attack enabled by the combination of VMFUNC and untrusted guest kernels
    ➔ Solution: gate EPT context

Contributions of the Paper

- Challenging issues
  - Issue: page table maintenance overhead to keep page tables at the same Guest Physical Address in default and non-default EPT contexts
    ➔ Solution: Anywhere Page Table (APT)
  - Issue: potential attack enabled by the combination of VMFUNC and untrusted guest kernels
    ➔ Solution: gate EPT context

- Flexible programming model

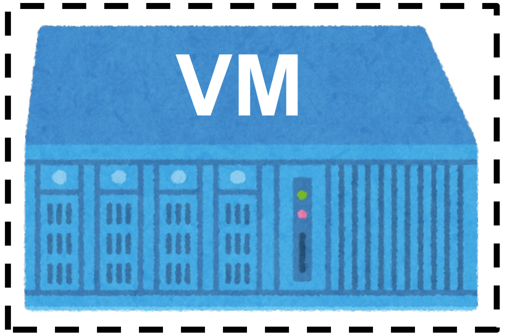
Contributions of the Paper

- Challenging issues
  - Issue: page table maintenance overhead to keep page tables at the same Guest Physical Address in default and non-default EPT contexts
➔ Solution: Anywhere Page Table (APT)
  - Issue: potential attack enabled by the combination of VMFUNC and untrusted guest kernels
➔ Solution: gate EPT context

- Flexible programming model

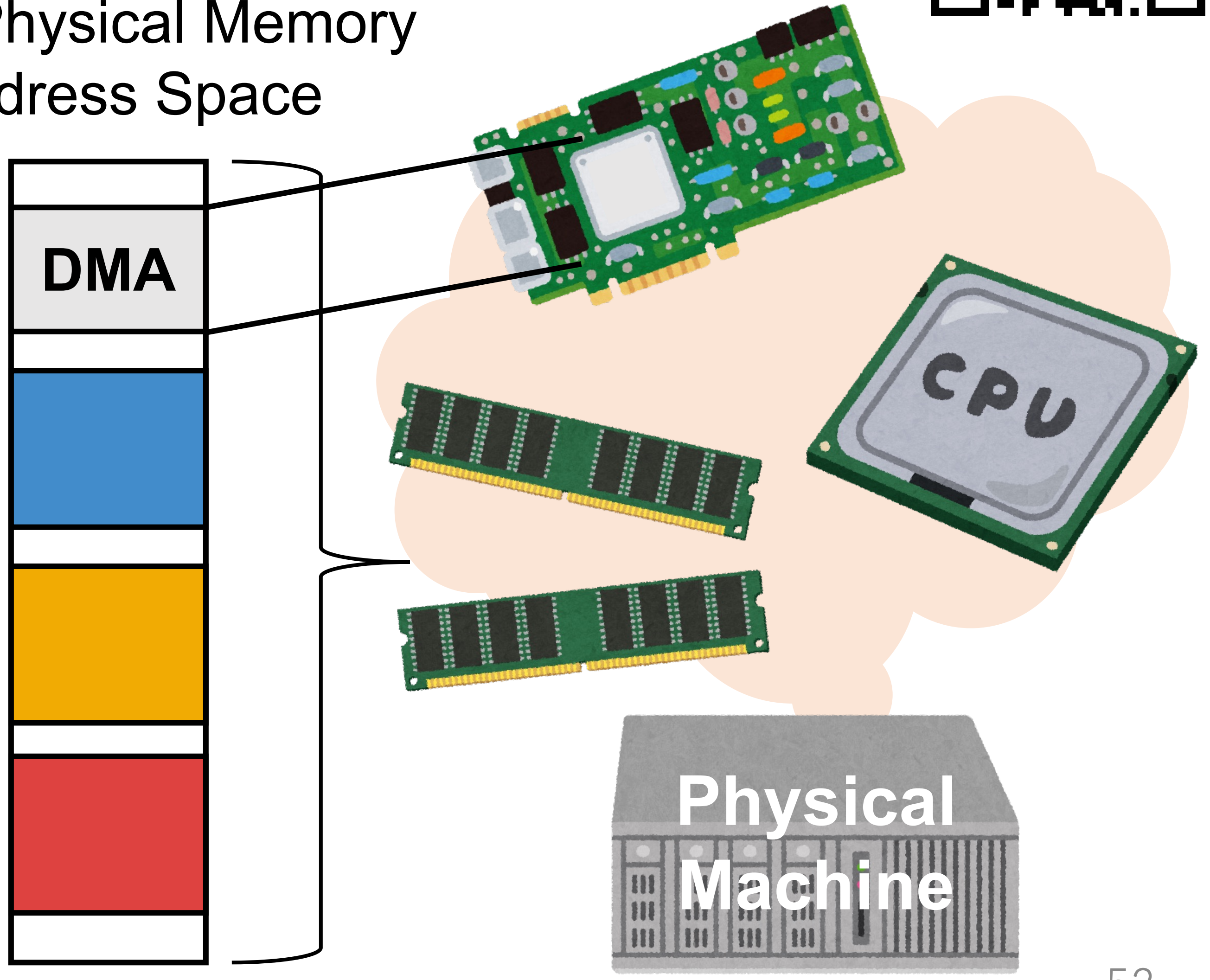
Please refer to the paper for details

Context Switch Overhead (VMCALL)

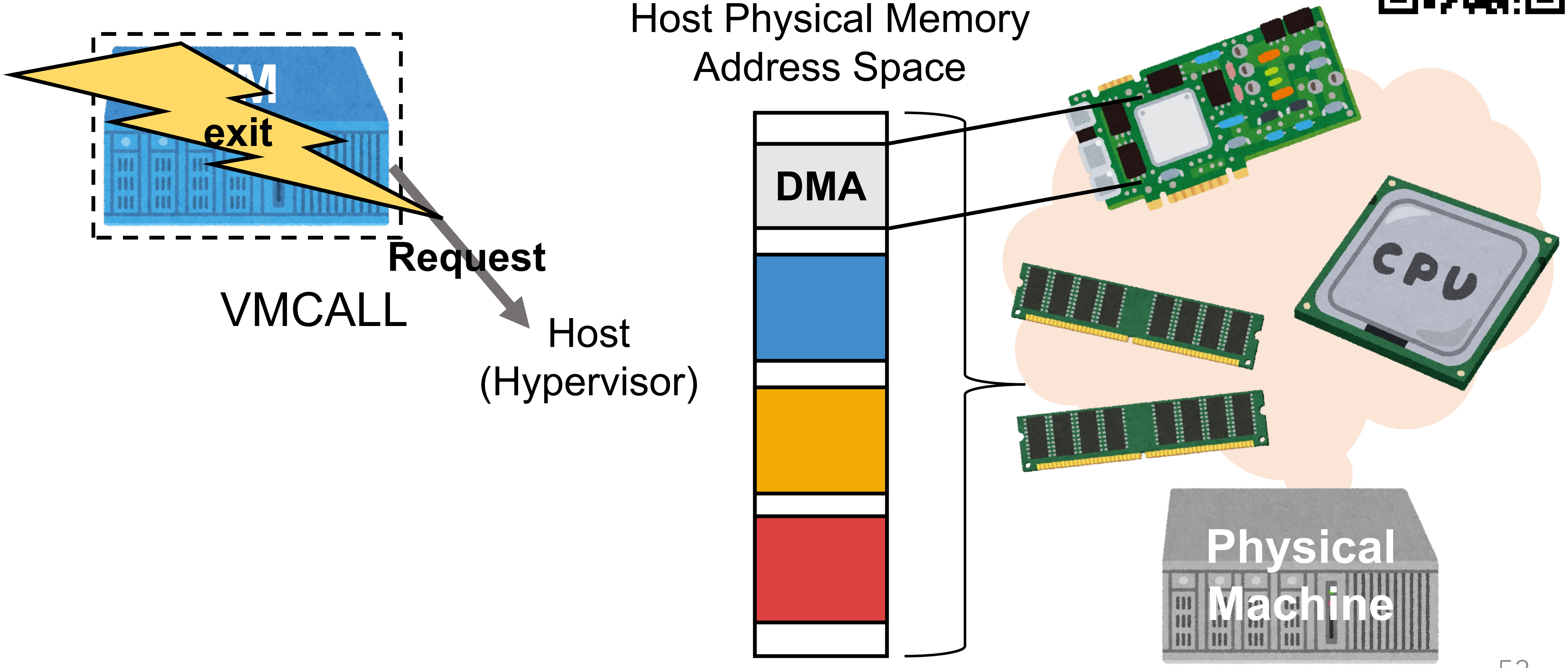


Host Physical Memory
Address Space

Host
(Hypervisor)



Context Switch Overhead (VMCALL)



Host Physical Memory
Address Space

exit

Request

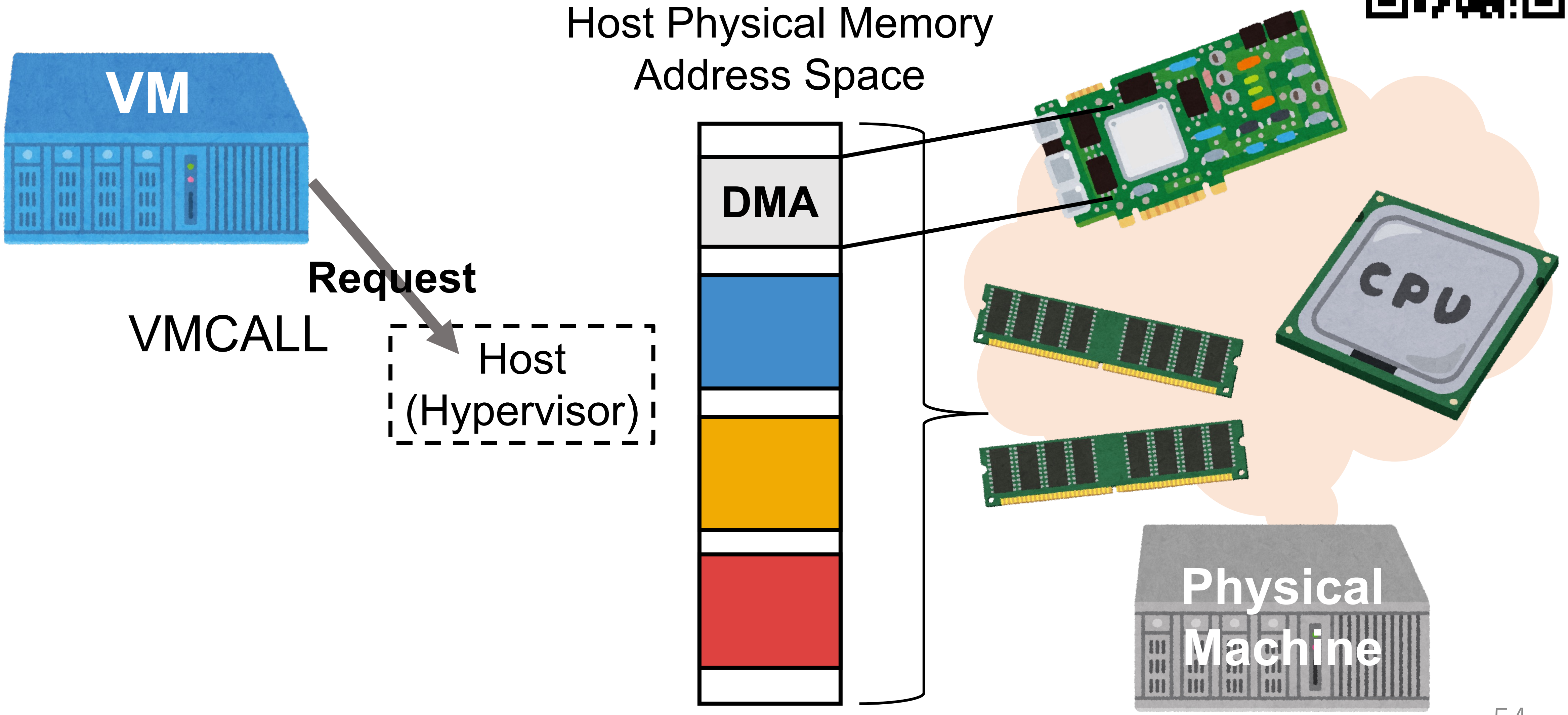
VMCALL

Host
(Hypervisor)

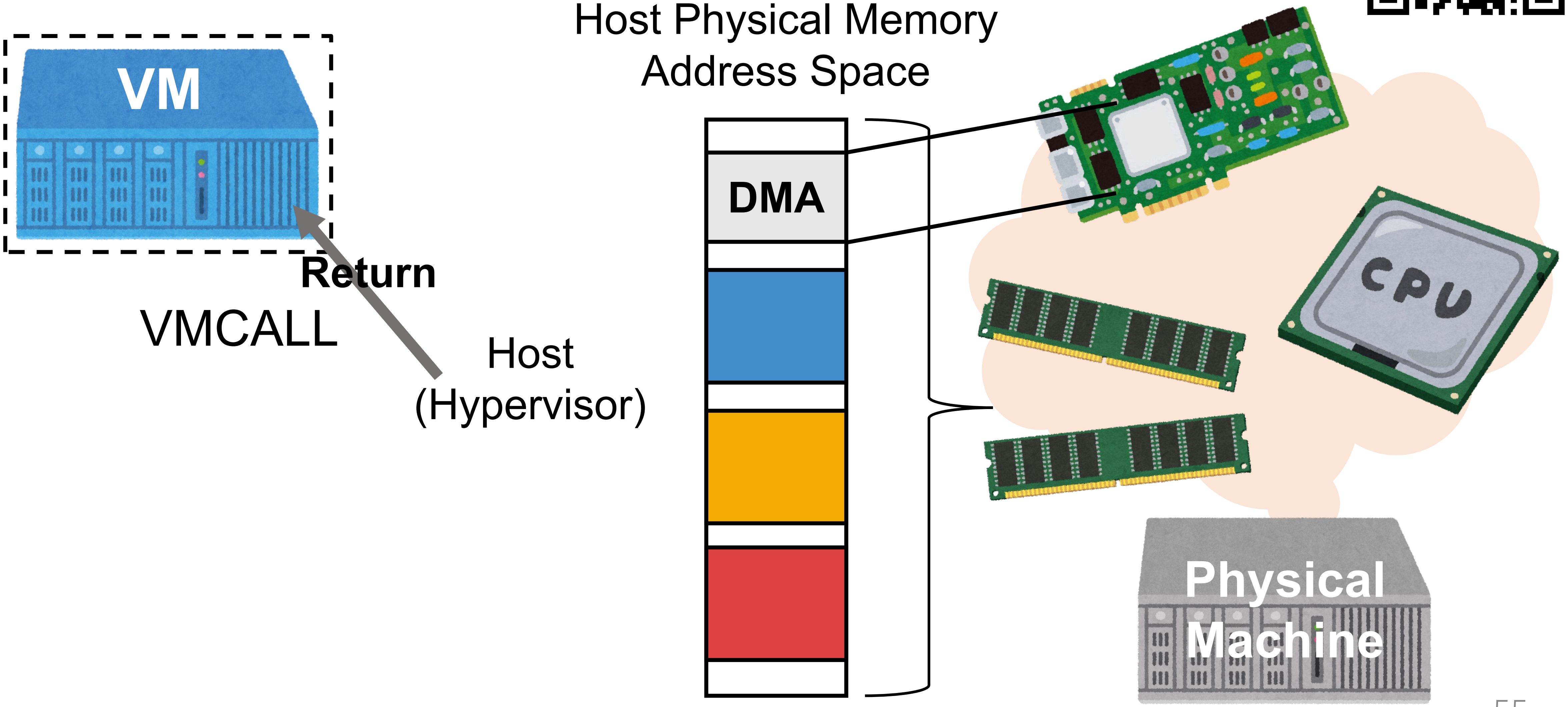
DMA

Physical
Machine

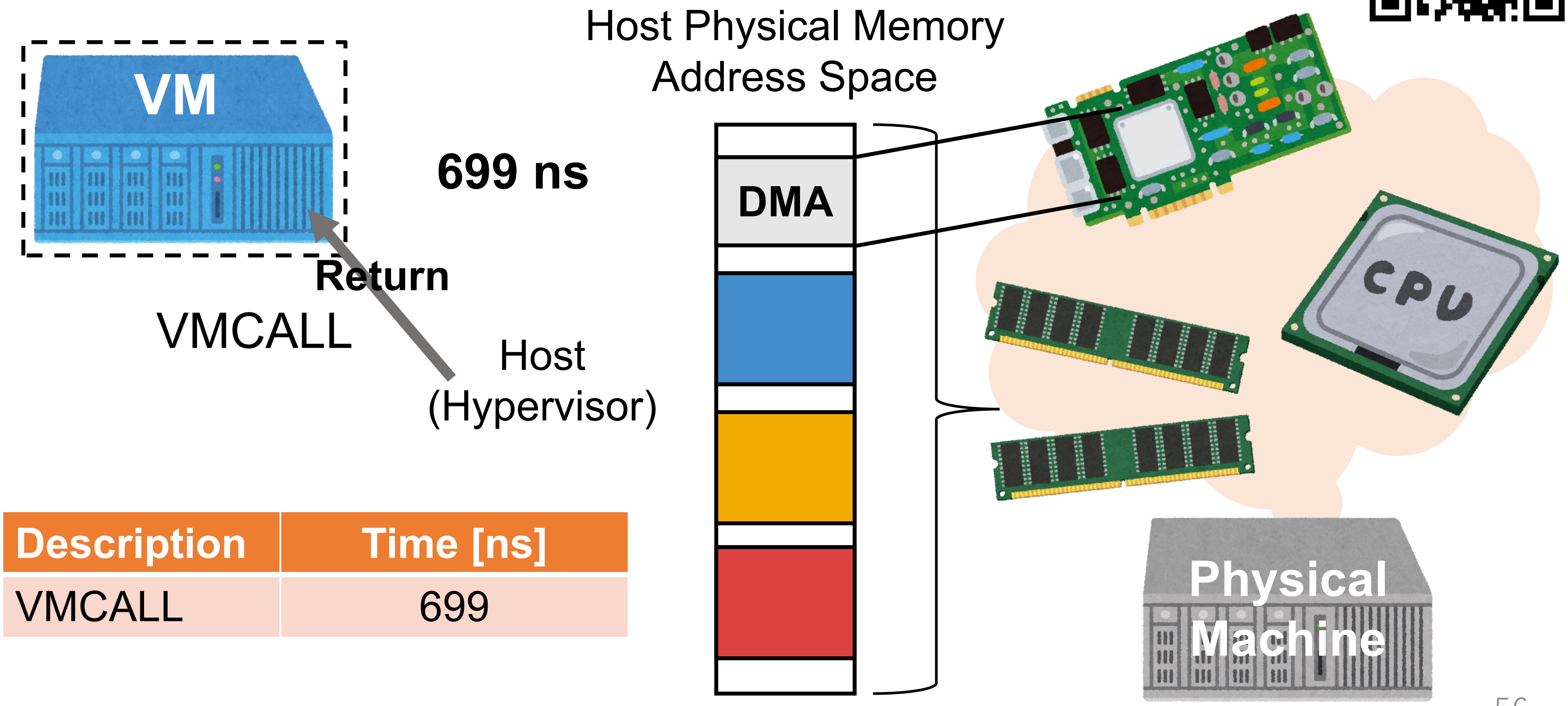
Context Switch Overhead (VMCALL)



