# Linux Networking on the s390 Architecture

A rich history, big boxes and reliability to the max



IBM

# Trademarks

The following are trademarks of the International Business Machines Corporation in the United States and/or other countries

- IBM® and the IBM logo
- z/Architecture®, IBM Z®
- z/VM®, z/OS®, z/TPF®, z/VSE®

A current list trademarks of IBM is available at

https://www.ibm.com/legal/copyright-trademark

The following are trademarks or registered trademarks of other companies or persons

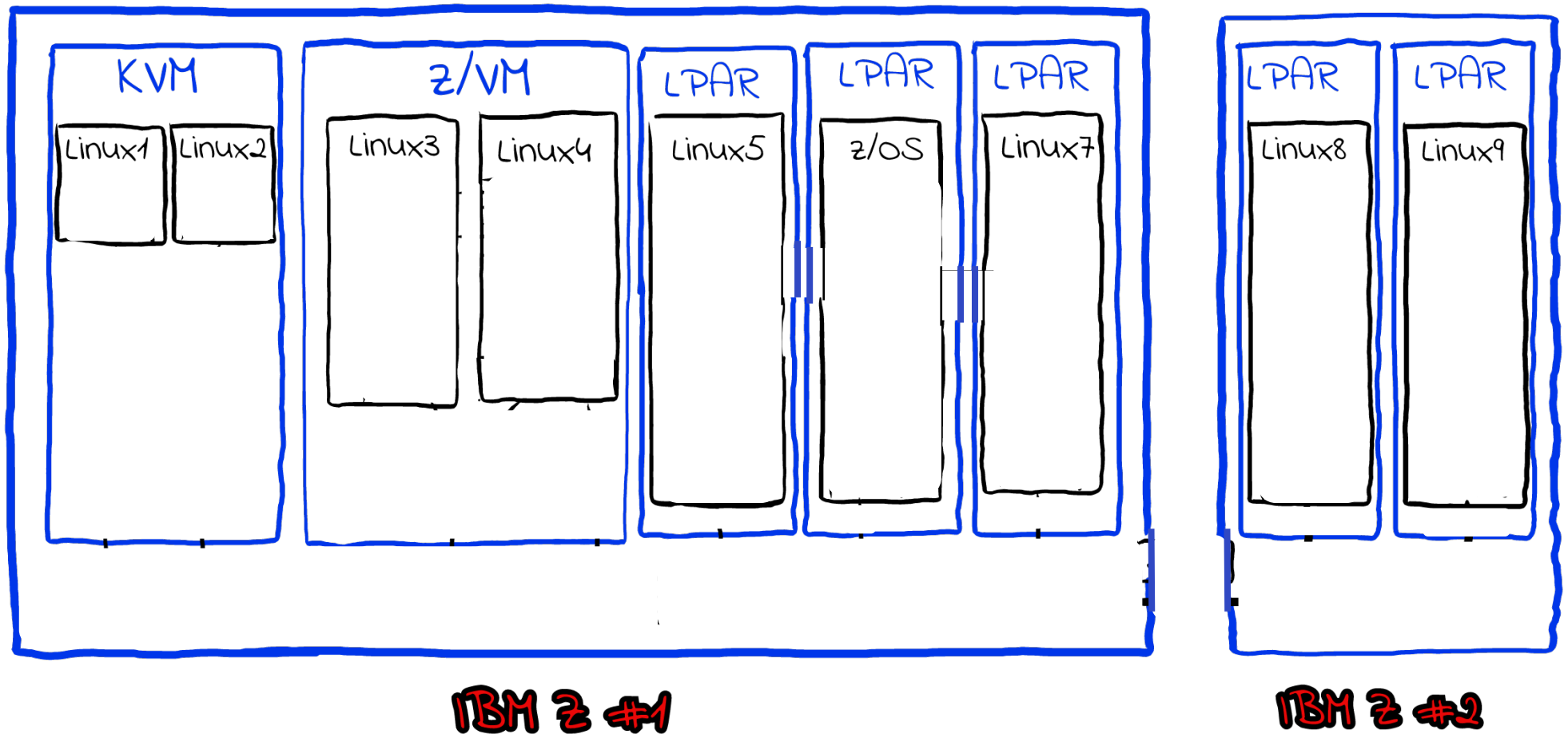
- Linux® is a registered trademark of Linus Torvalds in the United States and/or other countries
- PCIe® is a registered trademark of the PCI-SIG in the United States and/or other countries

Other company, product, or service names may be trademarks or services marks of others.

# The Hardware Platform

- Two flavors
  - IBM Z runs Linux, z/OS, z/TPF, z/VSE and z/VM
  - LinuxONE only runs Linux and Linux under z/VM
- Scale up in a single machine
  - Machine is divided into drawers for compute (CPC) and I/O
  - From 20U (1 CPC + 1 I/O) to 4 full racks (4 CPC + 12 I/O)
  - Single fully coherent SMP domain
  - From 4 to 200 usable cores + spares + offload cores
  - 4.6 or 5.2 GHz sustained clock
- RAS to the max
  - Hot plug everything (drawers, CPUs, memory…)
  - CPU snapshotting and hot sparing
  - PCIe link failover
- Operating Systems execute in Logical Partitions (LPARs)
- Direct descended of System/360 first released in 1964, z/Architecture since 2000
- Big-Endian only

