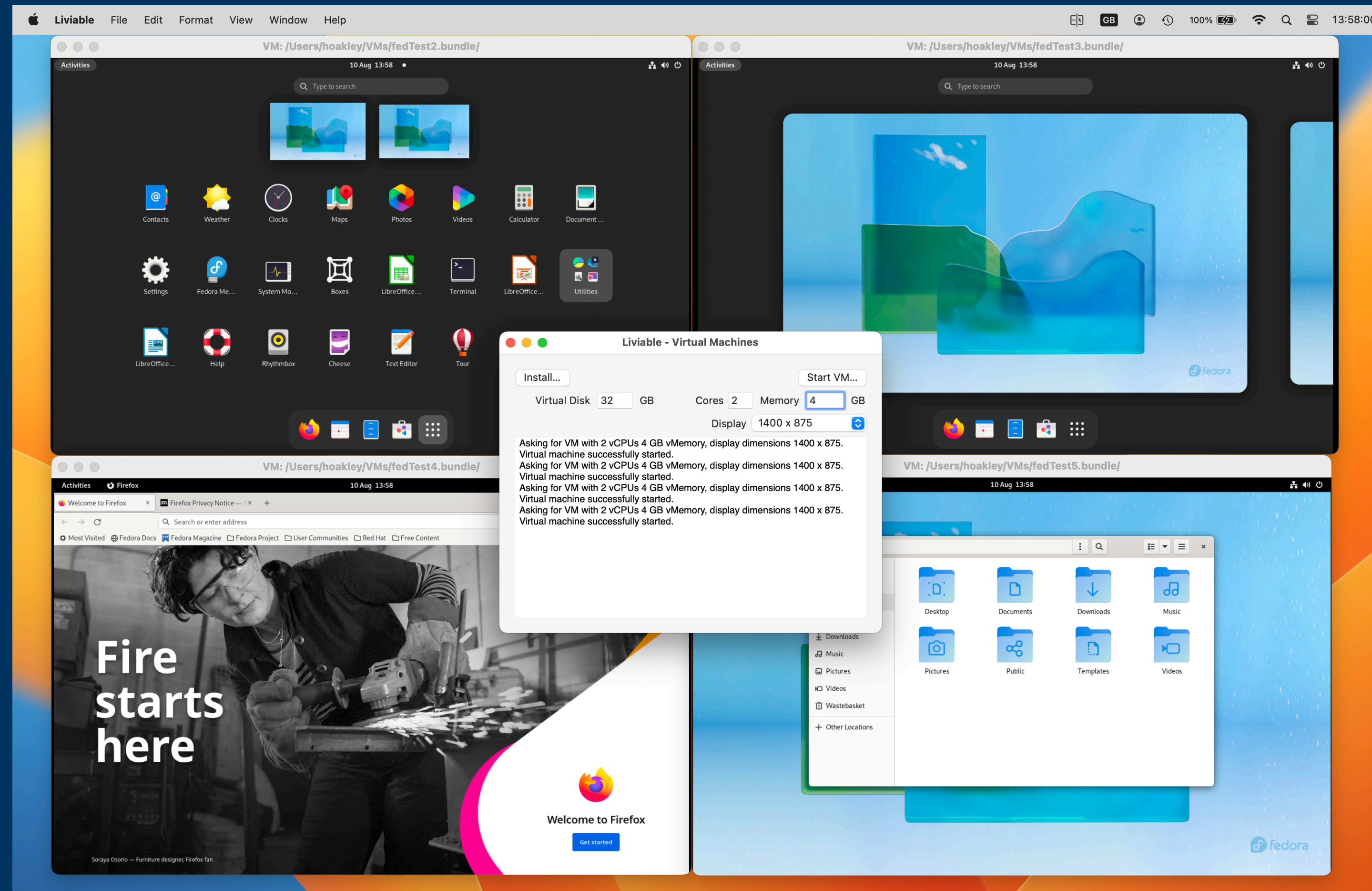


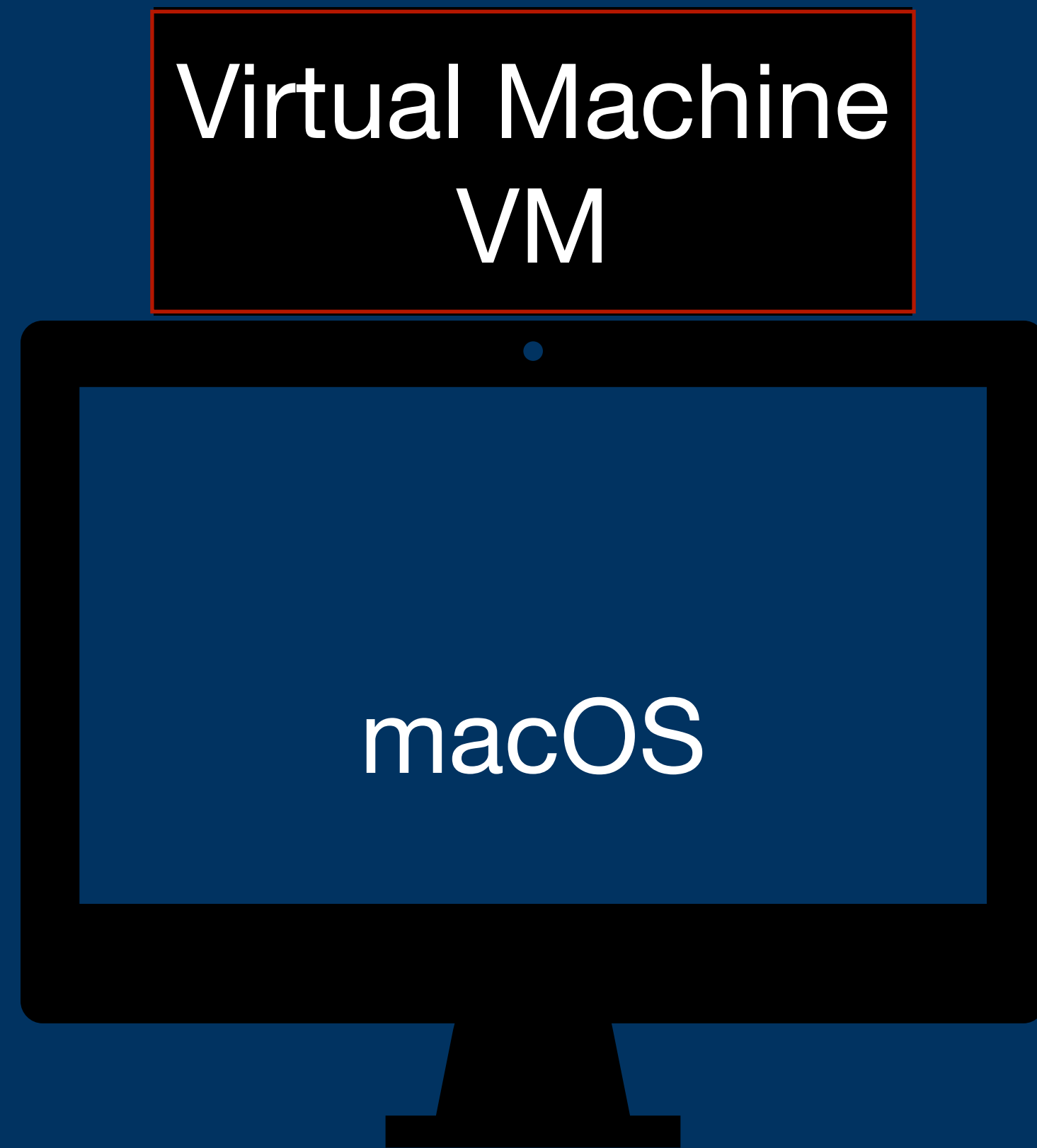
Virtualisation on Apple silicon Macs

