Rearchitecting System Software for the Cloud
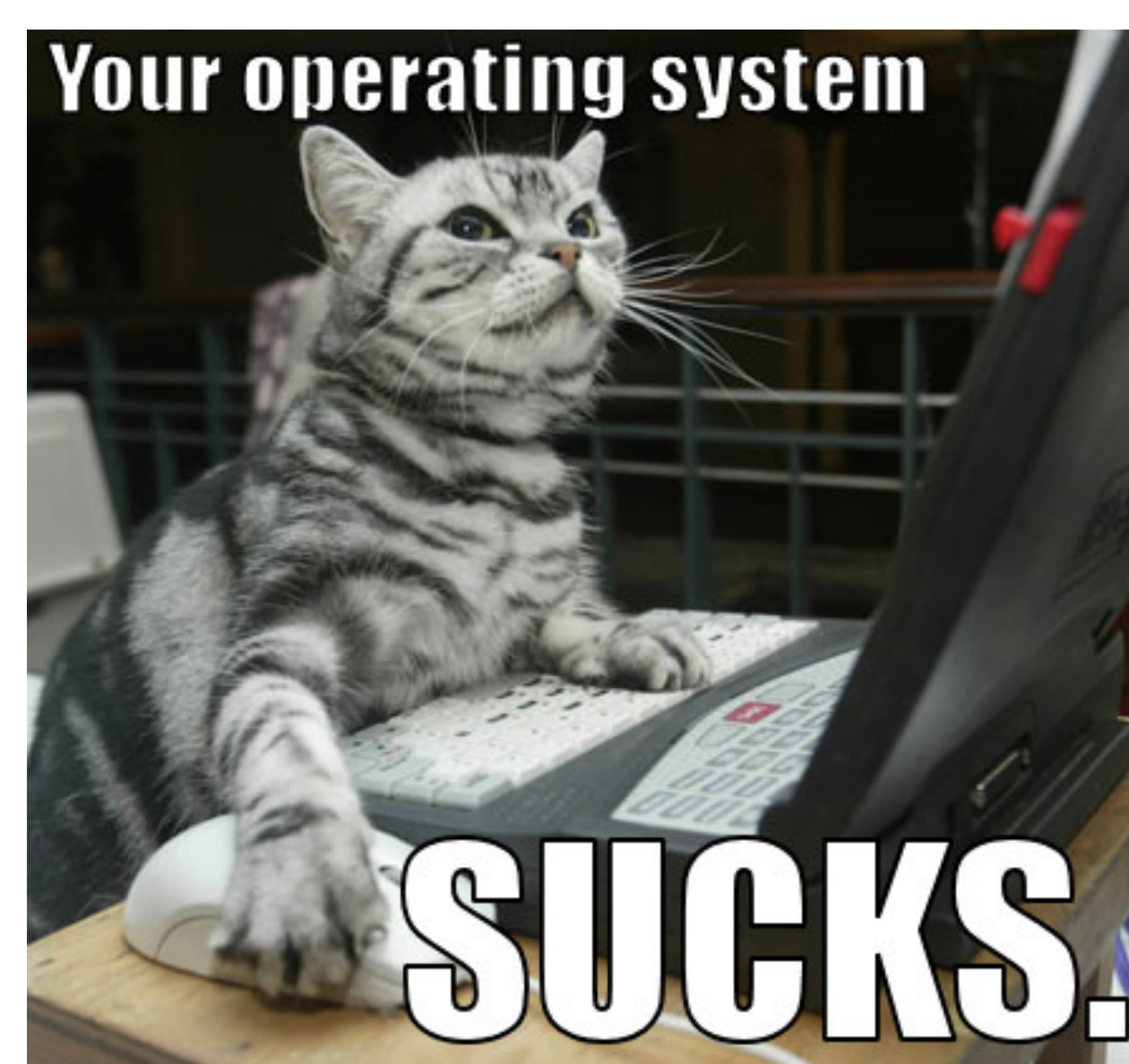
Muli Ben-Yehuda, **Dan Tsafrir**
Technion—Israel Institute of Technology

What is the Problem?

