

Toro, a Dedicated Unikernel for Microservices

www.torokernel.io

Matias Vara Larsen
matiasevara@gmail.com

Who am I?

- I am passionate about operating systems development and virtualization
- I studied Electronic Engineer at UNLP
- I did my PhD in formal languages and DSMLs at INRIA, Sophia Antipolis, France
- I worked at Citrix, Tttech, Huawei ...
- <https://github.com/MatiasVara>

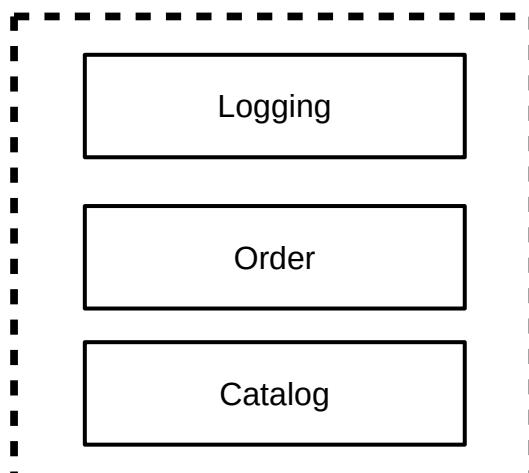


The Toro project since 2003

- 2003-2006, general purpose operating system
- 2006-2010, unikernel oriented to baremetal
- 2010-2017, unikernel to run as VM with support for emulated devices, e.g., e1000, ne2000, ide-disks
- 2017-2019, unikernel to run as VM with support for paravirtualized drivers, e.g., virtio-net, virtio-blk
- **2019-2020, unikernel to run as microVM with support for high-level paravirtualized drivers, e.g., virtio-fs and virtio-vsocket, main use-case: microservices**

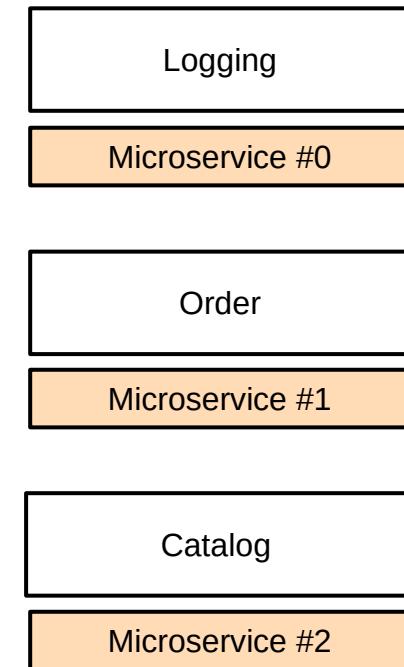
What are microservices?

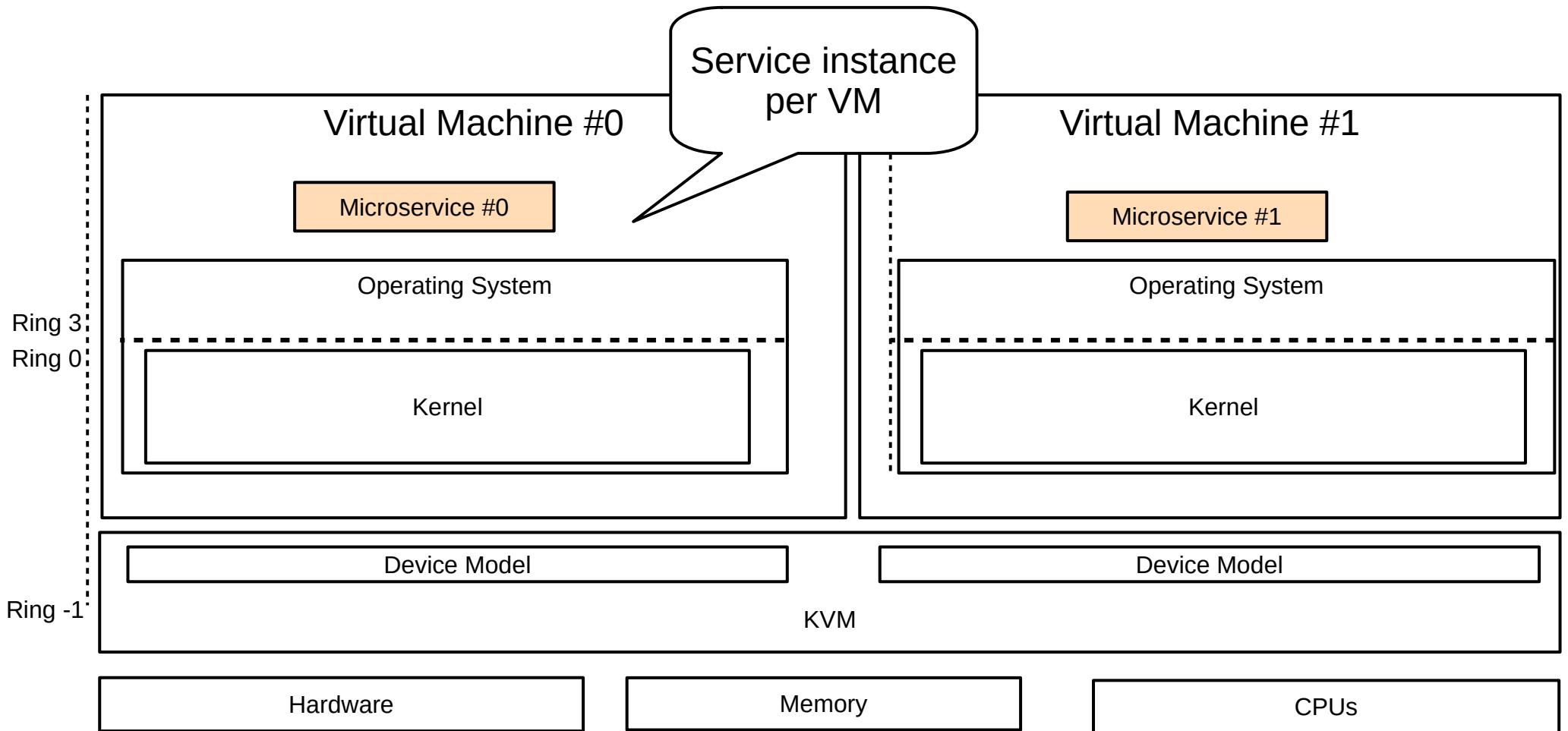
e.g., Amazon website



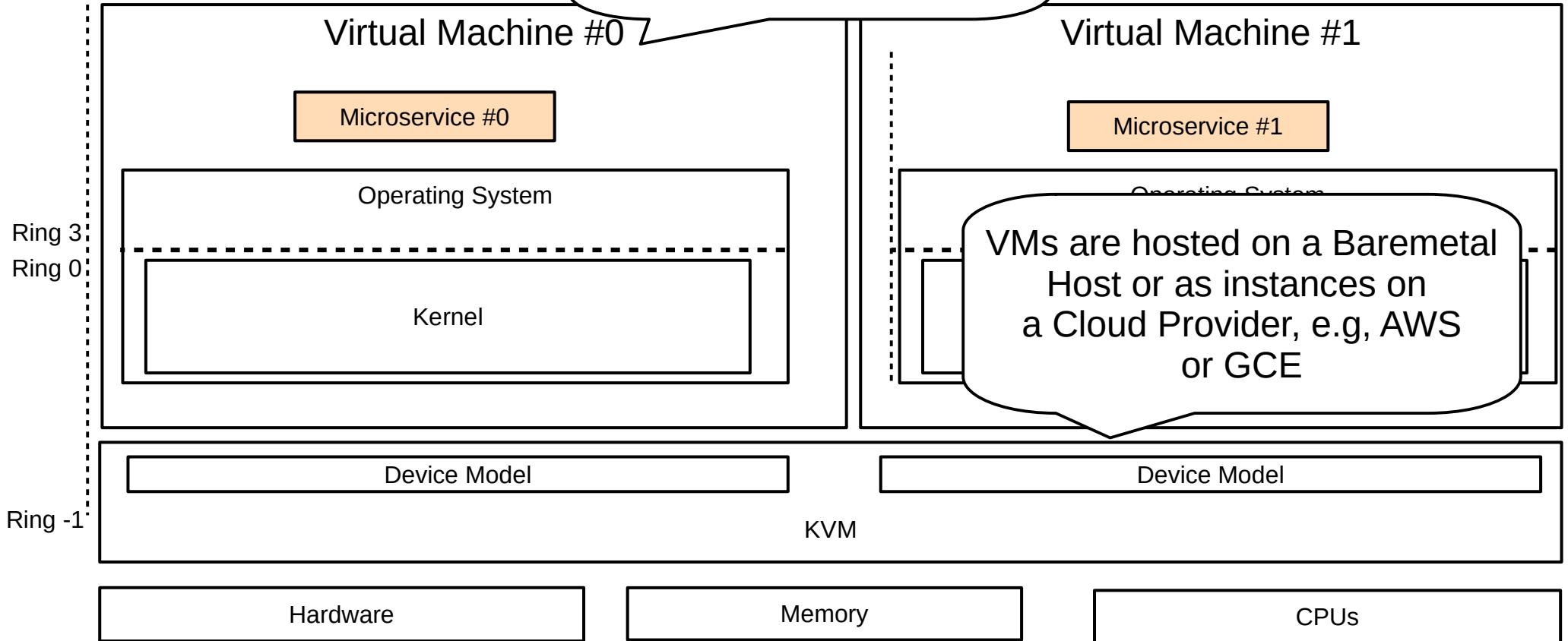
Monolithic Application

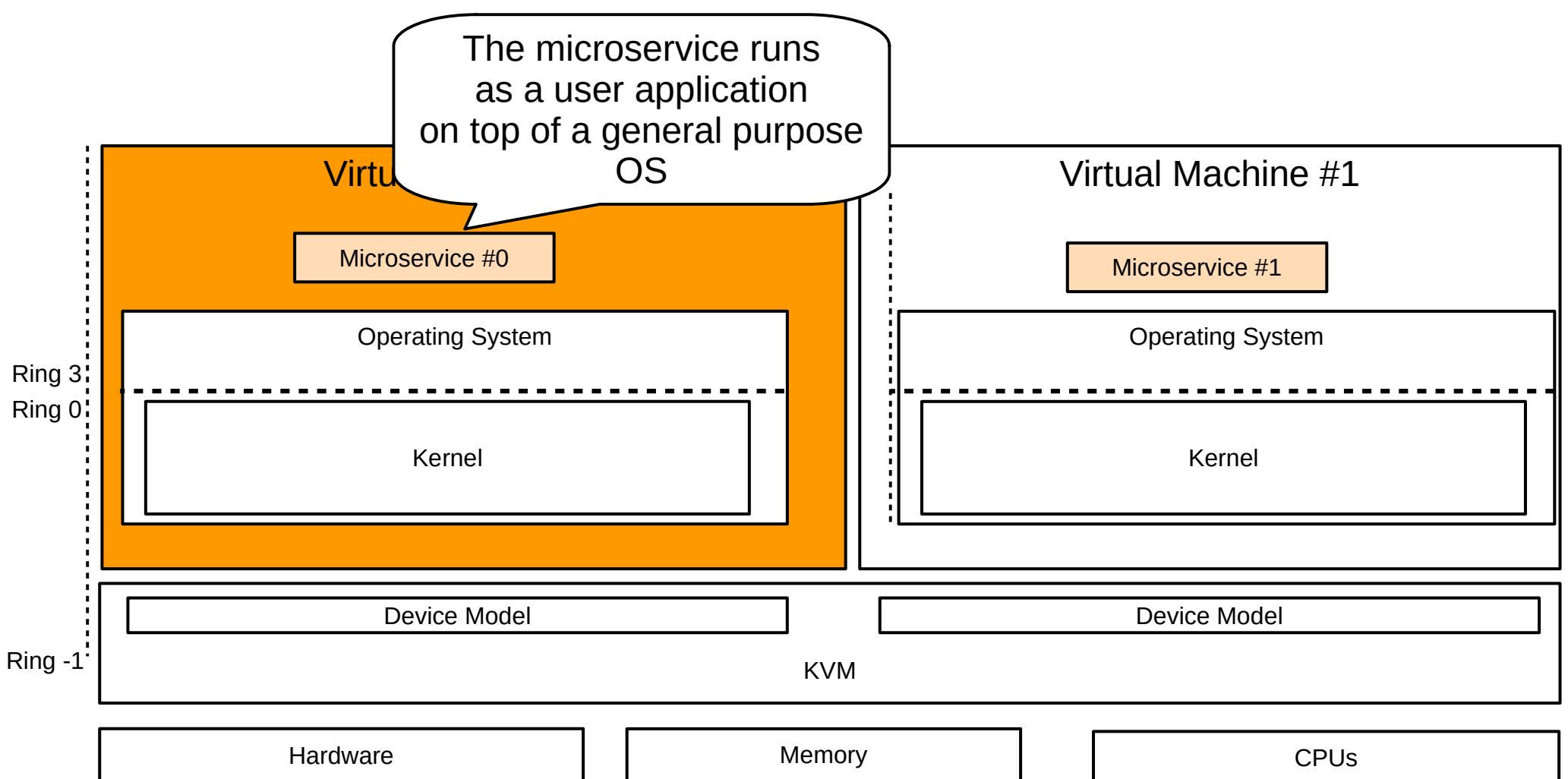
Decomposed Application
into Services

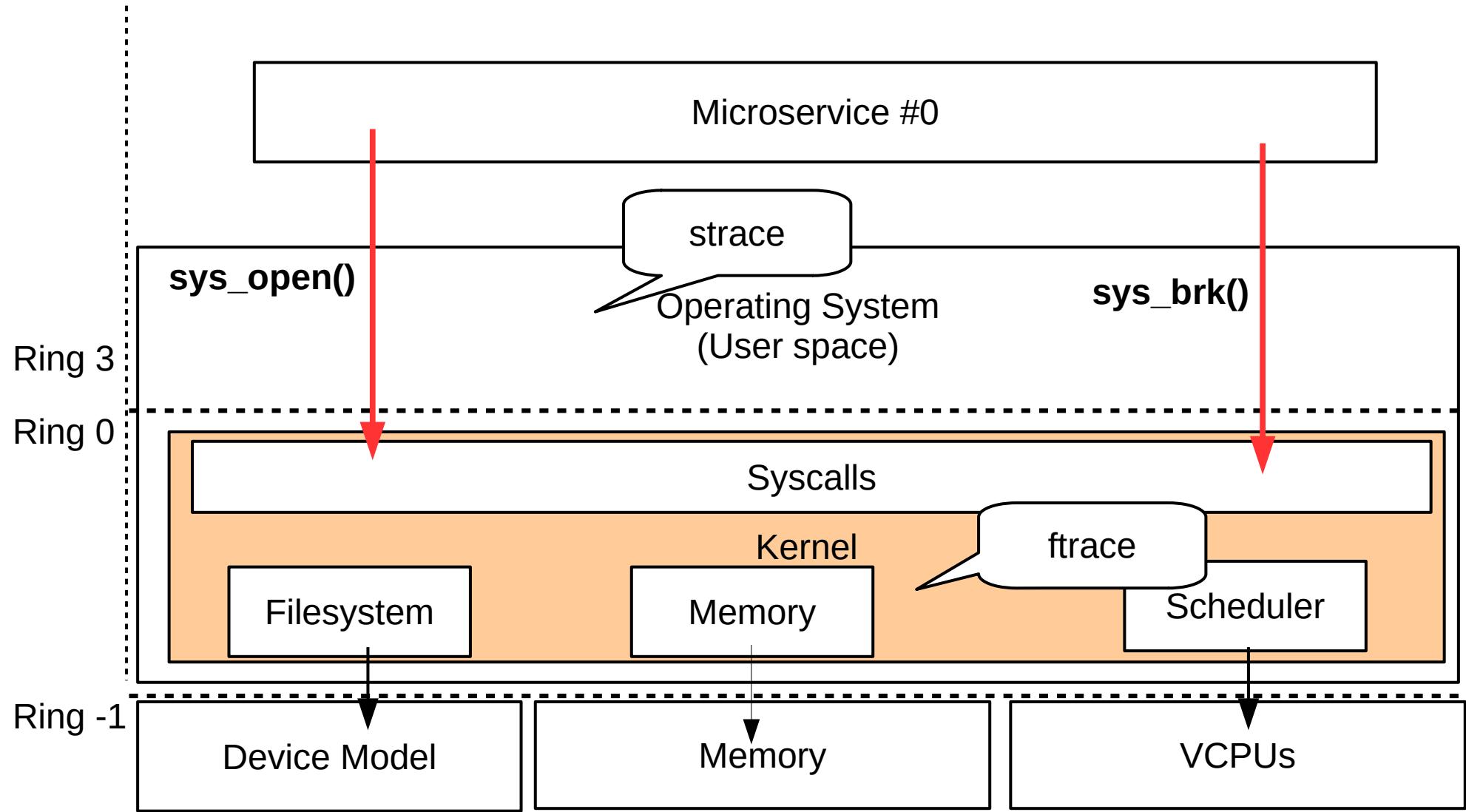




The use of VMs to host microservices allows to isolate different services

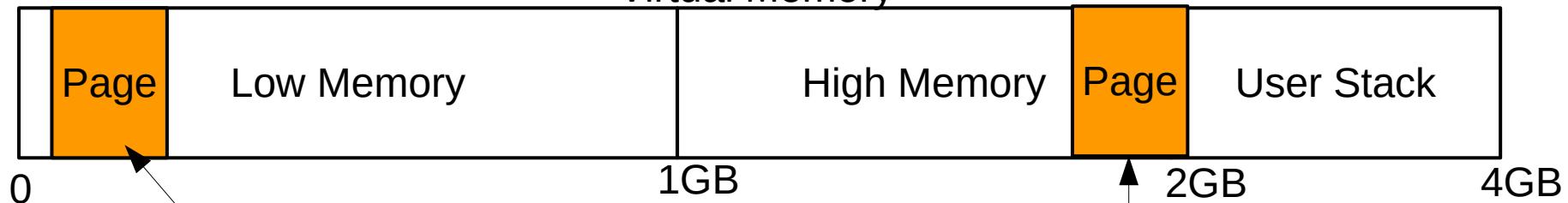






Paging

Virtual Memory



CPU cr3 reg (MMU)