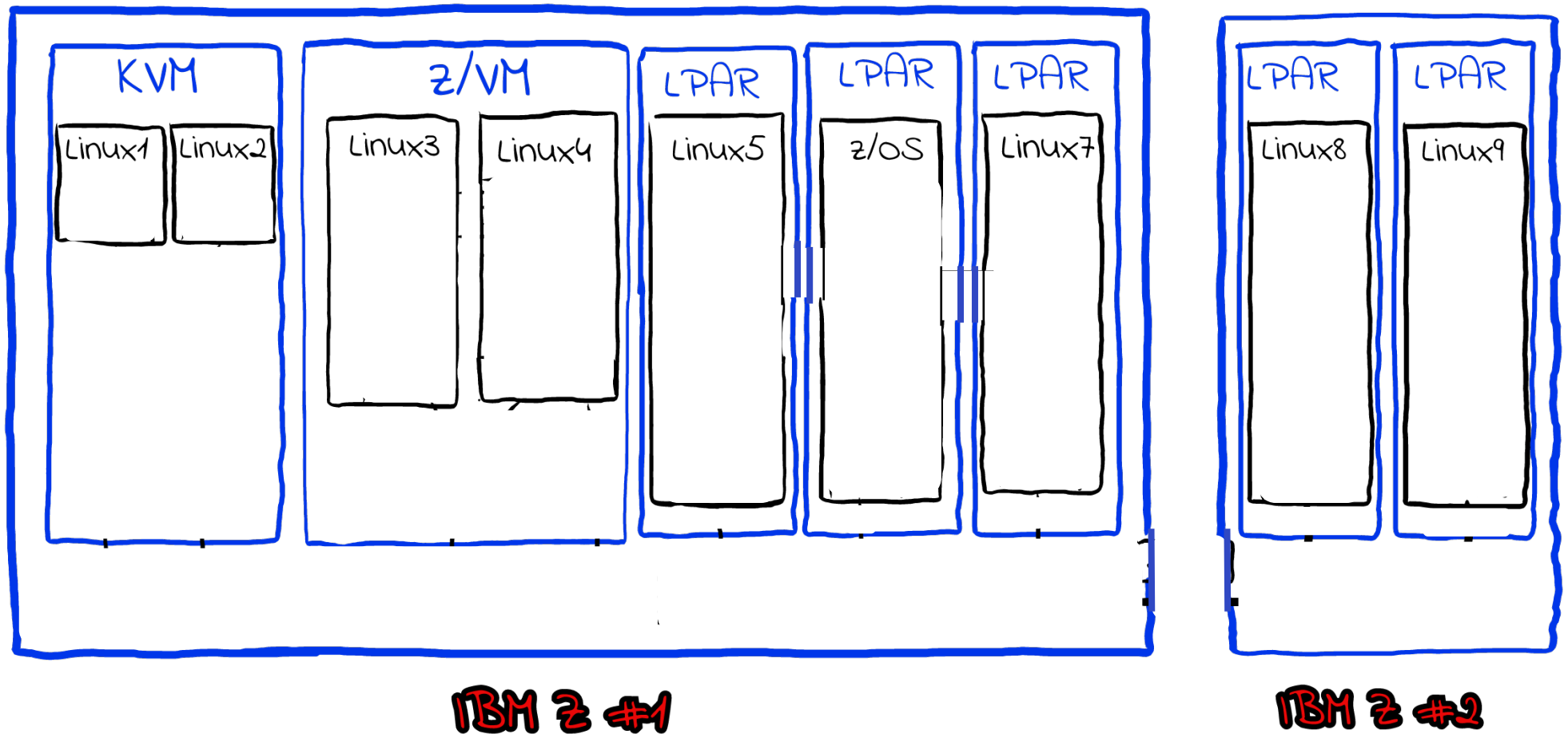
# Virtualization



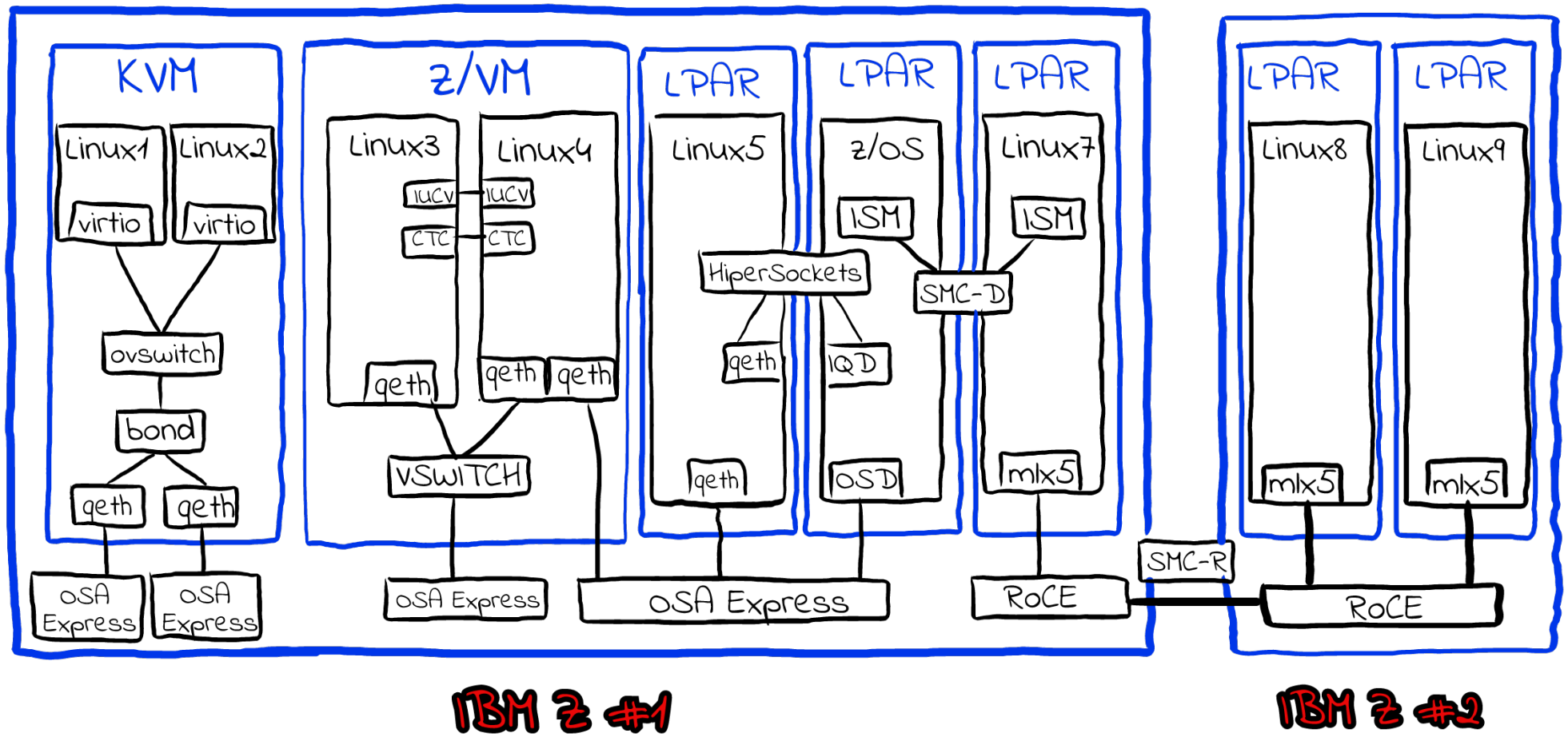
# Linux on IBM Z and LinuxONE

- Kernel Development
  - Patches for s/390 Linux available since 1999
  - Mainline first development practices
- Supported in all major enterprise distributions
- Fun facts
  - Last Big-Endian platform in enterprise distributions
  - Boot current Linux from virtual punch cards
  - Multi-level nested KVM
  - No GPUs/Framebuffers
  - 3270 as a Linux console

