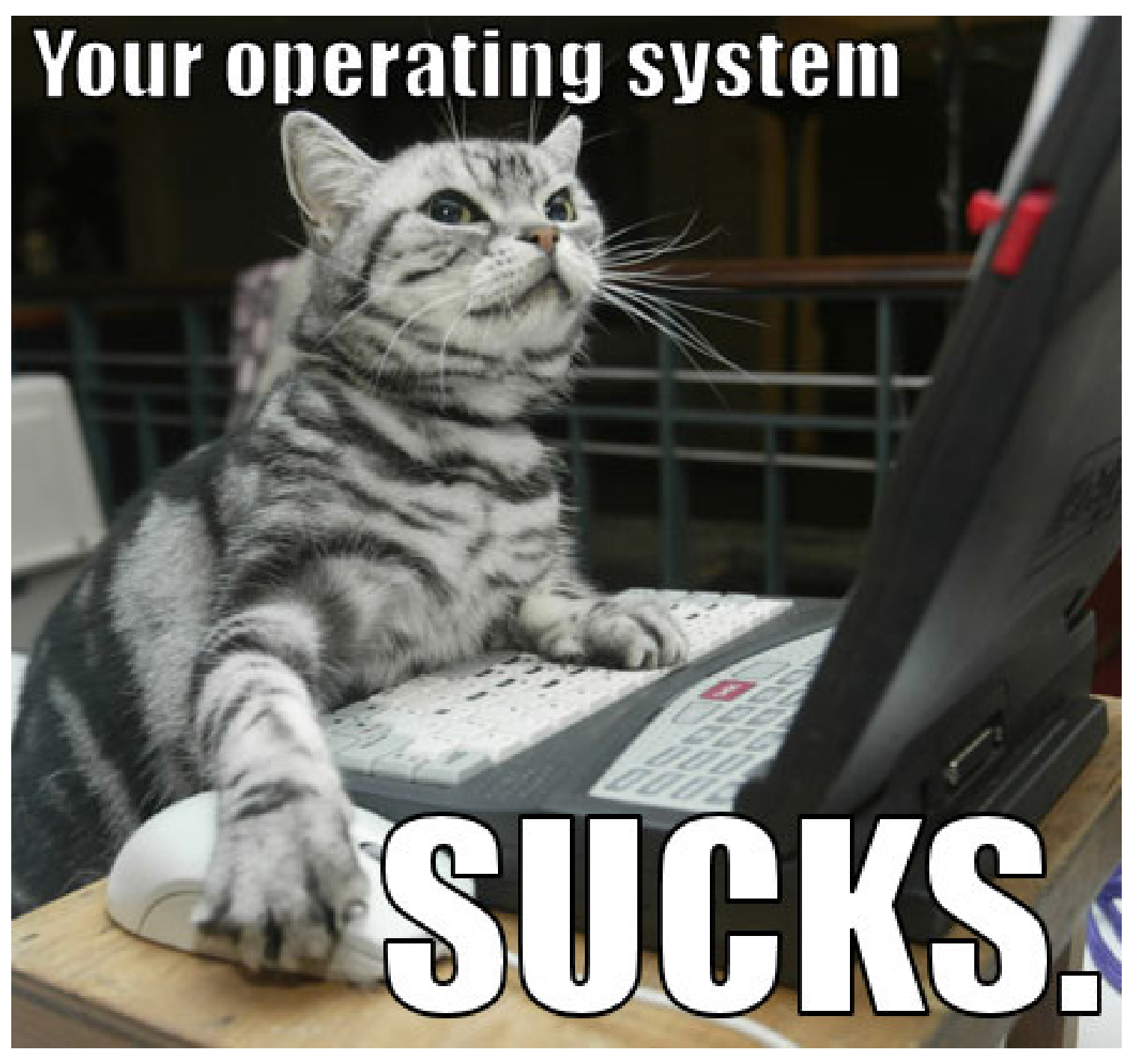
Rearchitecting System Software for the Cloud

Muli Ben-Yehuda, Dan Tsafrir
Technion—Israel Institute of Technology

What is the Problem?