Kernel PDT

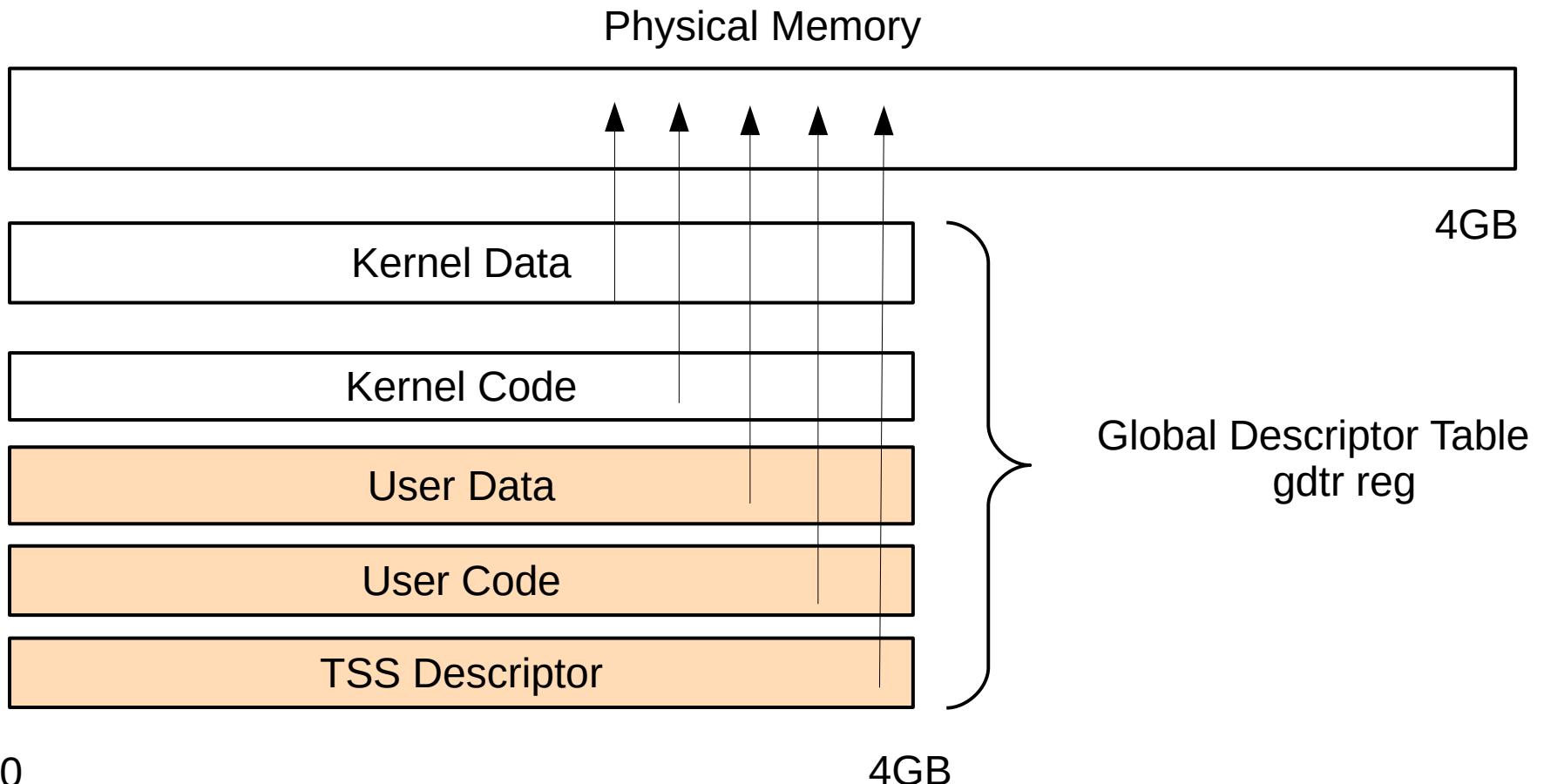
User PDT

Physical Memory

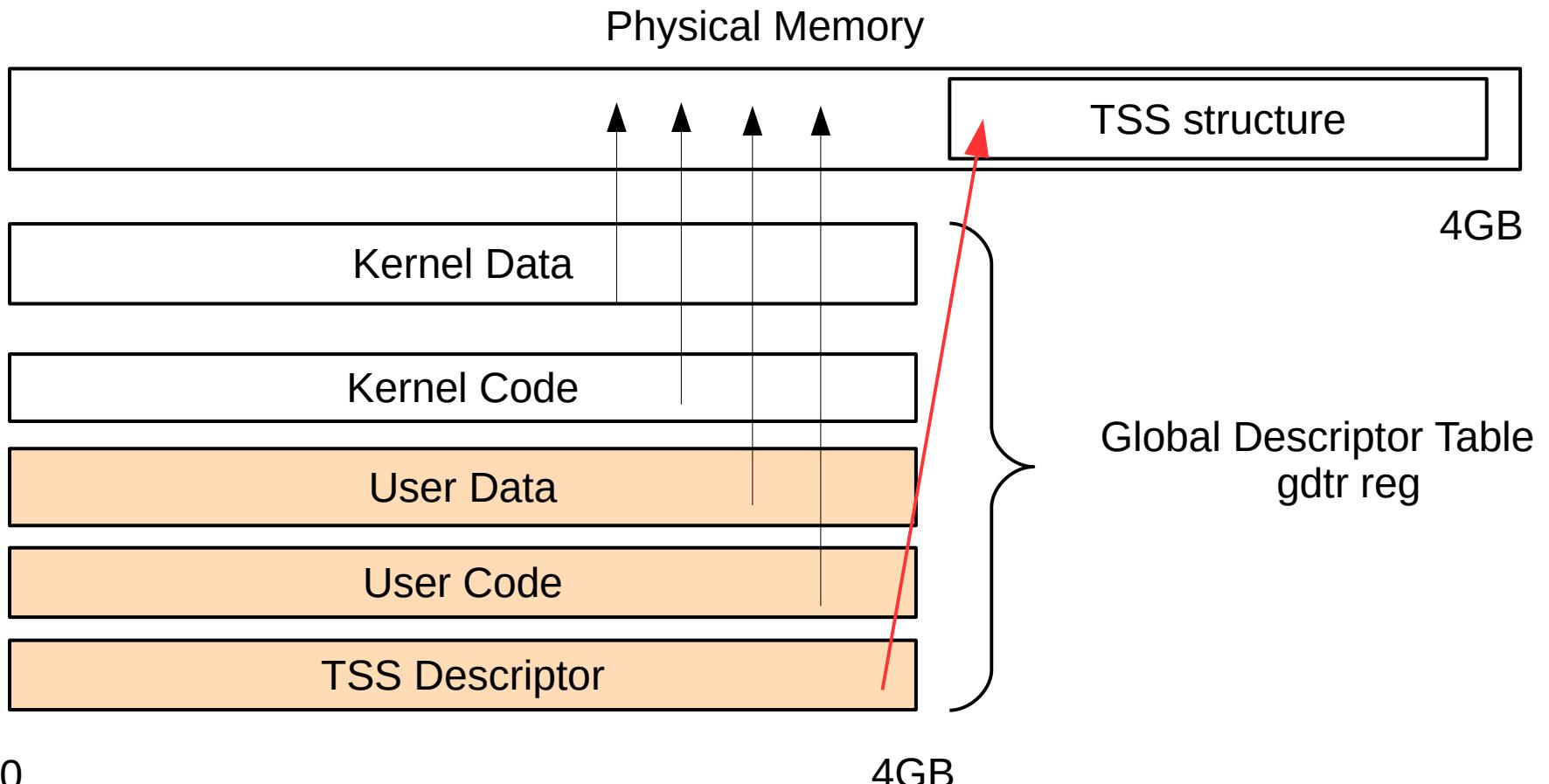
Page

Page

Segmentation



Segmentation



Processor's registers during a syscall

EAX=0ffbe000	EBX=001091e4	ECX=00000000	
ESI=00000000	EDI=00000000	EBP=0ffbdfbf	
EIP=00103381	EEI=00000007	[S APC]	CPL=0 TI=0 A20=1 SMM=0 HLT=1
FS =0010 00000000	ffffffffff	00cf9300	DPL=0 DS [-WA]
CS =0008 00000000	ffffffffff	00cf9a00	DPL=0 CS32 [-R-]
SS =0010 00000000	ffffffffff	00cf9300	DPL=0 DS [-WA]
DS =0010 00000000	ffffffffff	00cf9300	DPL=0 DS [-WA]
FS =001b 00000000	ffffffffff	00cff300	DPL=3 DS [-WA]
GS =001b 00000000	ffffffffff	00cff300	DPL=3 DS [-WA]
LDT=0000 00000000	00000000	00000000	
TR =0028 0f15	TSS register	1900	DPL=0 TI=1
GDT= 000020			
IDT= 00000000 000001b8			
CR0=80000019	CR2=00000000	CR3=0ffbcb000	CR4=00000000

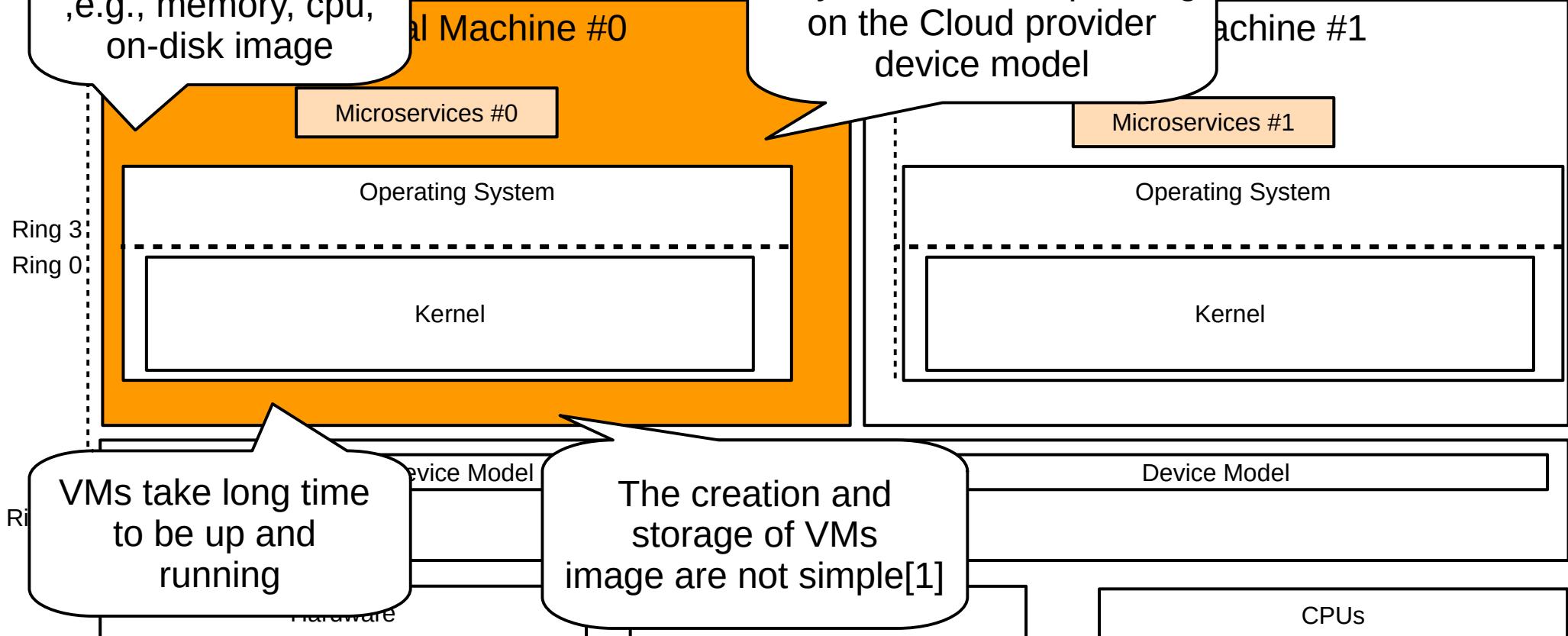
Code and Data for
Kernel's usage

Code and Data for
User's usage

Kernel PDT

Guests consume
a lot of resources
, e.g., memory, cpu,
on-disk image

A different set of drivers
may be needed depending
on the Cloud provider
device model

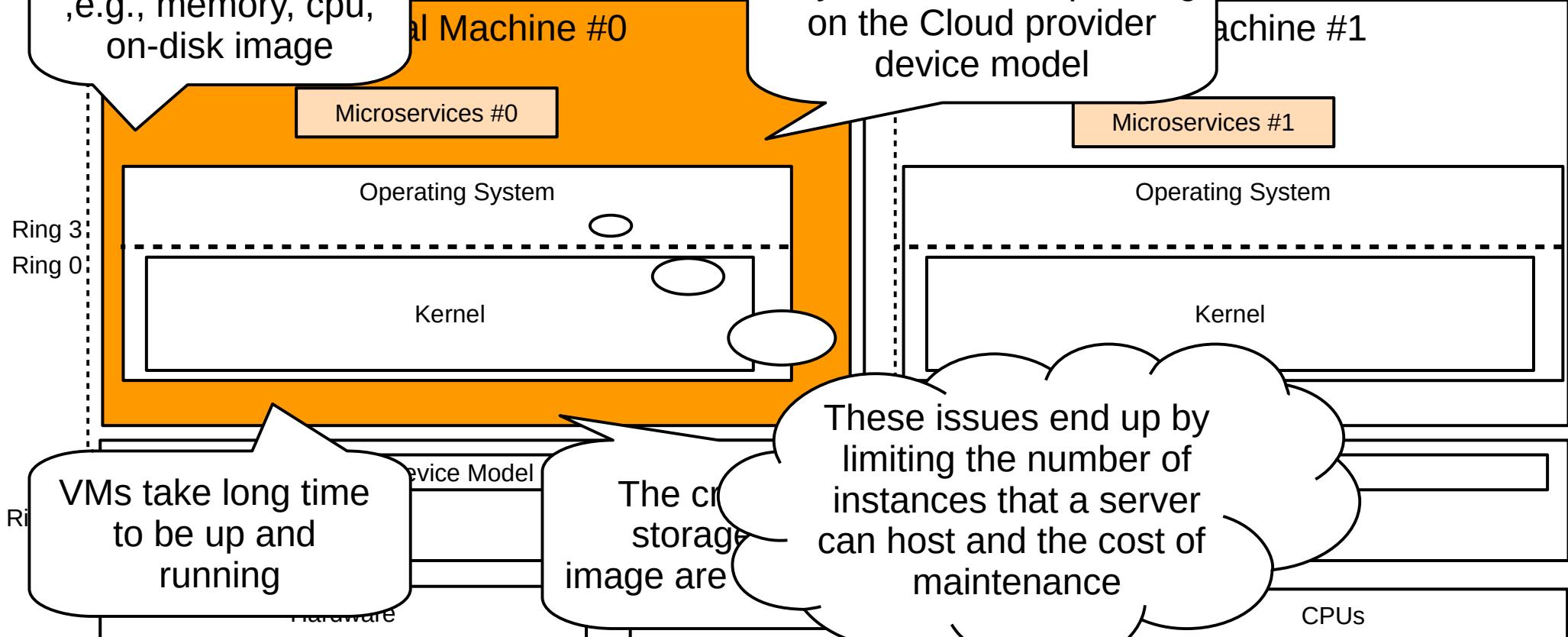


[1] “Demystifying container vs VM-based security: Security in plaintext”

(<https://cloud.google.com/blog/products/gcp/demystifying-container-vs-vm-based-security-security-in-plaintext>)

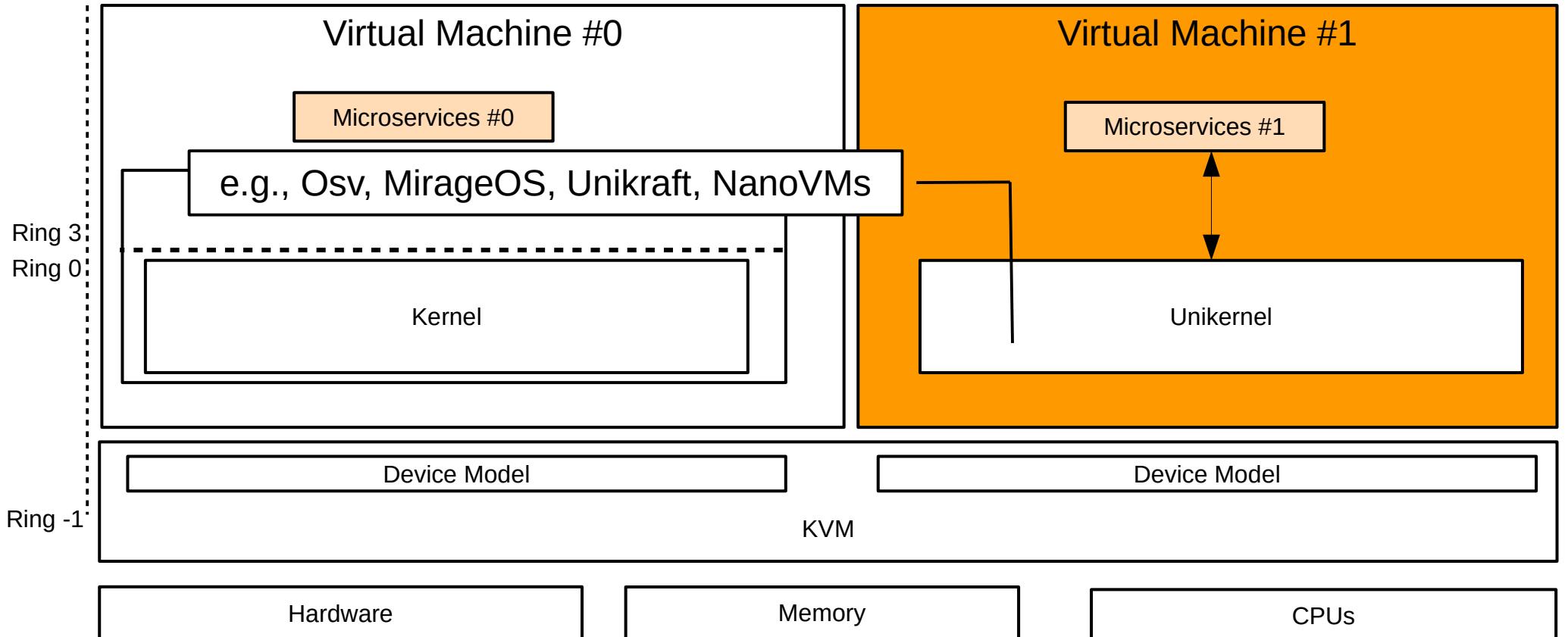
Guests consume a lot of resources, e.g., memory, cpu, on-disk image