How to run macOS and Linux without tears



Howard Oakley, The Eclectic Light Co <https://eclecticlight.co>

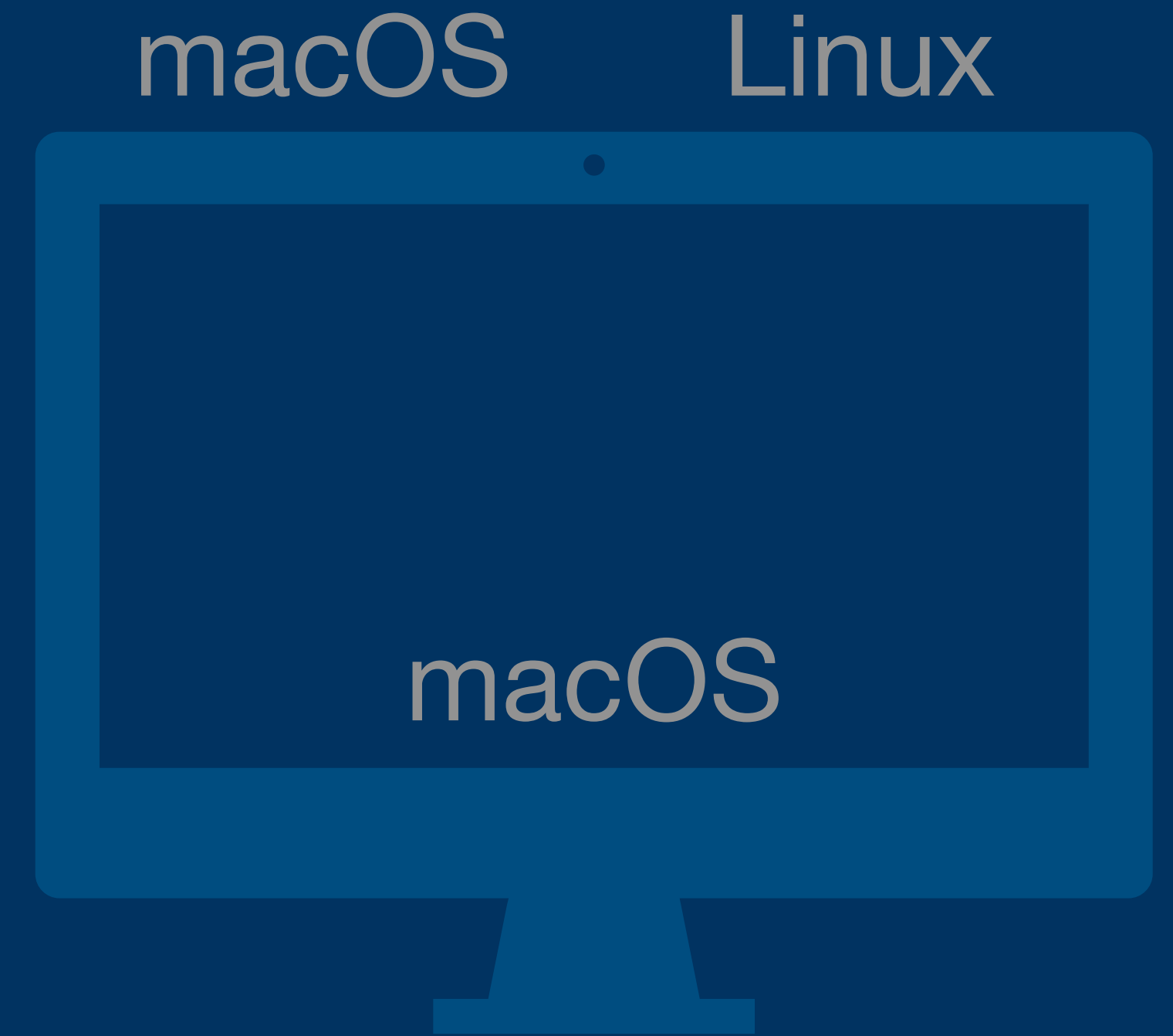