# Virtualization



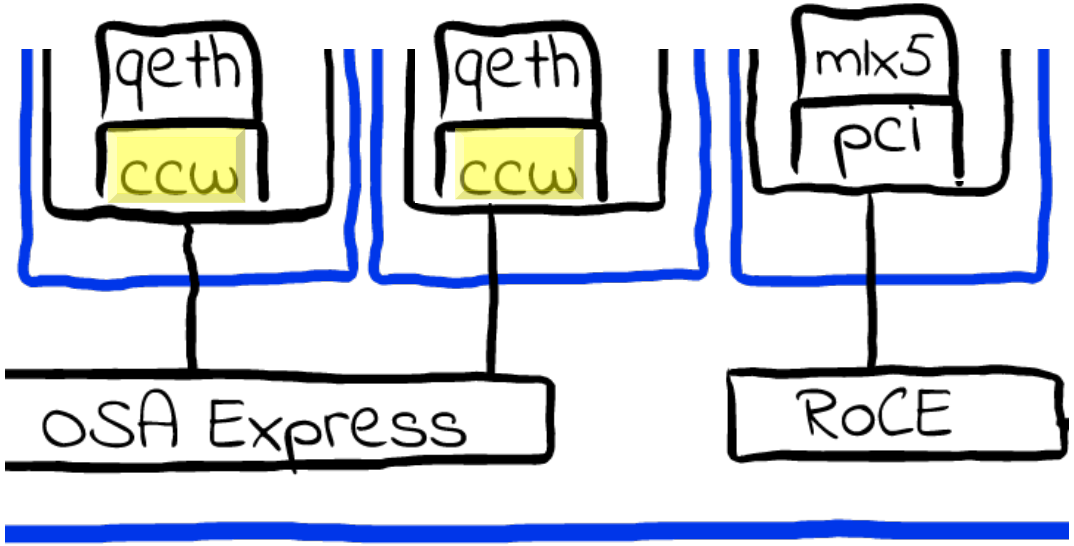
# S390 Networking in one Picture



# IBM Z Channel I/O

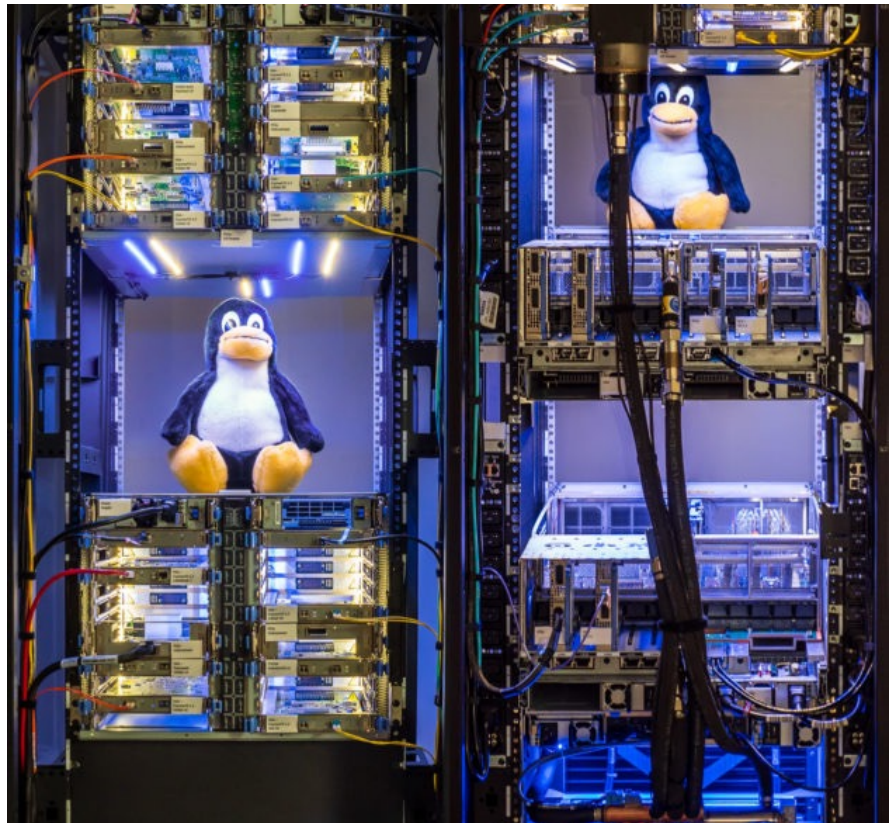
the foundation

- Channels:
  - Attach external devices including disks, tapes, networking
  - Abstracts HW details including physical transport
  - Virtualized and shareable between OS instances
  - Firmware handles discovery, monitoring, config changes, hot repair, etc.
  - Based on PCIe fabric on current systems
- In Linux channels integrate with the bus and device model
  - Bus types ccw, ccwgroup
  - Channel Command Words instead of MMIO



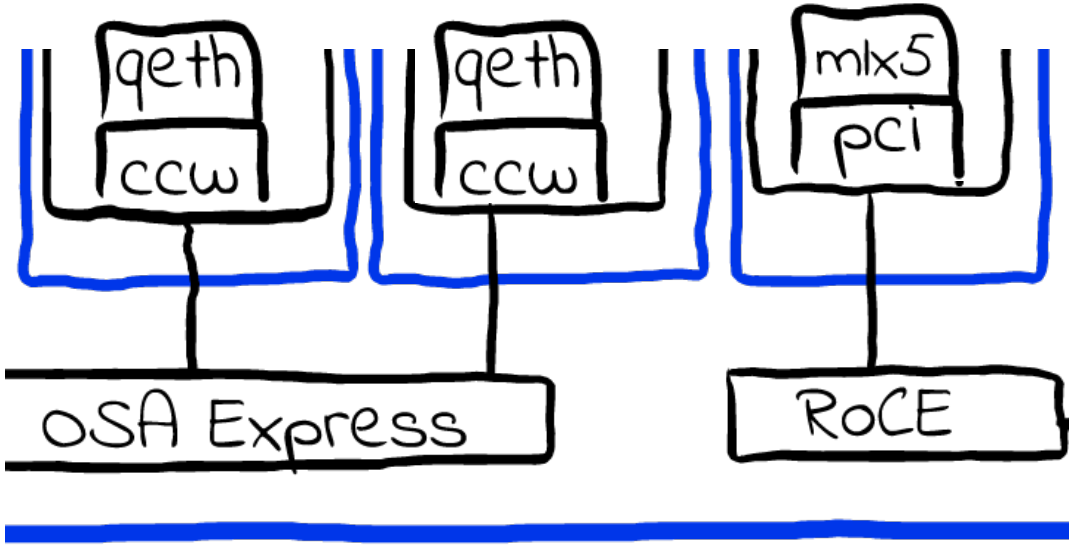
# IBM Z Channel I/O

- Channels:
  - Attach external devices including disks, tapes, networking
  - Abstracts HW details including physical transport
  - Virtualized and shareable between OS instances
  - Firmware handles discovery, monitoring, config changes, hot repair, etc.
  - Based on PCIe fabric on current systems
- In Linux channels integrate with the bus and device model
  - Bus types ccw, ccwgroup
  - Channel Command Words instead of MMIO