A different set of drivers may be needed depending on the Cloud provider device model

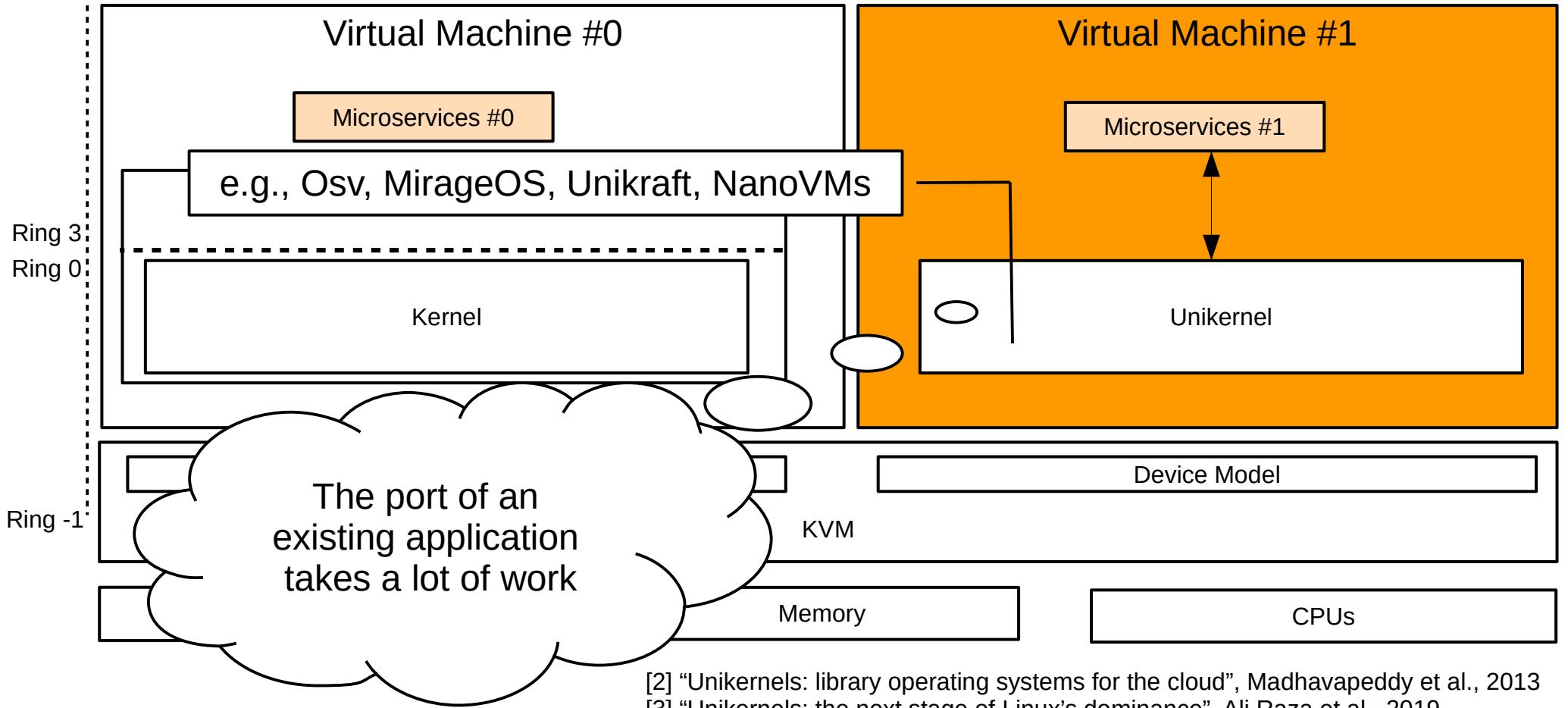


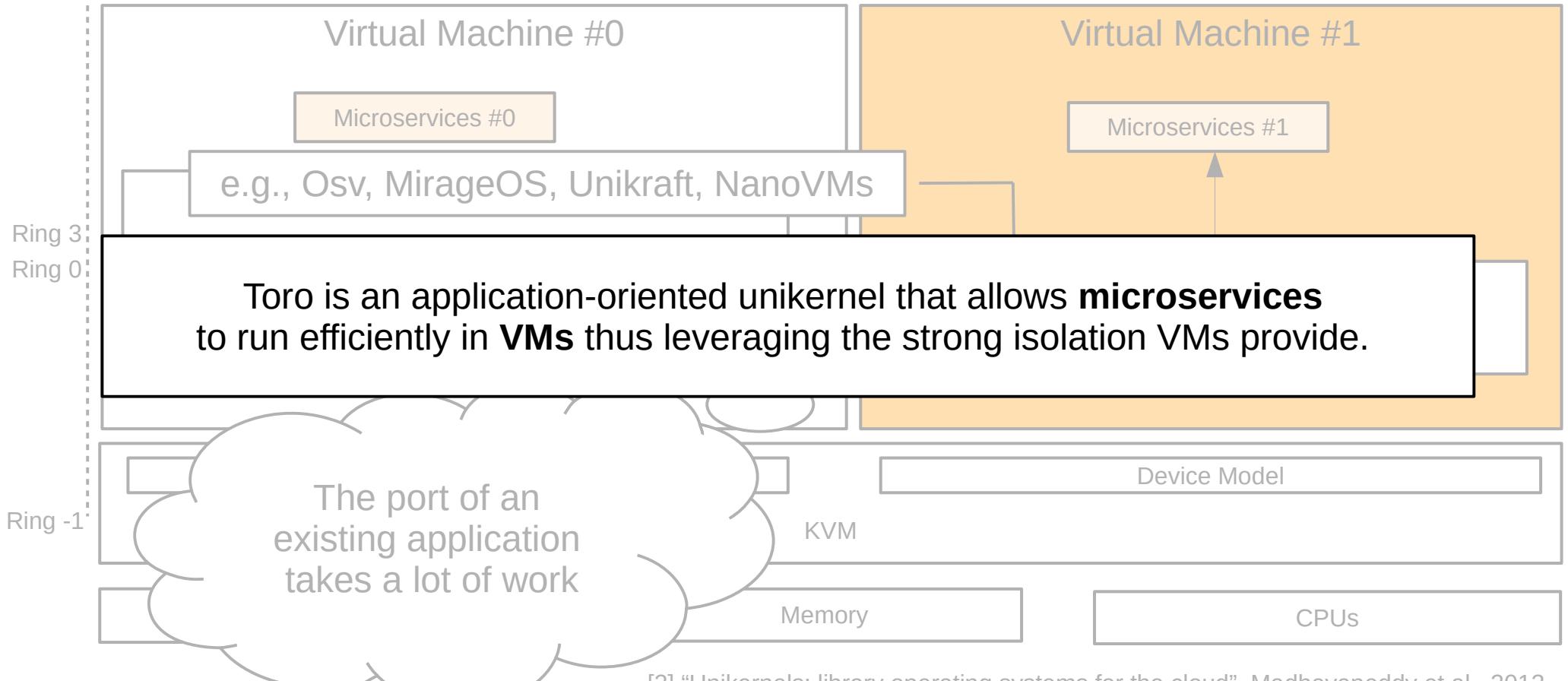
[1] “Demystifying container vs VM-based security: Security in plaintext”

(<https://cloud.google.com/blog/products/gcp/demystifying-container-vs-vm-based-security-security-in-plaintext>)



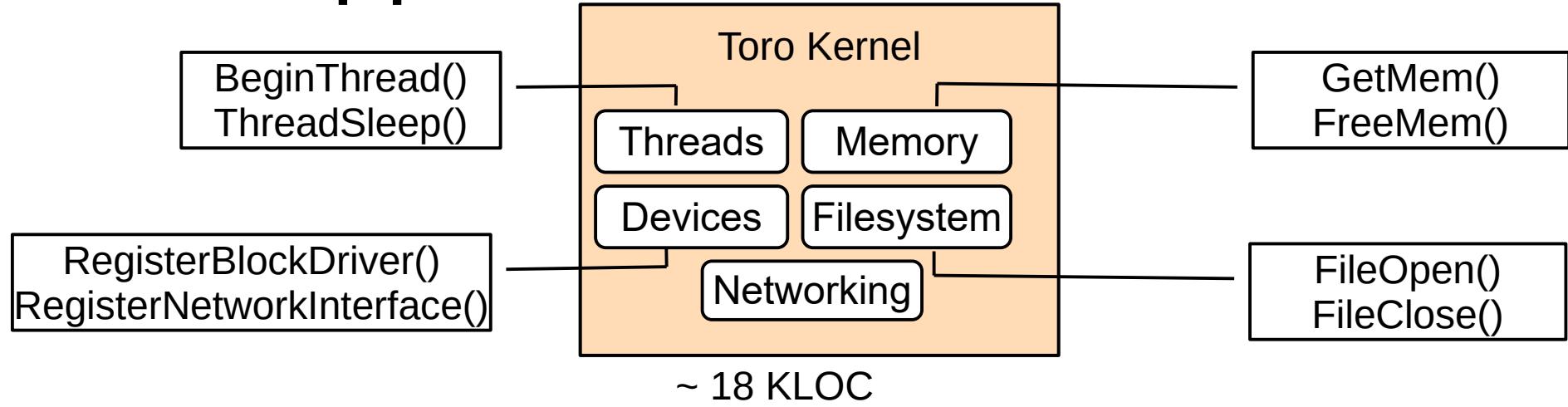
- [2] “Unikernels: library operating systems for the cloud”, Madhavapeddy et al., 2013
[3] “Unikernels: the next stage of Linux’s dominance”, Ali Raza et al., 2019





[2] “Unikernels: library operating systems for the cloud”, Madhavapeddy et al., 2013
[3] “Unikernels: the next stage of Linux’s dominance”, Ali Raza et al., 2019

Application-oriented Kernel

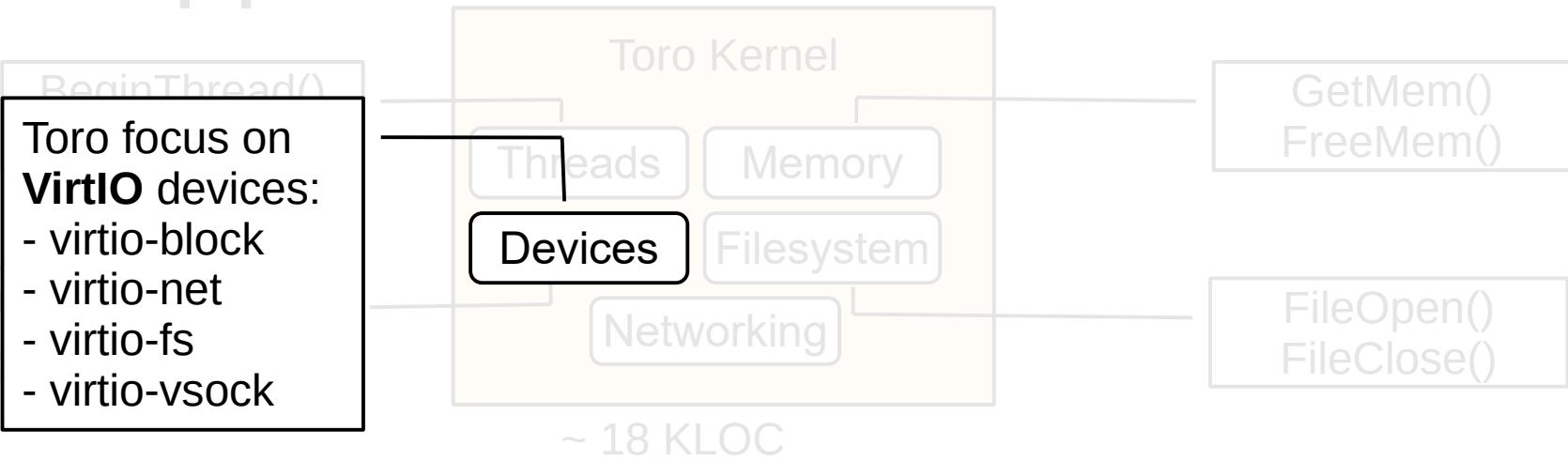


Toro is an embedded kernel including five units:

- Process
- Memory
- Filesystem
- Networking
- Devices, e.g., Block Device, Network Device