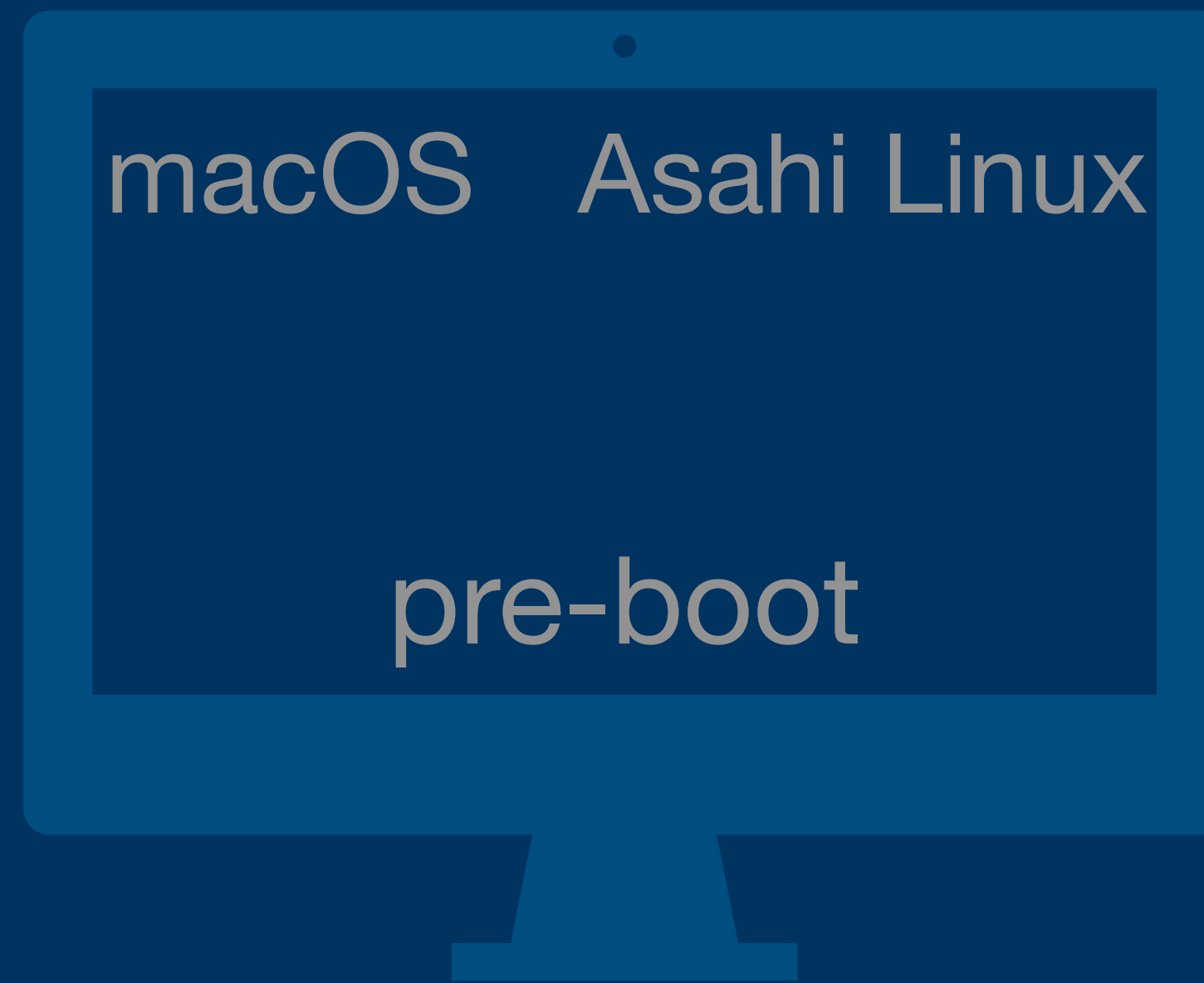
Terminology



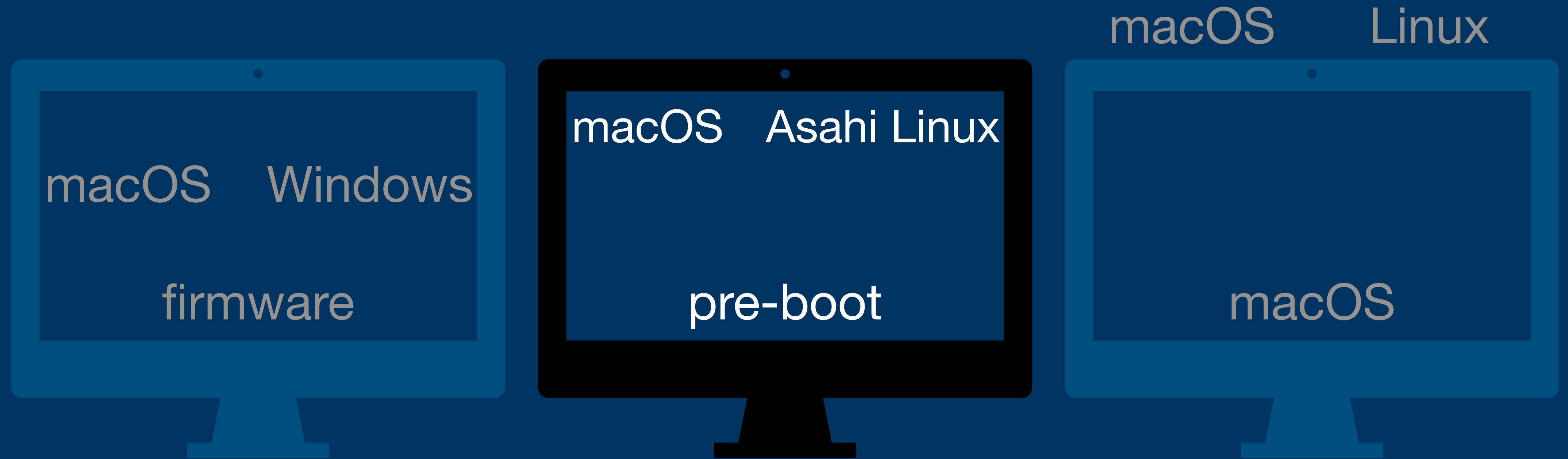
Guest

Host