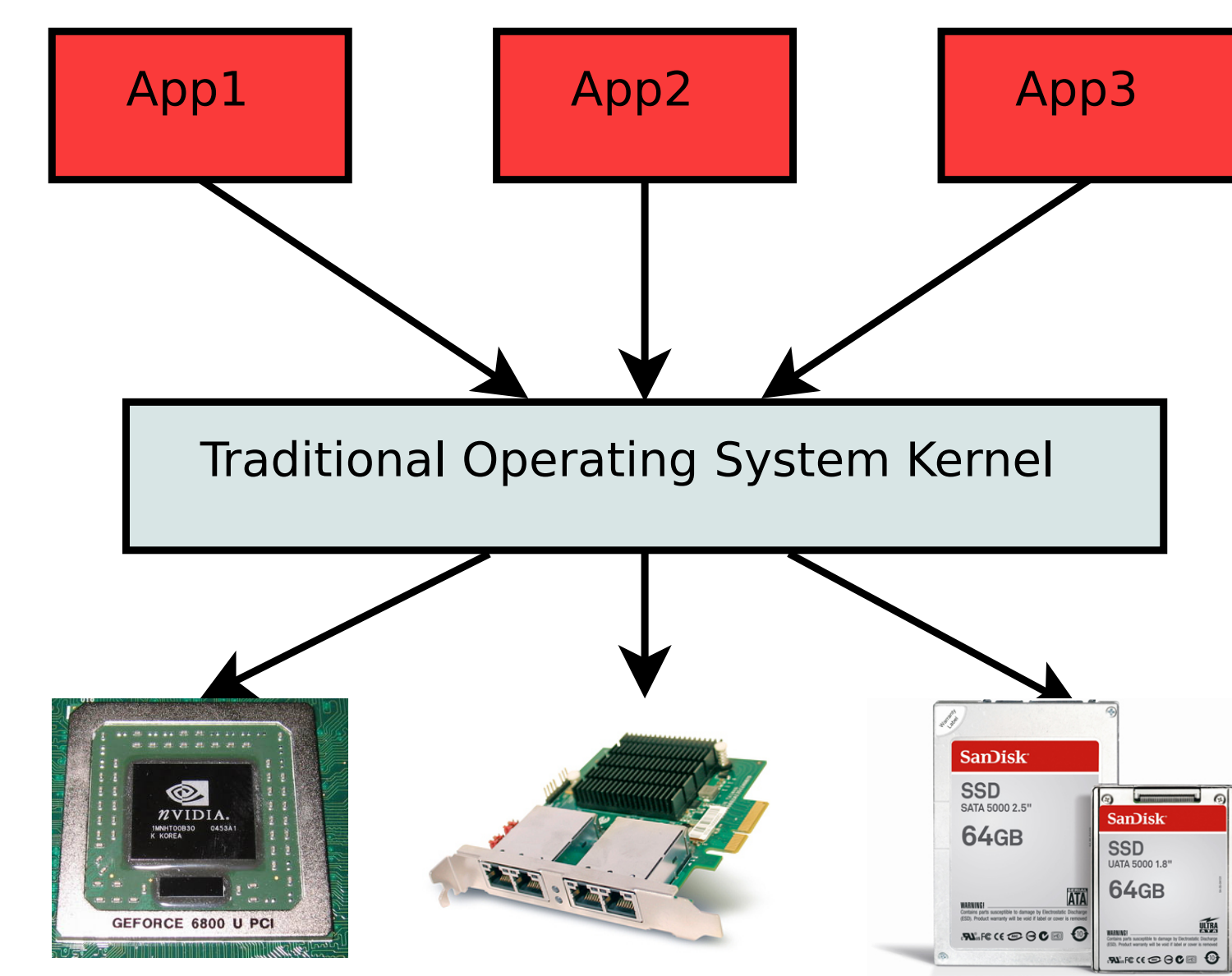- Using traditional OS's in the cloud, where users pay per use, is expensive.