Each unit provides minimalist APIs accessible from the embedded application

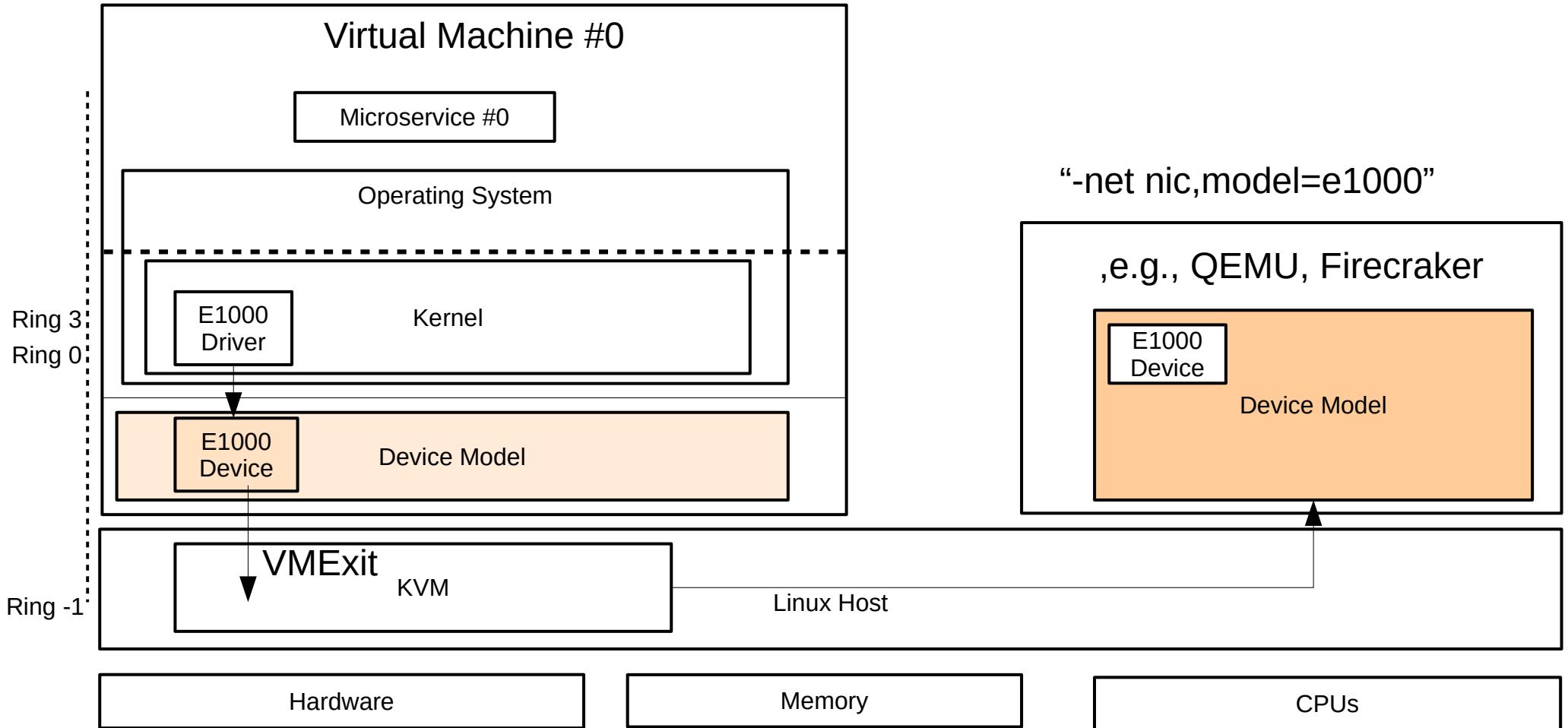
Application-oriented Kernel

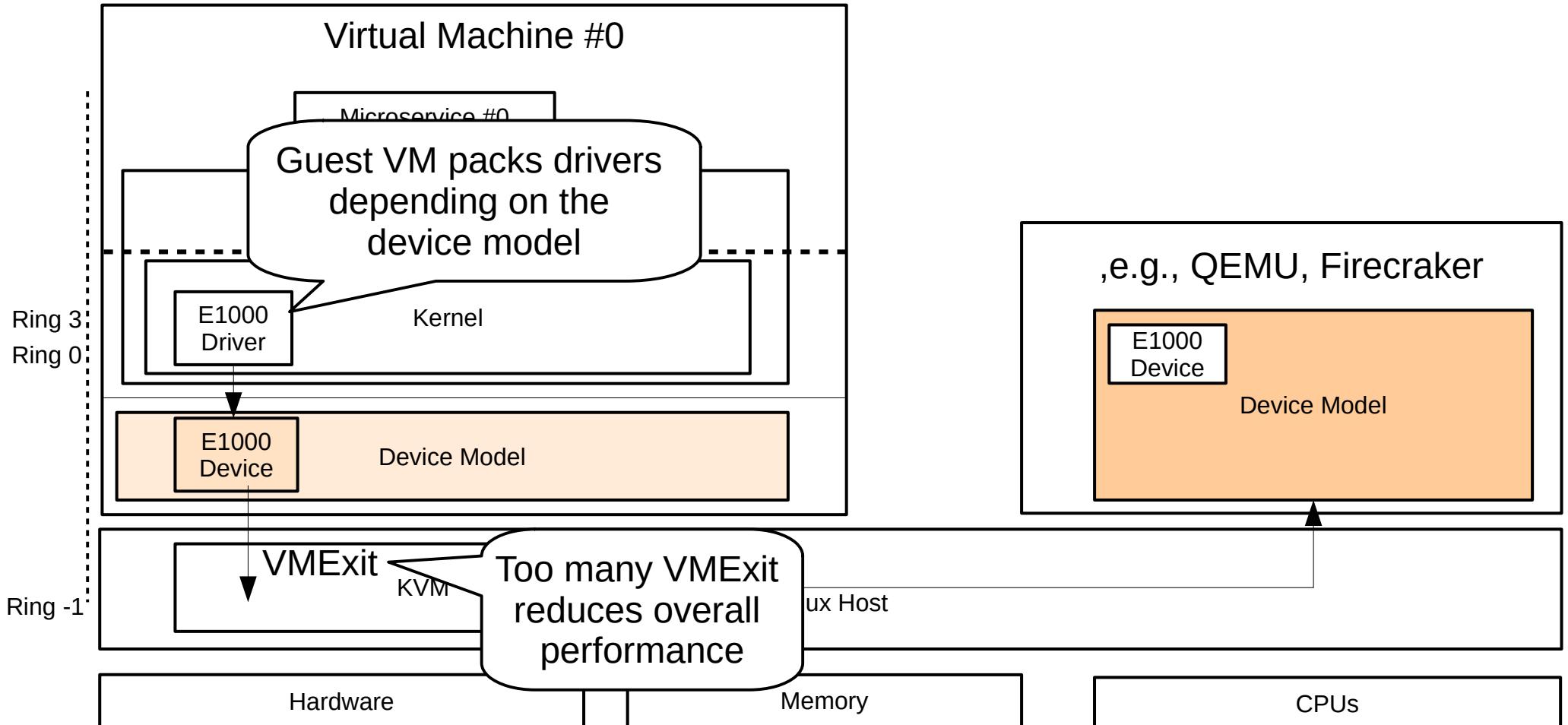


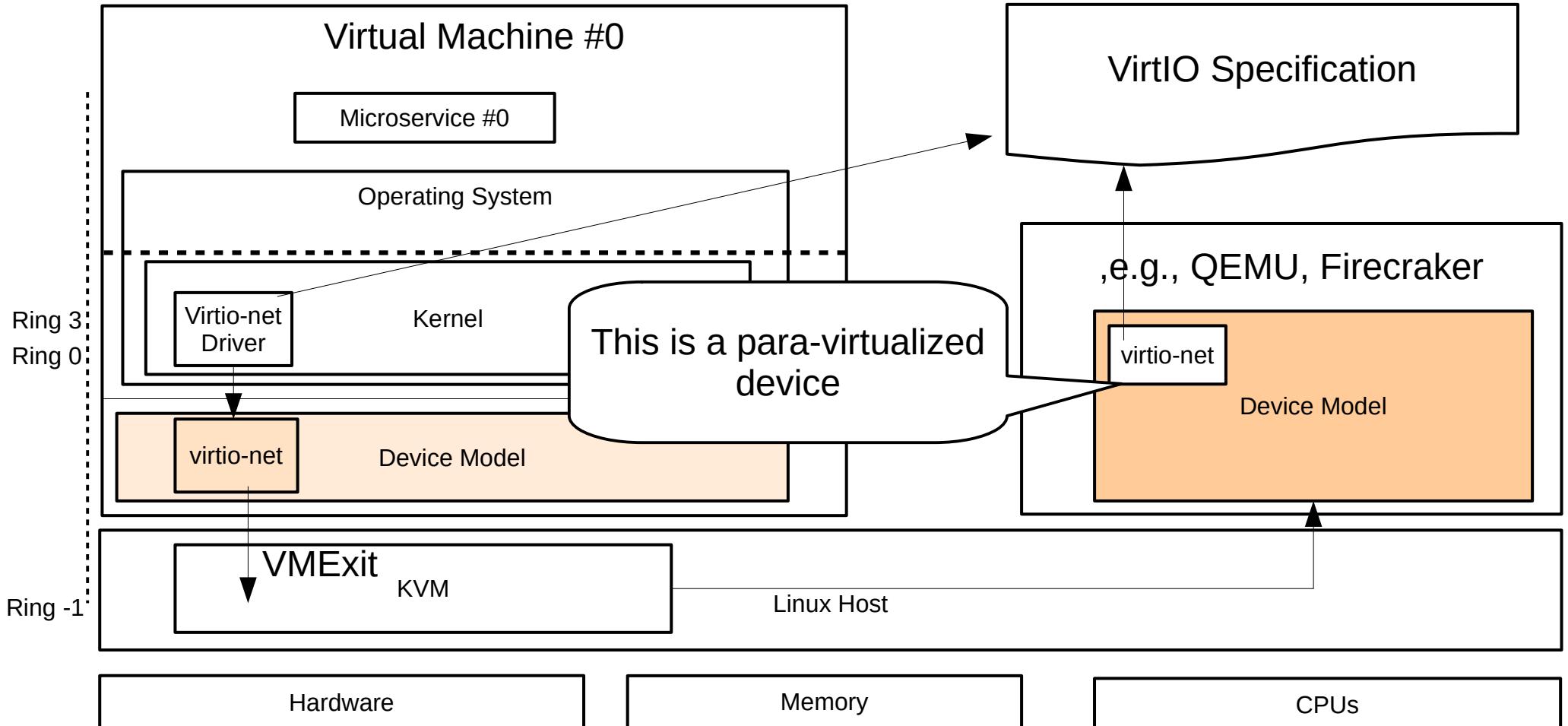
Toro is an embedded kernel including five units:

- Process
- Memory
- Filesystem
- Networking
- Devices, e.g., Block Device, Network Device

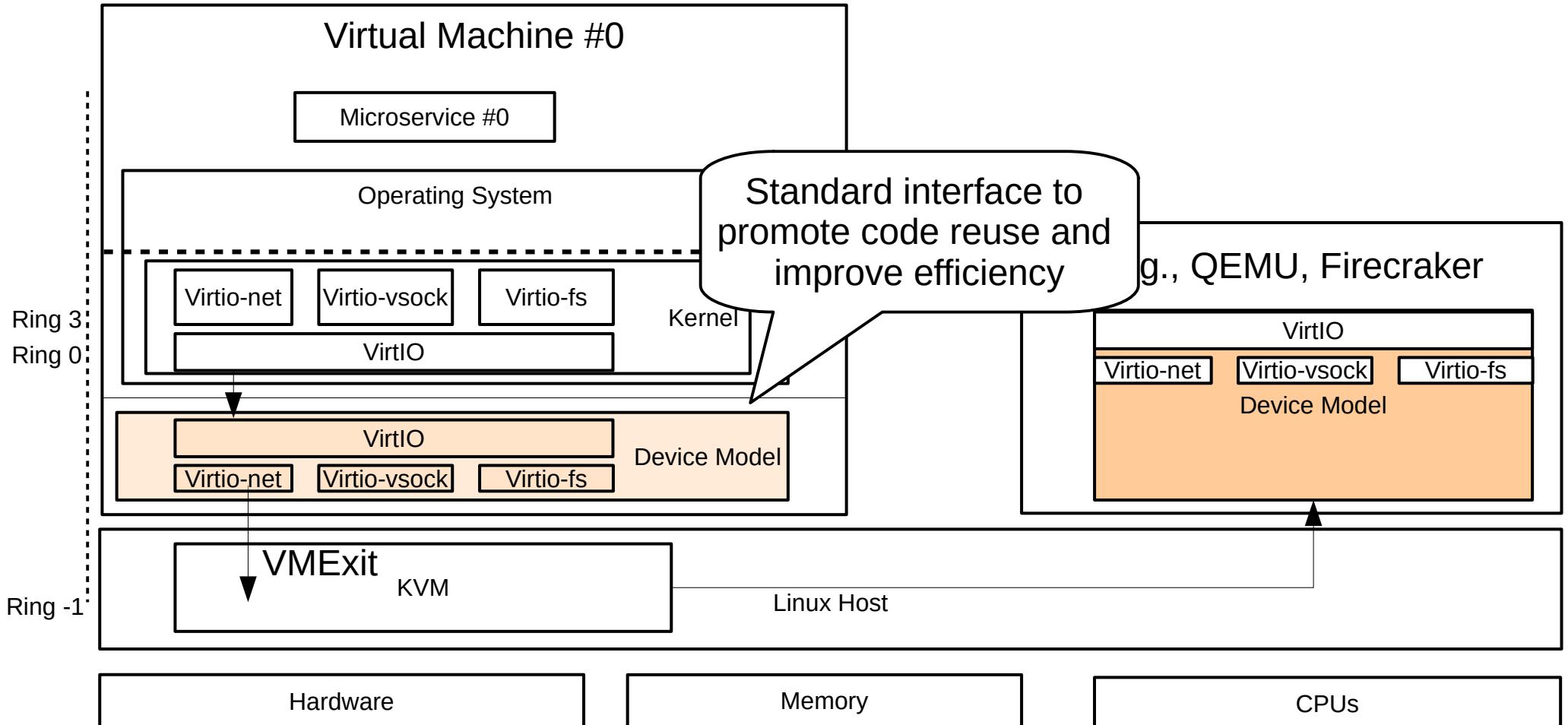
Each unit provides minimalist APIs accessible from the embedded application

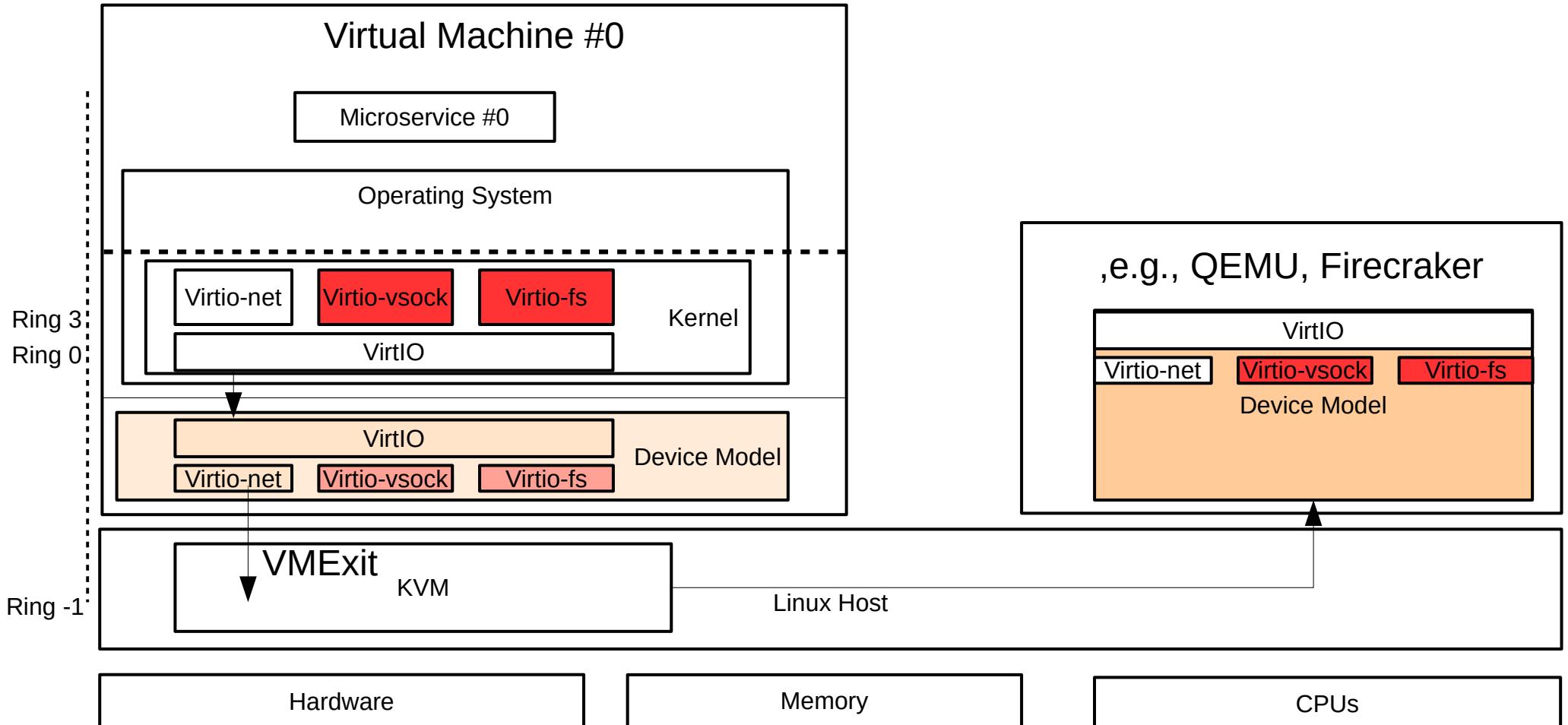






“virtio: towards a de-facto standard for virtual I/O devices”, Rusty Russell, IBM OzLabs.





“Building a Cloud Infrastructure to Deploy Microservices as Microvm Guests”, KVMForum’20
 “Leveraging virtiofs and vsocket in Toro unikernel”, DevConf’20