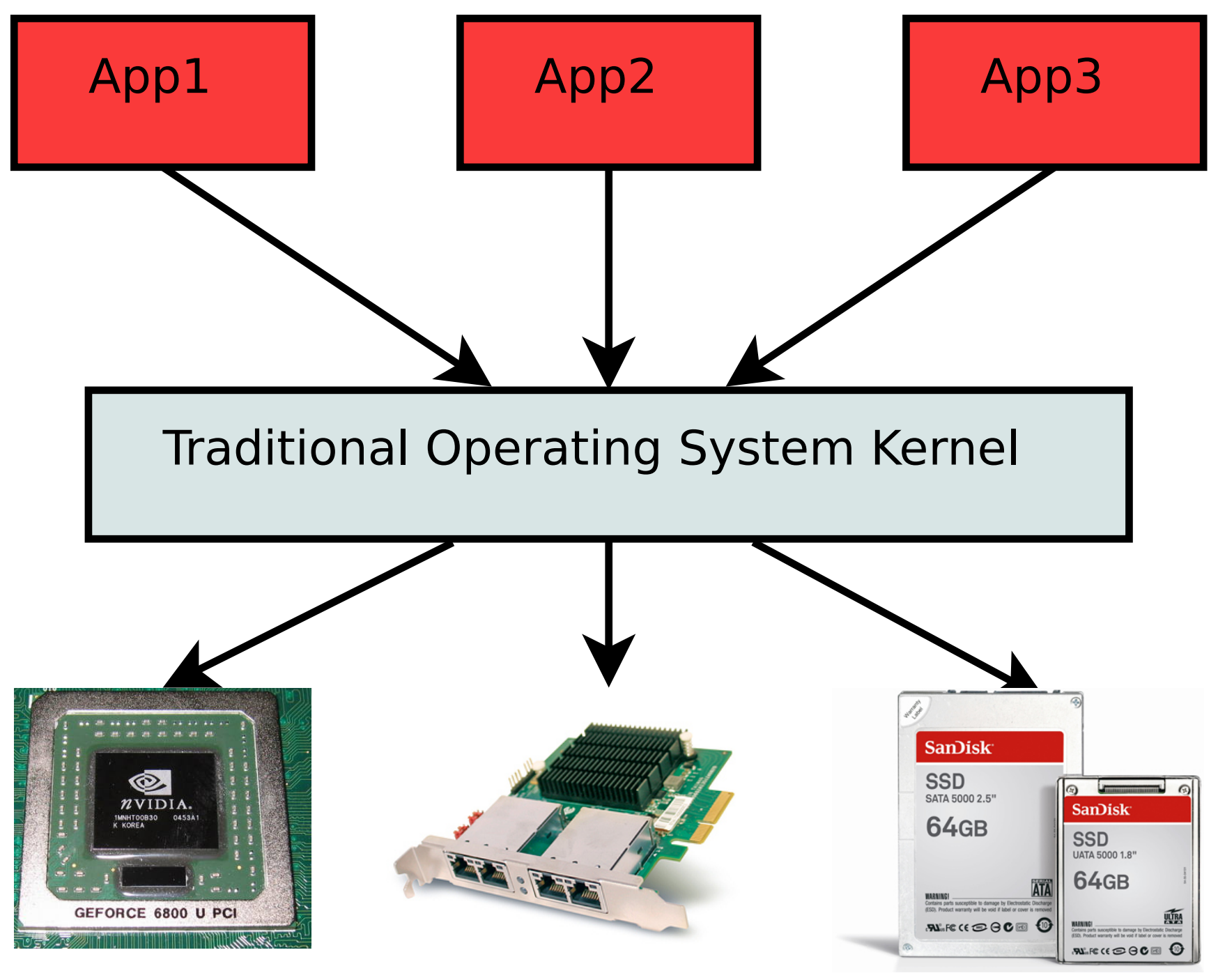
- Using traditional OS's in the cloud—see RaaS poster nearby—is expensive.