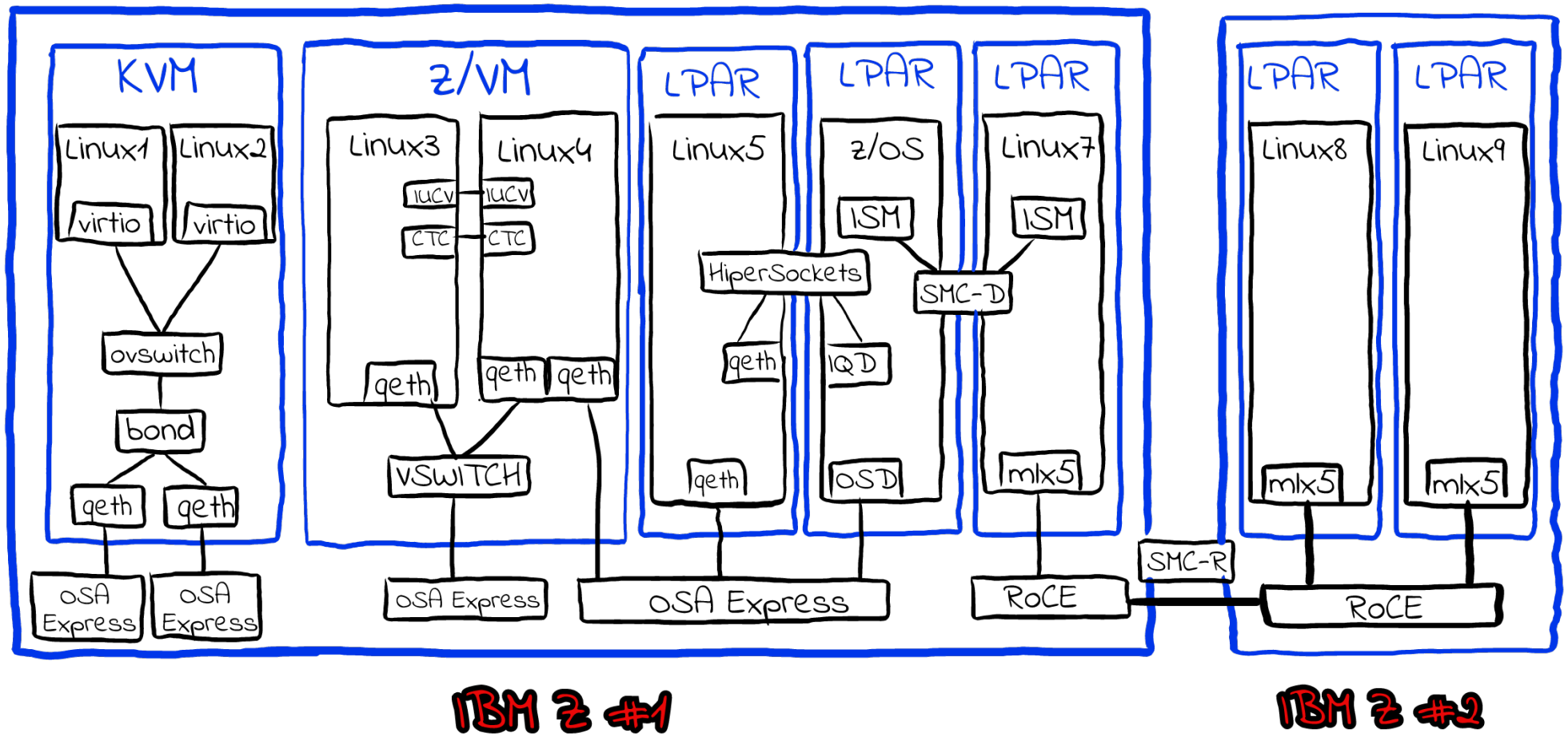


# IBM Z Channel I/O

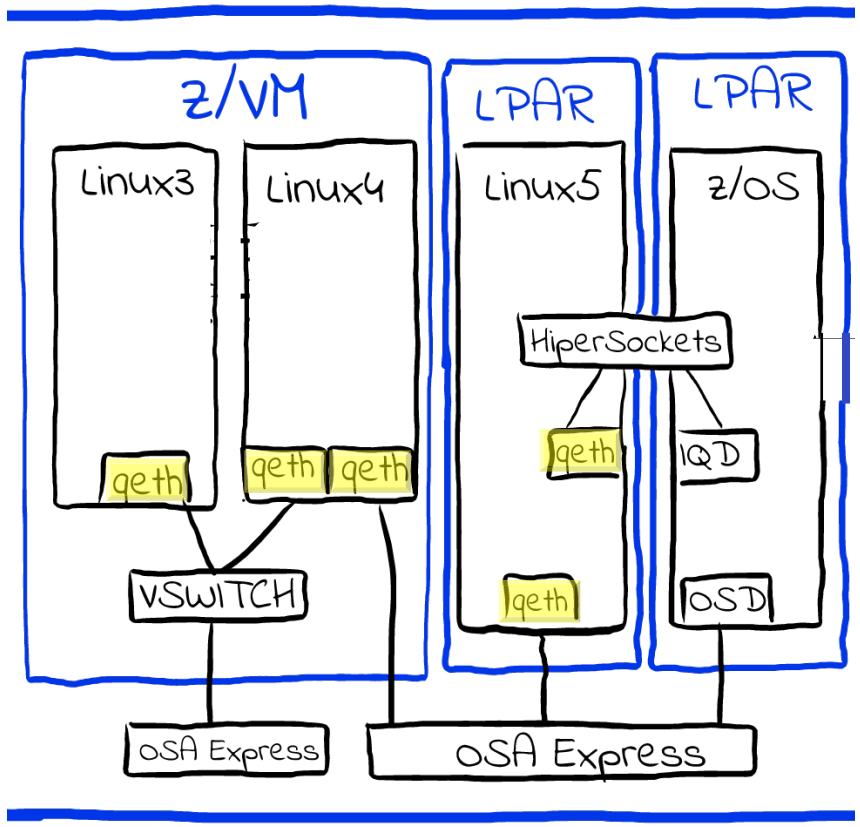
- Channels:
  - Attach external devices including disks, tapes, networking
  - Abstracts HW details including physical transport
  - Virtualized and shareable between OS instances
  - Firmware handles discovery, monitoring, config changes, hot repair, etc.
  - Based on PCIe fabric on current systems
- In Linux channels integrate with the bus and device model
  - Bus types ccw, ccwgroup
  - Channel Command Words instead of MMIO