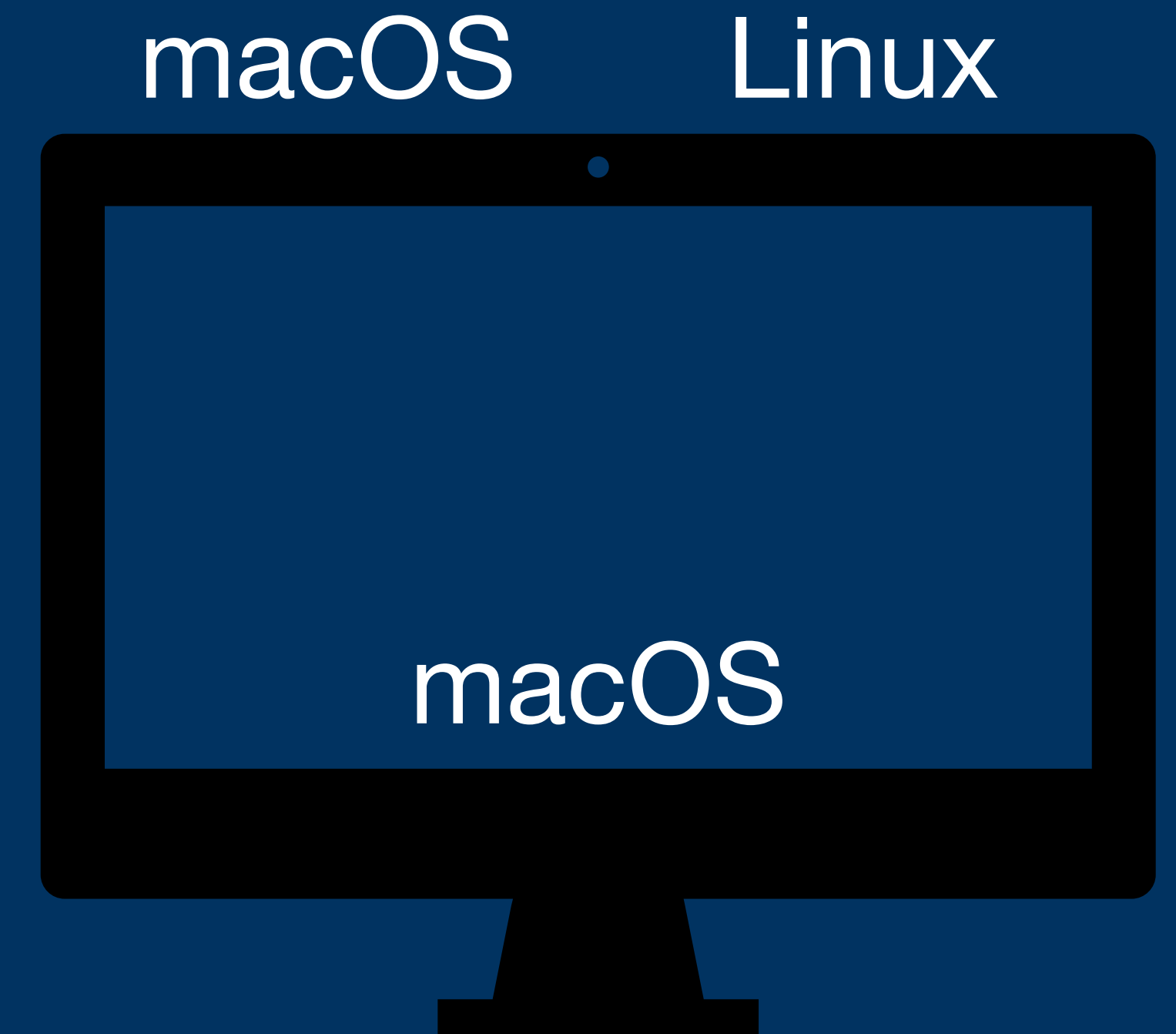
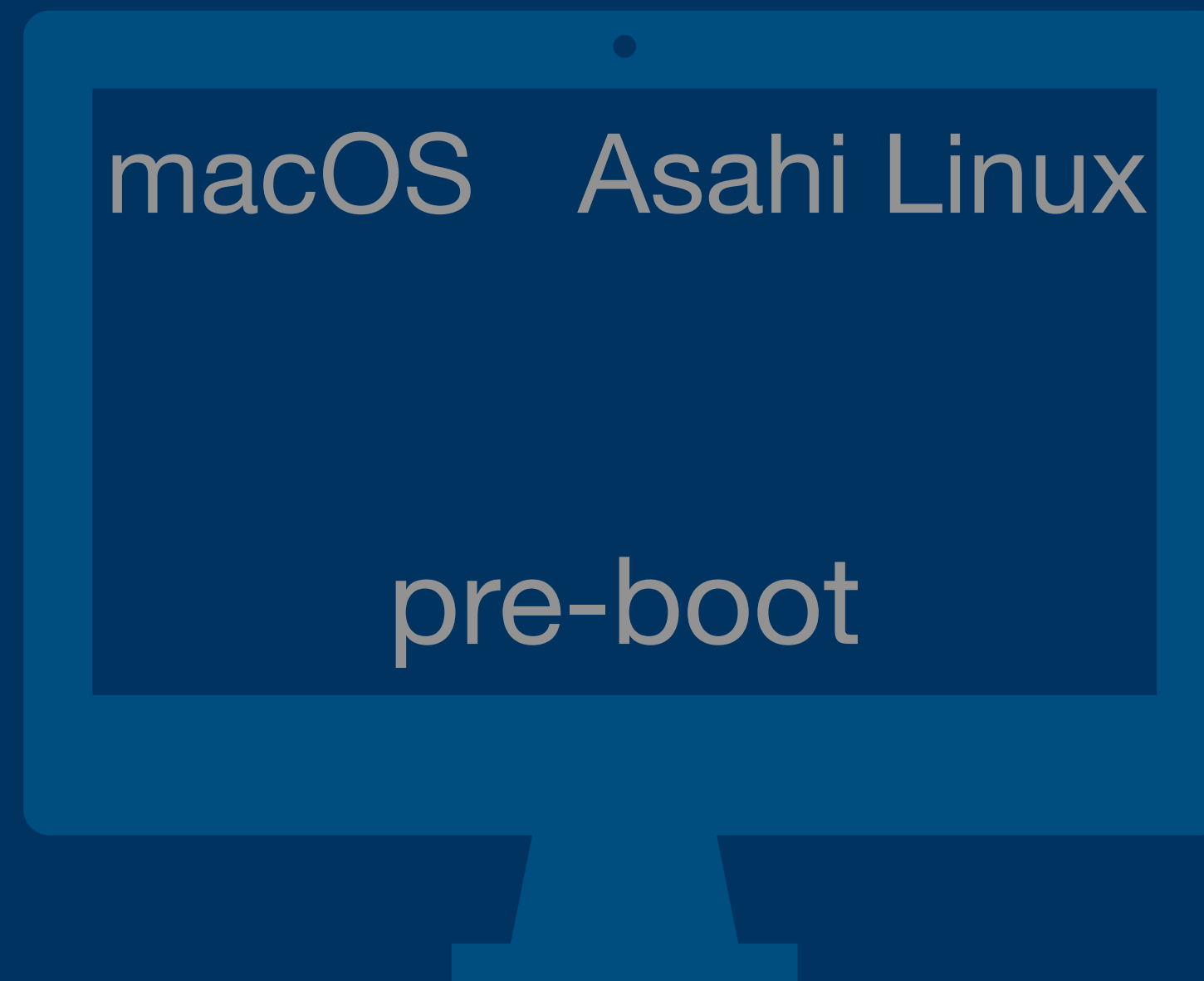
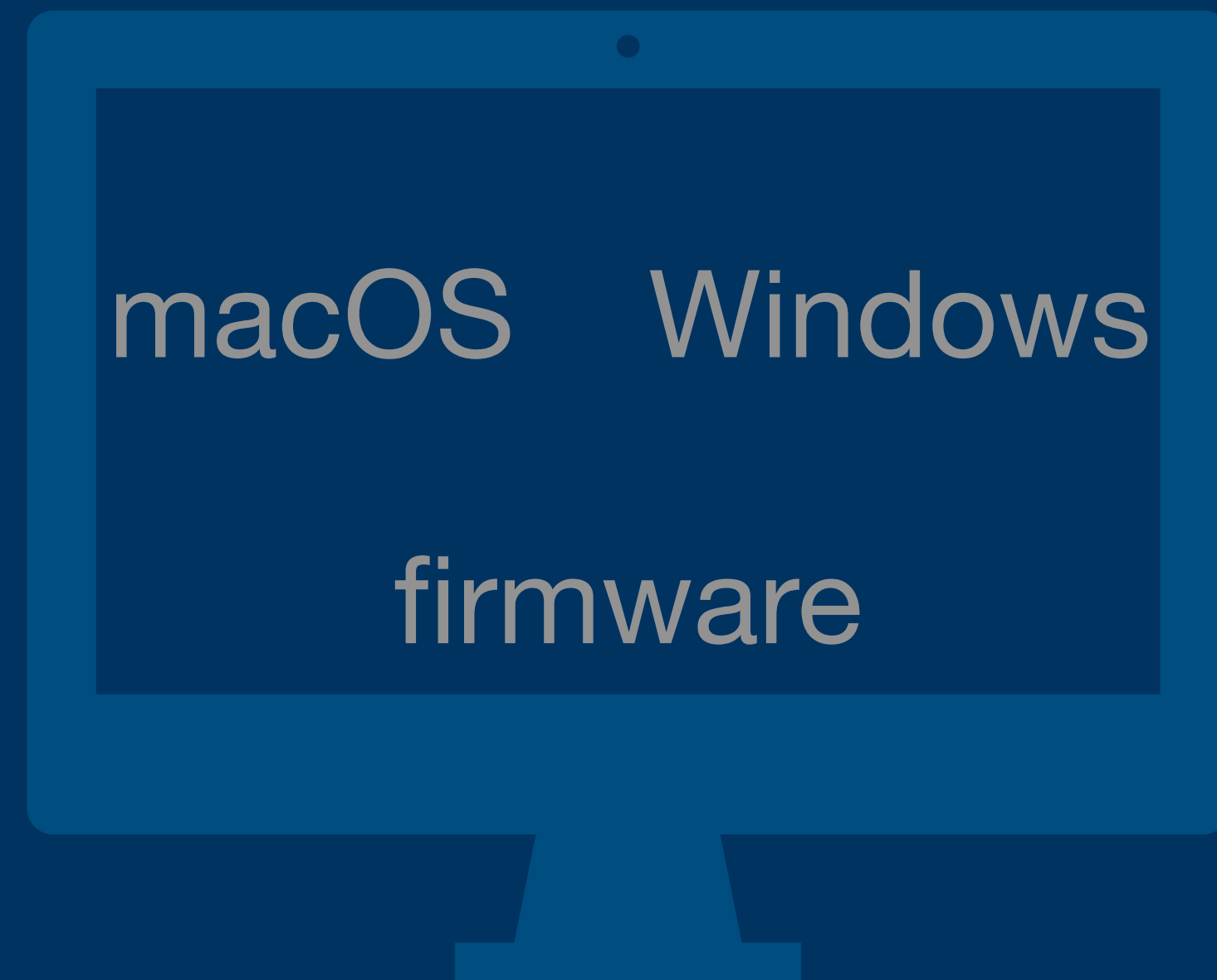
Boot Camp