Context Switch Overhead (VMCALL)

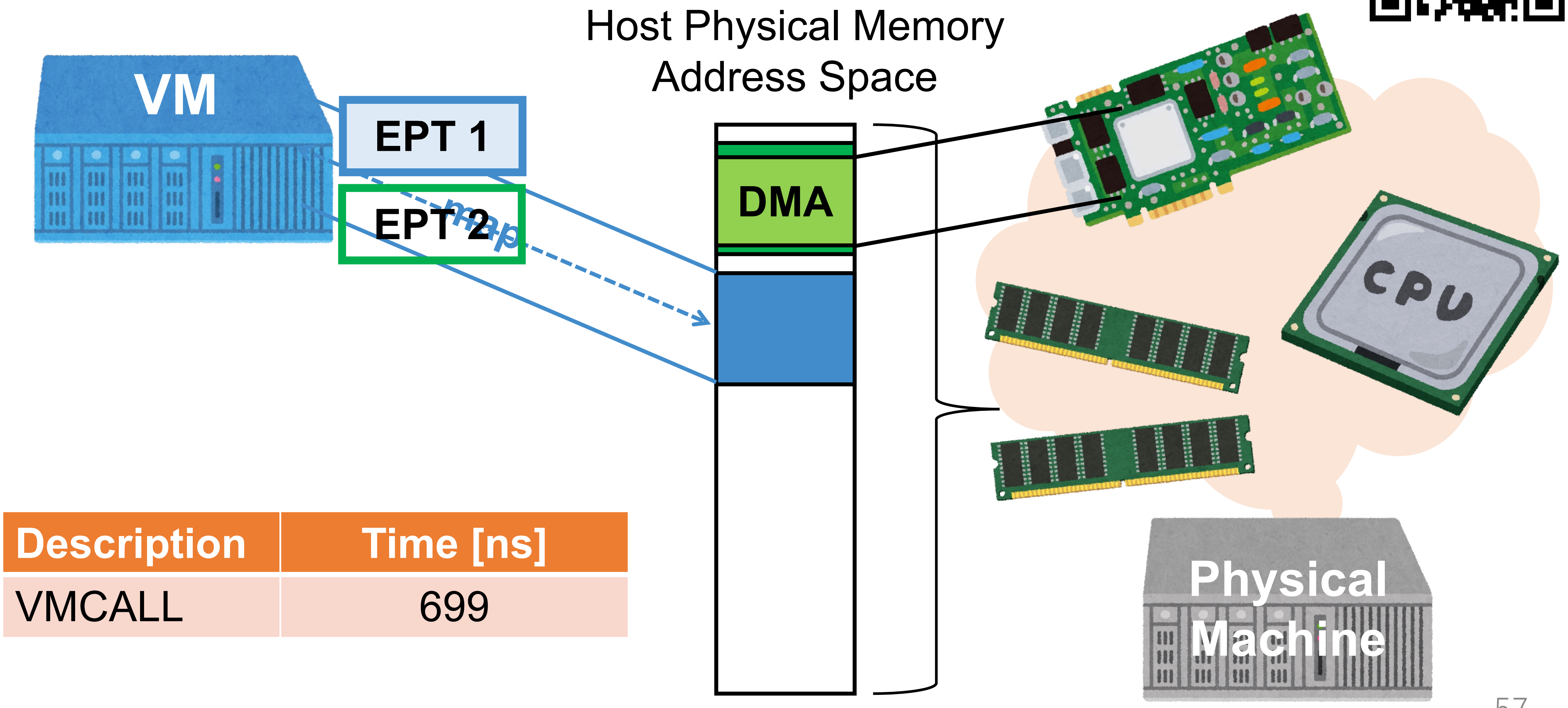


Context Switch Overhead (VMCALL)



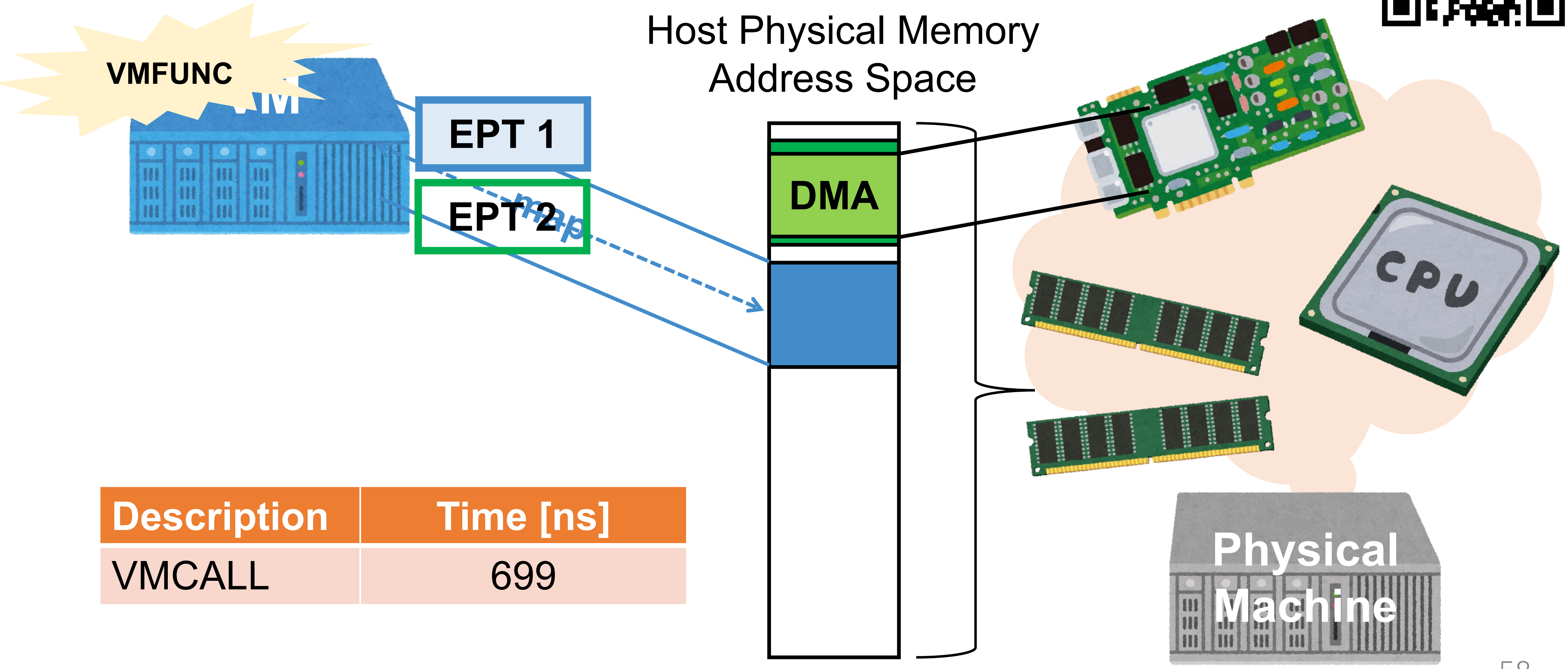
Description	Time [ns]
VMCALL	699

Context Switch Overhead (ELISA)



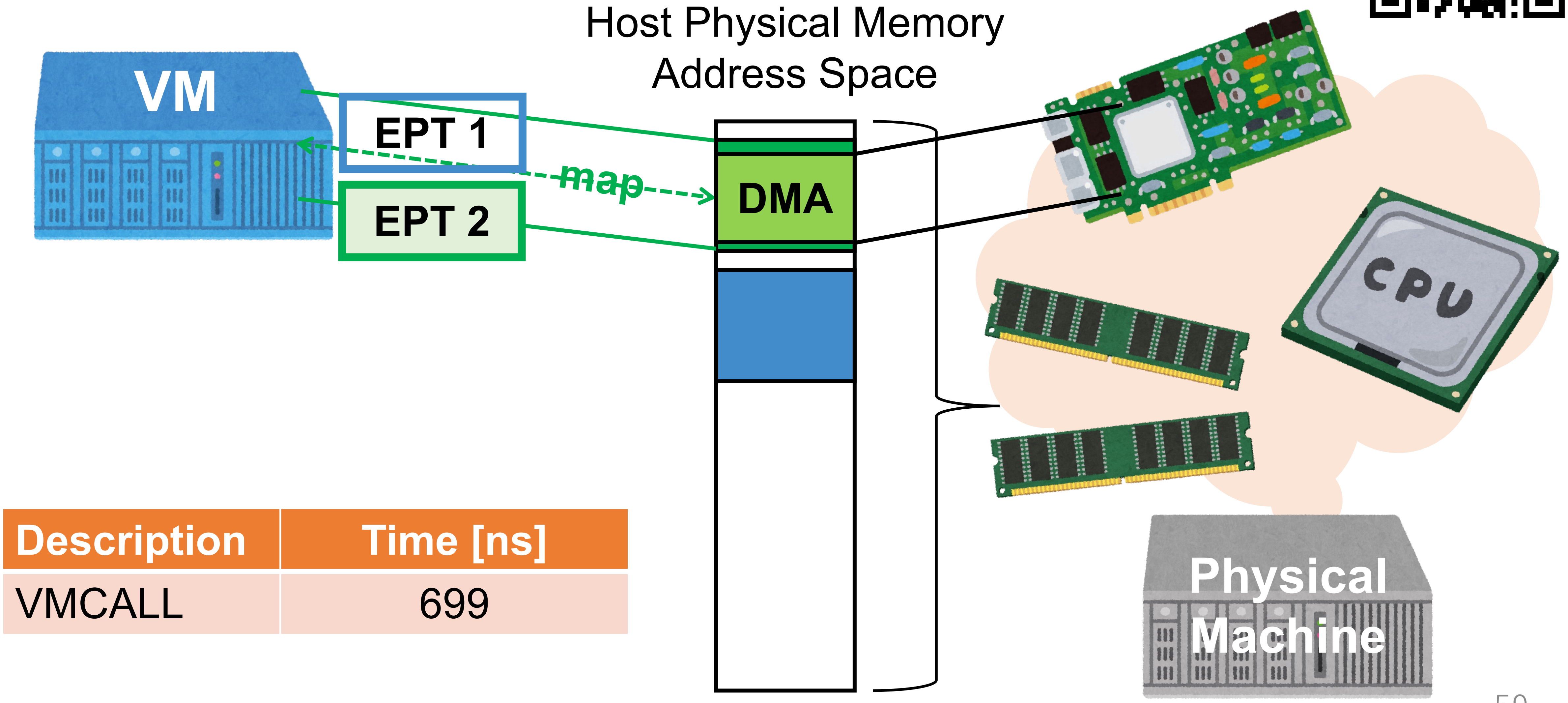
Description	Time [ns]
VMCALL	699

Context Switch Overhead (ELISA)



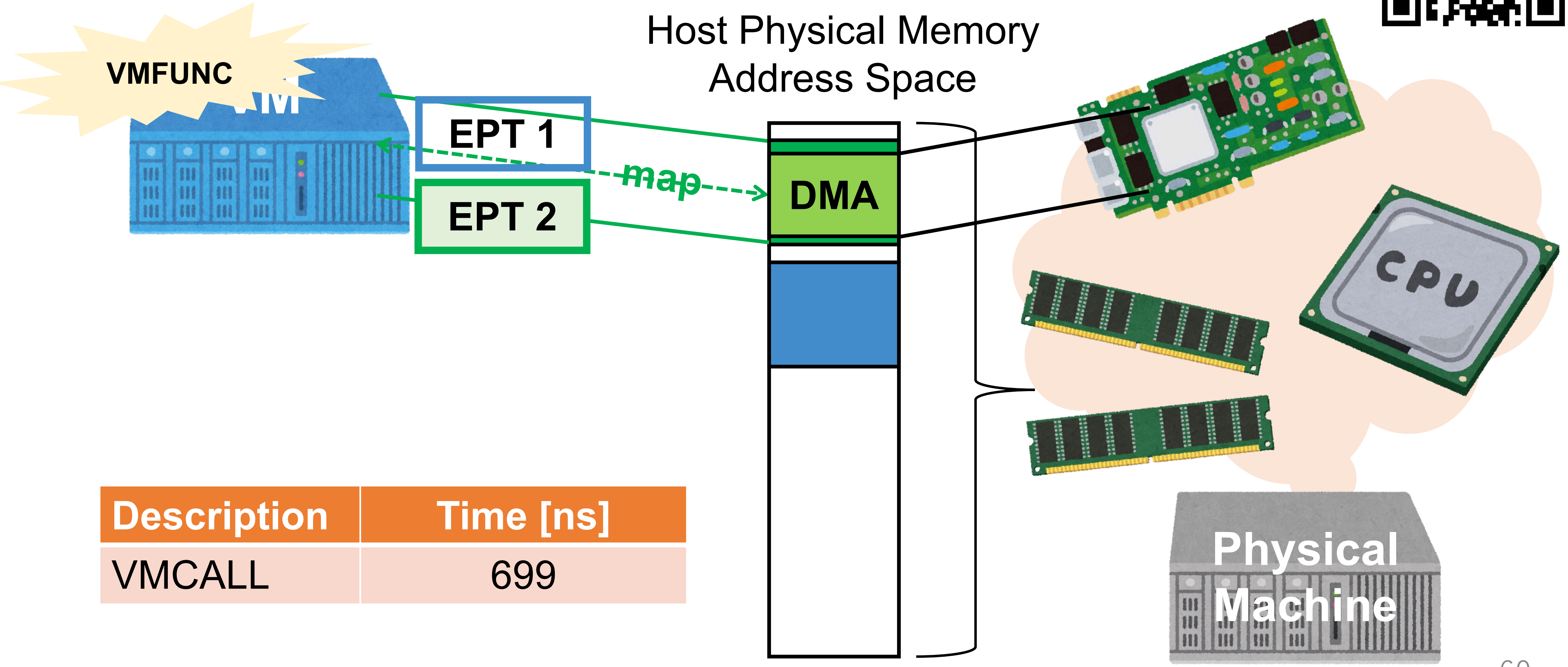
Description	Time [ns]
VMCALL	699

Context Switch Overhead (ELISA)



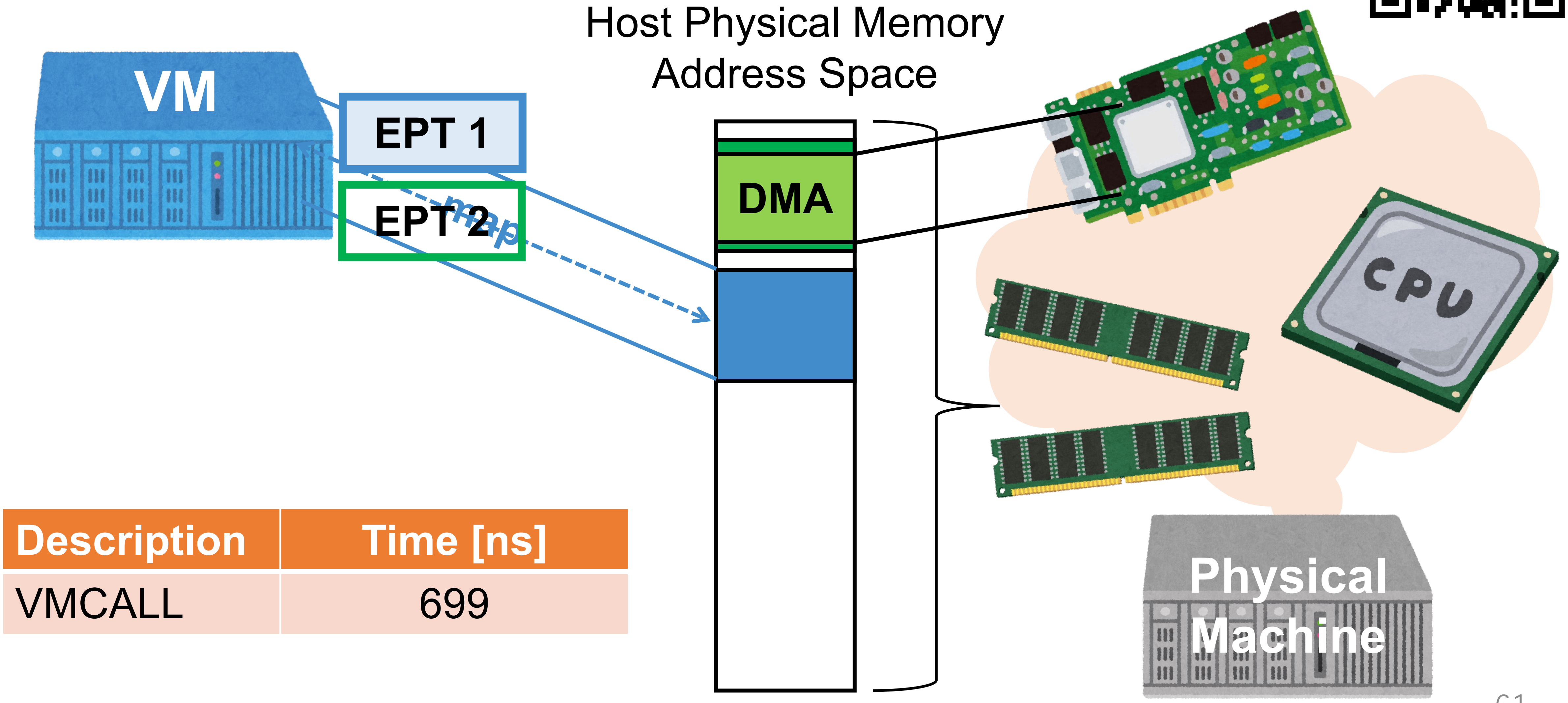
Description	Time [ns]
VMCALL	699

Context Switch Overhead (ELISA)