# S390 Networking in one Picture



# S390 Networking in one Picture

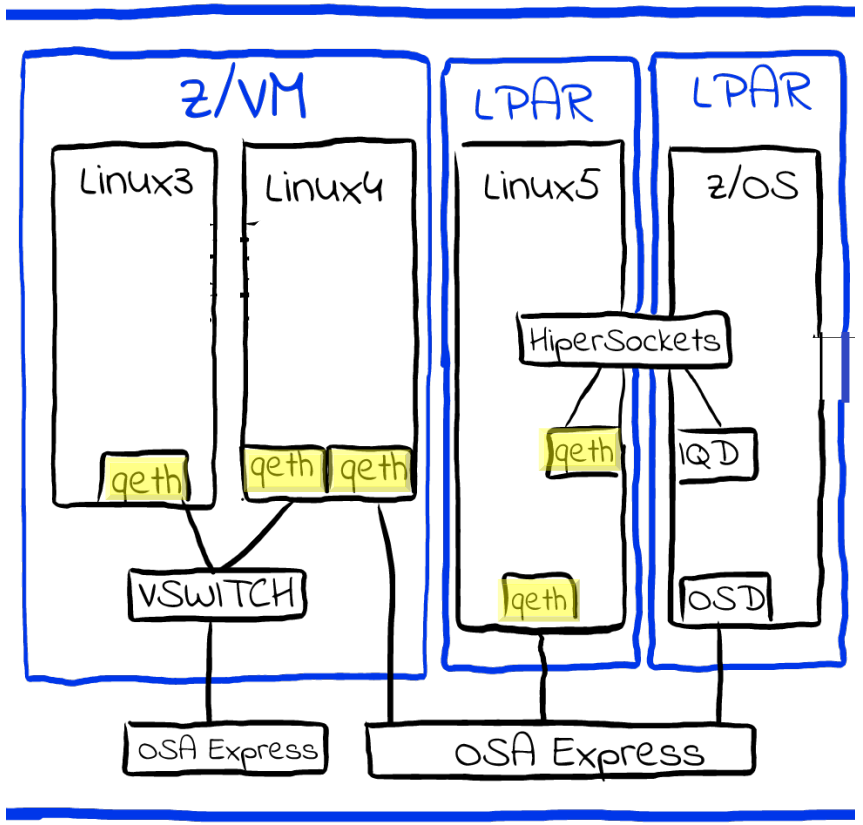



IBM z #1

# *qeth* QDIO ethernet

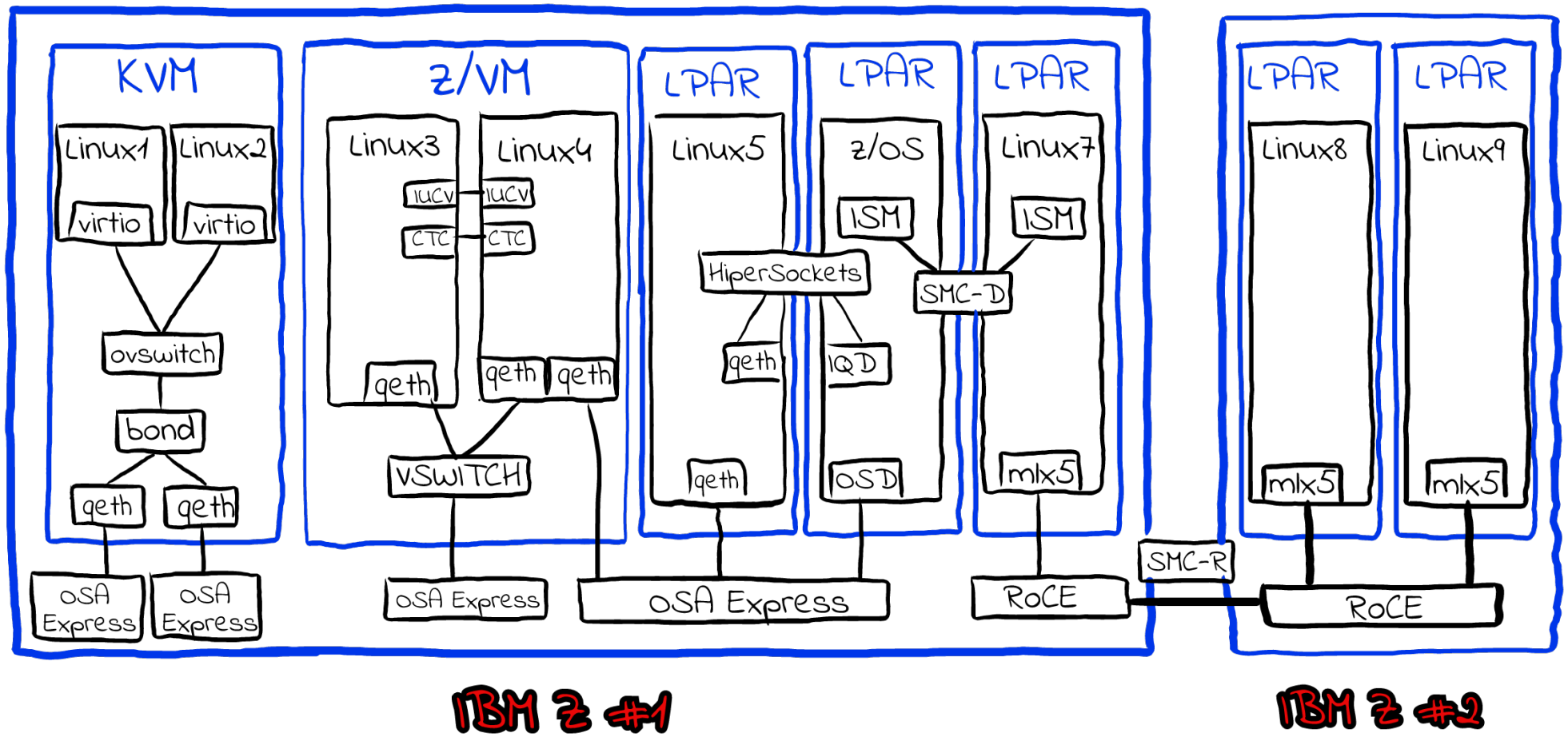
bread and butter

- Queued direct IO (QDIO) extends the channel architecture
  - Data plane of ring buffers pointing to 128x16x4k data
  - HW support for 1st and 2nd level guest address translation
- Network devices:
  - OSA-Express cards
  - HiperSockets (FW emulated networks between LPARs)
  - z/VM VSWITCH (and guest LAN)
- Layer 3: IP devices, ARP offloading, special features
- Layer 2: Ethernet devices, better fit for Linux network stack
- Not all interfaces provide all features



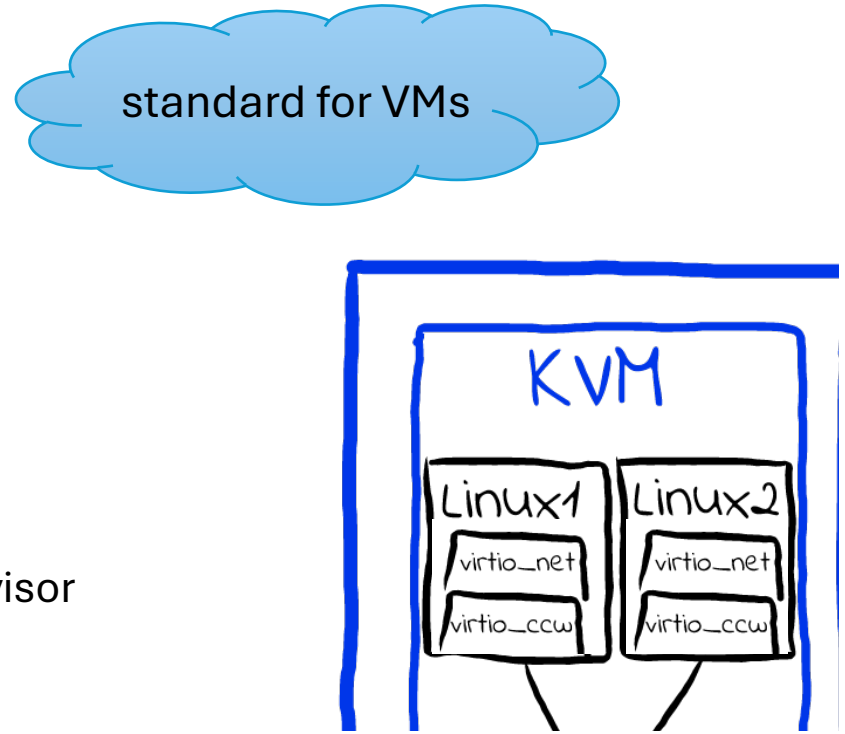

IBM Z #1

# S390 Networking in one Picture



# Other s390 network drivers

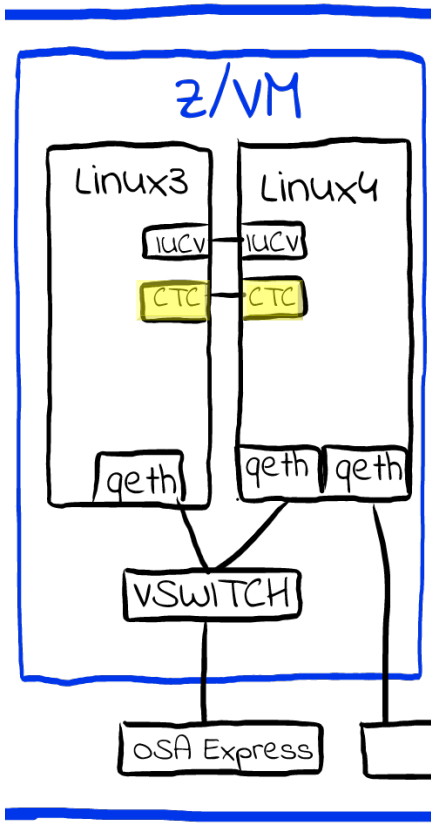
- *virtio-net*
  - Same as everywhere except using virtio-ccw
- *ctc* Channel-to-Channel
  - Directly connected channels
  - P2P between 2 z/VM guests
- *iucv* Inter-User-Communication-Vehicle
  - Mechanism for data transfer:
    - Between z/VM guests or z/VM guest and z/VM Hipervisor
    - No ccw bus, no device, direct calls to hypervisor (net/iucv)
  - Addressing by guest ID (user in z/VM parlance)
  - Also used for consoles (drivers/tty/hvc/hvc_iucv.c)
- *lcs* LAN channel station
  - Old message-based Firmware interface
  - Predecessor of OSA (qeth); no queues



# Other s390 network drivers

- *virtio-net*
  - Same as everywhere except using virtio-ccw
- *ctc* Channel-to-Channel
  - Directly connected channels
  - P2P between 2 z/VM guests
- *iucv* Inter-User-Communication-Vehicle
  - Mechanism for data transfer:
    - Between z/VM guests or z/VM guest and z/VM Hipervisor
    - No ccw bus, no device, direct calls to hypervisor (net/iucv)
  - Addressing by guest ID (user in z/VM parlance)
  - Also used for consoles (drivers/tty/hvc/hvc_iucv.c)
- *lcs* LAN channel station
  - Old message-based Firmware interface
  - Predecessor of OSA (qeth); no queues