Virtual Machine #0

Microservice #0

Operating System

Stack TCP/IP

Virtio-net

Virtual Filesystem

Virtio-block

Kernel

virtio-net

virtio-block

Device Model

KVM

Linux Host

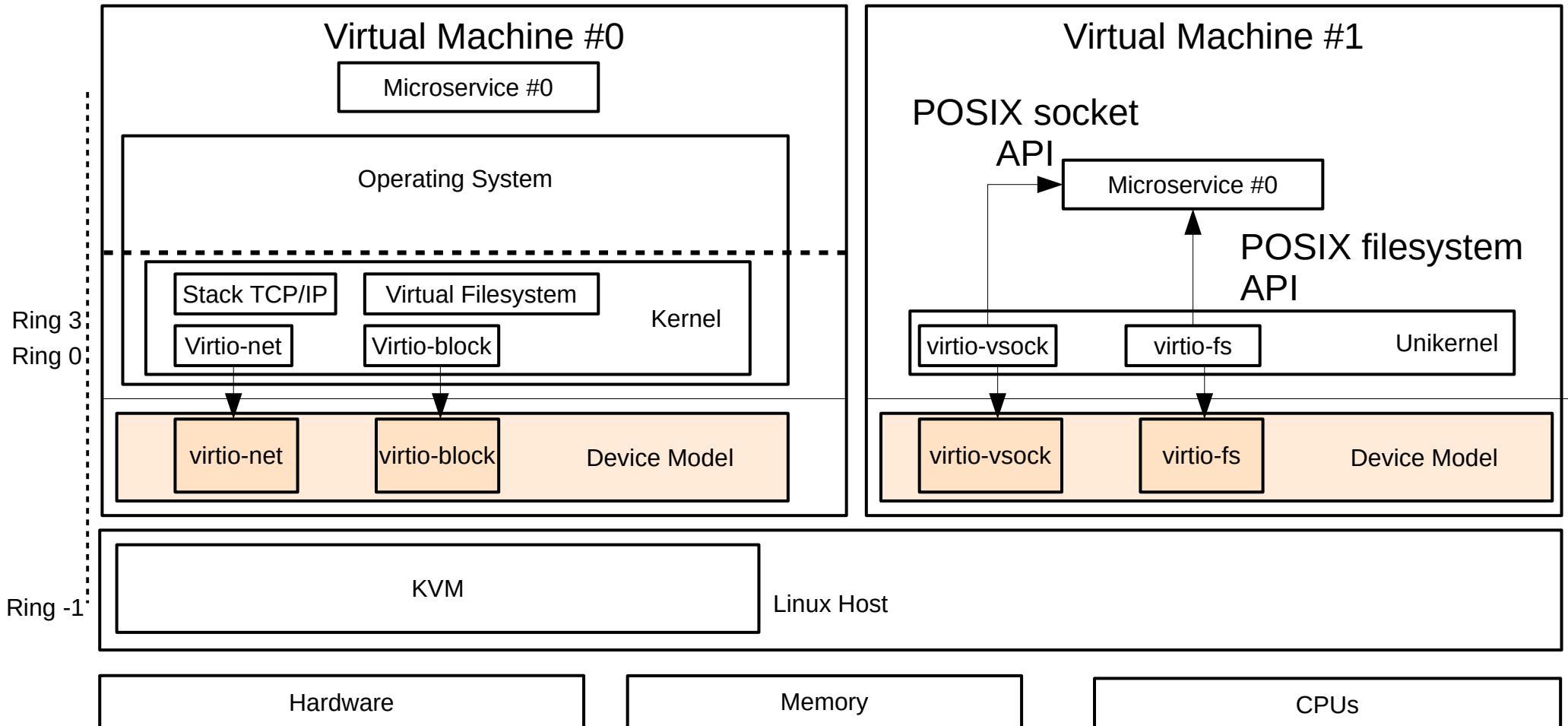
Hardware

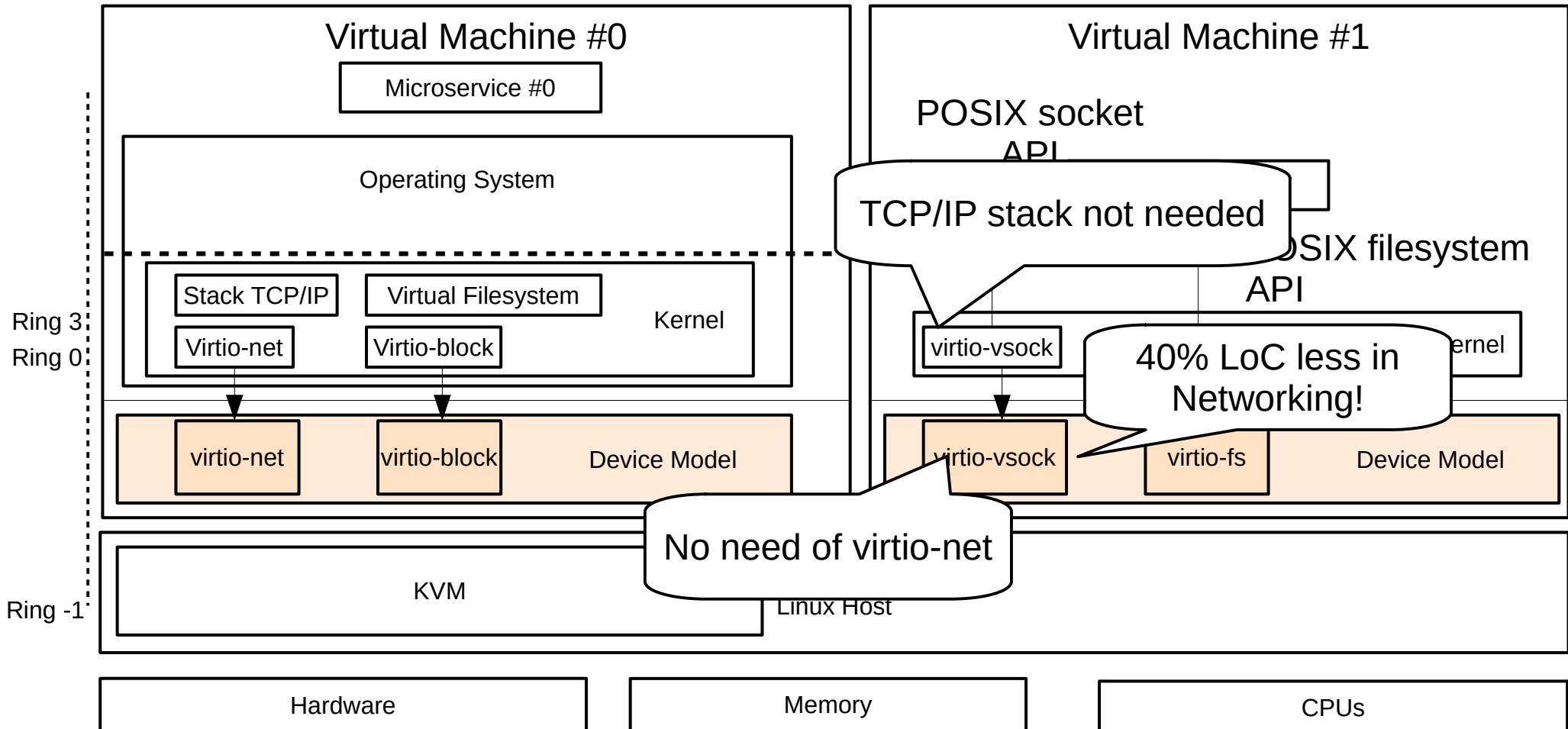
Memory

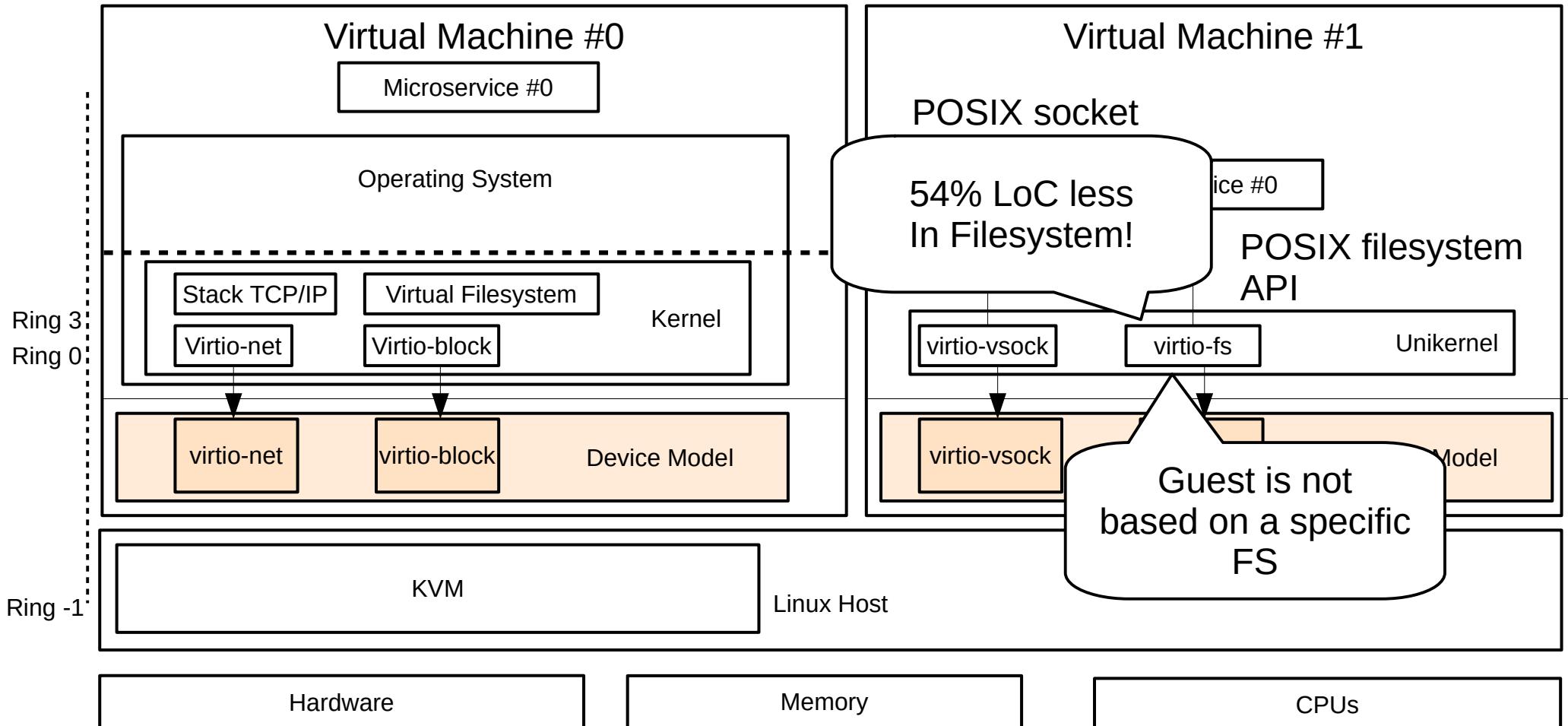
CPUs

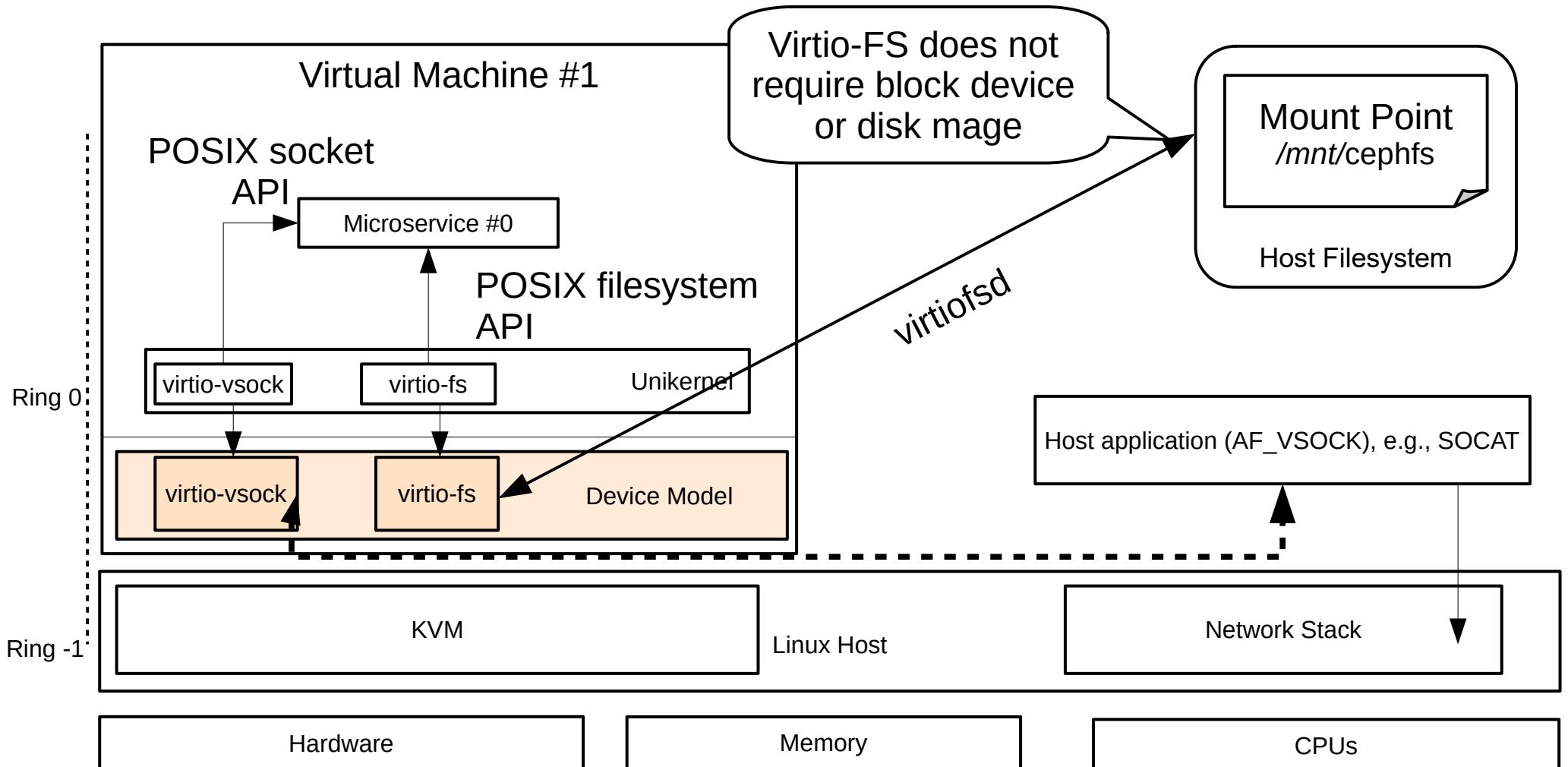
Ring 3
Ring 0

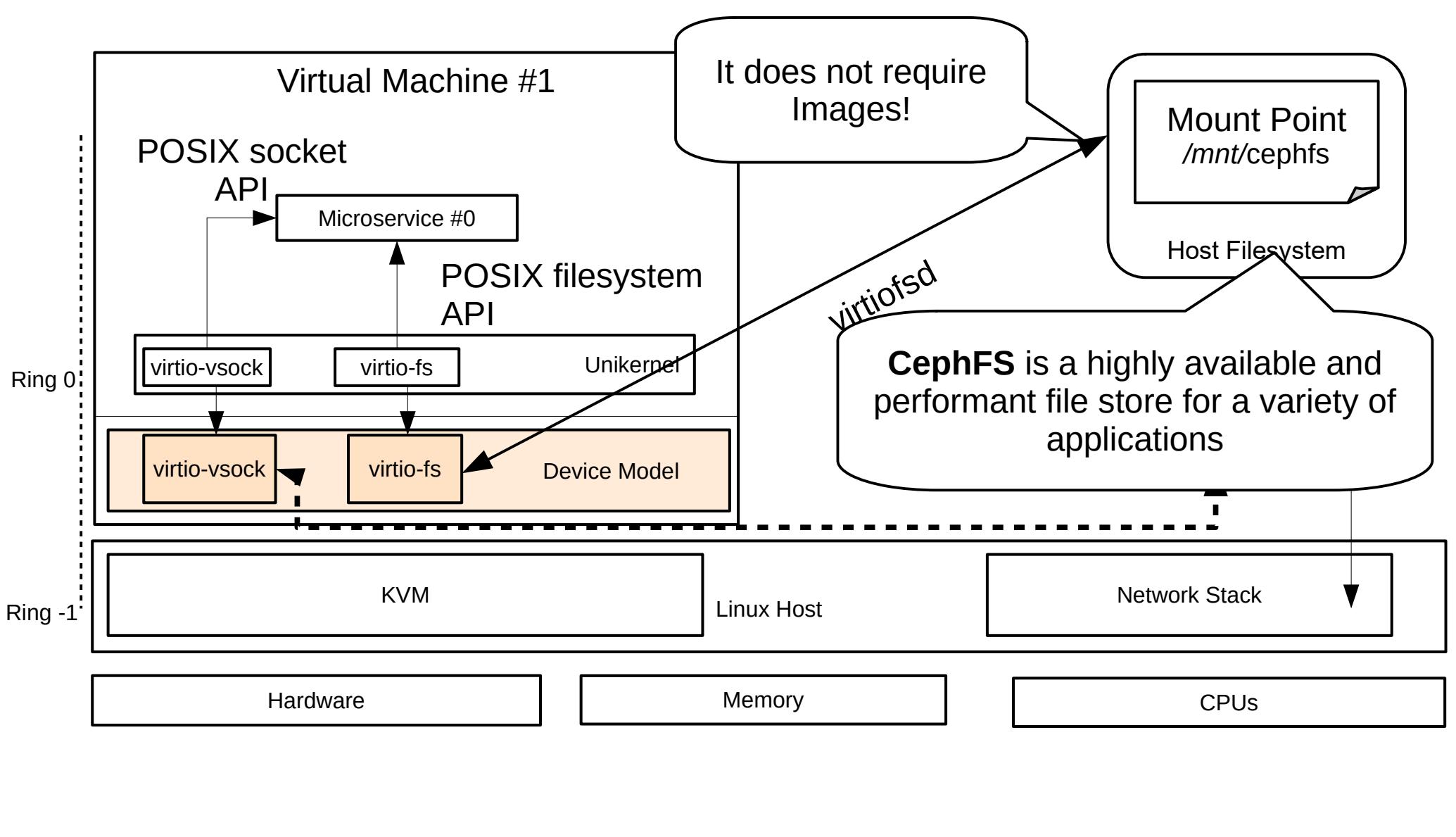
Ring -1



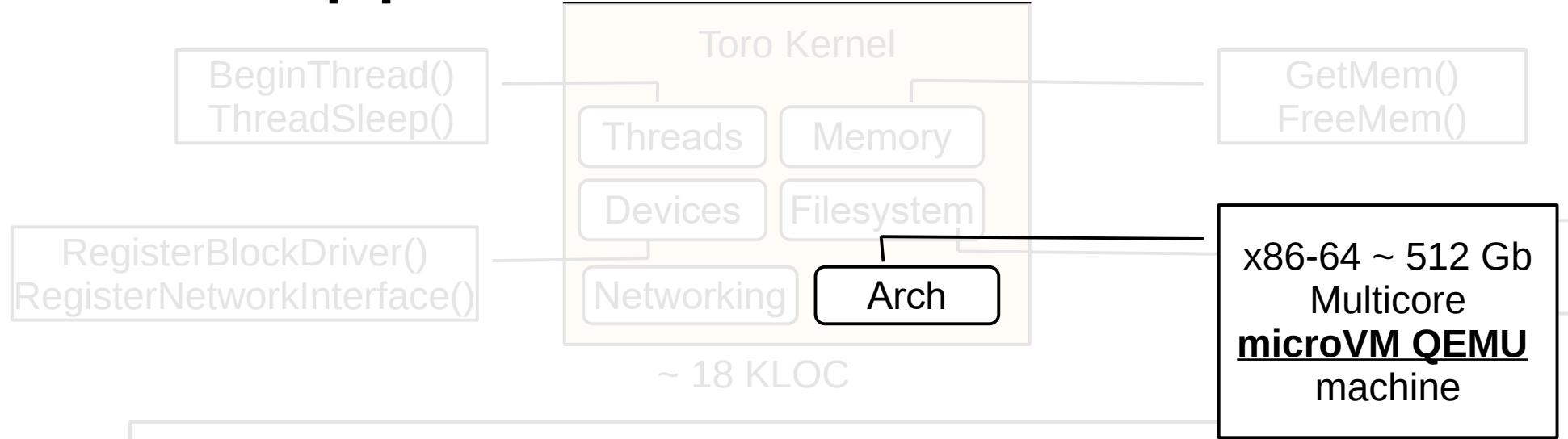








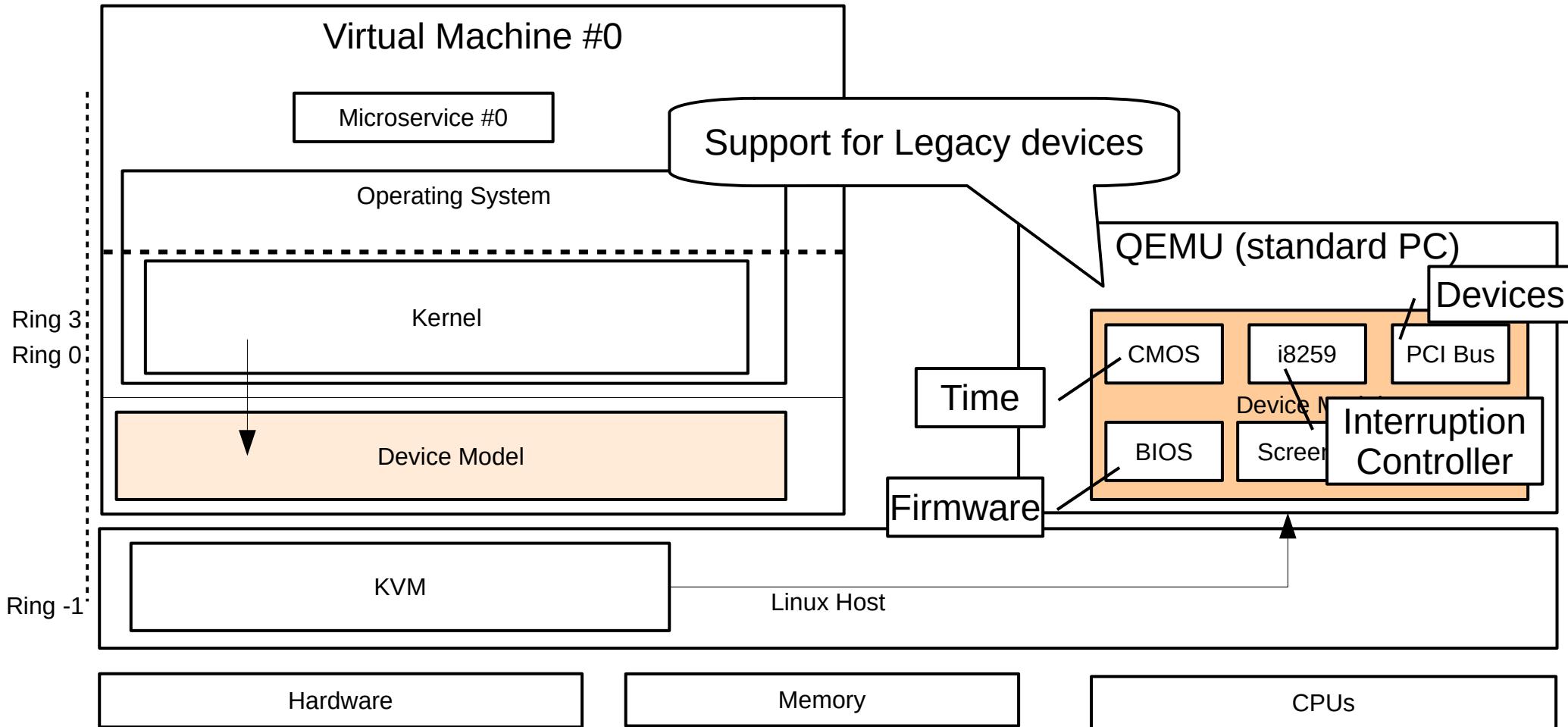
Application-oriented Kernel

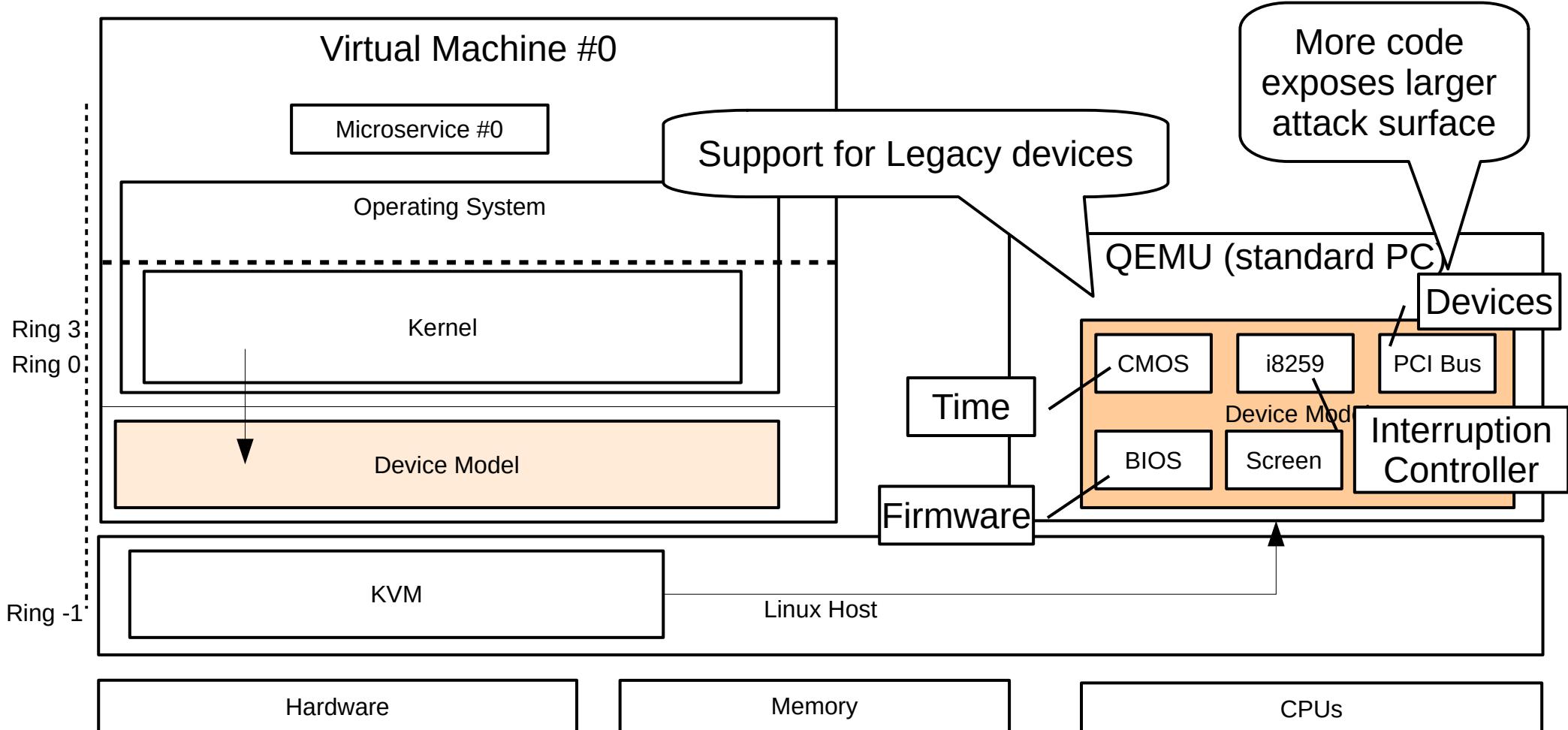