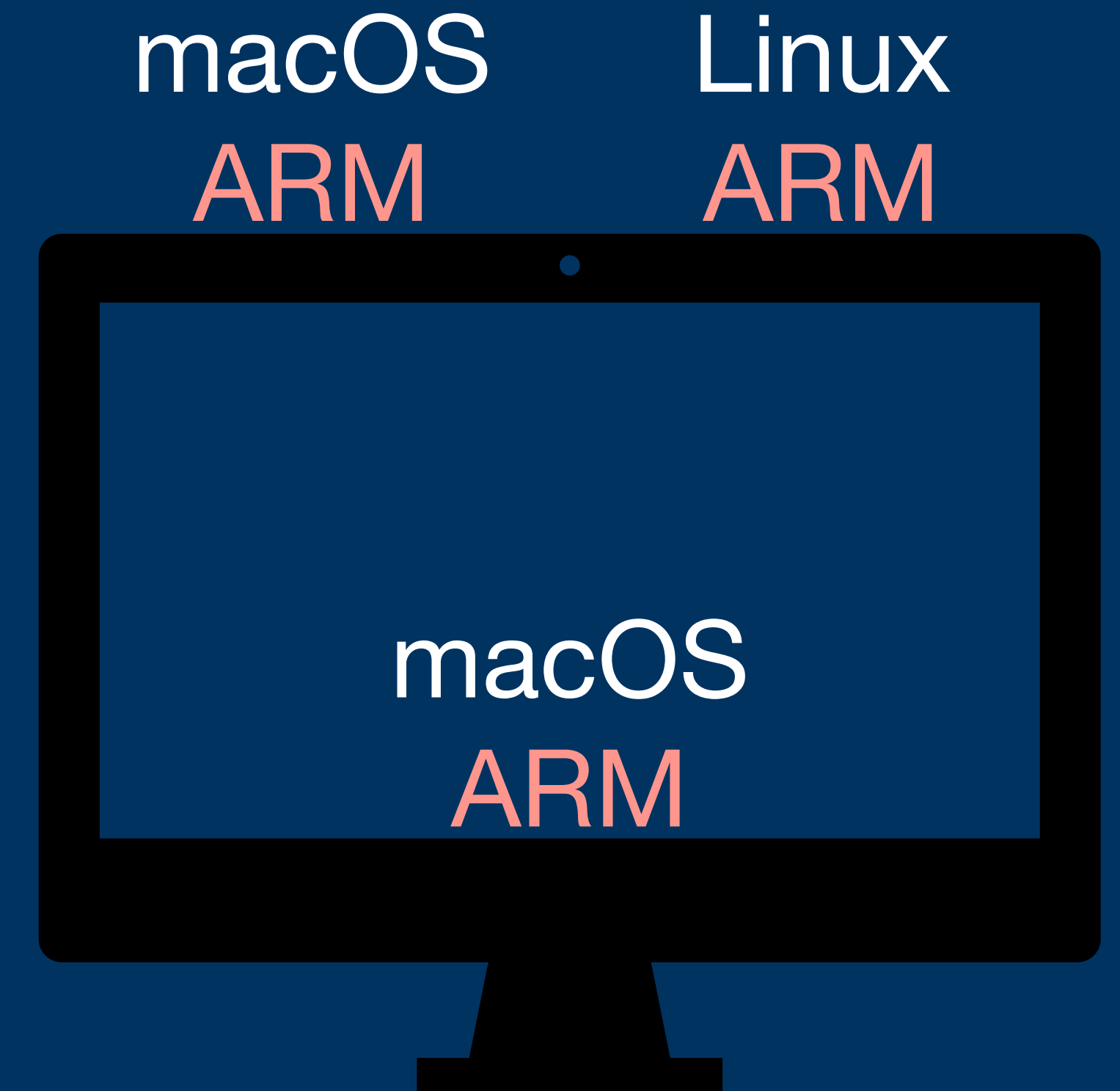
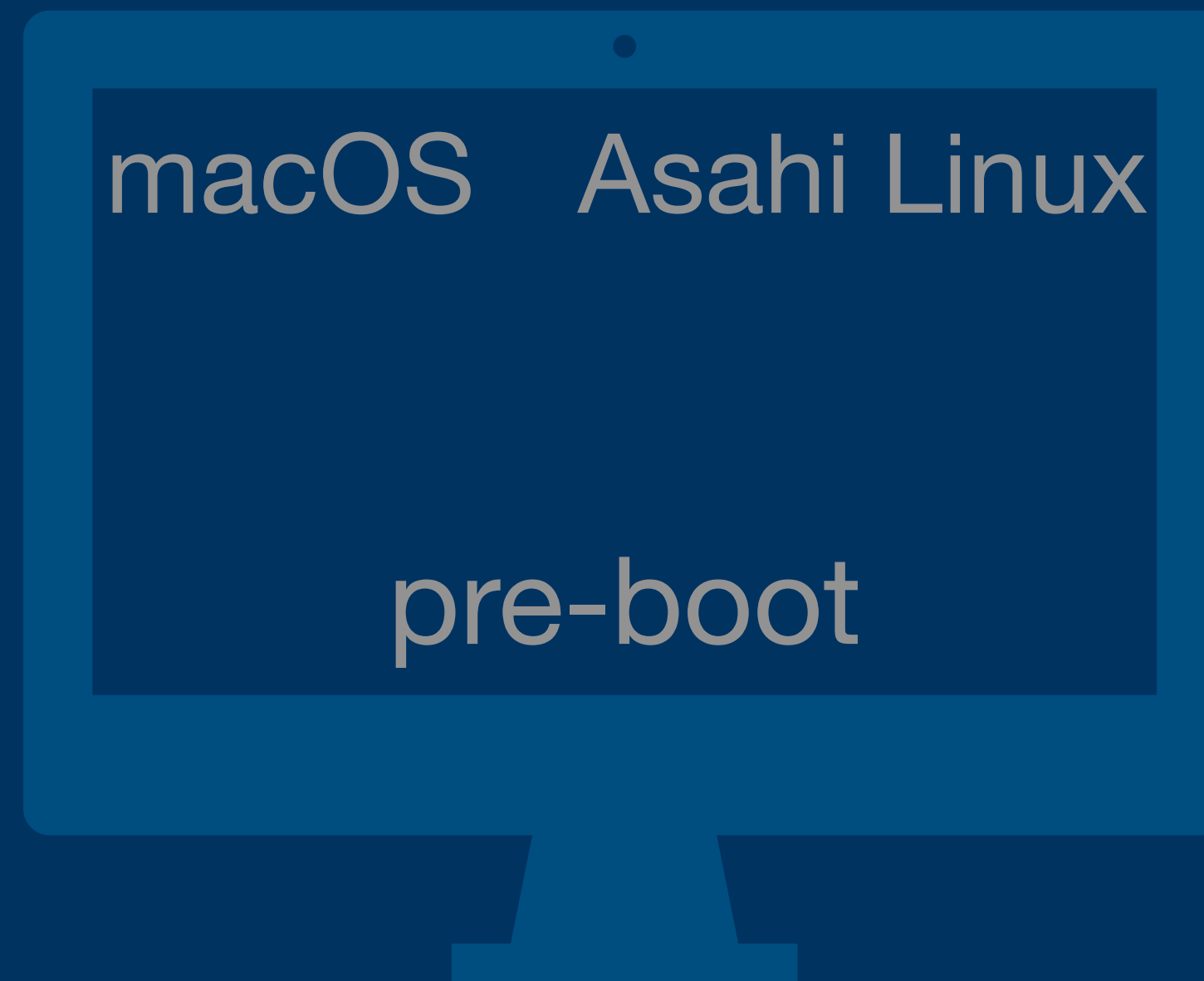
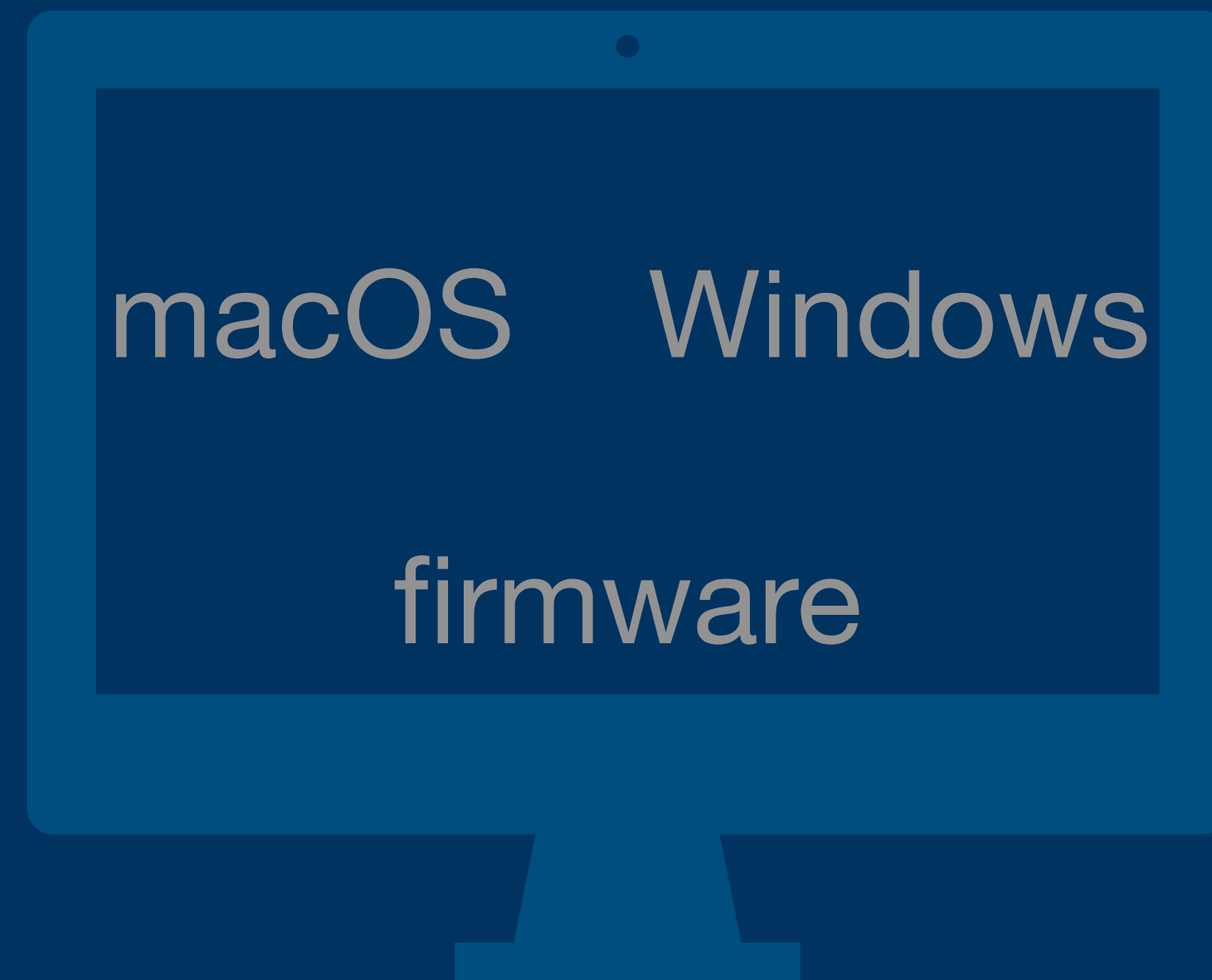
Dual boot