Today's Operating Systems



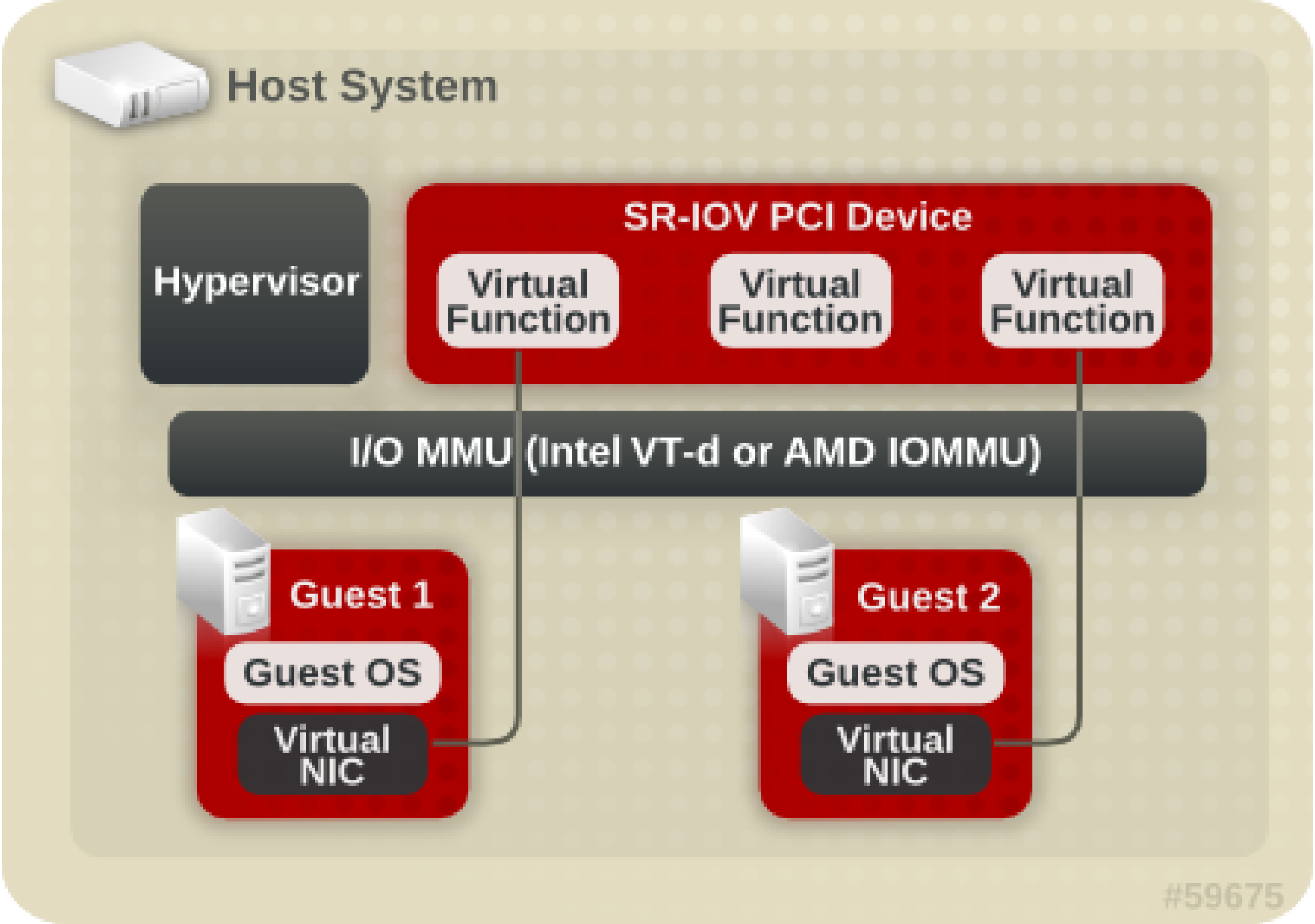
- Today's operating systems are inefficient ⇒ need better sys. software.

Traditional OS Structure



- Traditional operating systems were designed to share I/O devices.