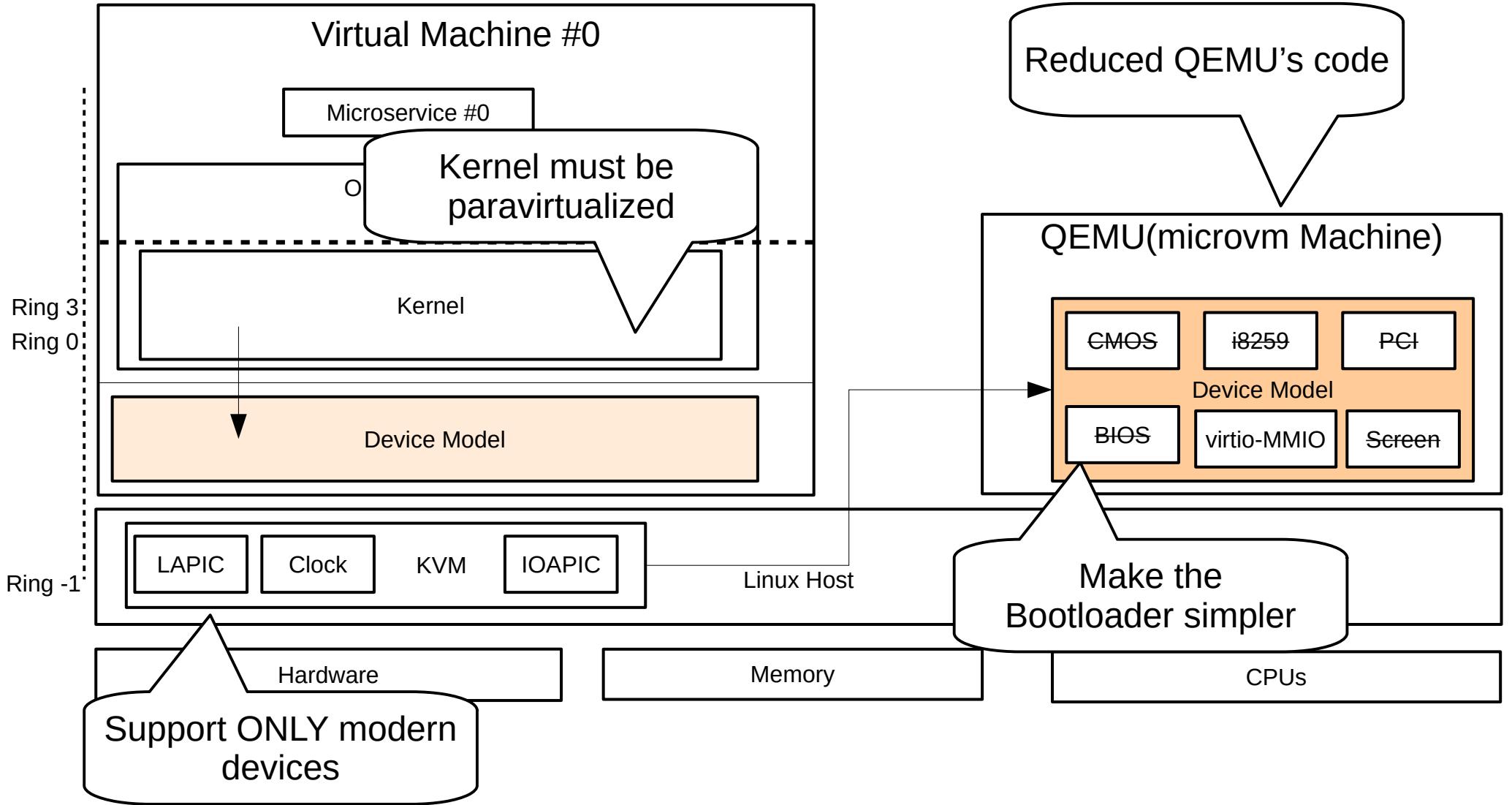
Toro is an embedded kernel including five units:

- Process
- Memory
- Filesystem
- Networking
- Devices, e.g., Block Device, Network Device

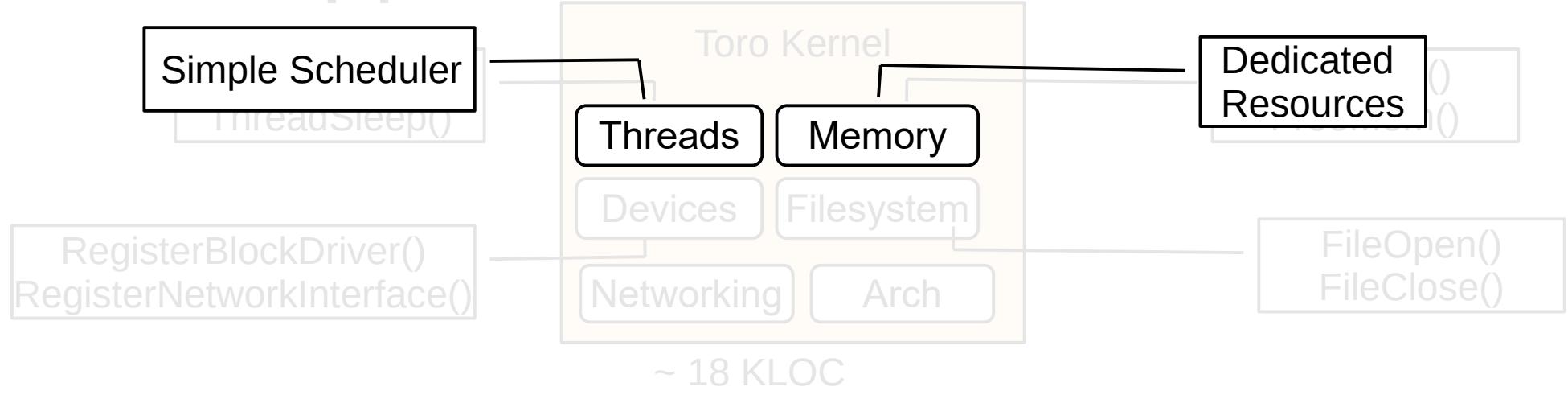
Each unit provides minimalist APIs accessible from the embedded application







Application-oriented Kernel

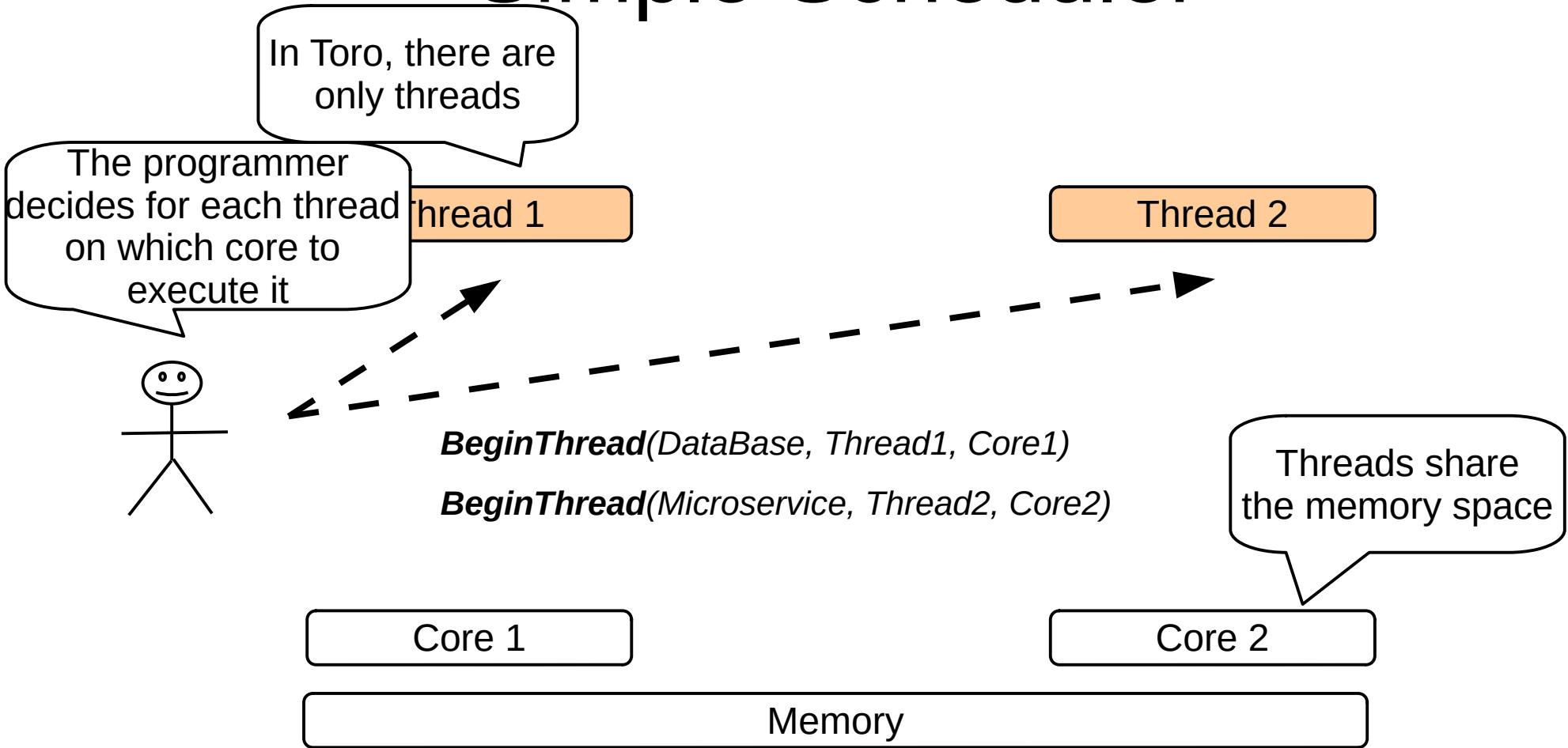


Toro is an embedded kernel including five units:

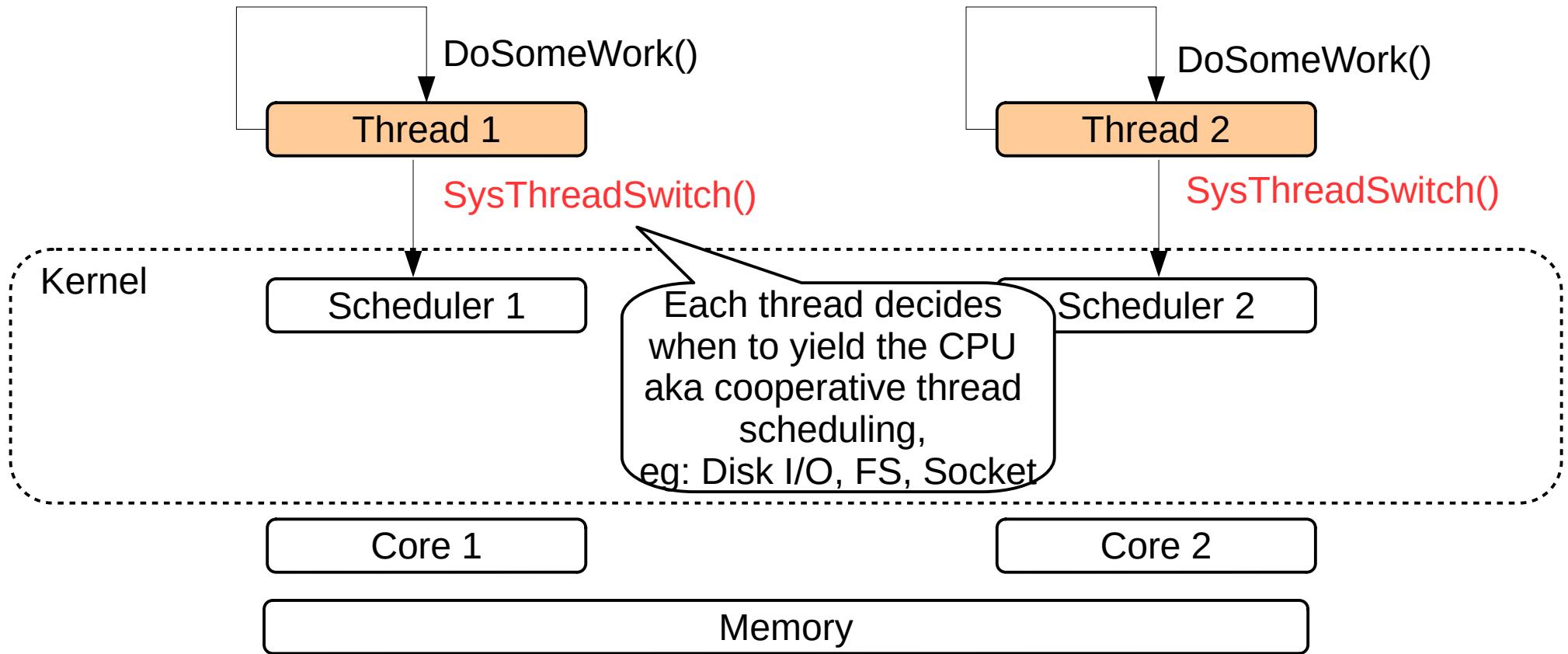
- Process
- Memory
- Filesystem
- Networking
- Devices, e.g., Block Device, Network Device