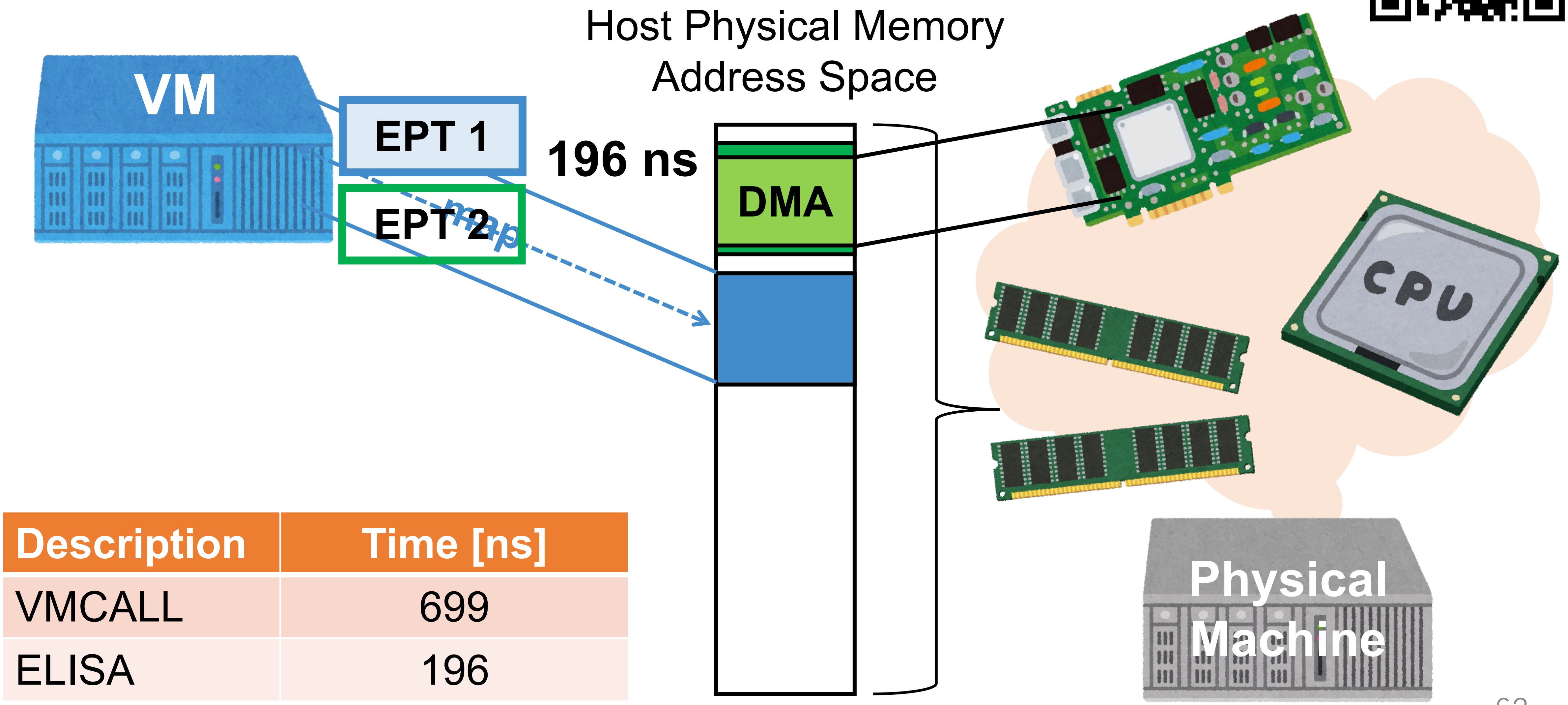
Description	Time [ns]
VMCALL	699

Context Switch Overhead (ELISA)



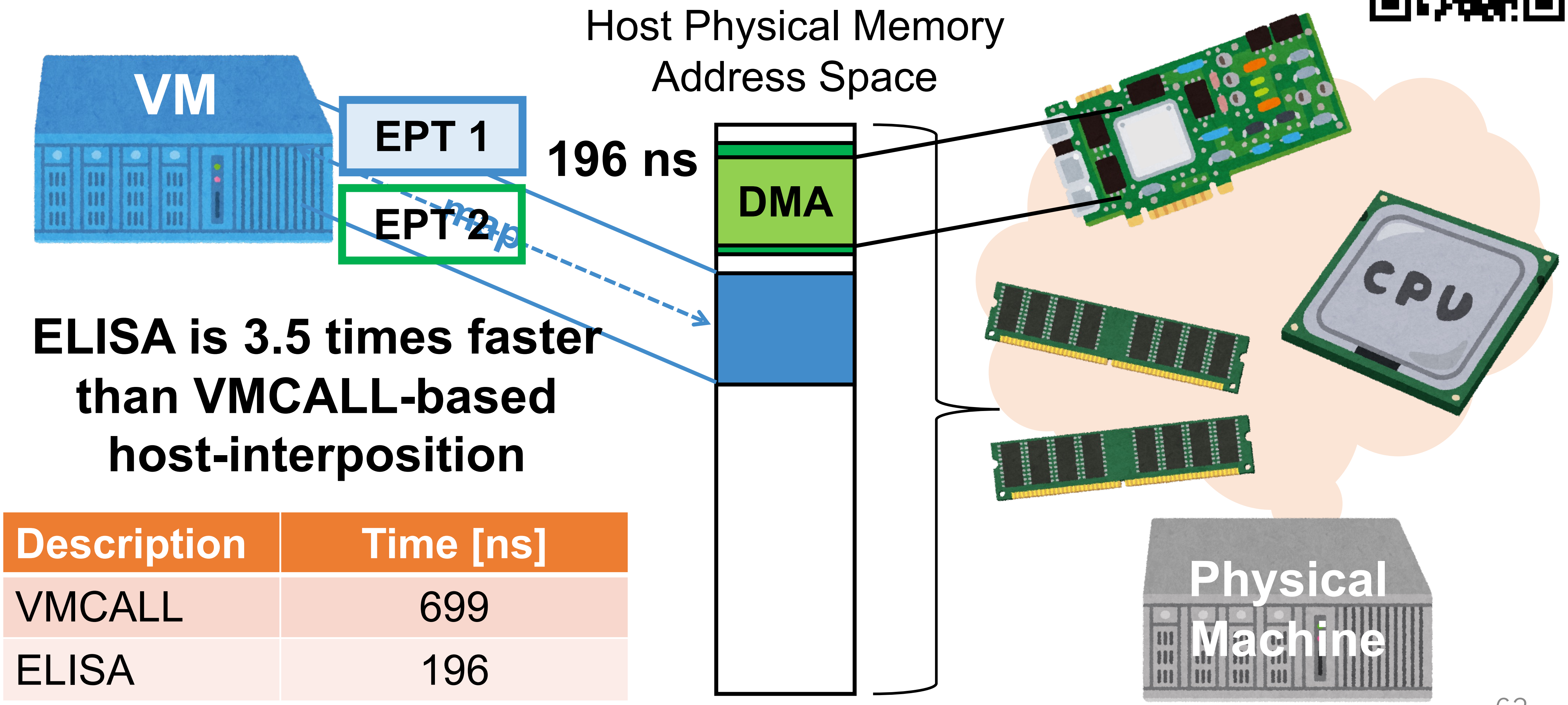
Description	Time [ns]
VMCALL	699

Context Switch Overhead (ELISA)



Description	Time [ns]
VMCALL	699
ELISA	196

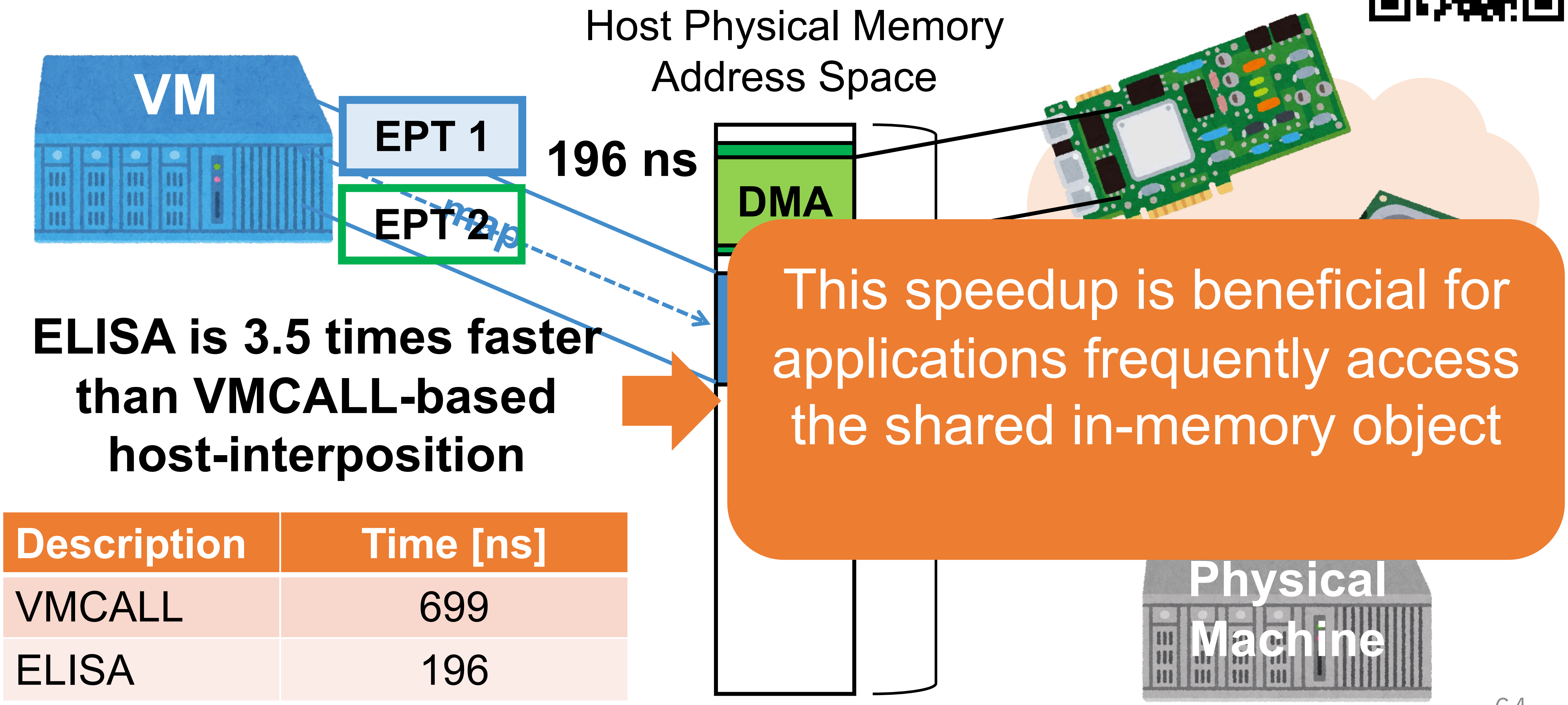
Context Switch Overhead (ELISA)



ELISA is 3.5 times faster than VMCALL-based host-interposition

Description	Time [ns]
VMCALL	699
ELISA	196

Context Switch Overhead (ELISA)

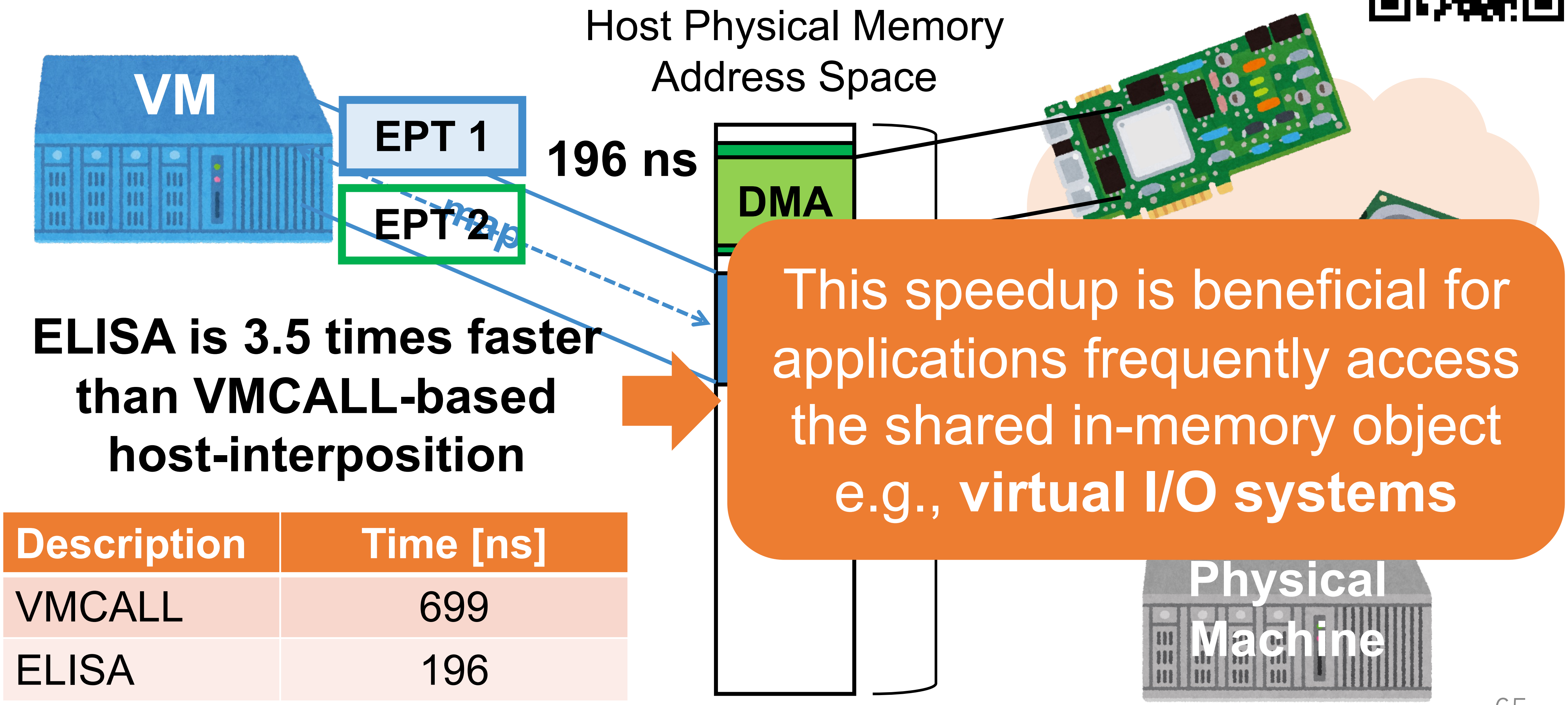


ELISA is 3.5 times faster than VMCALL-based host-interposition

Description	Time [ns]
VMCALL	699
ELISA	196

This speedup is beneficial for applications frequently access the shared in-memory object

Context Switch Overhead (ELISA)

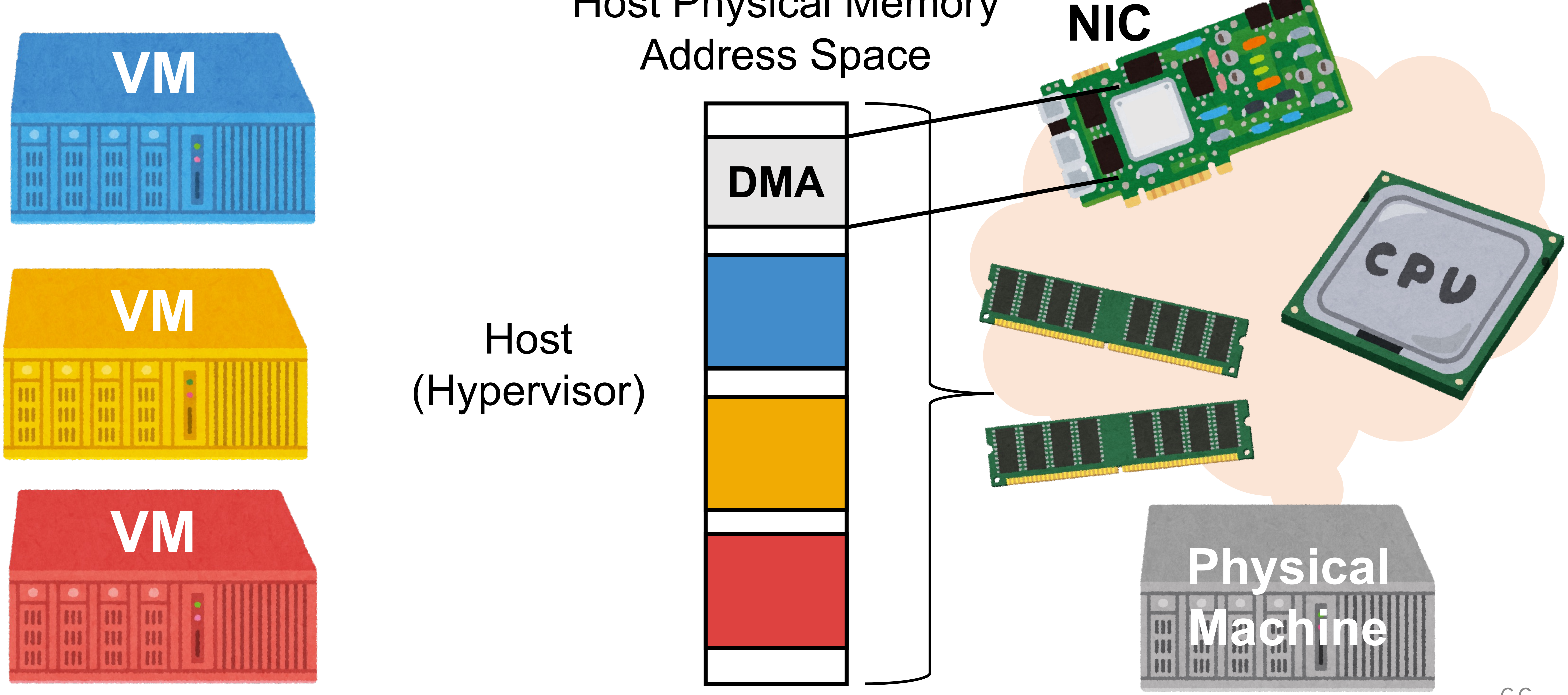


ELISA is 3.5 times faster than VMCALL-based host-interposition

Description	Time [ns]
VMCALL	699
ELISA	196

This speedup is beneficial for applications frequently access the shared in-memory object e.g., **virtual I/O systems**