Machine Virtualization



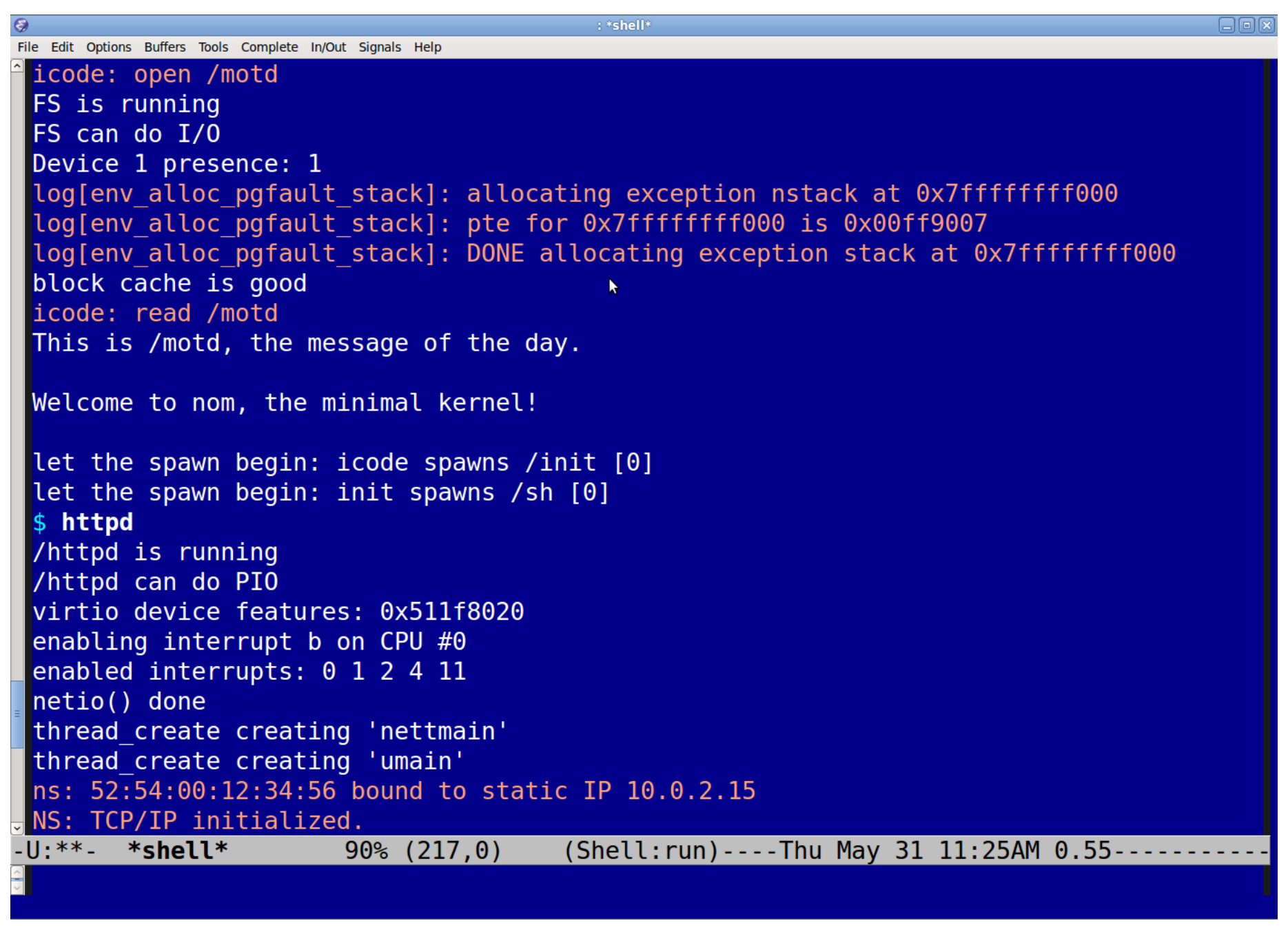
- SR-IOV devices can be shared by multiple contexts.

Benefits of nom

- All applications bypass the kernel completely on the I/O path.
- Small, simple, and secure kernel.
- Applications customize their I/O stacks to fit their needs.
- Applications adapt to changing costs of different resources quickly.