- Will become more expensive as providers move to pay-per-resource (RaaS).
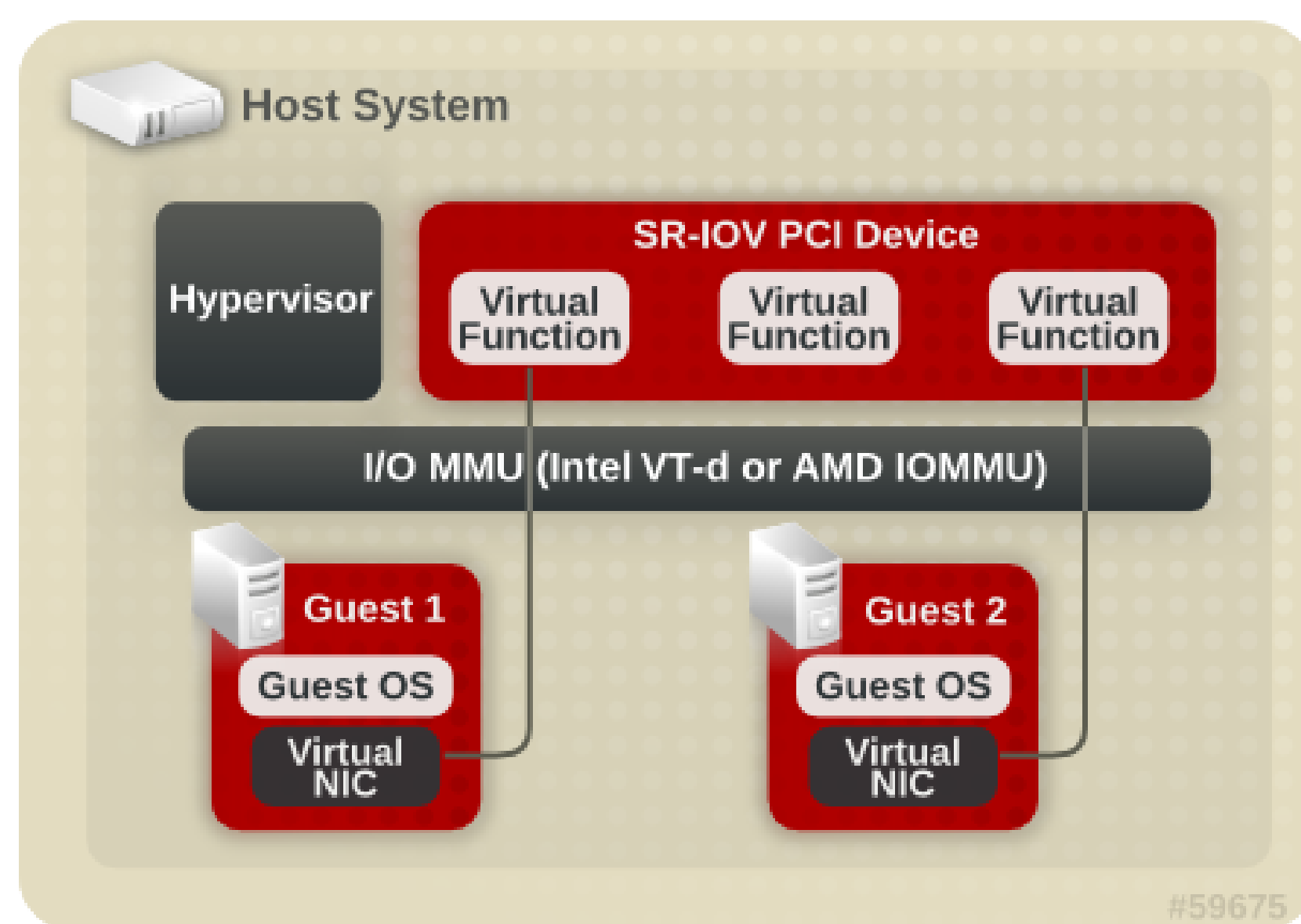
Today's OS's Are Inefficient

- Applications constrained by kernel abstractions and implementation choices.

- Single I/O stack for all apps, single device driver for all apps.

Traditional OS Structure

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

- Also designed to abstract underlying resources from applications.