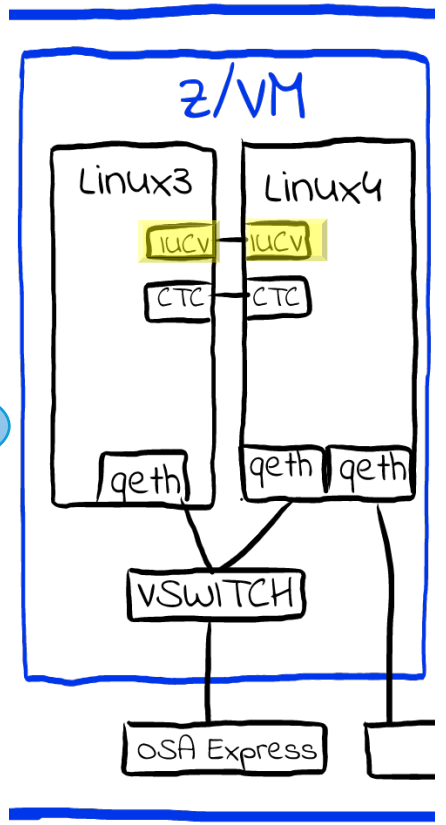
old and simple



# Other s390 network drivers

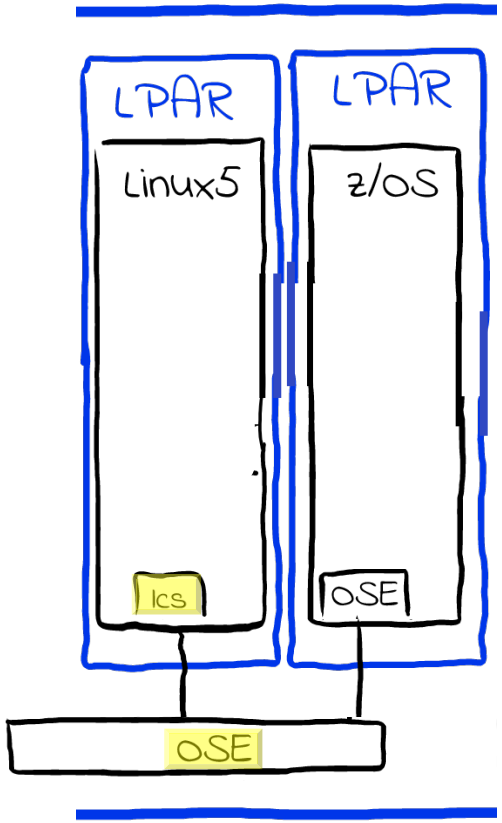
- *virtio-net*
  - Same as everywhere except using virtio-ccw
- *ctc* Channel-to-Channel
  - Directly connected channels
  - P2P between 2 z/VM guests
- *iucv* Inter-User-Communication-Vehicle
  - Mechanism for data transfer:
    - Between z/VM guests or z/VM guest and z/VM Hipervisor
    - No ccw bus, no device, direct calls to hypervisor (net/iucv)
  - Addressing by guest ID (user in z/VM parlance)
  - Also used for consoles (drivers/tty/hvc/hvc_iucv.c)
- *lcs* LAN channel station
  - Old message-based Firmware interface
  - Predecessor of OSA (qeth); no queues

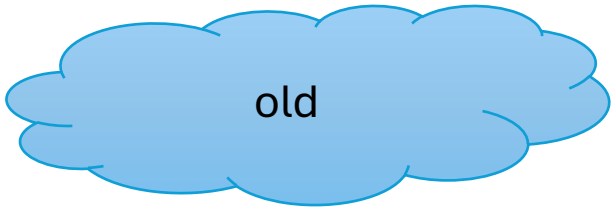
virtual and simple



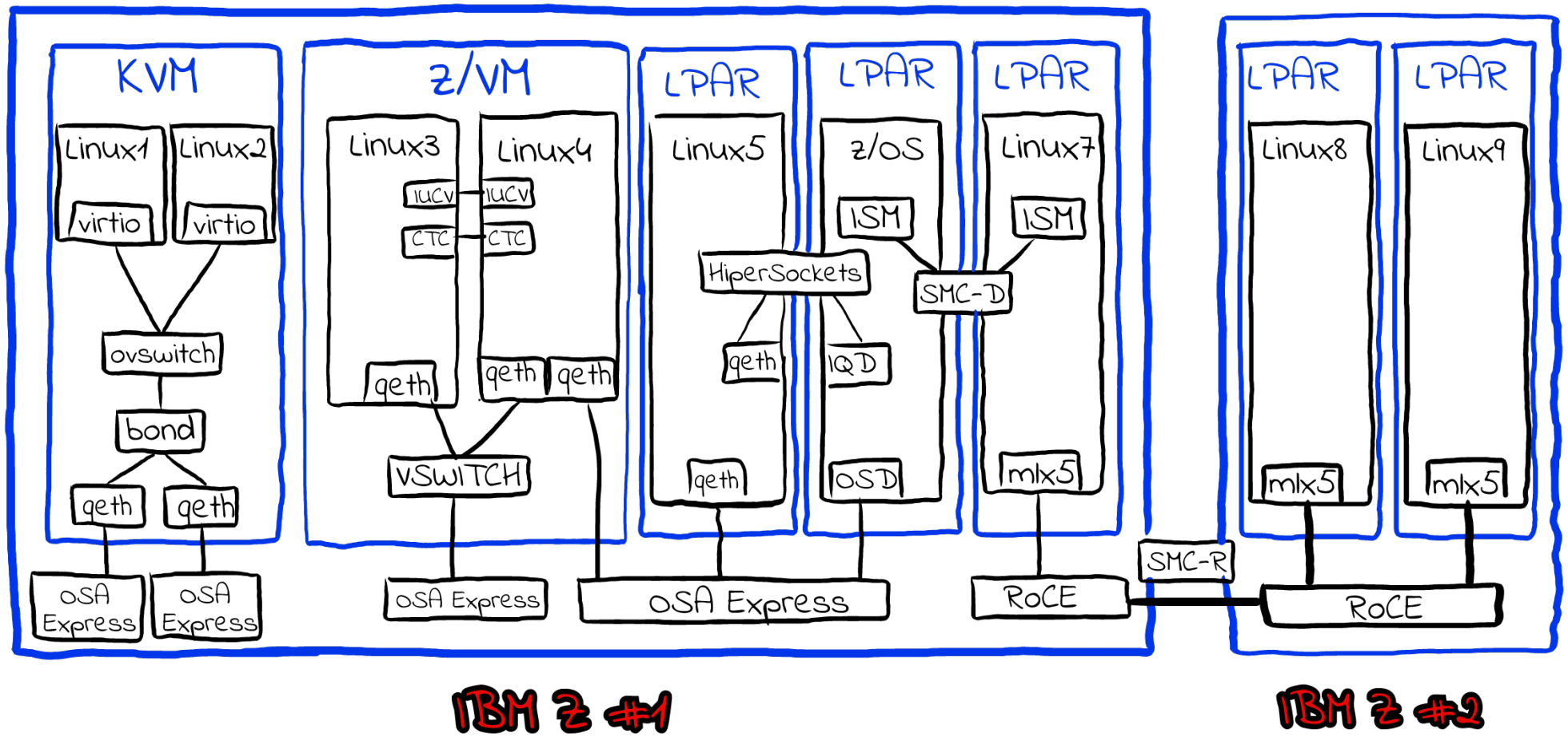
# Other s390 network drivers

- *virtio-net*
  - Same as everywhere except using virtio-ccw
- *ctc* Channel-to-Channel
  - Directly connected channels
  - P2P between 2 z/VM guests
- *iucv* Inter-User-Communication-Vehicle
  - Mechanism for data transfer:
    - Between z/VM guests or z/VM guest and z/VM Hipervisor
    - No ccw bus, no device, direct calls to hypervisor (net/iucv)
  - Addressing by guest ID (user in z/VM parlance)
  - Also used for consoles (drivers/tty/hvc/hvc_iucv.c)
- *lcs* LAN channel station
  - Old message-based Firmware interface
  - Predecessor of OSA (qeth); no queues