nom is Work in Progress

- Runs on x86-64 bare-metal and QEMU


: *shell*
File Edit Options Buffers Tools Complete In/Out Signals Help

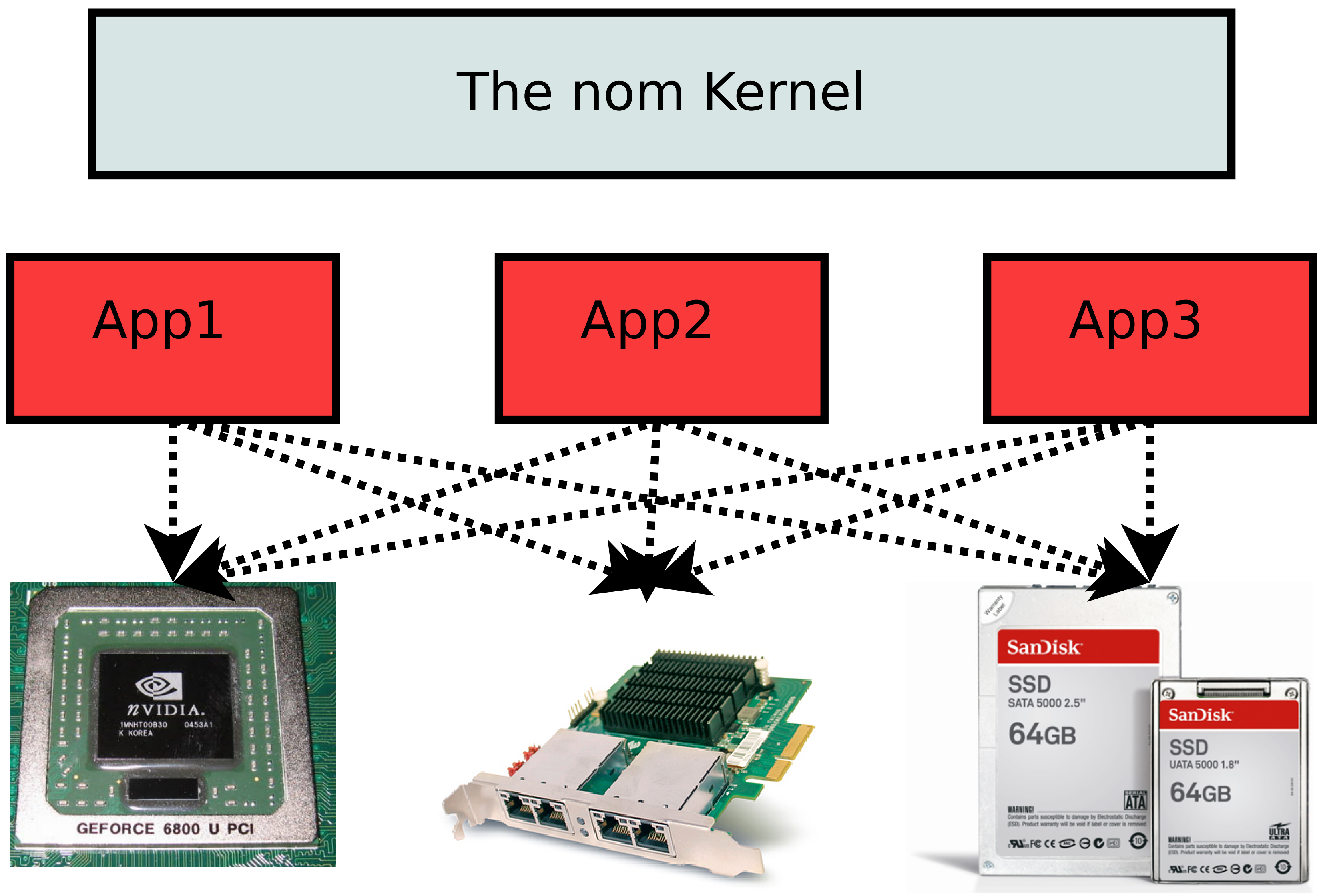
```
icode: open /motd
FS is running
FS can do I/O
Device 1 presence: 1
log[env_alloc_pgfault_stack]: allocating exception nstack at 0x7ffffffff000
log[env_alloc_pgfault_stack]: pte for 0x7ffffffff000 is 0x00ff9007
log[env_alloc_pgfault_stack]: DONE allocating exception stack at 0x7ffffffff000
block cache is good
icode: read /motd
This is /motd, the message of the day.

Welcome to nom, the minimal kernel!

let the spawn begin: icode spawns /init [0]
let the spawn begin: init spawns /sh [0]
$ httpd
/httpd is running
/httpd can do PIO
virtio device features: 0x511f8020
enabling interrupt b on CPU #0
enabled interrupts: 0 1 2 4 11
netio() done
thread_create creating 'nettmain'
thread_create creating 'umain'
ns: 52:54:00:12:34:56 bound to static IP 10.0.2.15
NS: TCP/IP initialized.
-U:**-  *shell*      90% (217,0)     (Shell:run)----Thu May 31 11:25AM 0.55----------
```

