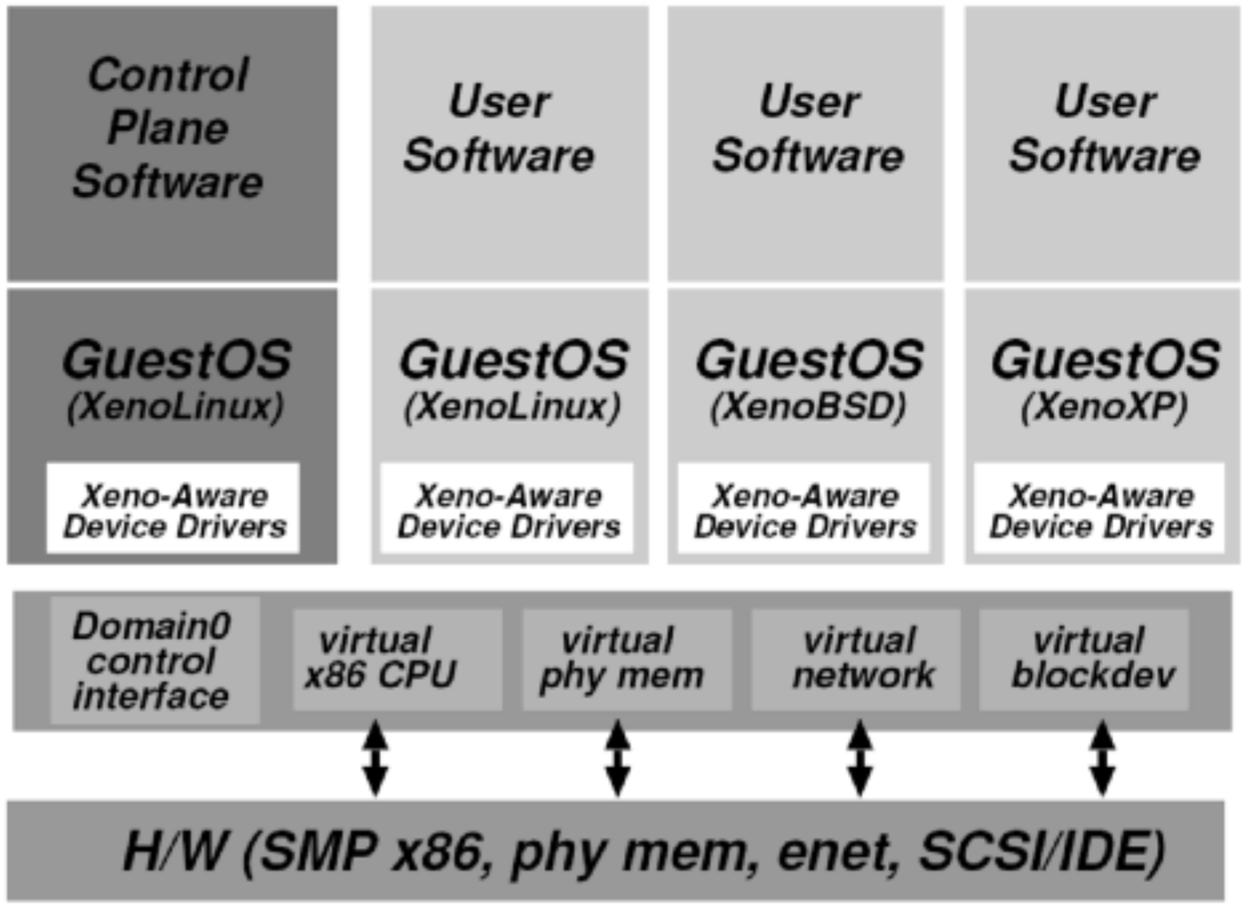
VMM case study 1 - *Xen*



Early versions use "paravirtualization"

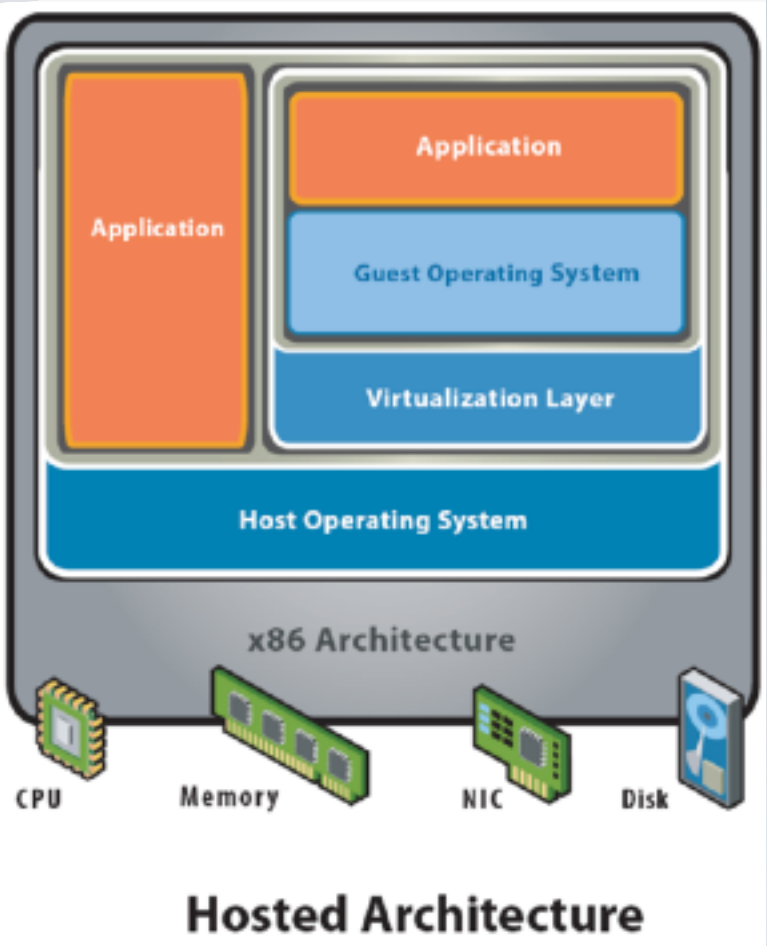
- Fancy word for "we have to modify & recompile the OS"
- Since you're modifying the OS, make life easy for yourself
- Create a VMM interface to minimize porting and overhead

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

VMM case study 2 - *VMware*



VMware uses software virtualization

- Dynamic binary rewriting translates code executed in VM
  - Most instructions translated identically, e.g. `movl`
  - Rewrite privileged instructions with emulation code (may trap), e.g. `popf`
- Think JIT compilation for JVM, but full binary x86 to IR code to safe subset of x86
- Incurs overhead, but can be well-tuned (small % hit)

✓ VMware workstation uses hosted model

- VMM runs unprivileged, installed on base OS (+ driver)
- Relies upon base OS for device functionality

✓ VMware ESX server uses hypervisor model similar to Xen, but no guest domain/OS