VM Networking by Host-interposition



Host Physical Memory
Address Space

NIC

VM

VM

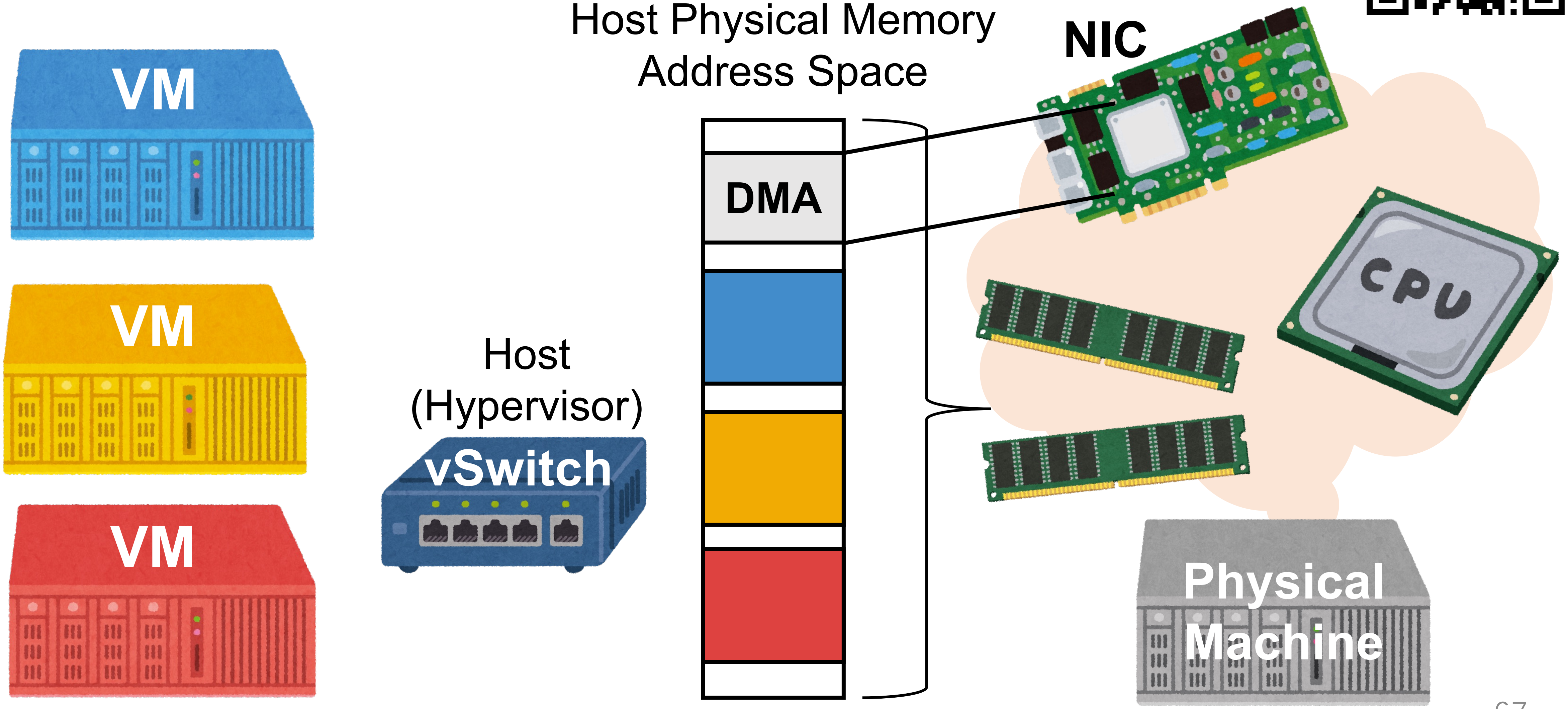
VM

Host
(Hypervisor)