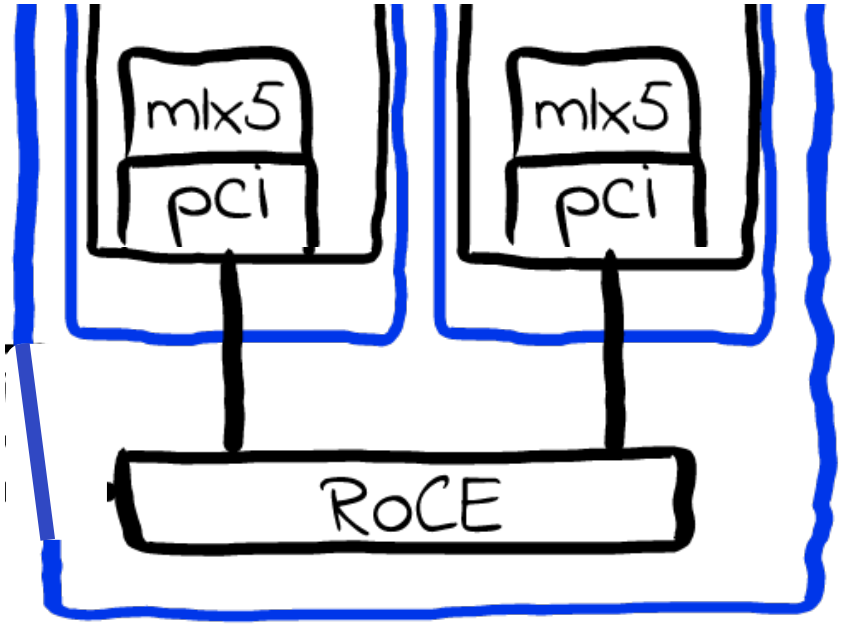



# S390 Networking in one Picture

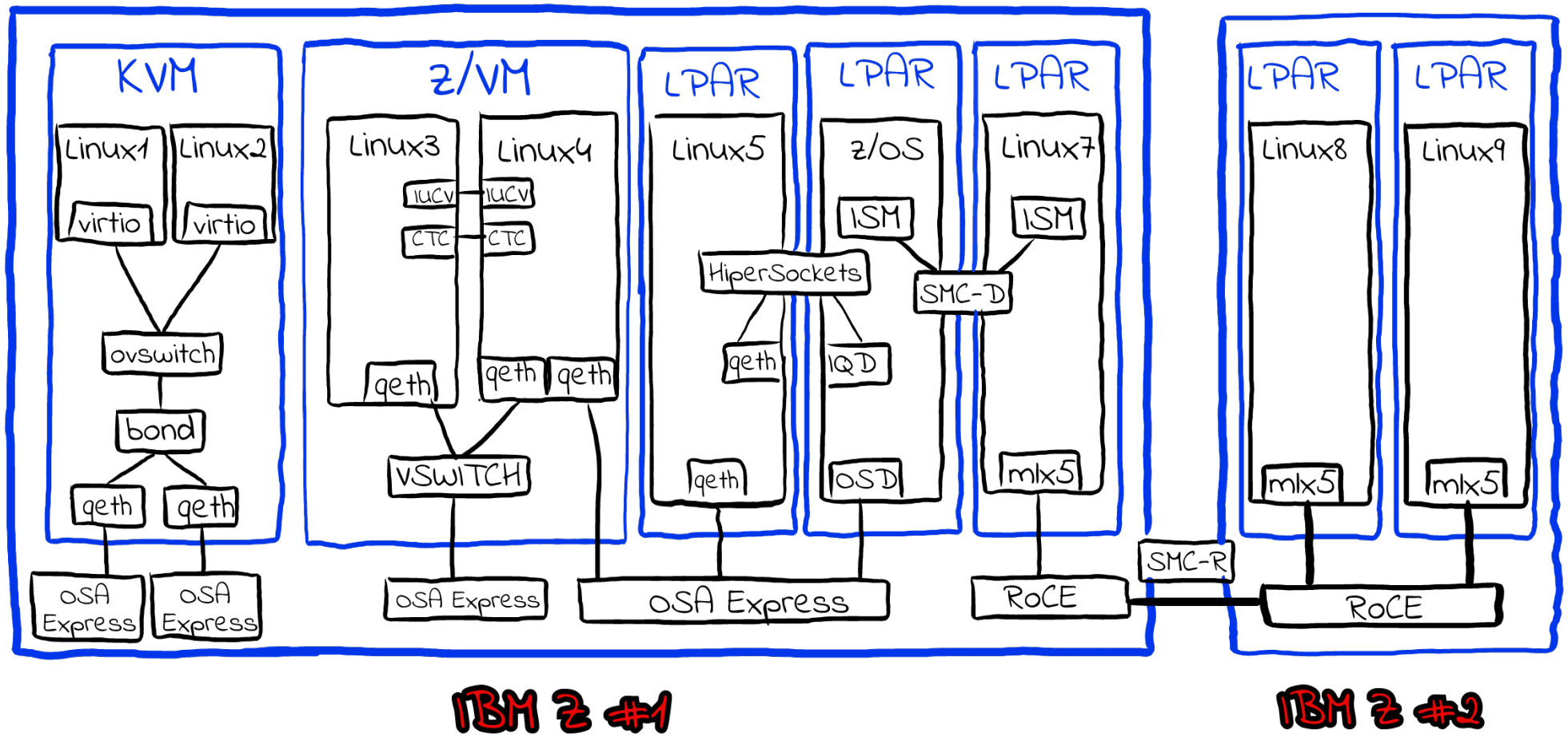


# Native PCIe Devices


- Use common drivers including mlx5 and nvme and PCI subsystem
- S390 has no MMIO but PCI instructions
  - *pcistg reg,handle,bar,len*
    - Classic variant with an opaque 32 bit function handle not an address
    - Linux "makes up" addresses + indirection to allow use in readq()/writeq()
  - *pcistg**i** reg,ioaddr,len*
    - MIO variant with a virtual address translating to a so-called MIO address
    - Designed to fit Linux APIs including ioremap() + readq()/writeq()
    - Supports re-ordering via ioremap_wc()
- Virtualization impacts
  - PCI probing done by firmware OSs get a list of PCI functions
  - VF BARs mapped by firmware
  - IOMMU emulation + shadowing for second level VMs
  - Firmware interaction for AER, Automatic Link failover, hotplug