Support for Virtualization
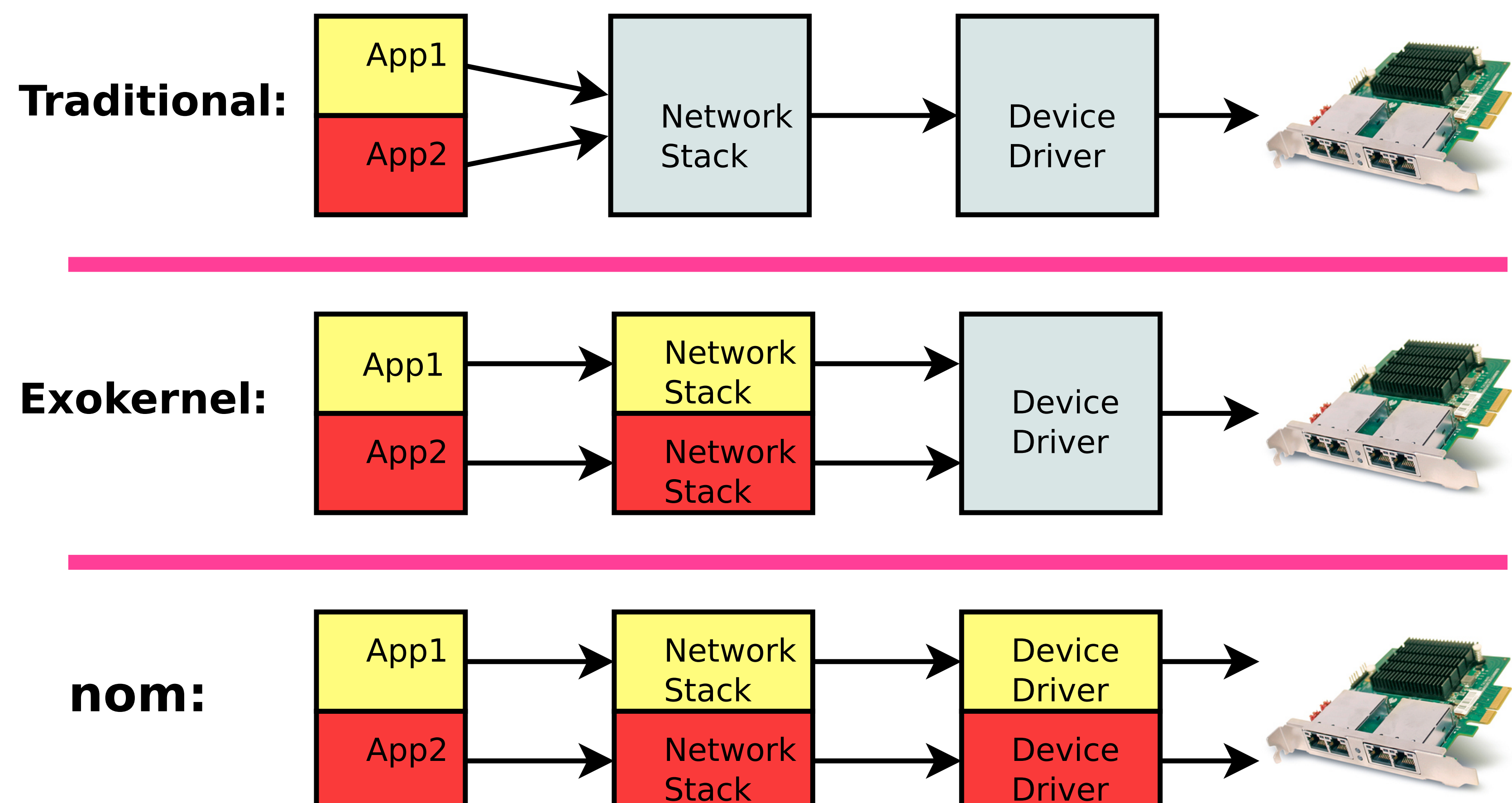
- Architectural support for machine virtualization (e.g., VT-x, VT-d, SR-IOV) provides opportunity to rethink OS structure.

- SR-IOV PCI devices can be accessed directly by multiple software entities.

A nom Packet's Progress

- The nom kernel provides every application with direct and secure access to its own I/O device (SR-IOV Virtual Functions) using architectural support for machine virtualization.

- Each nom application can use its own specialized I/O stacks and device drivers, tailored and optimized for its purpose.

- nom notifies applications as market prices of resources change; applications can adapt their behavior accordingly.

nom is Work in Progress

- Runs on SMP x86-64 bare-metal and QEMU

- Intel & Mellanox SR-IOV network devices

- Applications perform direct and secure PIO & MMIO, program devices to DMA & raise interrupts safely.

- For more information: muli@cs.technion.ac.il

Some Research Questions

- How can applications best benefit from constantly changing resource prices and availability?

- Are there fundamental differences between and OS and a hypervisor?

- Are SR-IOV devices secure?

Related Work

- Exokernels.

- Virtual machine direct device assignment ("PCI passthrough").

- Kernel-bypass I/O: VIA, Quadrics, Infiniband.

- RaaS cloud computing.