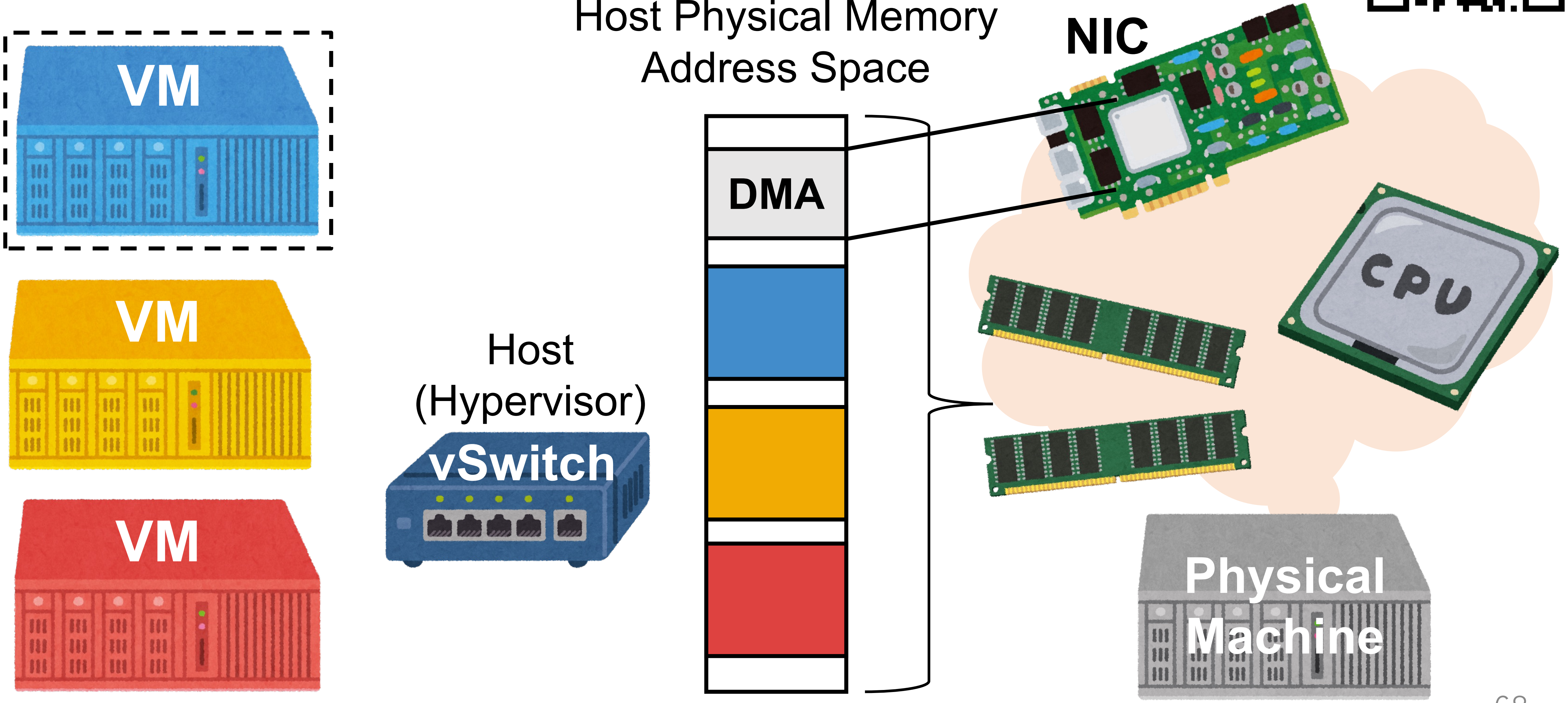
DMA

Physical
Machine

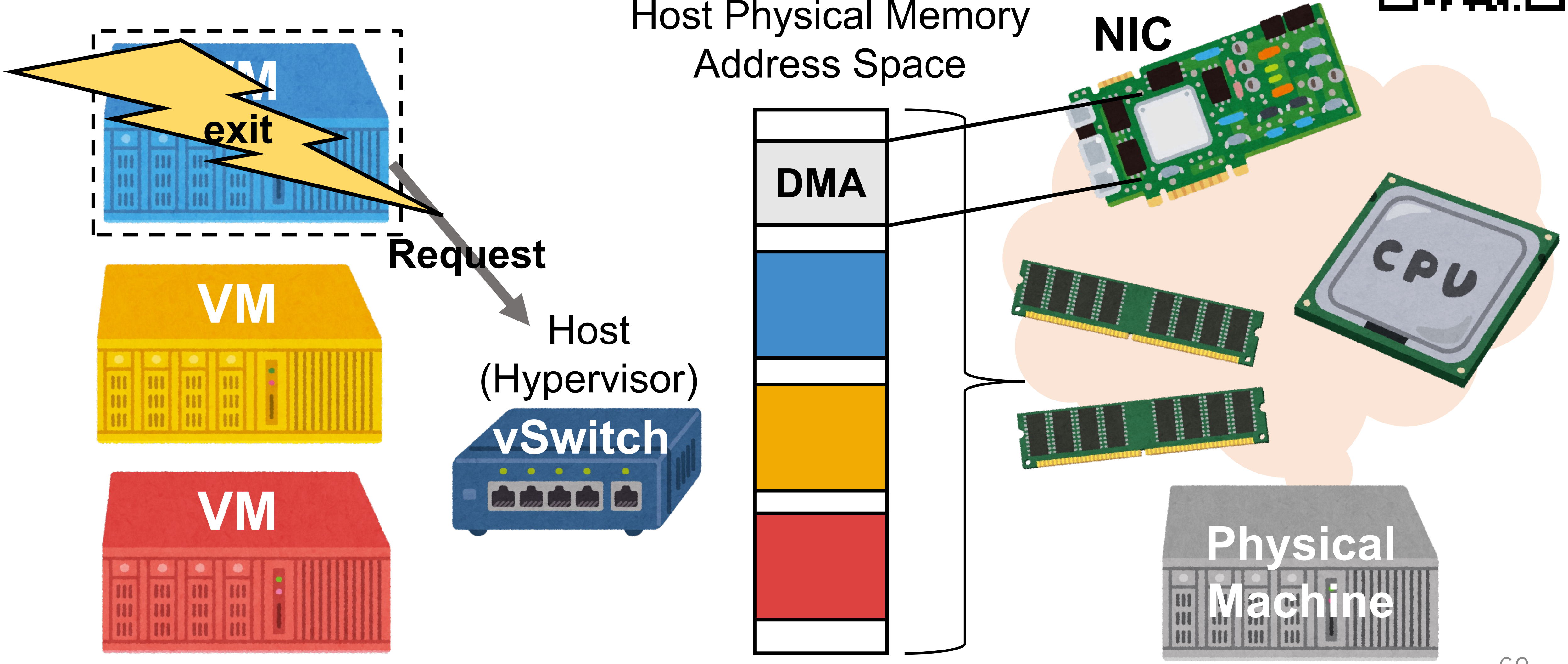
VM Networking by Host-interposition



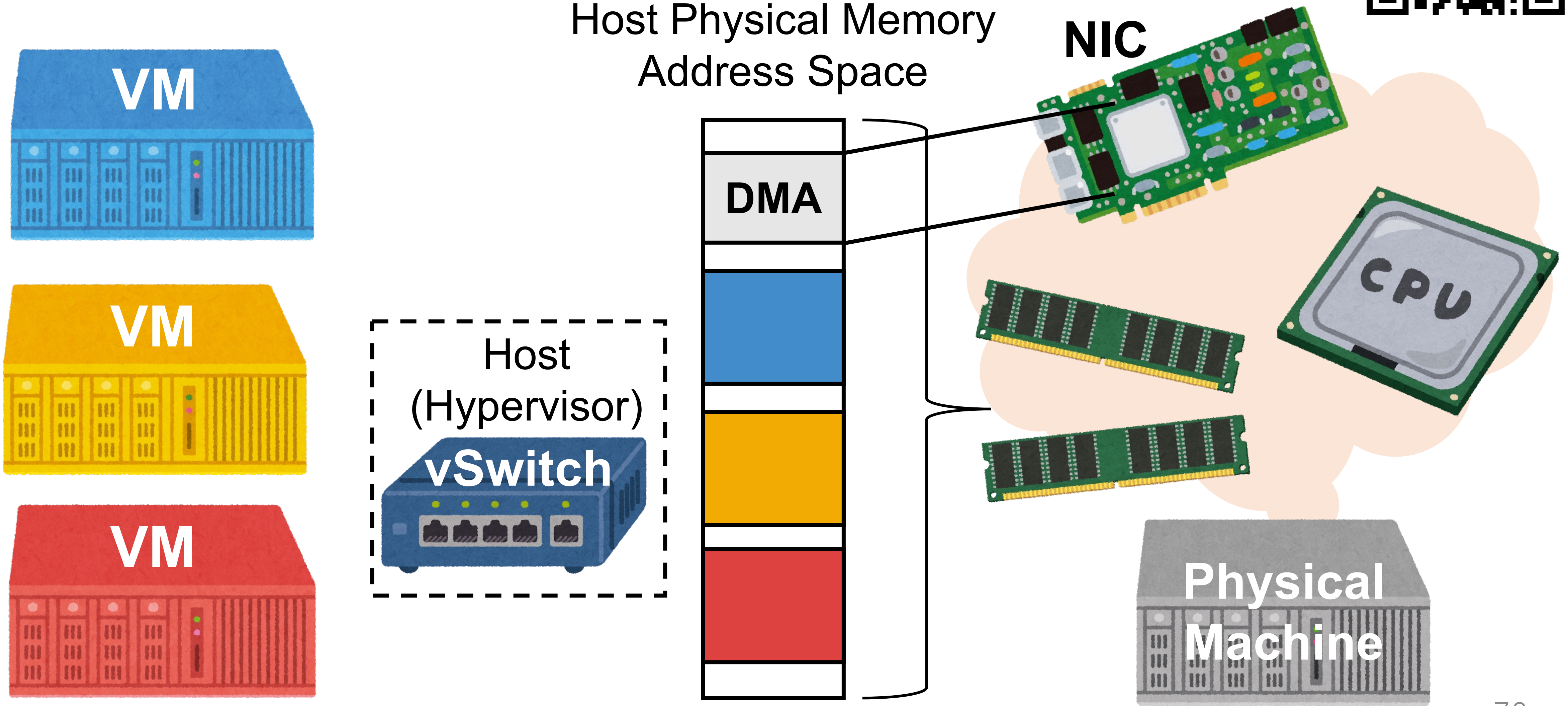
VM Networking by Host-interposition



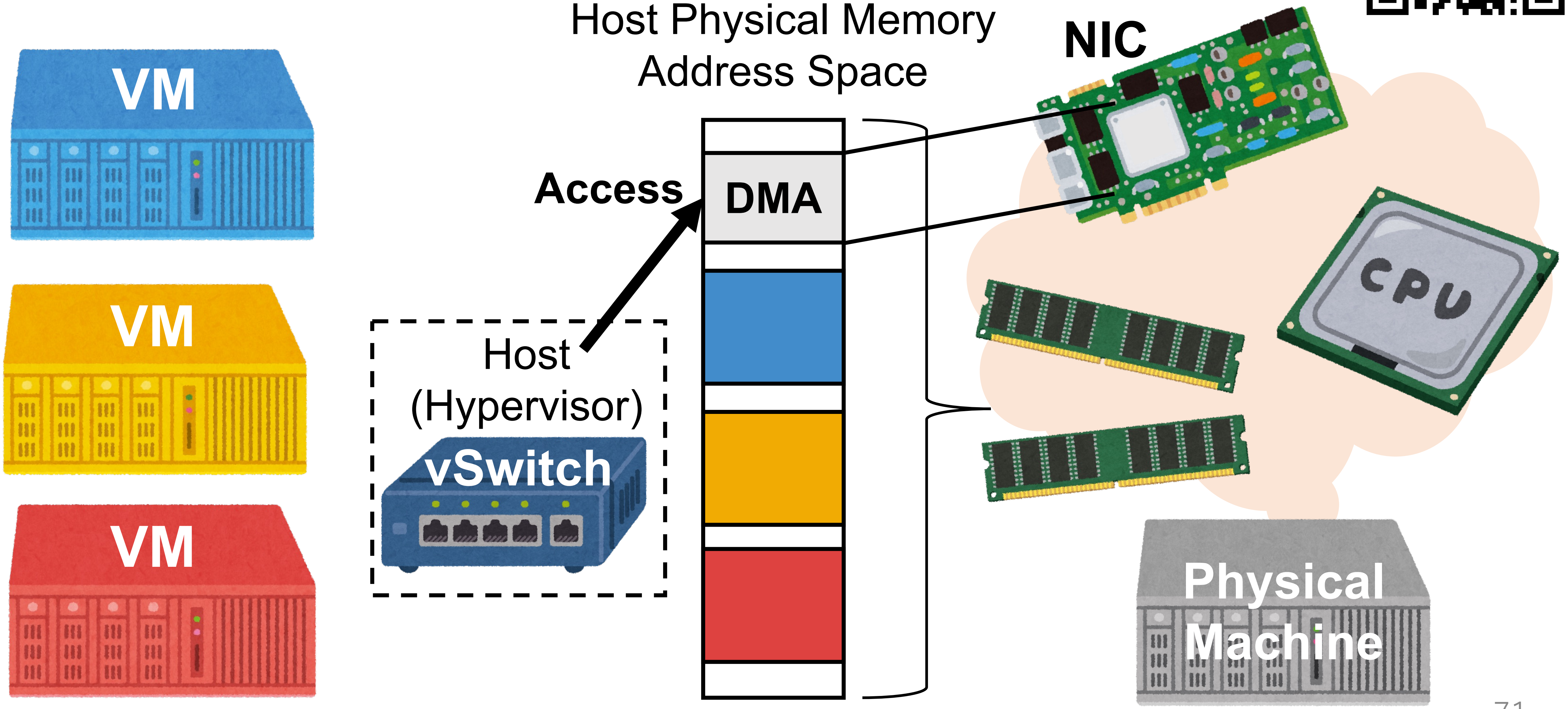
VM Networking by Host-interposition



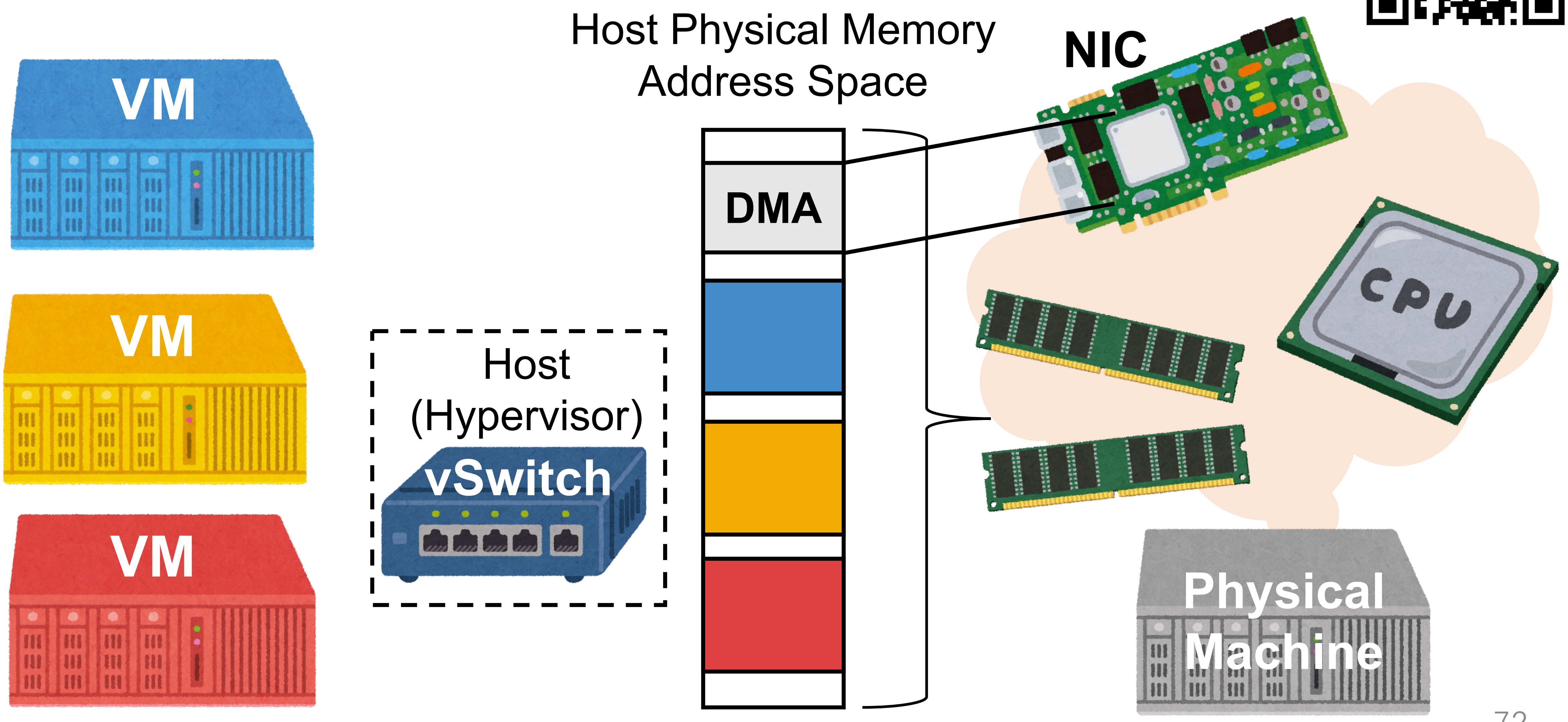
VM Networking by Host-interposition



VM Networking by Host-interposition

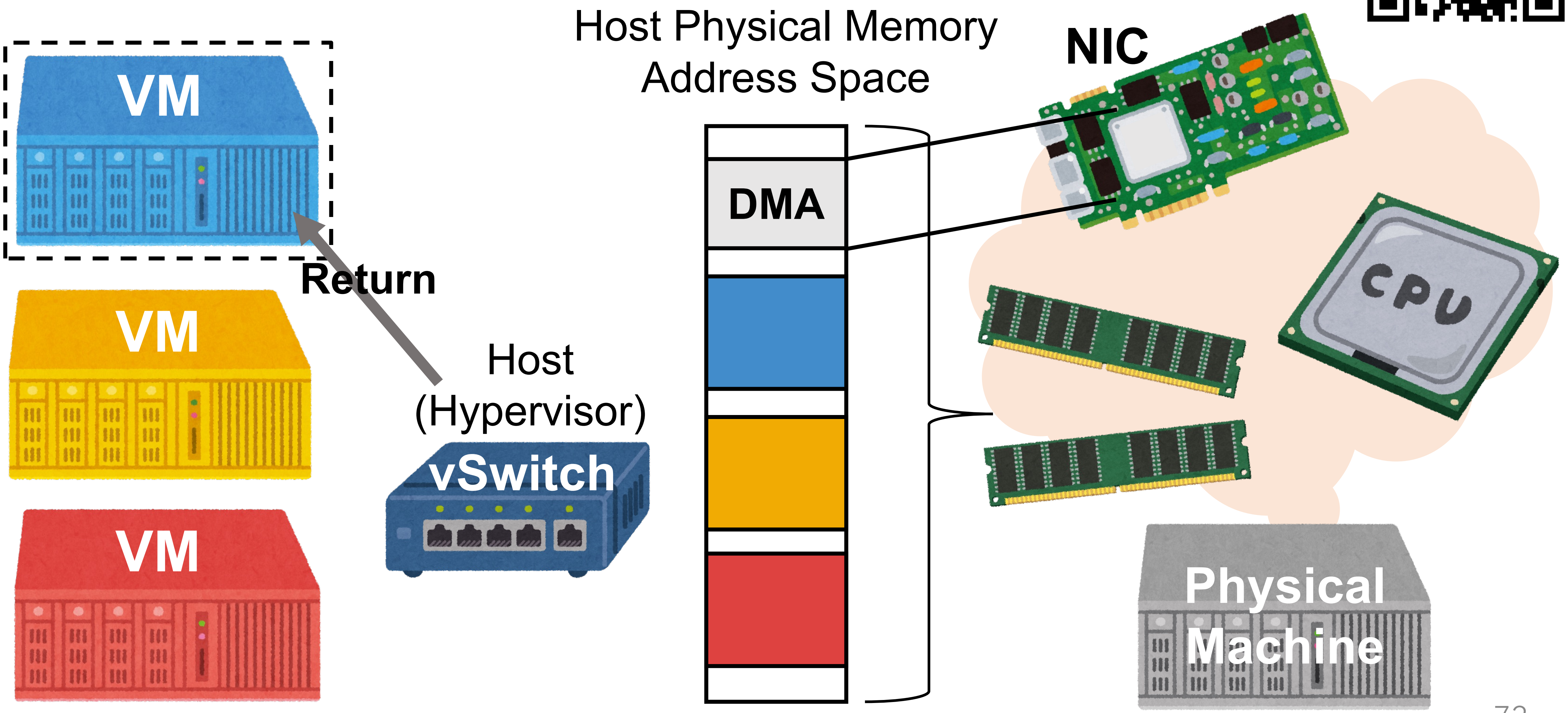


VM Networking by Host-interposition

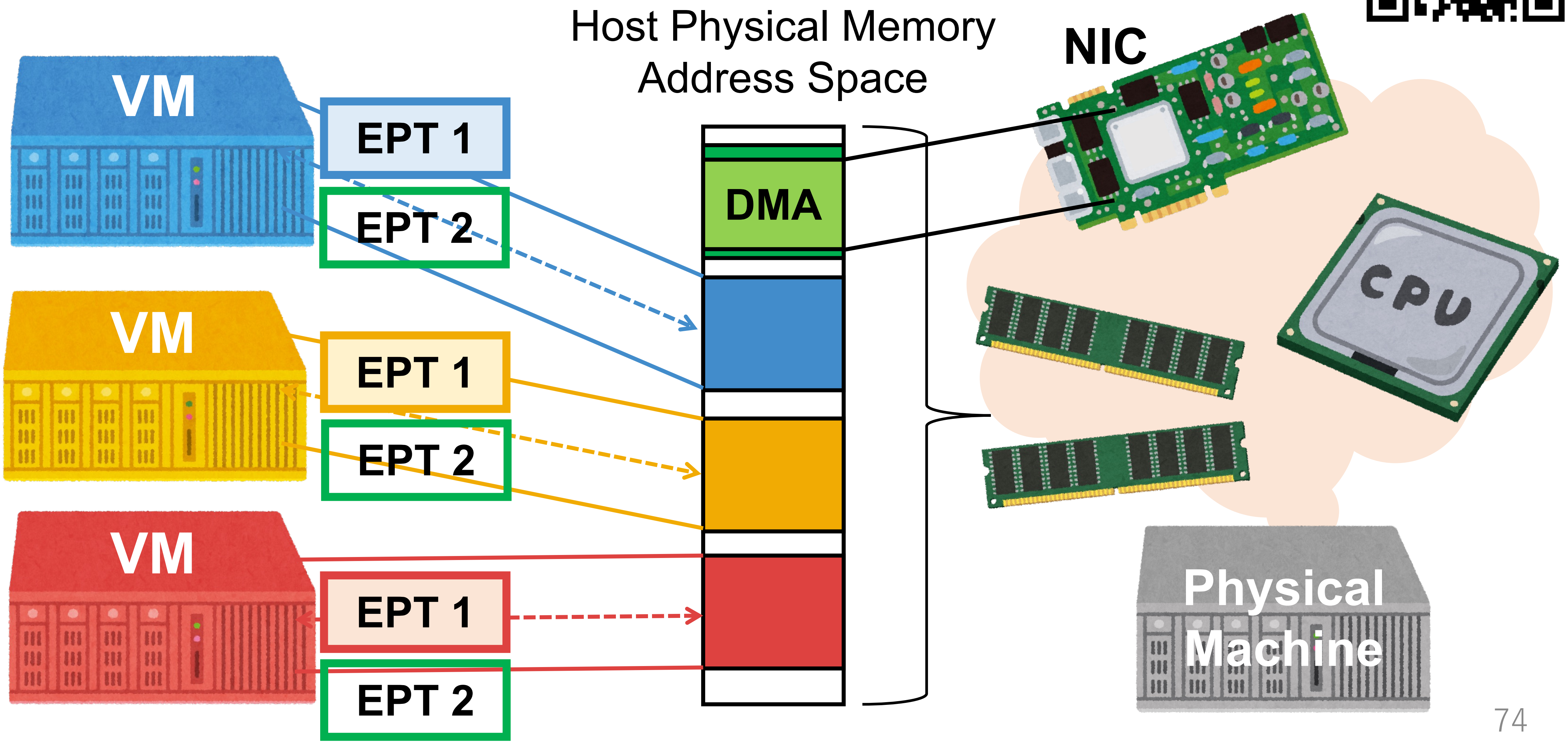


Host Physical Memory Address Space

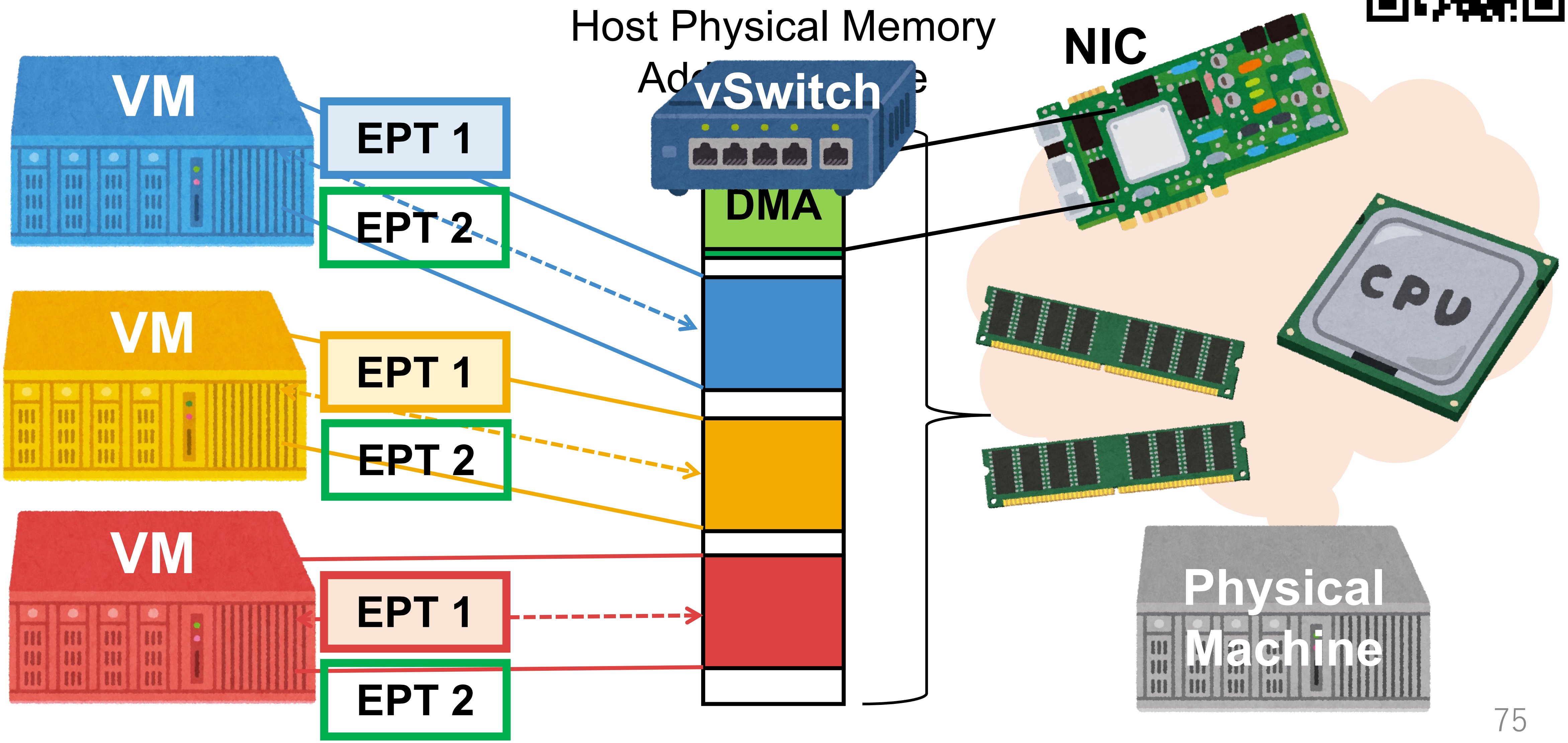
VM Networking by Host-interposition