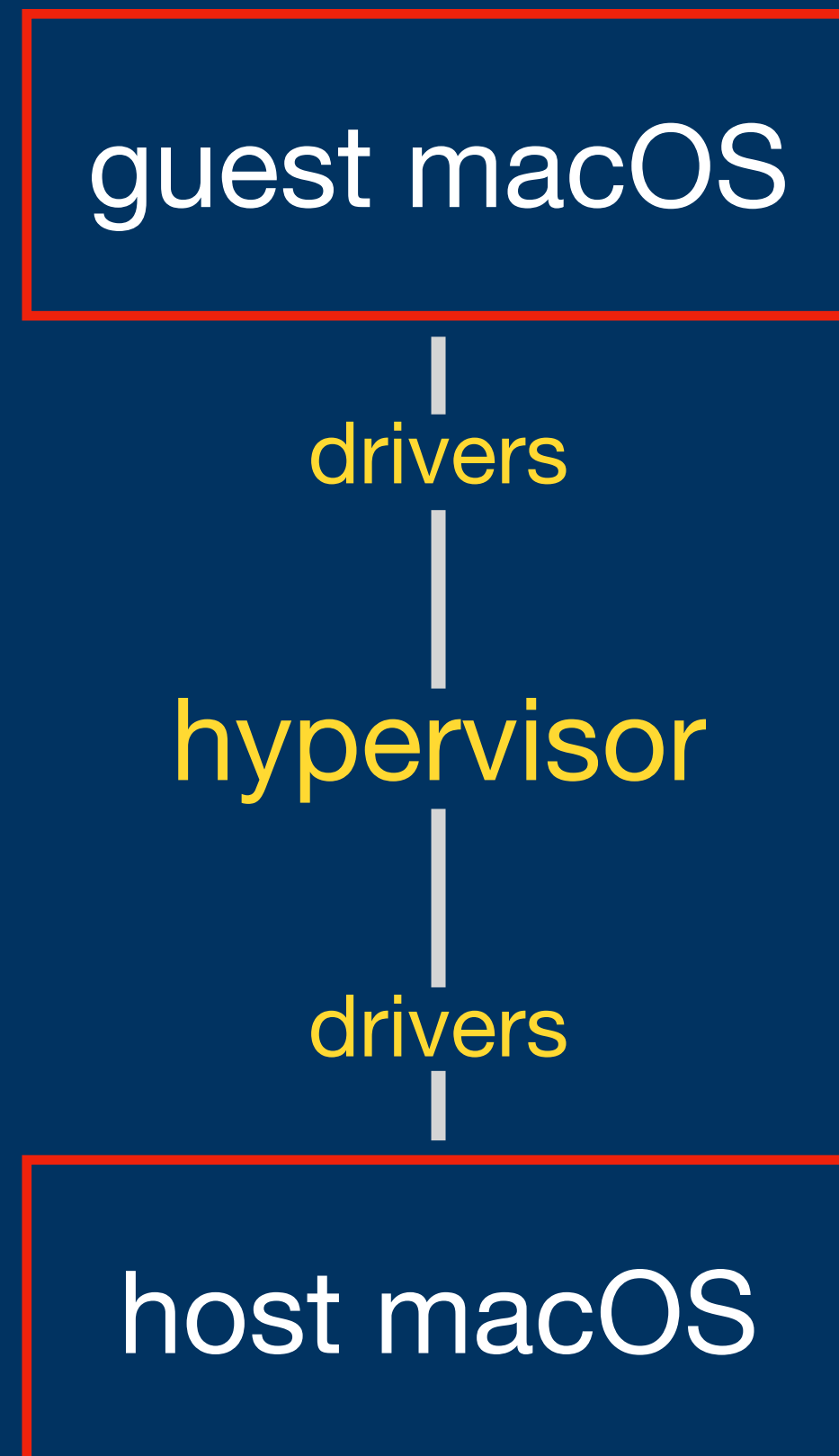
Virtualisation



Virtualisation



Hypervisor or lightweight virtualisation

