Virtualizing I/O - Three Models

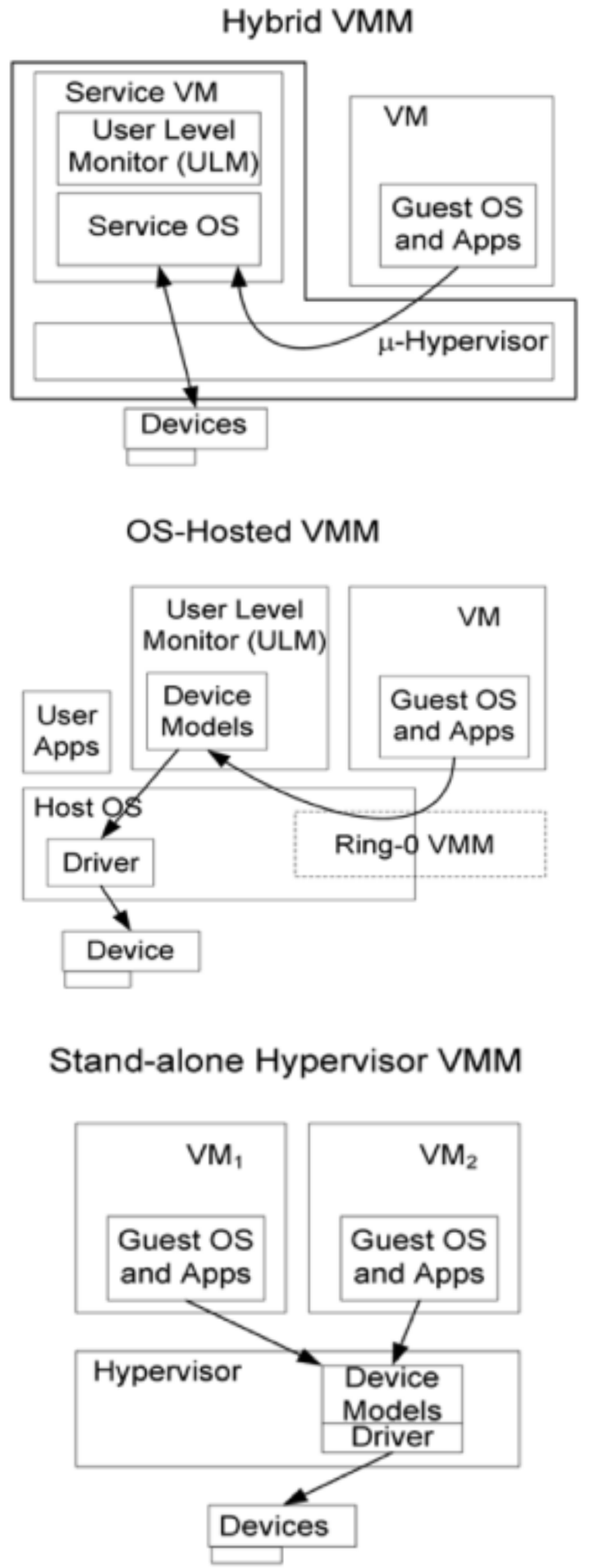
Xen : modify OS to use low-level I/O interface (hybrid)

- Define generic devices with simple interface: virtual disk, virtual NIC, etc.
- Ring buffer of control descriptors, pass pages back and forth
- Handoff to trusted domain running OS with real drivers

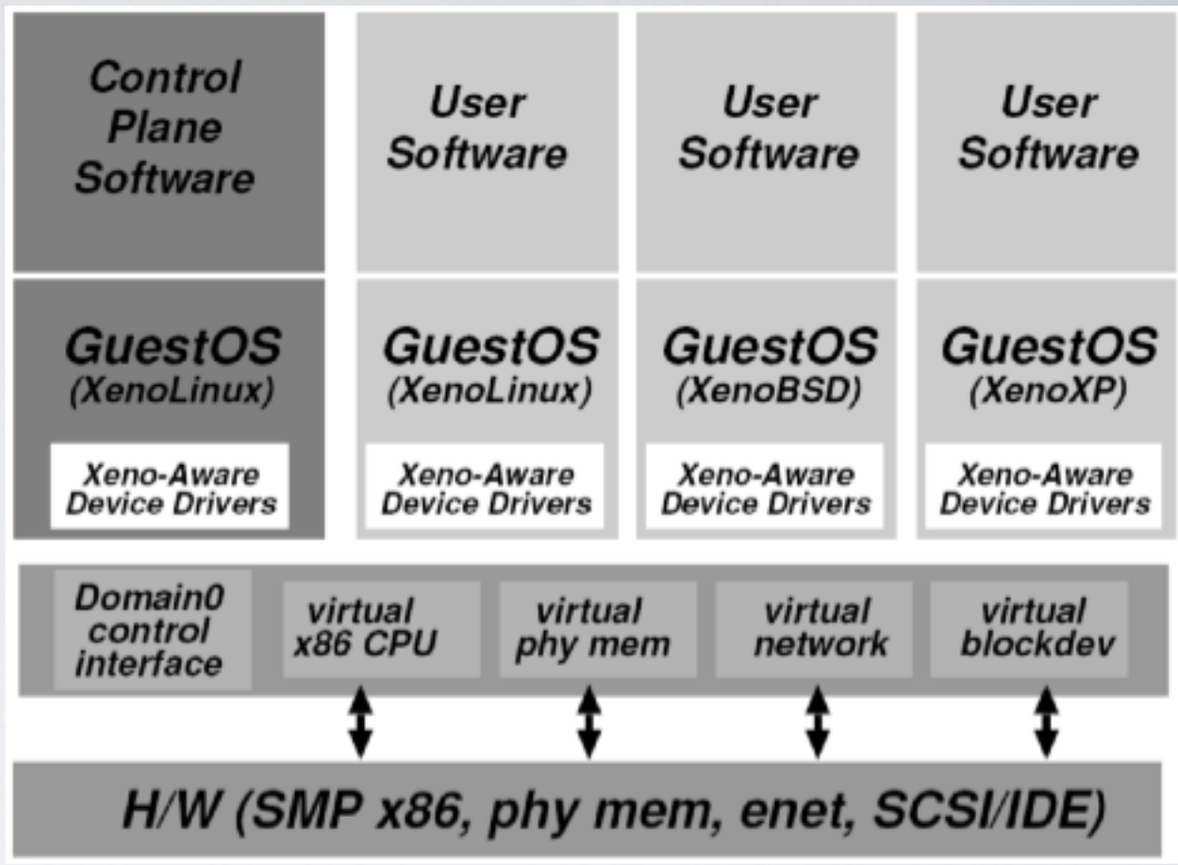
VMware : VMM supports generic devices (hosted)

- E.g. AMD Lance chipset/PCNet Ethernet device
- Load driver into OS in VM, OS uses it normally
- Driver knows about VMM, cooperates to pass the buck to a real device driver
  (e.g., on underlying host OS)

VMware ESX Server: drivers run in VMM (hypervisor)



VMM case study 1 - *Xen*



Early versions use "paravirtualization"

- Fancy word for "we have to modify & recompile the OS"
- Since you're modifying the OS, make life easy for yourself
- Create a VMM interface to minimize porting and overhead

Xen hypervisor (VMM) implements interface

- VMM runs at privilege, VMs (domains) run unprivileged
- Trusted OS (Linux) runs in own domain (Domain0)
  use Domain0 to manage system, operate devices, etc.

✓ Most recent version of Xen does not require OS mods because of Intel/AMD hardware support - commercialized via XenSource, but also open source