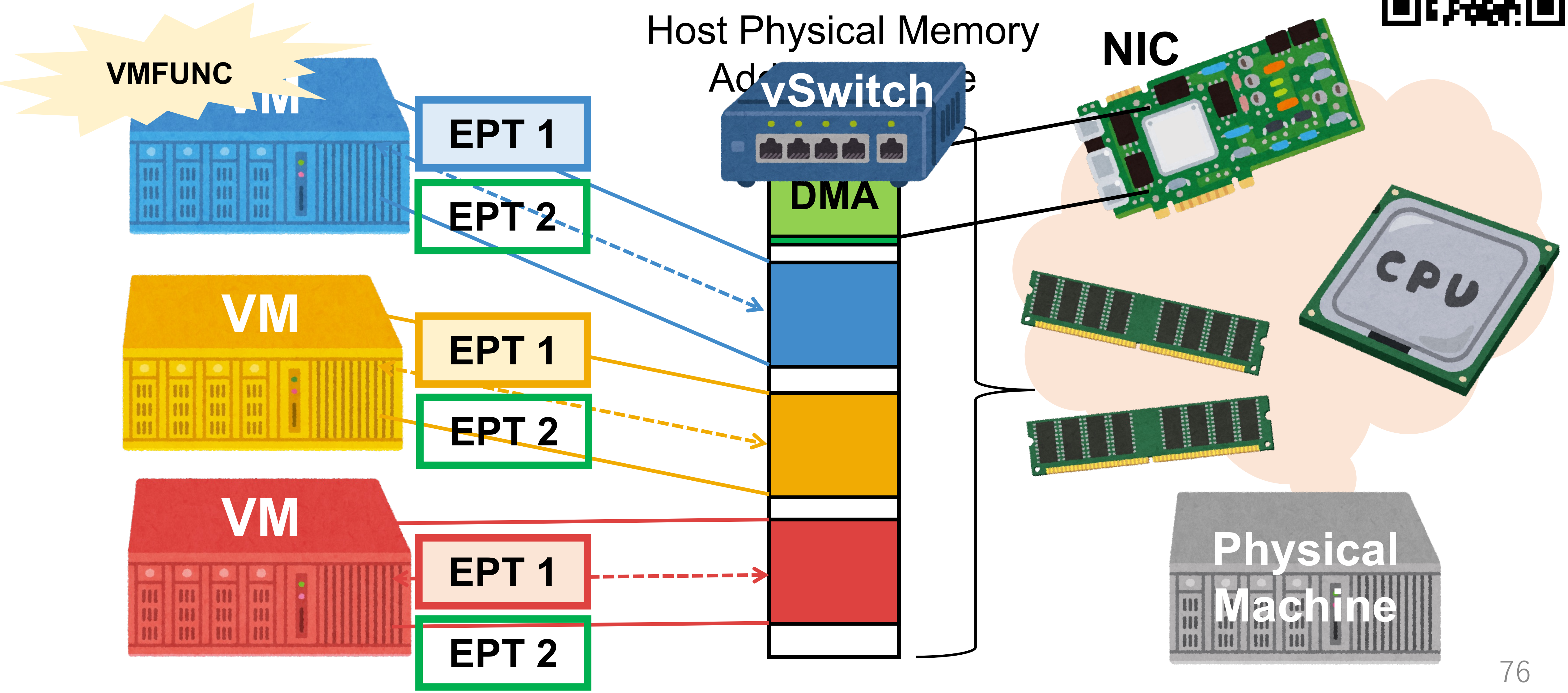
ELISA-based VM Networking System



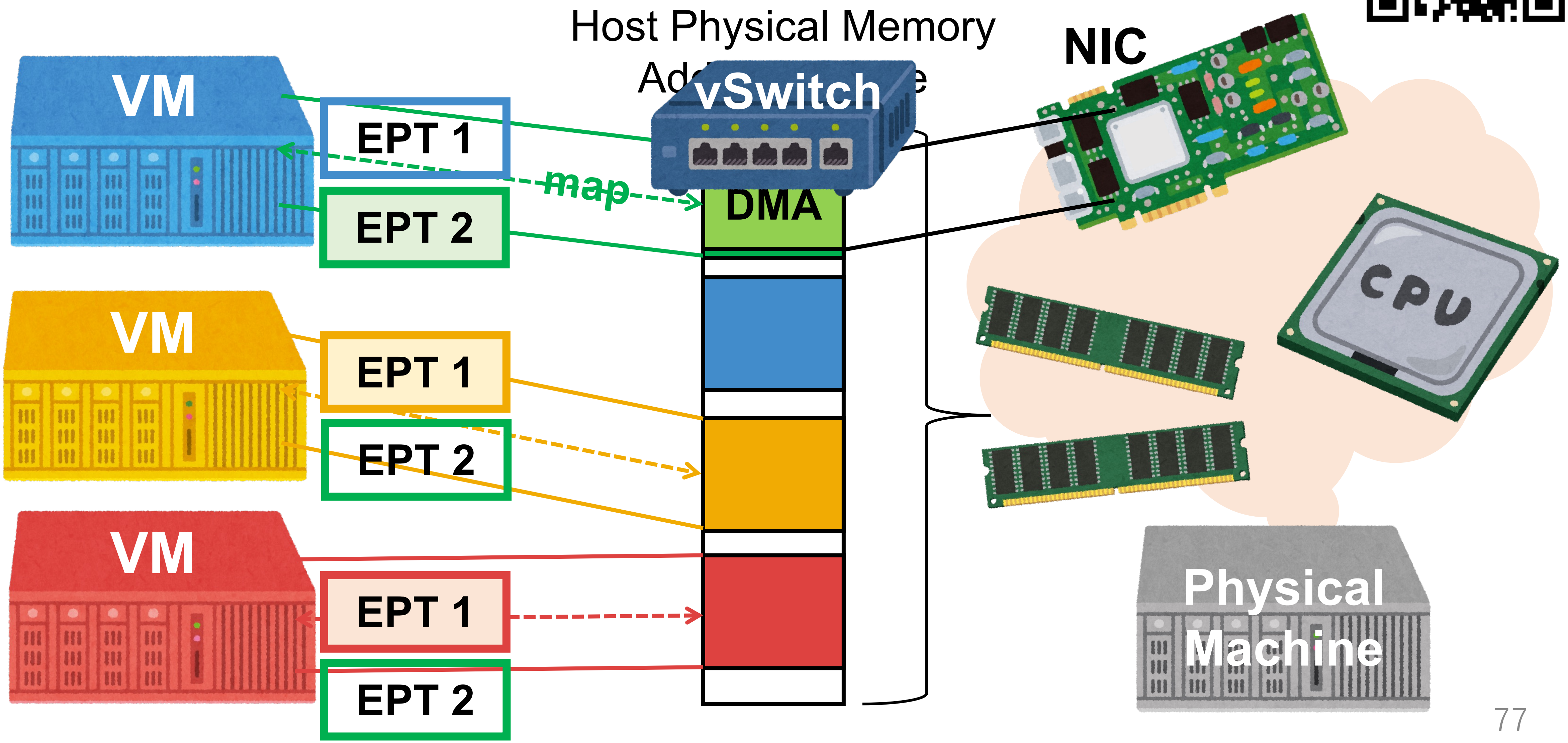
ELISA-based VM Networking System



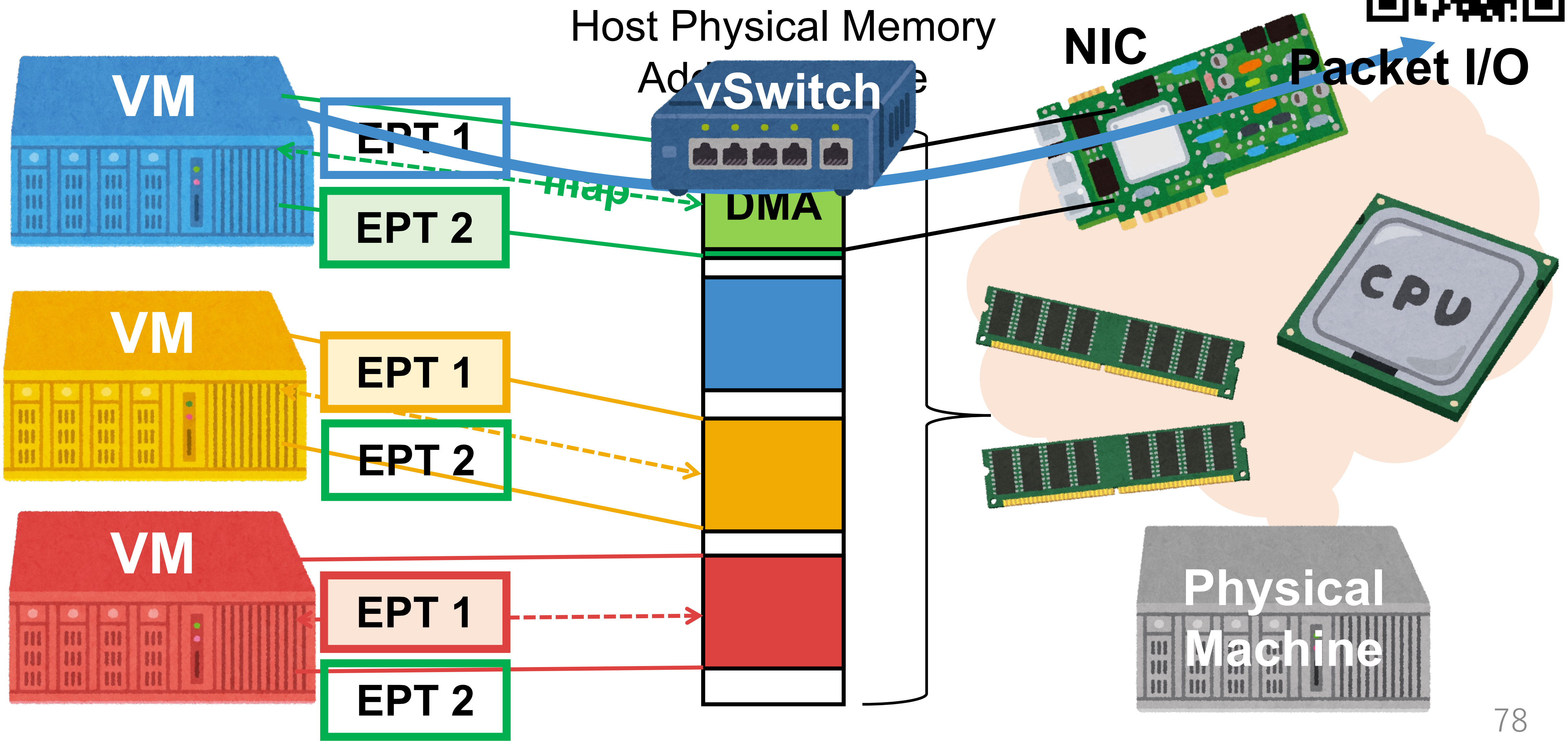
ELISA-based VM Networking System



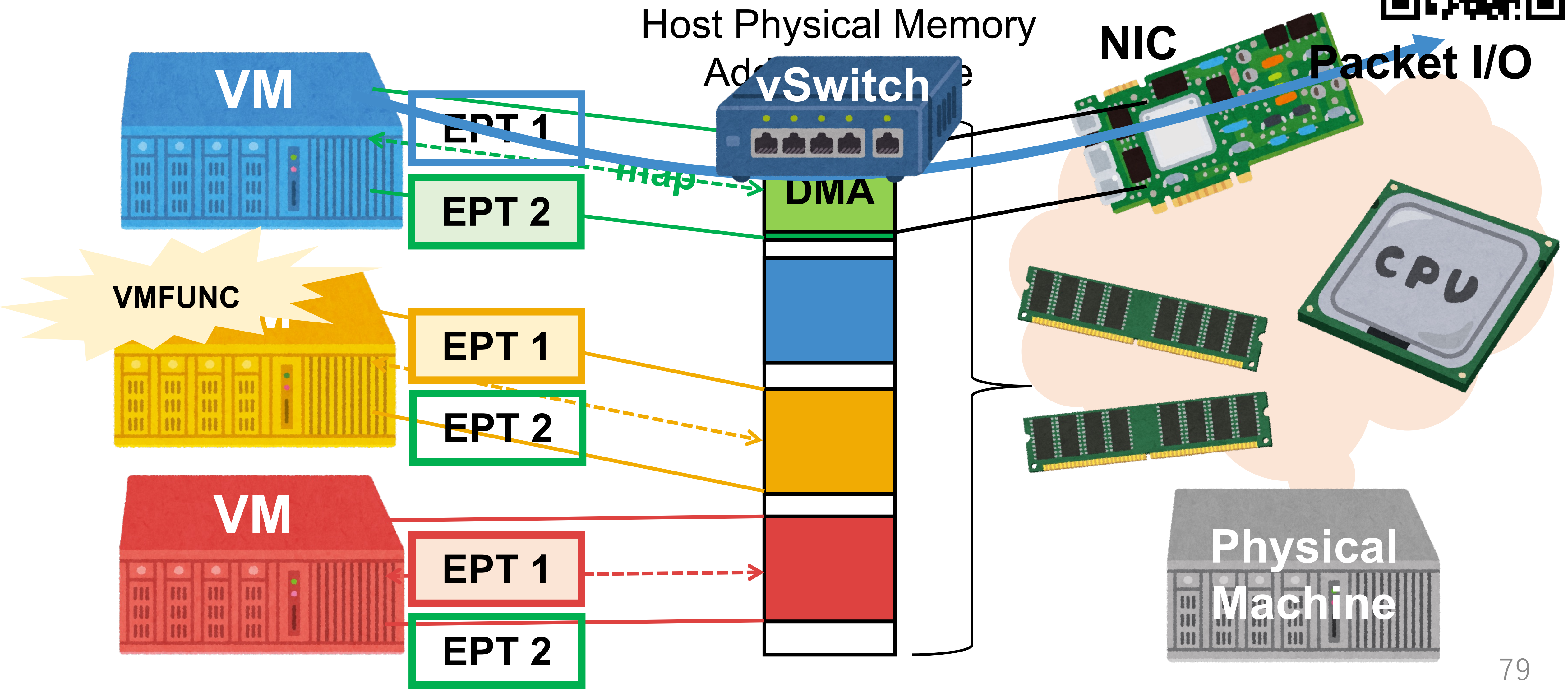
ELISA-based VM Networking System



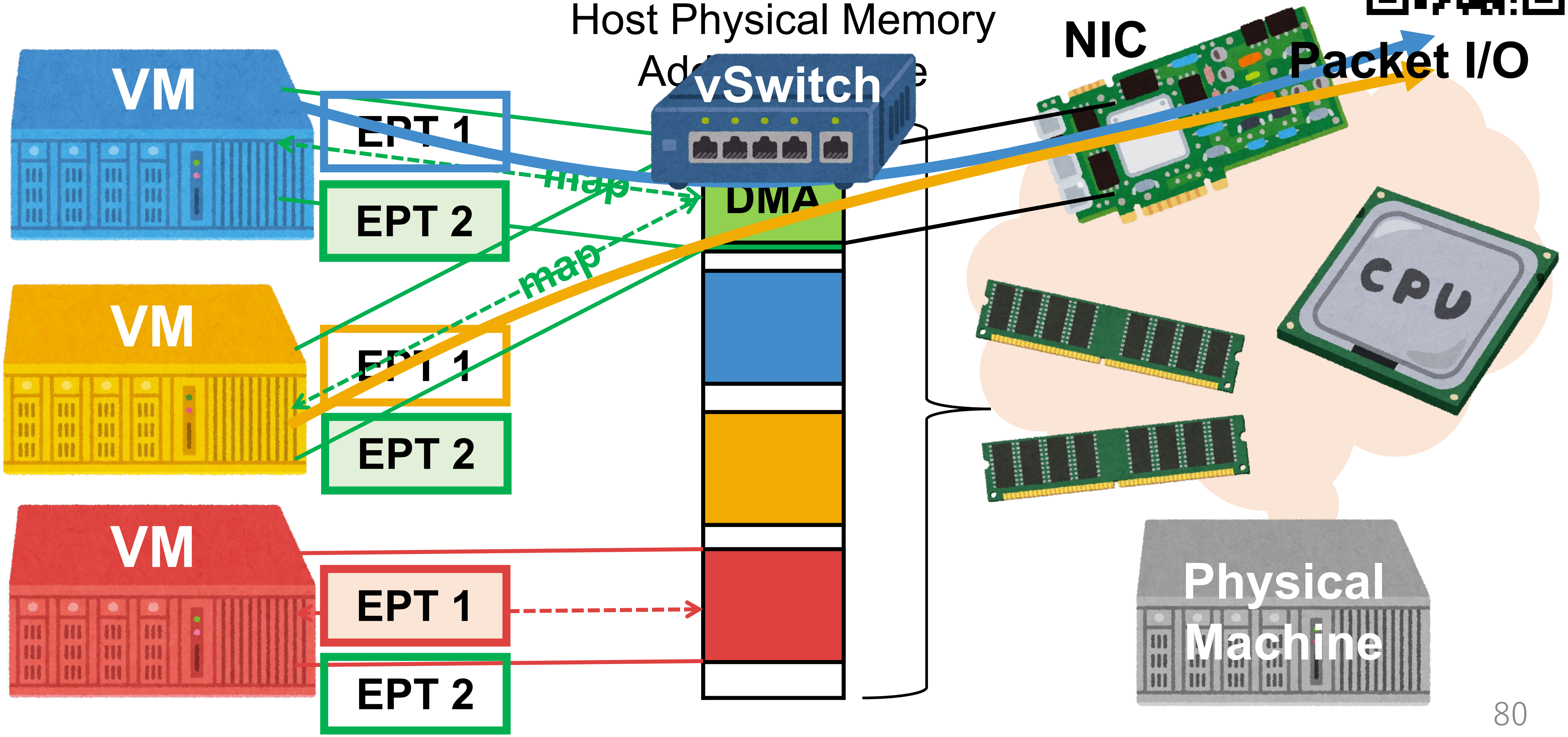
ELISA-based VM Networking System



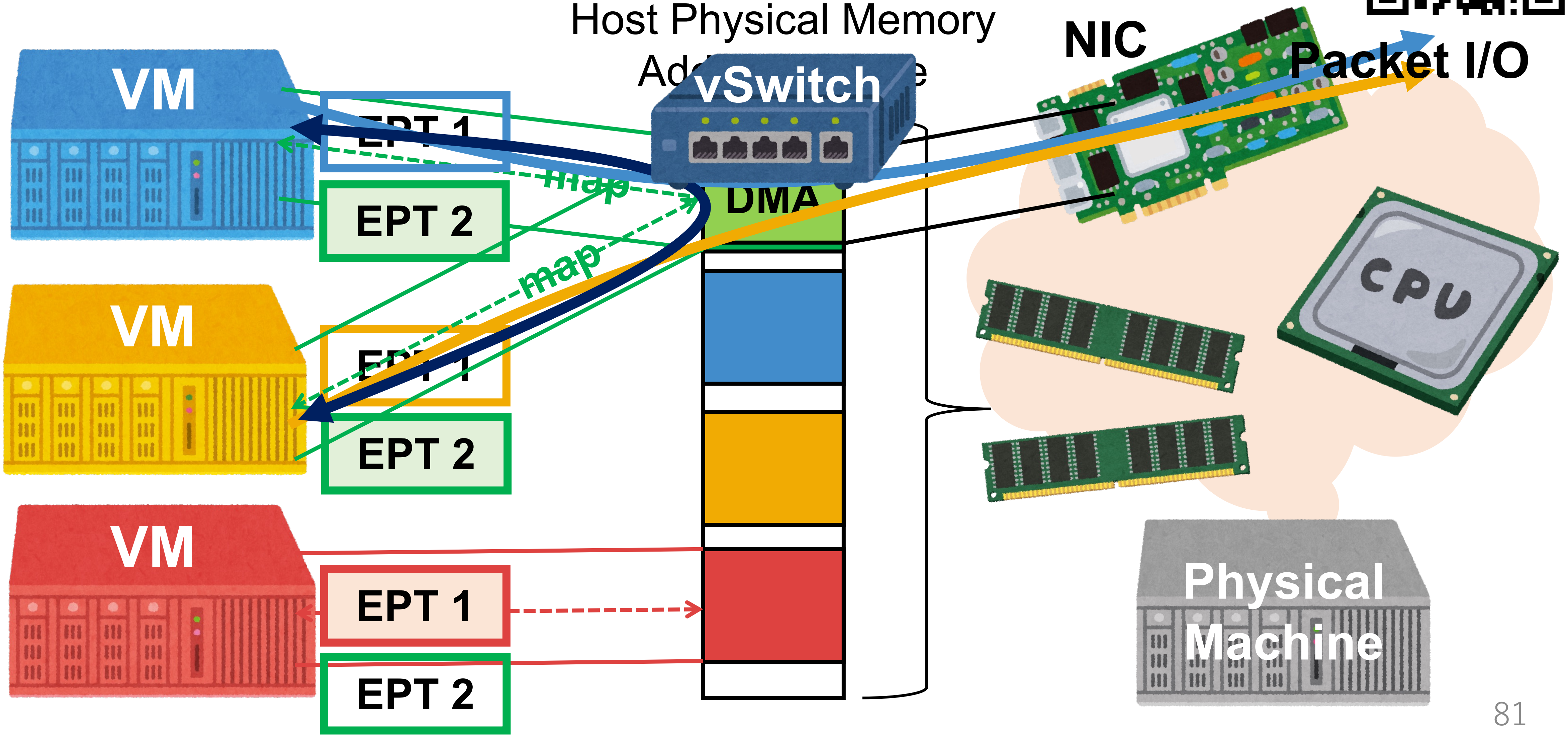
ELISA-based VM Networking System



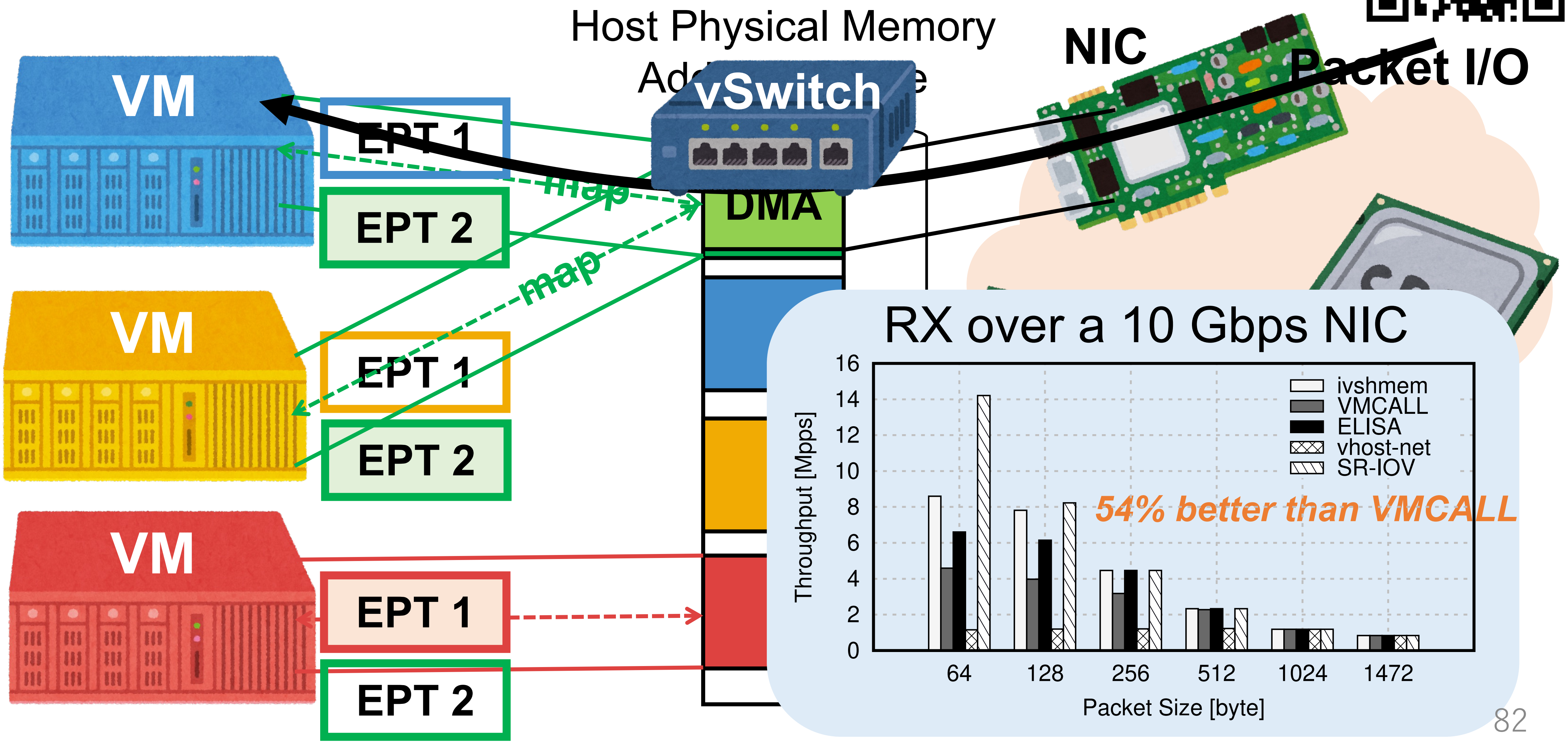
ELISA-based VM Networking System



ELISA-based VM Networking System

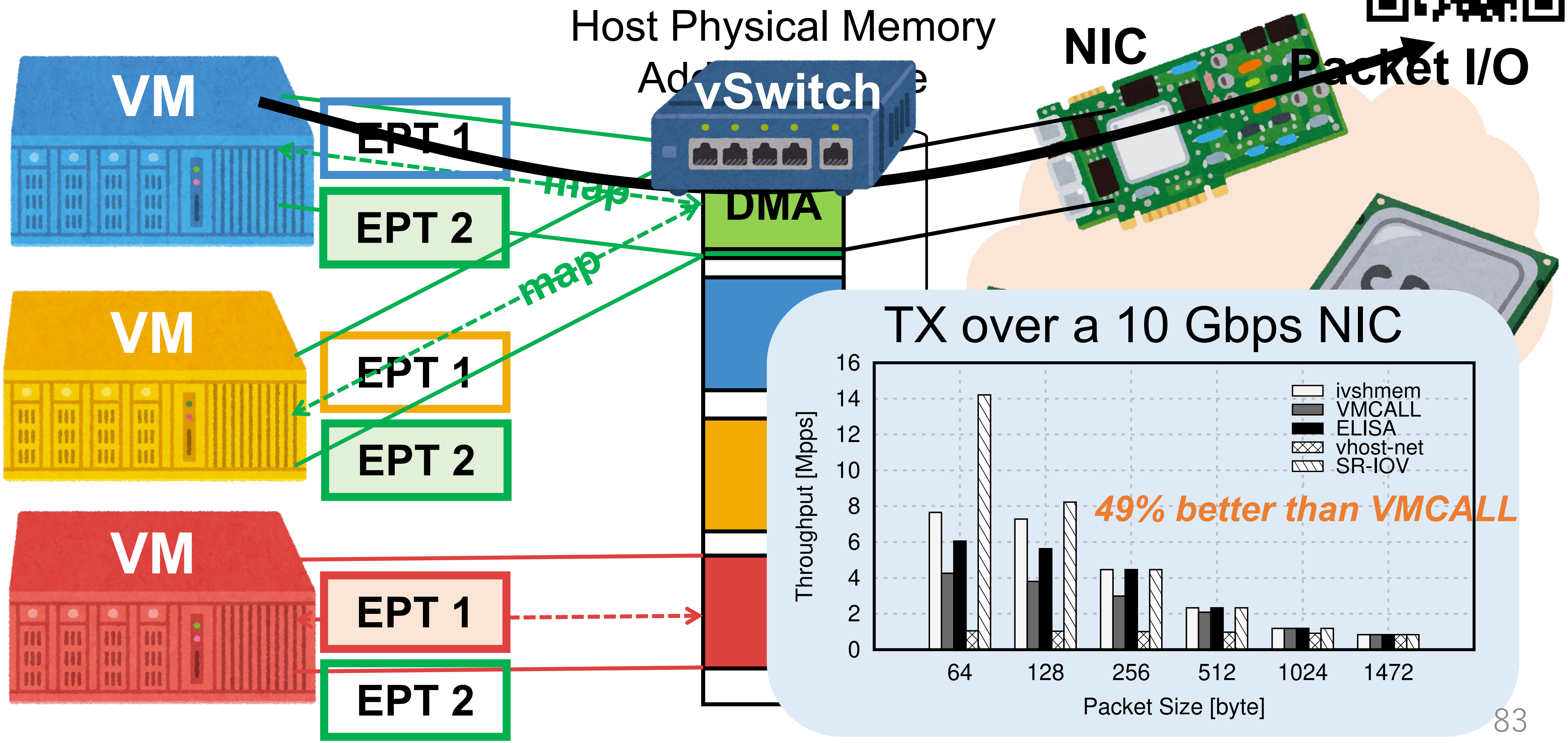