Each unit provides minimalist APIs accessible from the embedded application

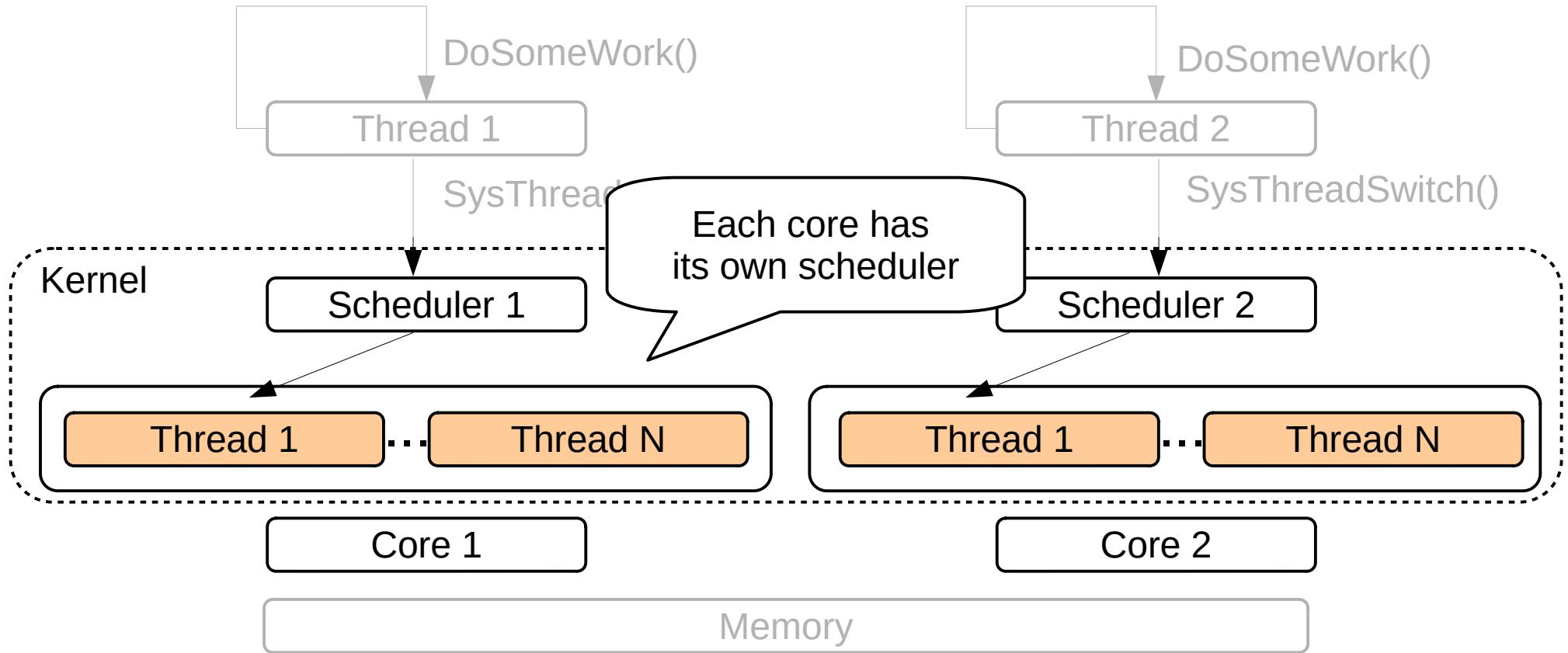
Simple Scheduler



Simple Scheduler

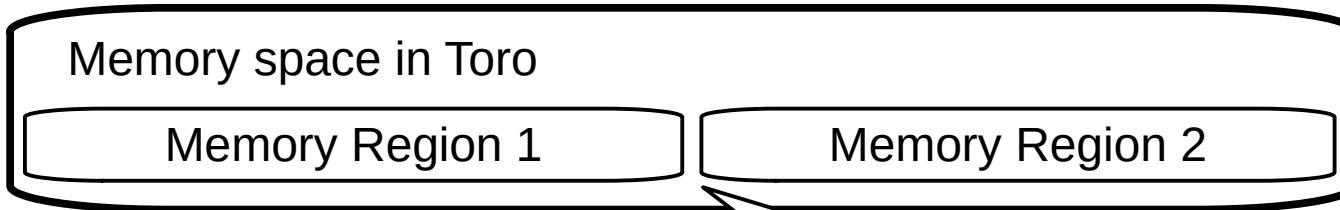


Simple Scheduler



Dedicated Resources

Memory



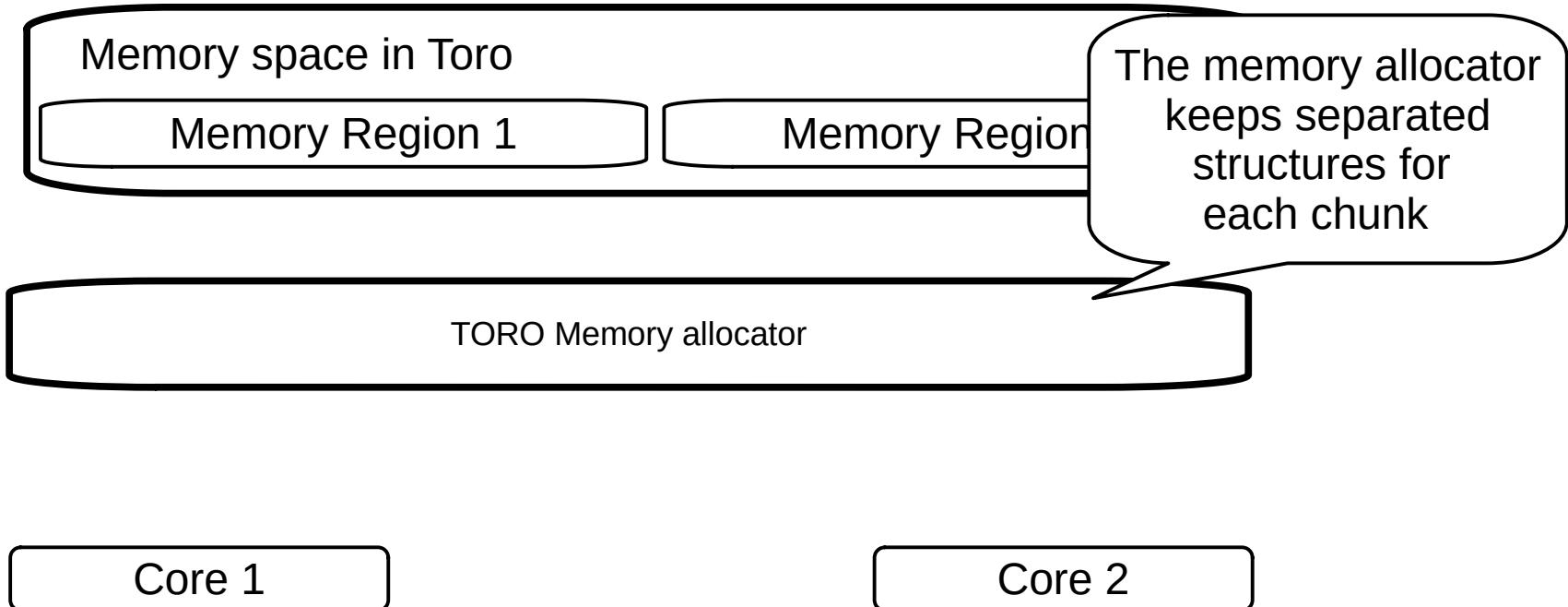
Toro reserves
the same amount
of memory for each
core