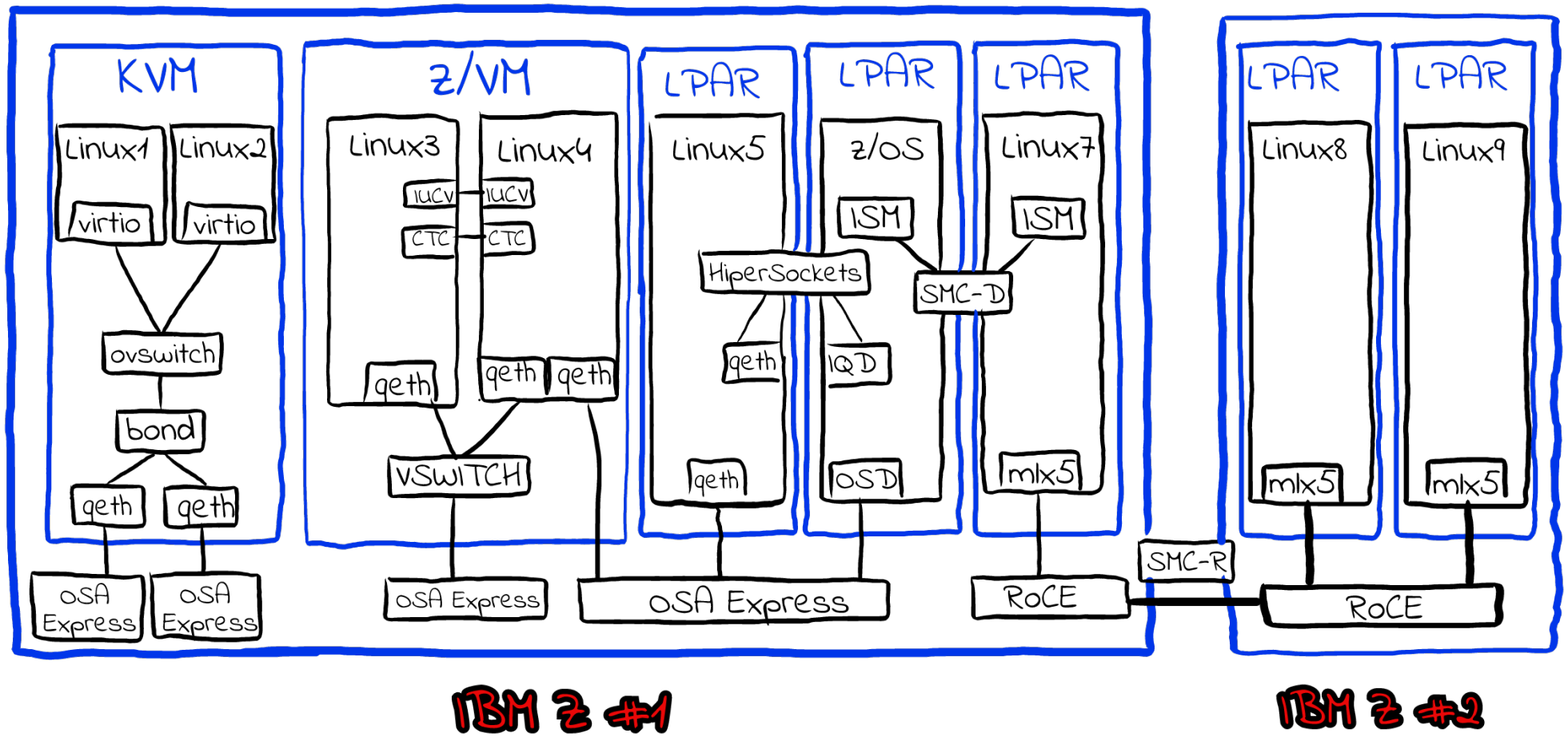
# S390 Networking in one Picture



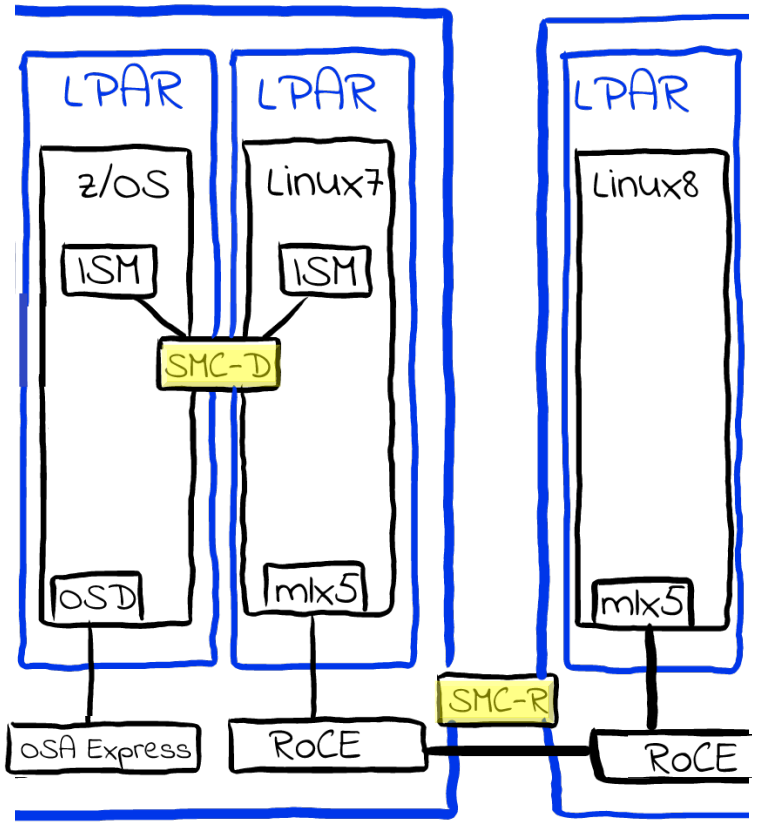
# Network Environments

- Network environments are diverse and often complex
  - 100+ physical network ports
  - 100s of VMs/partitions on multiple hypervisors
  - In-Place upgrades are common
  - Multiple internal networking technologies
  - External switches are customer owned and controlled
- Exploit proximity without compromising isolation
  - Hipersocket Converged Interface (HSCI)
  - Shared Memory Communications (SMC)

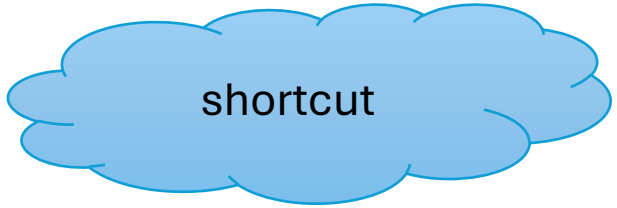
# S390 Networking in one Picture



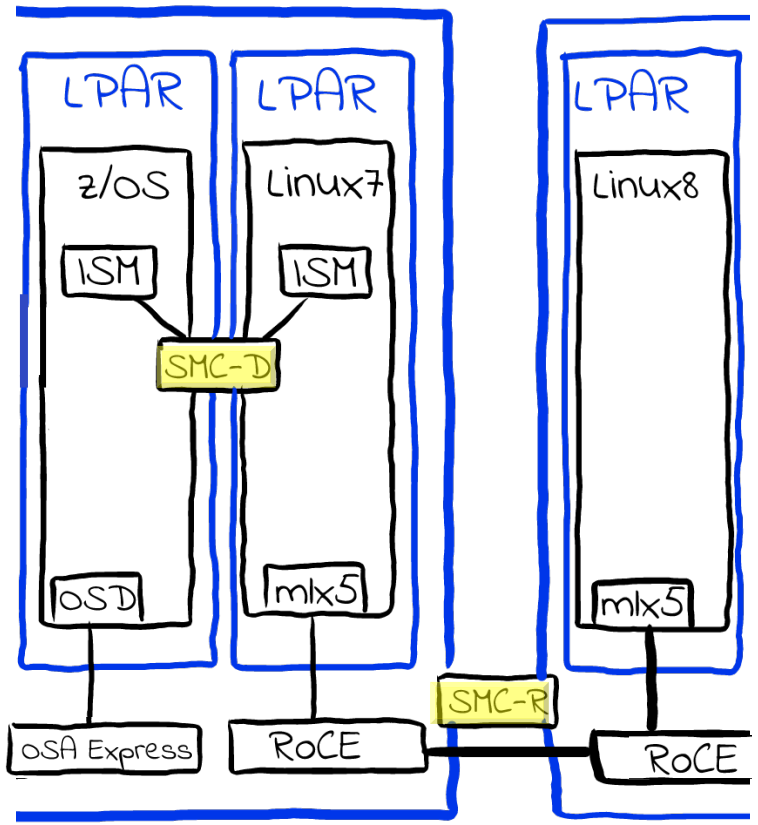
# S390 Networking in one Picture



# Shared Memory Communications (SMC)




- Aims to accelerate TCP/IP by exploiting proximity when available
  - SMC-D
    - Exploits when peers are on the same machine
    - TCP/IP upgraded to firmware mediated cross-partition memcpy()
    - Now available outside s390 across containers with CONFIG_SMC_LO
  - SMC-R
    - Exploits when peers share RDMA capable network
    - TCP/IP upgraded to RDMA (RoCE)
    - Not bound to s390 though some extra convenience features
- Enable in applications
  - Manually with AF_SMC or IPPROTO_SMC (in linux-next)
  - Transparently with LD_PRELOAD or BPF (requires IPPROTO_SMC)



# Alive, kicking and innovating

- Common challenges
  - High variability in environments
  - Large scale virtualization
  - Increasing network layering
    - Container, Orchestrator, Hypervisor, Machine, VLAN/VXLAN/SDN
- Unique Solutions
  - Shared NICs before SR-IOV
  - SMC-R/SMC-D
- Continuing to modernize and increasing standardization
  - Plan to shift to PCIe based network devices for direct access on Linux
  - SMC-D is expanding to other architectures