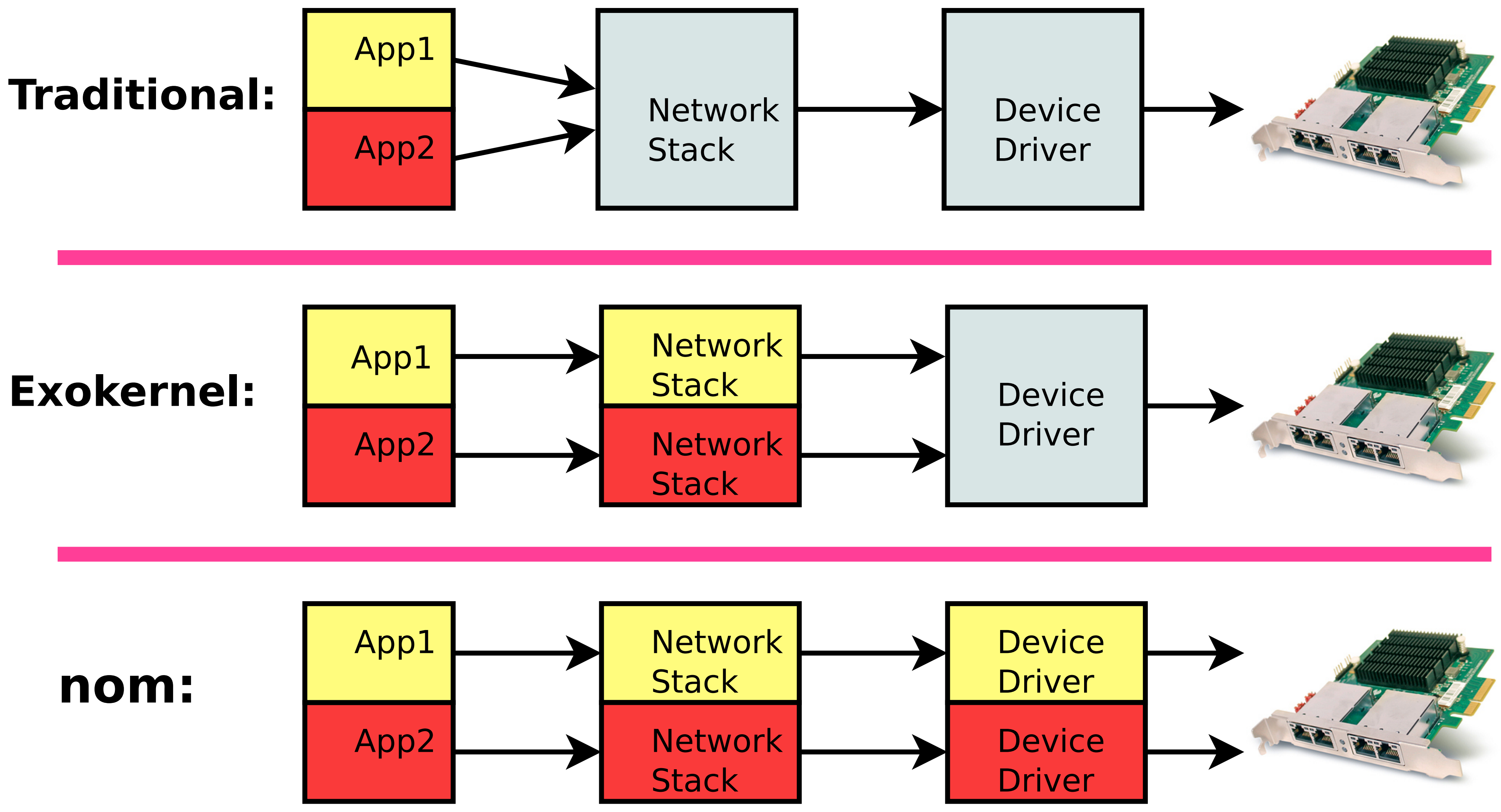
- SMP support
- Intel, Mellanox SR-IOV devices
- PIO using iopl/VMCS exception bitmap
- MMIO using page-table mapping
- DMA using IOMMUs
- Direct interrupt injection [Gordon12]

The nom Operating System



- The nom kernel provides every application with direct access to its own devices using architectural support for machine virtualization.

A Packet's Progress



Related Work

- Exokernel: [Engler95], [Kaashoek97], [Ganger02]
- Virtual machine device assignment: [LeVasseur04], [Ben-Yehuda06], [Gordon12]
- Userspace I/O, in particular VIA, Quadrics, and Infiniband.

Current Research Projects

- How should applications adapt to changing resource availability?
- What is the difference between an OS and a hypervisor?
- What is the difference between an application and a virtual machine?
- Are SR-IOV devices secure?