Imitation of Software and Hardware

Simulation, Emulation, Virtualization, and Translation

Teaching Evaluations

Drawing: Networking

- Take three minutes to draw “networking”

- Some reminders
  - Layers (physical, link, network, transport, (security), application)
  - Protocols (e.g., Ethernet, IP, TCP, HTTP)
  - Packets (headers and payloads)
  - Network hardware (NICs, routers, etc.)
  - Sockets
  - Efficiency vs reliability
  - TCP (handshakes, acknowledgements, windows, fairness)

Imitation

- We can imitate/fake all the hardware and software we’ve discussed

In relation to operating systems and hardware

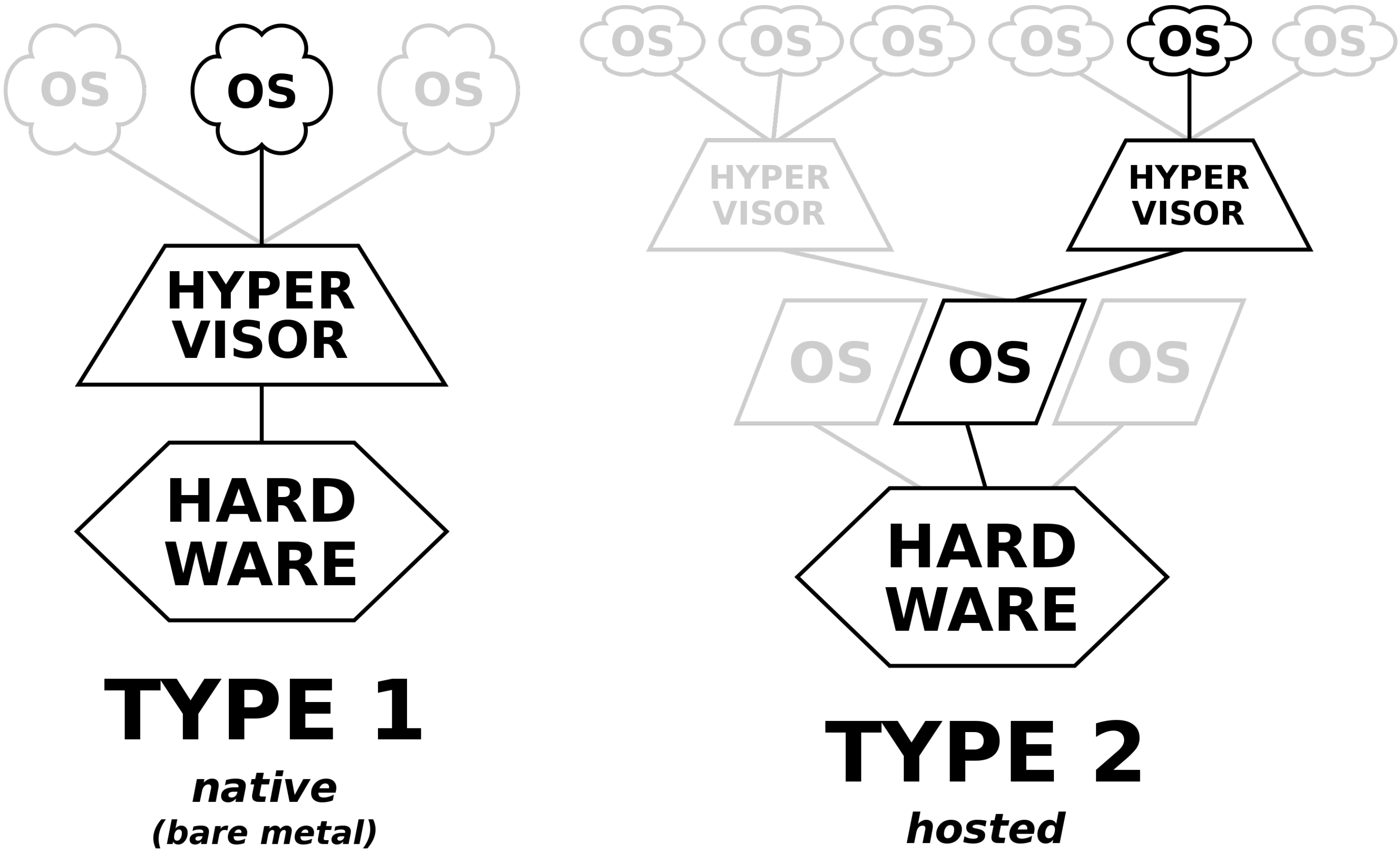
- Simulation: fake a system at some level of abstraction
- Emulation: run instructions designed for a different architecture
- Virtualization:
- Translation: translate some instructions and run others natively

Simulation

Emulation

Virtualization

Hypervisor Types



Common Technologies

- Cygwin: rewrite of Linux/Unix functionality (GNU, POSIX) for Windows
- MinGW: less complete rewrite; more focused on GCC
- Wine: translates Windows API calls to POSIX on the fly
- WSL1: translates POSIX API calls to Windows on the fly
- WSL2: uses a virtualization approach
- QEMU: generic machine emulator
- Other: KVM, Hyper-V, Proxmox, Xen

QEMU

- Run an OS and its applications on an otherwise incompatible host
- Can use a combination of emulation and virtualization
- Emulates
  - CPUs (x86, PowerPC, ARM, RISC-V, MIPS, etc.)
  - Network
  - Hard drives
  - Serial interfaces
  - USB
  - Display
  - Among other things…

```
qemu-system-x86_64 linux.img
```

Teaching Evaluations