ELISA-based VM Networking System



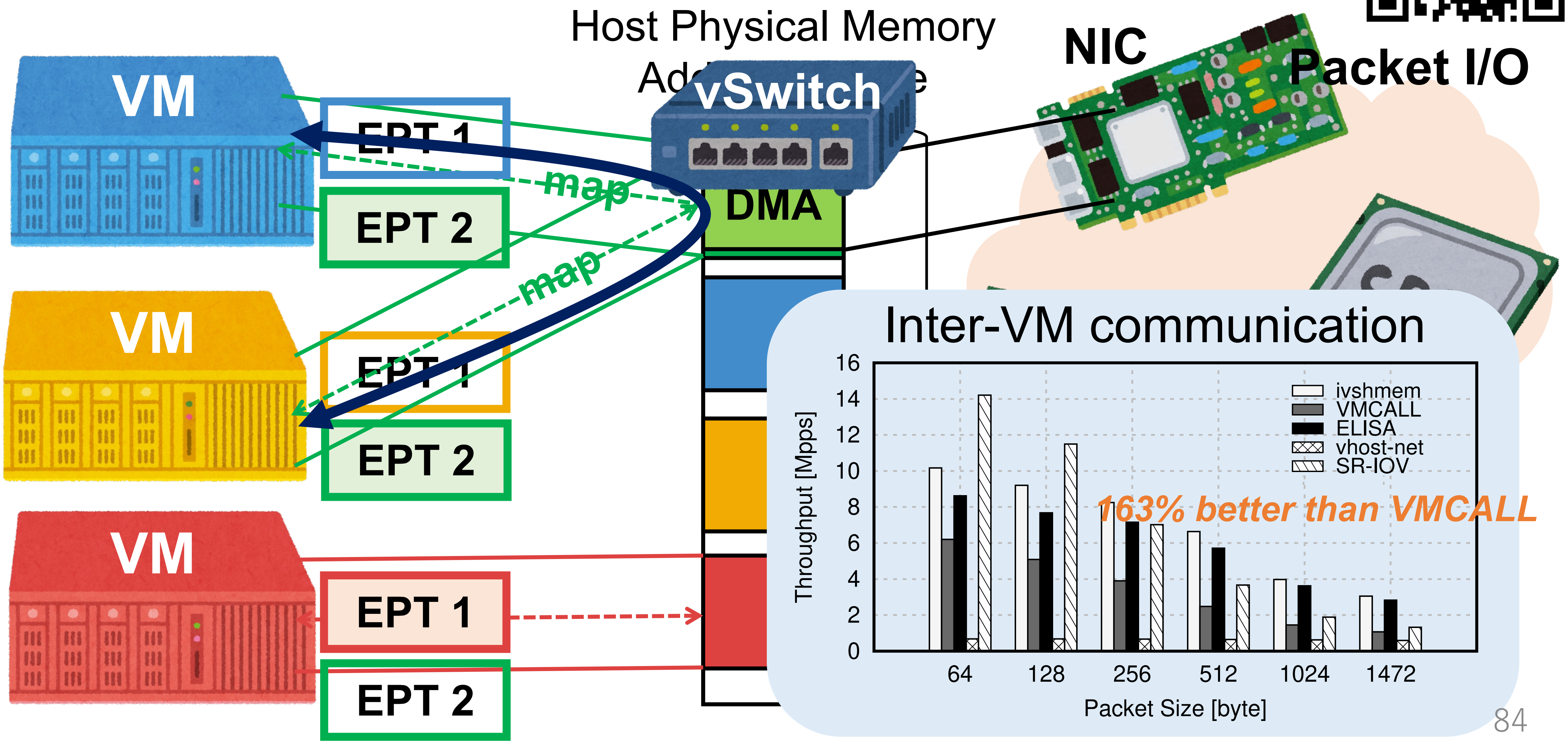
54% better than VMCALL

ELISA-based VM Networking System



49% better than VMCALL

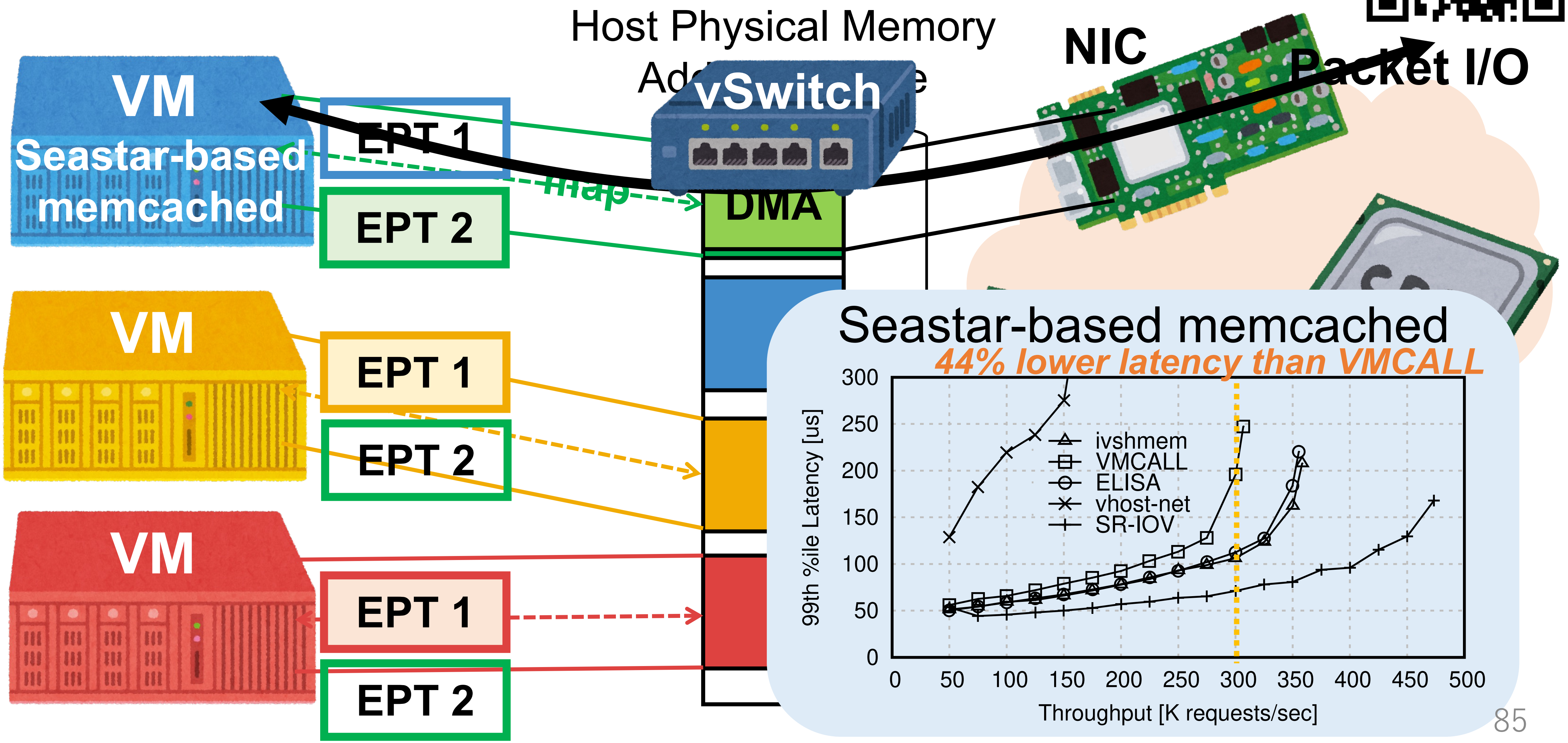
ELISA-based VM Networking System



Inter-VM communication

163% better than VMCALL

ELISA-based VM Networking System



Seastar-based memcached

44% lower latency than VMCALL

Summary

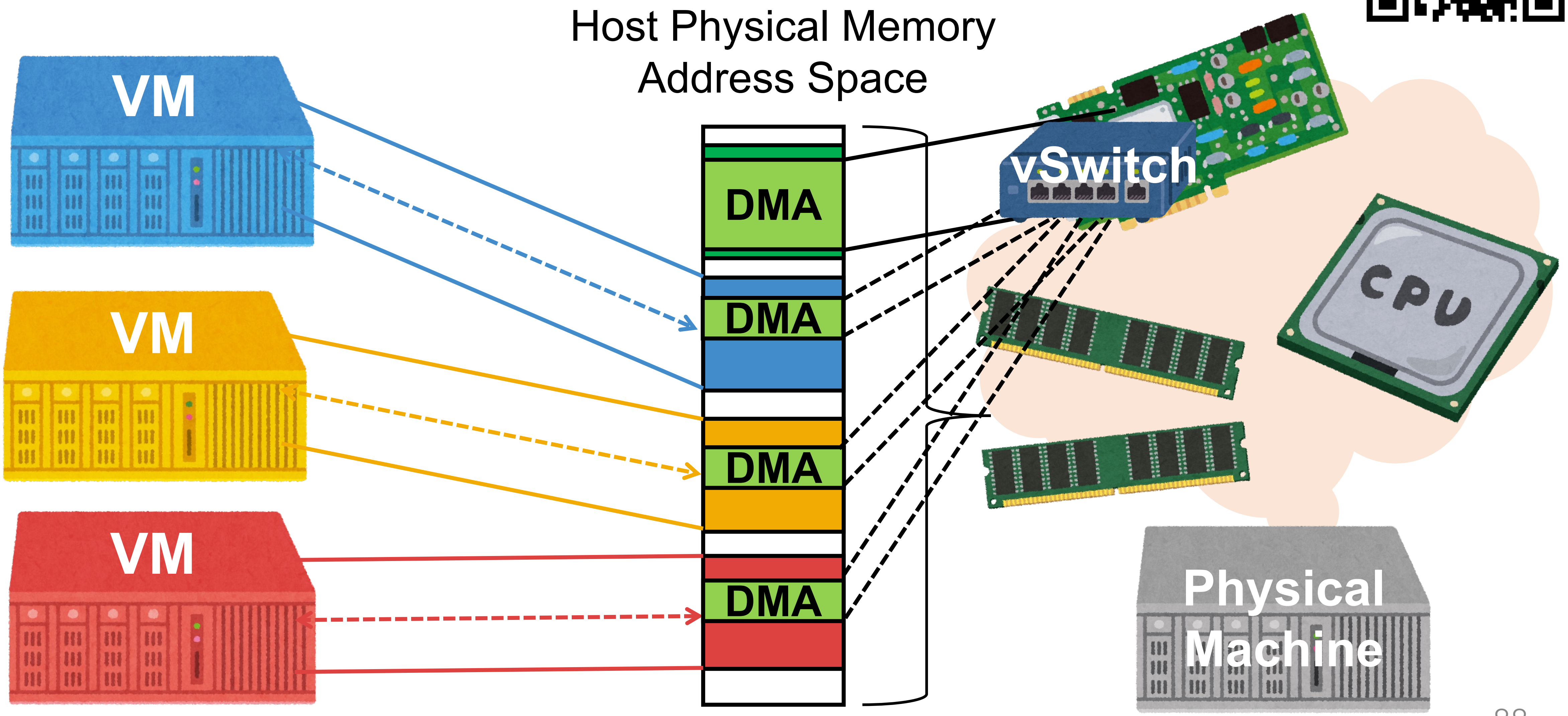
- ELISA is an in-memory object sharing scheme for VMs

- ELISA employs EPT separation and VMFUNC to achieve isolation at a low overhead

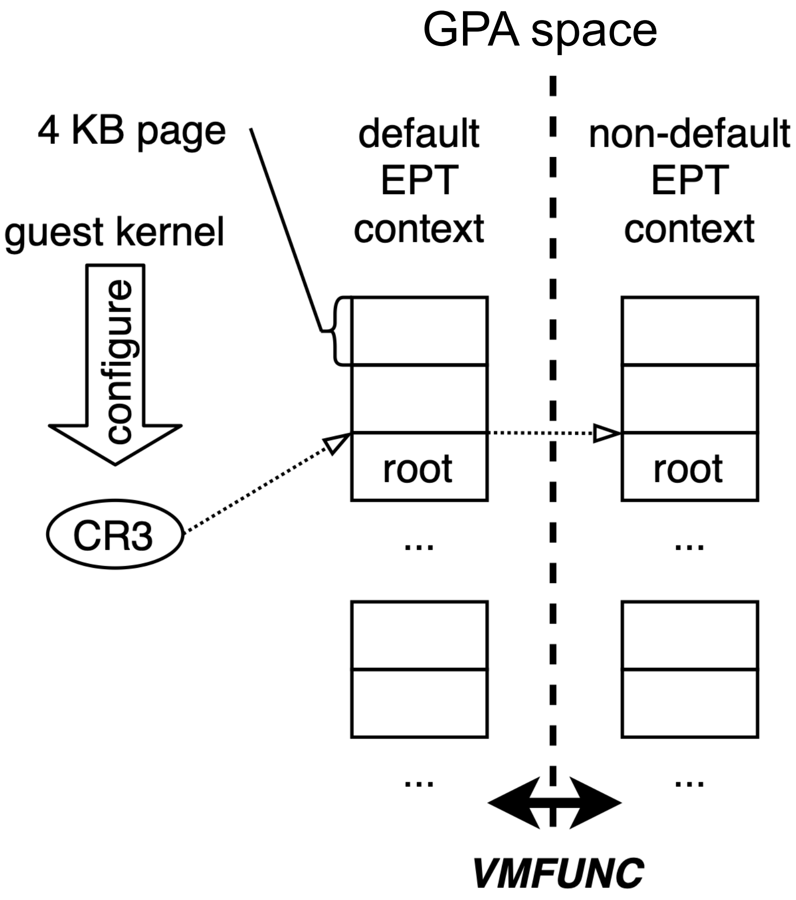
Thank you for your listening

Questions?