Core 1

Core 2

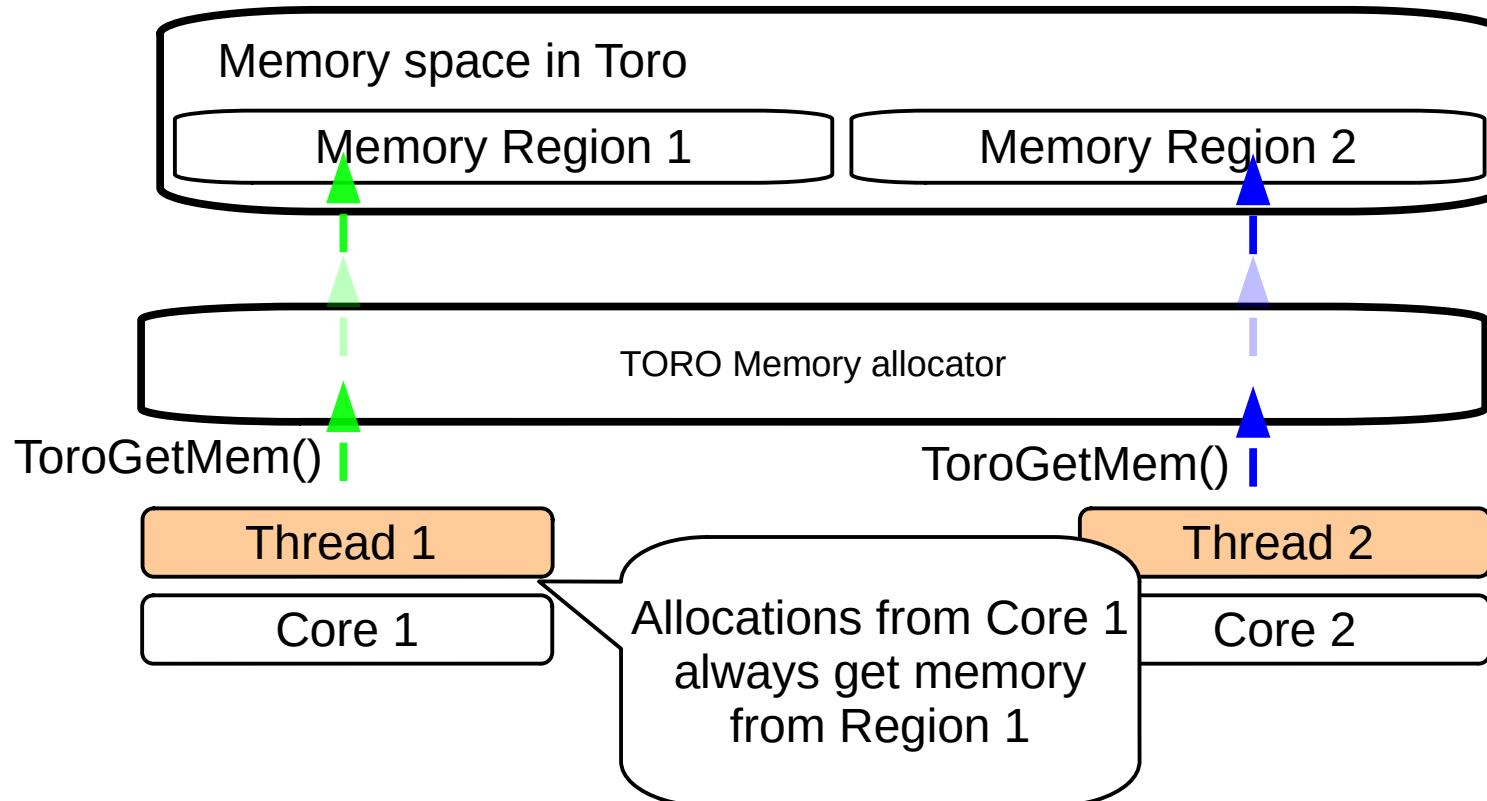
Dedicated Resources

Memory

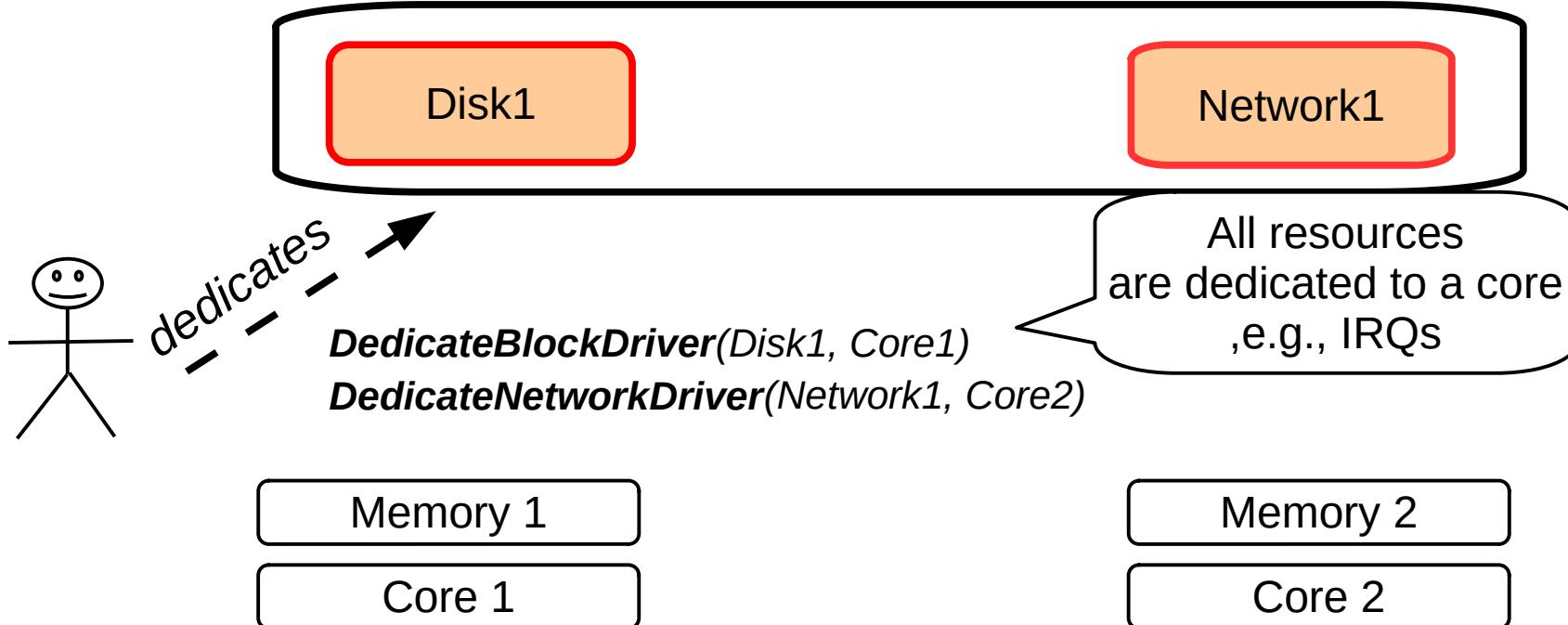


Dedicated Resources

Memory

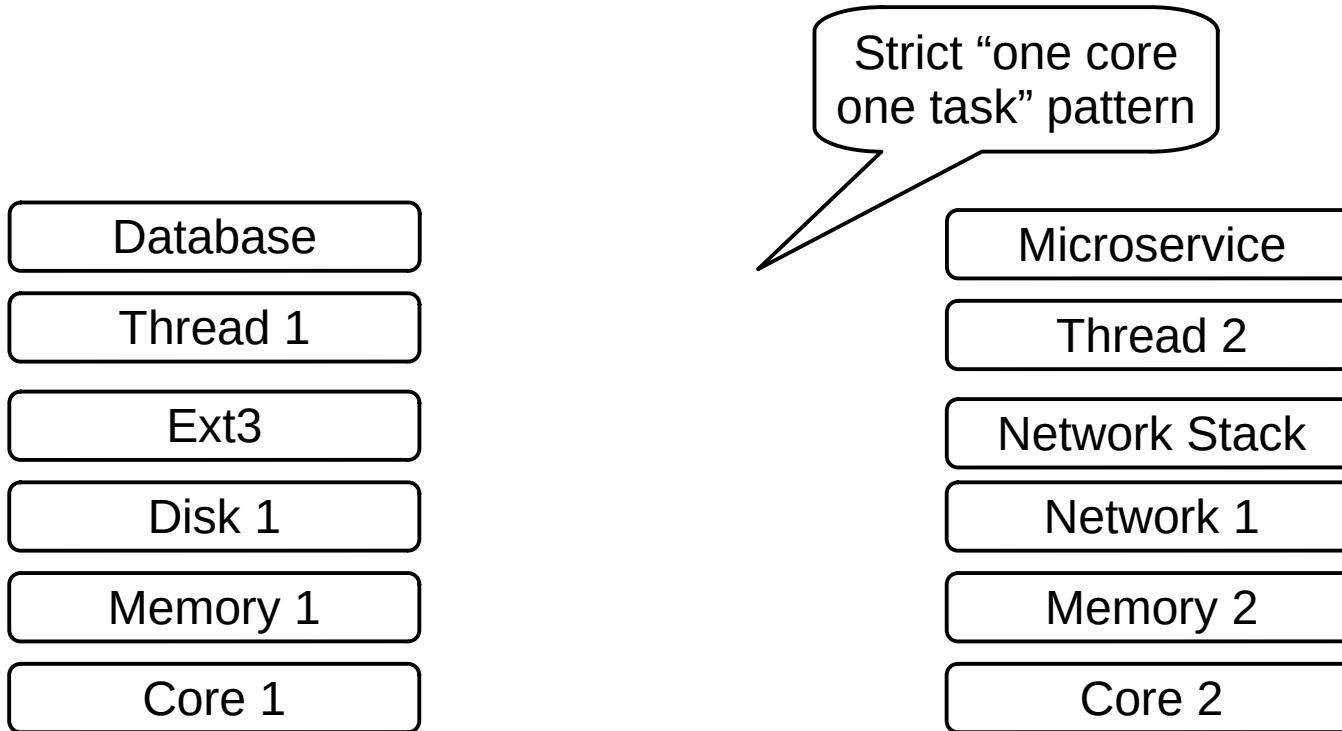


Dedicated Resources Devices



Dedicated Resources

Filesystems and Networking



Dedicated Resources

Filesystems and Networking

Strict “one core
one task” pattern

“Chat is easy, show me the code.” - Linus Torvalds

Ext3

Disk 1

Memory 1

Core 1

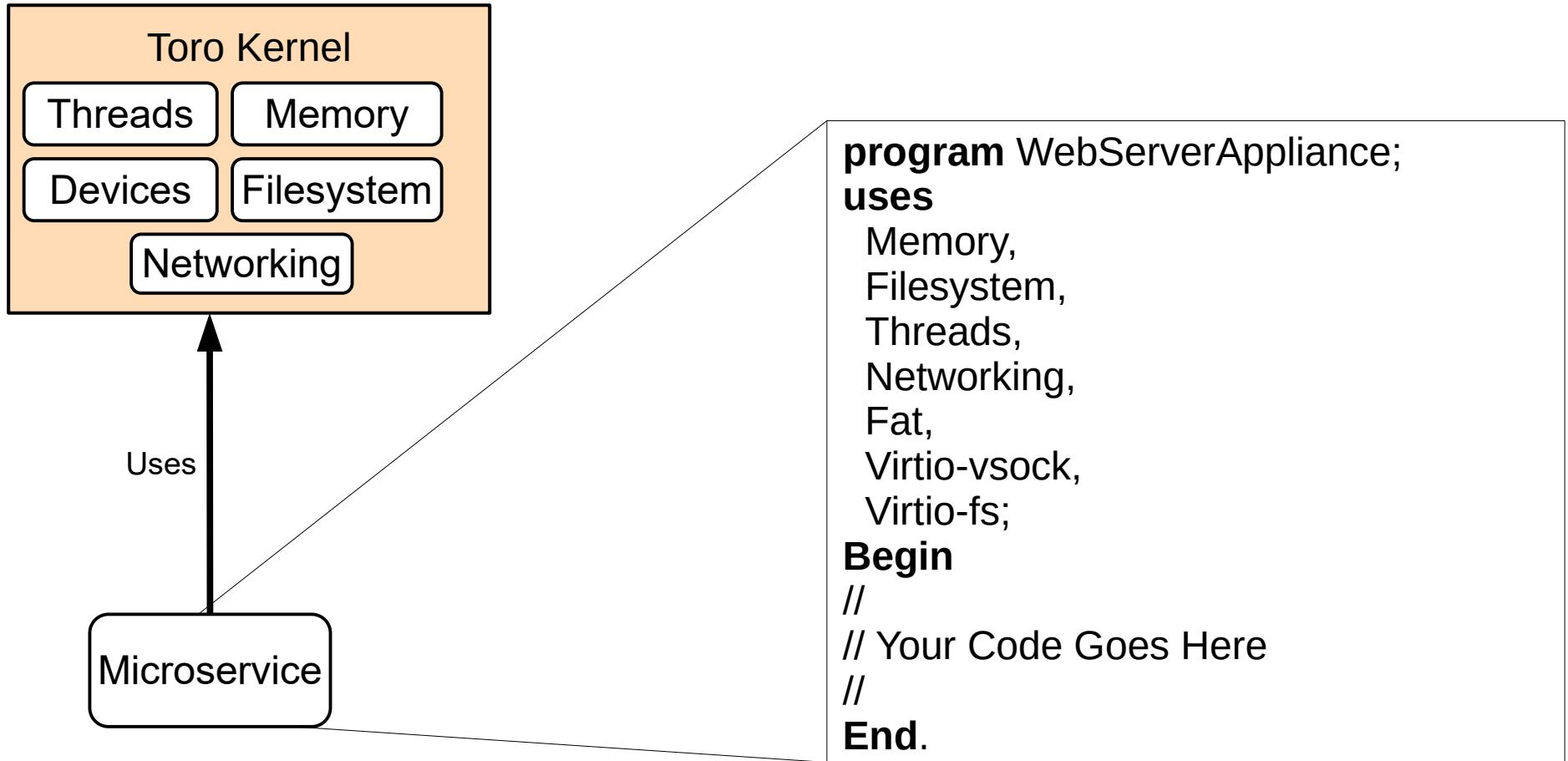
Network Stack

Network 1

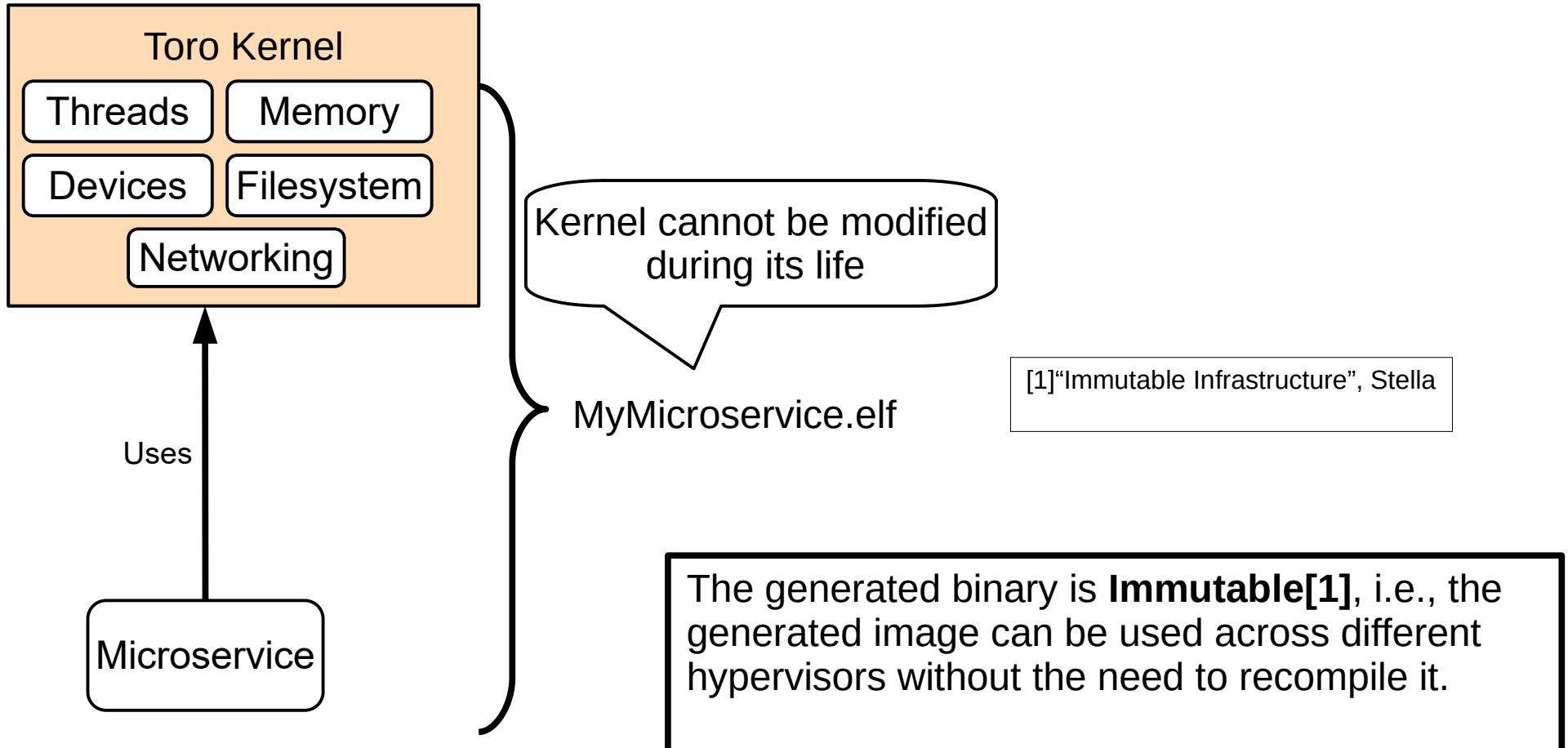
Memory 2

Core 2

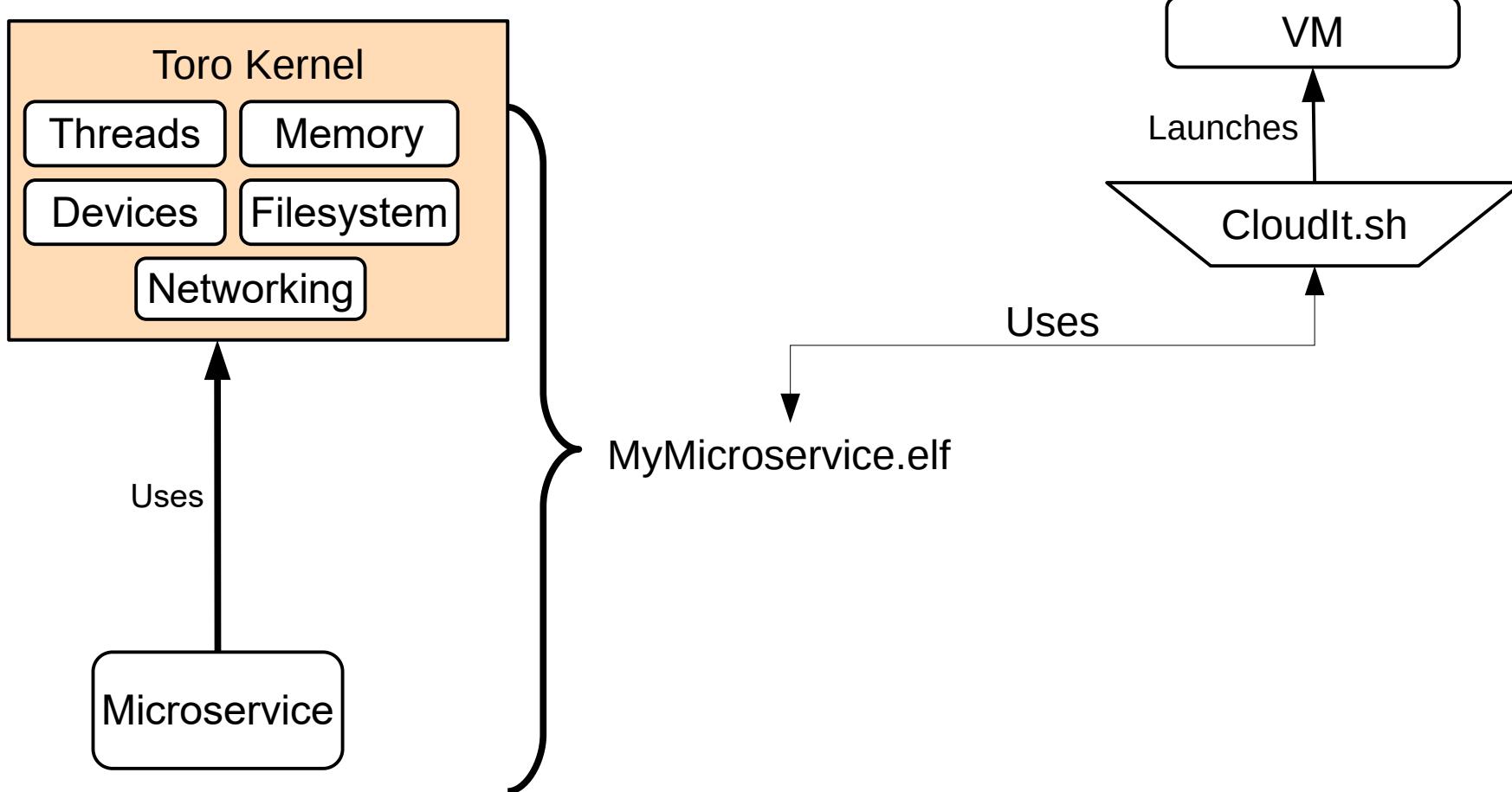
Application-oriented Kernel



Application-oriented Kernel

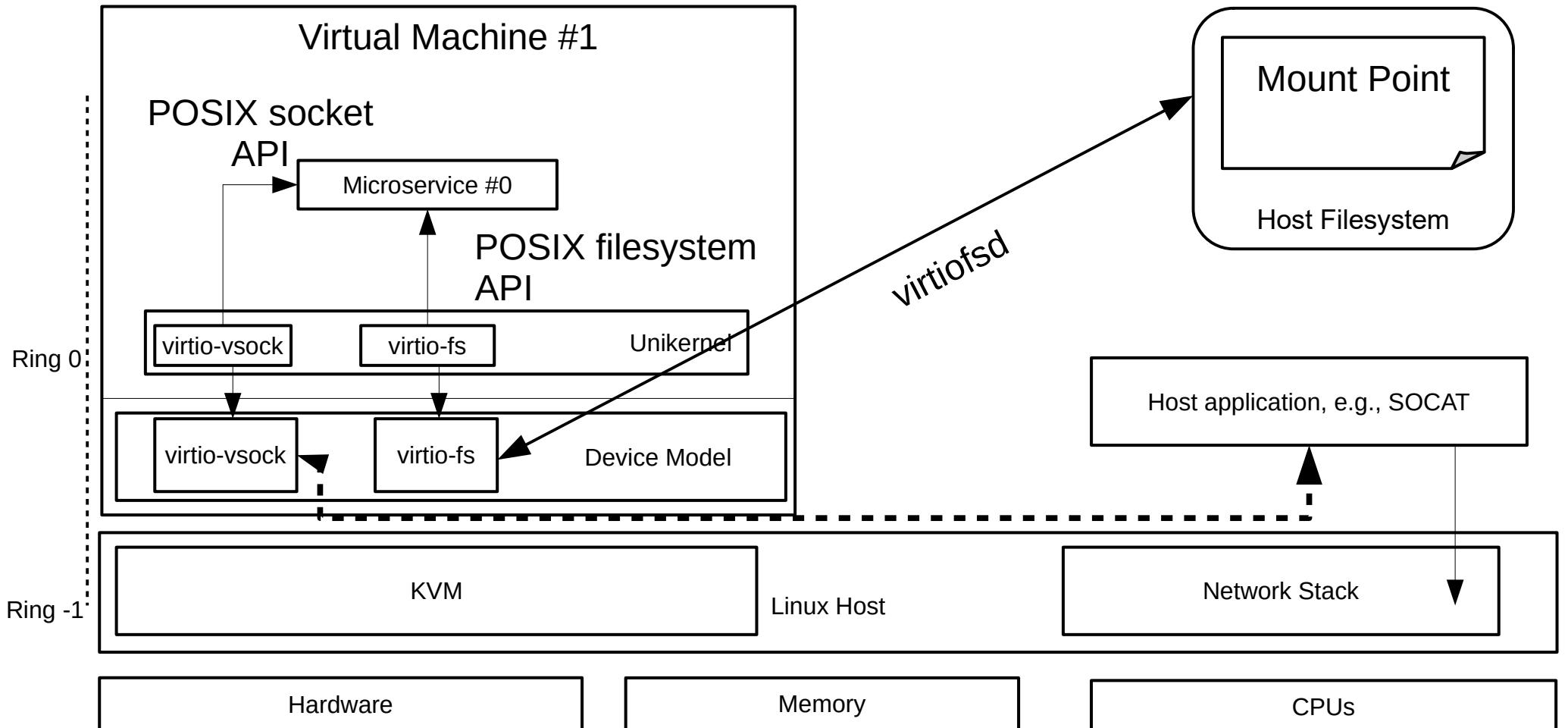


Application-oriented Kernel



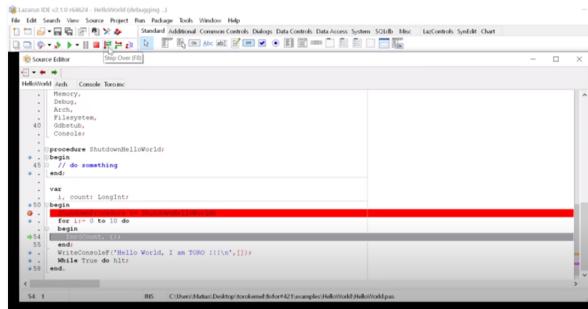
The WebServer Appliance

- Simple microservice serving files over HTTP protocol
 - Find it at <https://github.com/torokernel/torokernel> among other examples
 - This appliance is used to host Toro's website (<http://www.torokernel.io> and click on “View on Toro”)



Development Host

IDE



Compiler

GDBClient

StaticWebServer.elf



GDB Remote Protocol

Deployment Host

StaticWebServer

ToroKernel

GDBStub

virtio-fs

virtio-console

virtio-vsock

QEMU micromv

VM Context

KVM

Linux

Watch demo at <https://youtu.be/X8608cxZv1Y>

How to contribute?

- <https://github.com/torokernel/torokernel/wiki/How-to-Contribute>
- Start with issues labeled as #helpwanted
- Fork+ Pull request
- An email to torokernel@googlegroups.com with a patch attached

Books that I enjoyed

- “Modern Operating Systems”, Andrew Tanenbaum
- “Operating Systems: Internals and Design Principles”, William Stallings
- “Understanding the Linux Kernel”, Daniel P Bovet
- “The Pragmatic Programmer”, David Thomas and Andrew Hunt
- “The Mythical Man-month”, Fred Brooks
- “Software Architecture in Practice”, Len Bass, Paul Clements, and Rick Kazman
- “The Art of Doing Science and Engineering, Learning to Learn”, Richard W. Hamming
- “Introduction to Algorithm”, Cormen

Q&A

Thanks!



www.torokernel.io