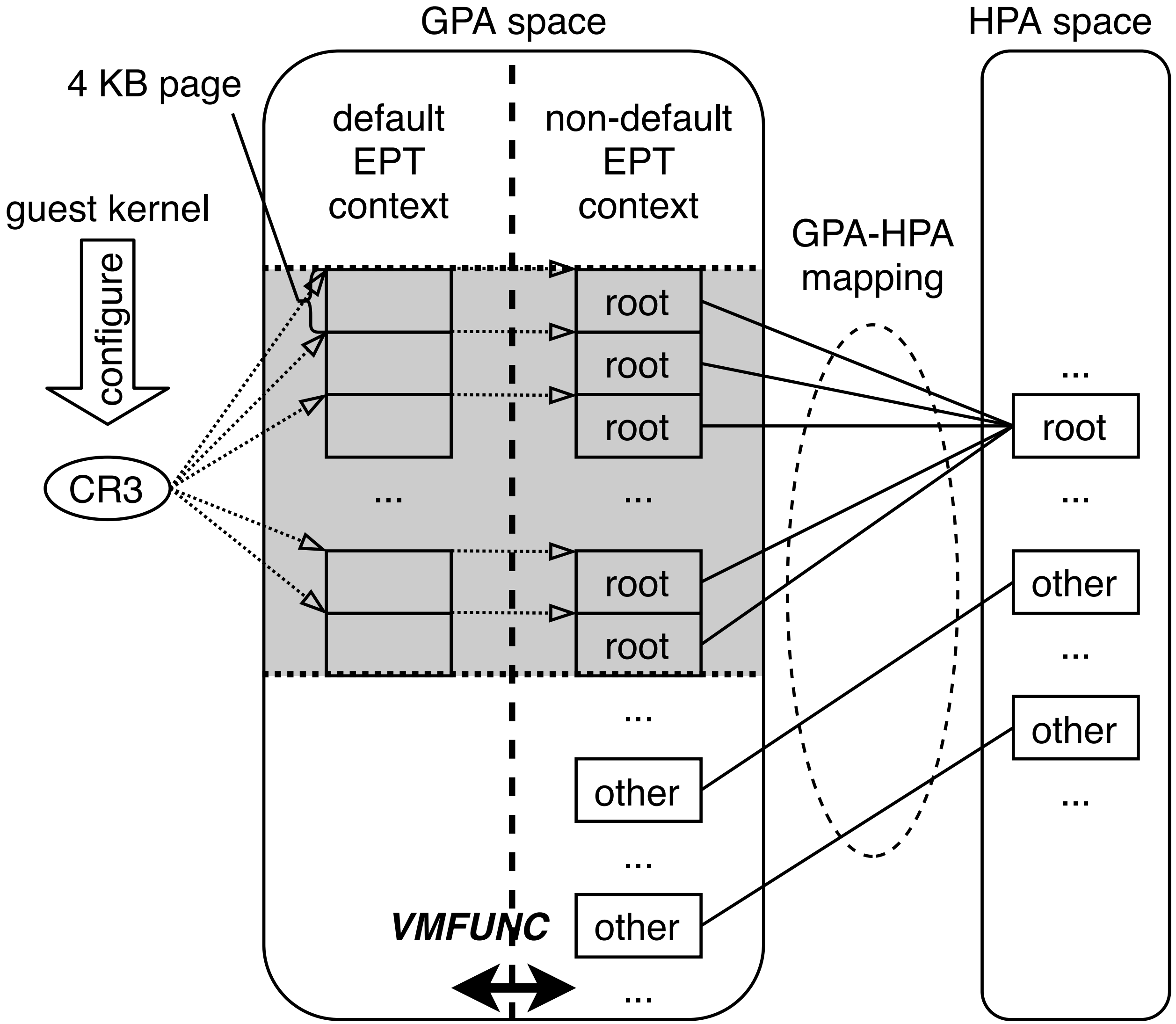
SR-IOV


Host Physical Memory
Address Space
VM
VM
VM
DMA
DMA
DMA
DMA
vSwitch
CPU
Physical
Machine

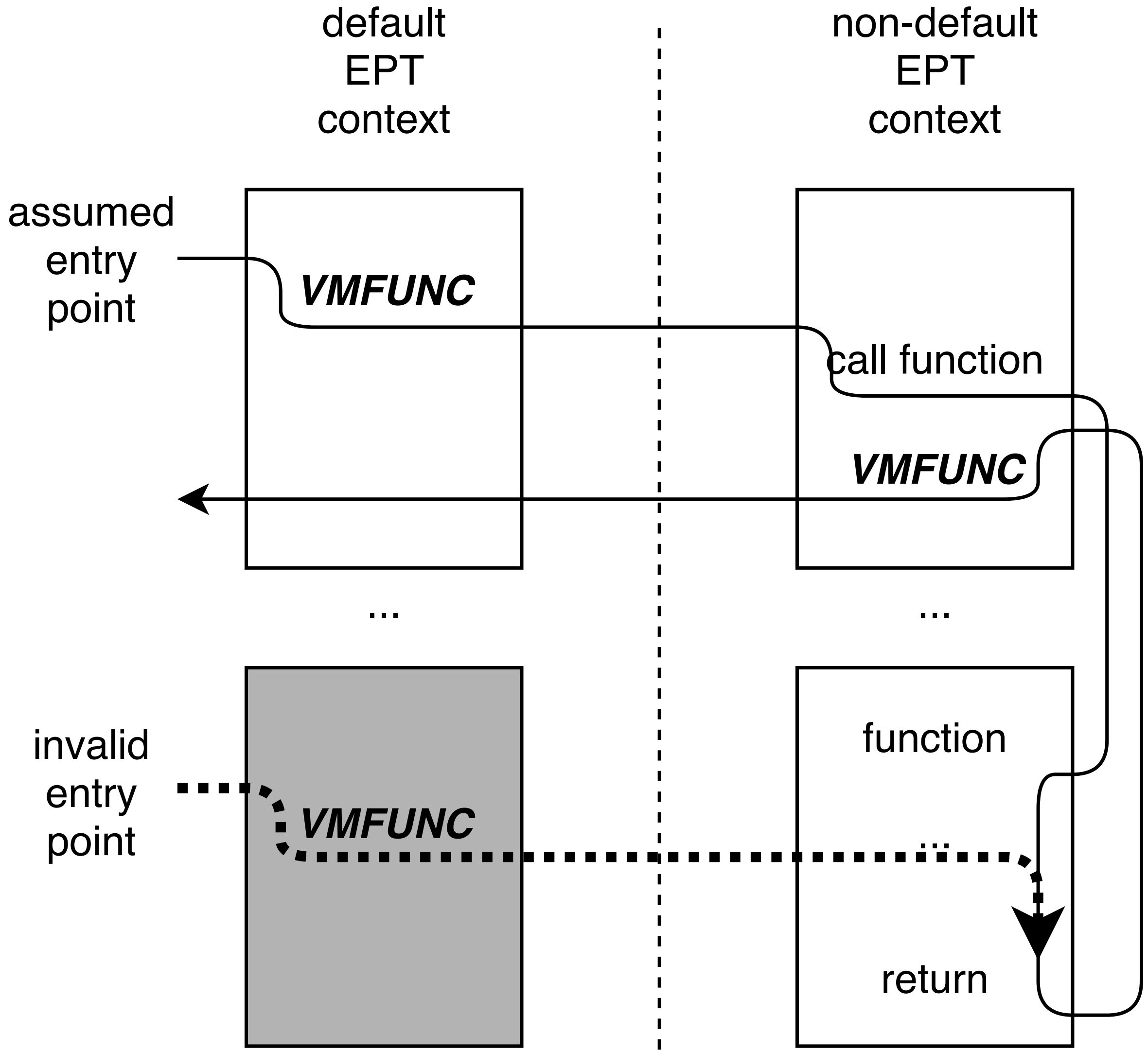
Page Table Maintenance Overhead



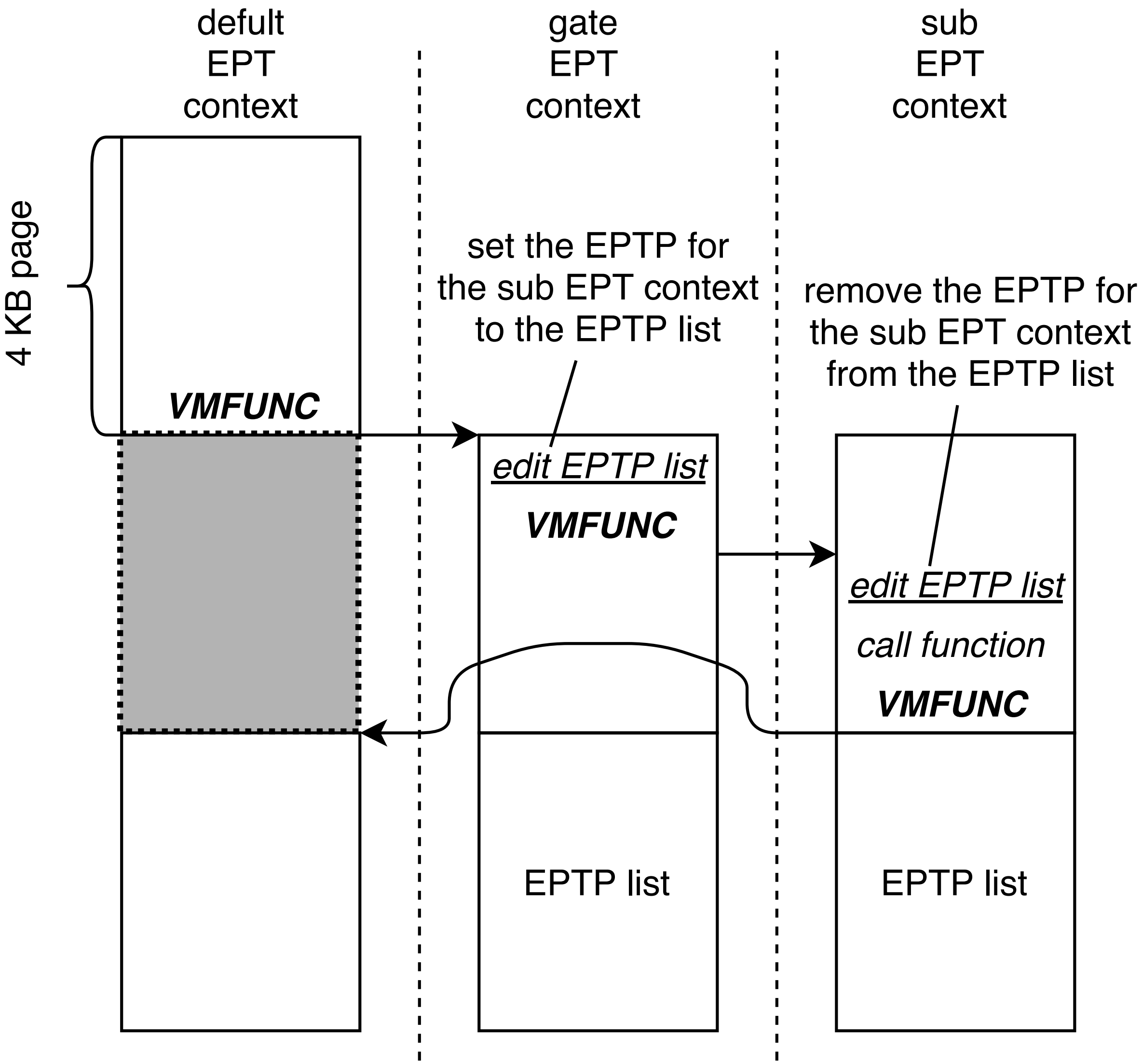
Anywhere Page Table (APT)



Potential Attack

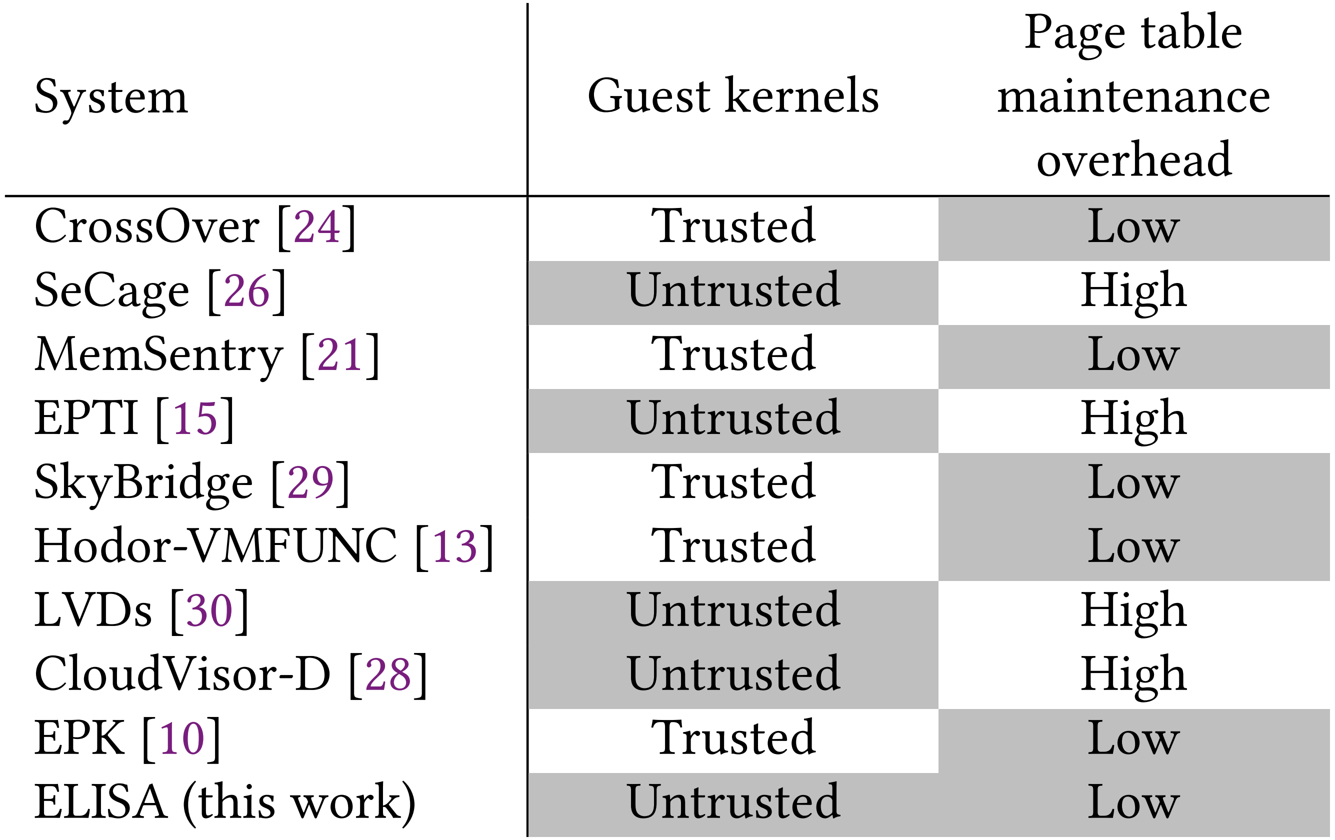


Gate EPT Context



Comparison

Table 2: Properties of VMFUNC-based systems. Shaded parts indicate the desired properties.



System	Guest kernels	Page table maintenance overhead
CrossOver [24]	Trusted	Low
SeCage [26]	Untrusted	High
MemSentry [21]	Trusted	Low
EPTI [15]	Untrusted	High
SkyBridge [29]	Trusted	Low
Hodor-VMFUNC [13]	Trusted	Low
LVDs [30]	Untrusted	High
CloudVisor-D [28]	Untrusted	High
EPK [10]	Trusted	Low
ELISA (this work)	Untrusted	Low