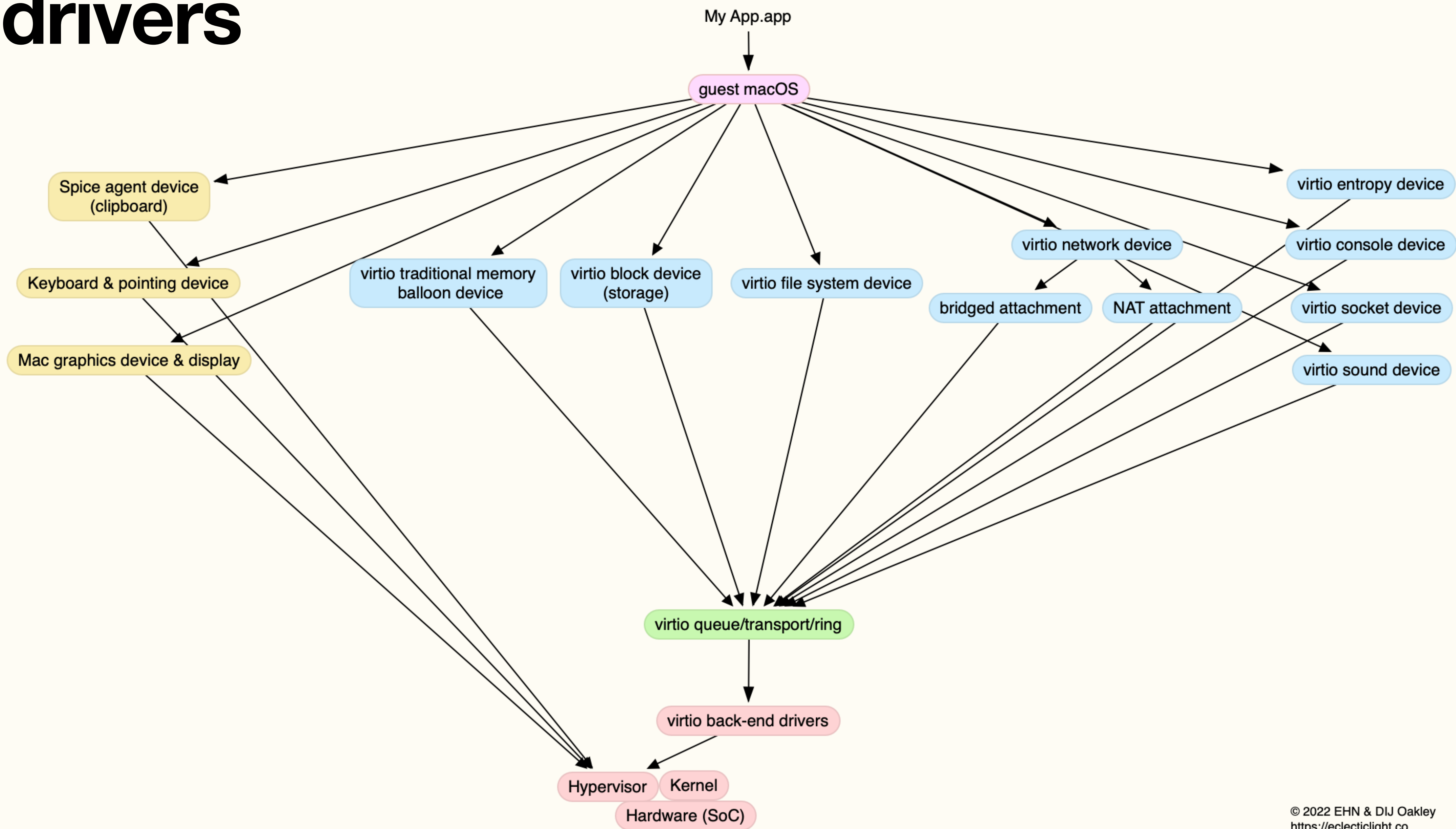


ARM Virtualisation Extensions

- EL2 hypervisor
 - stage 2 translation for control of access to memory-mapped resources
 - EL1/0 instruction and register access trapping, for trap-and-emulate
 - virtual exception generation

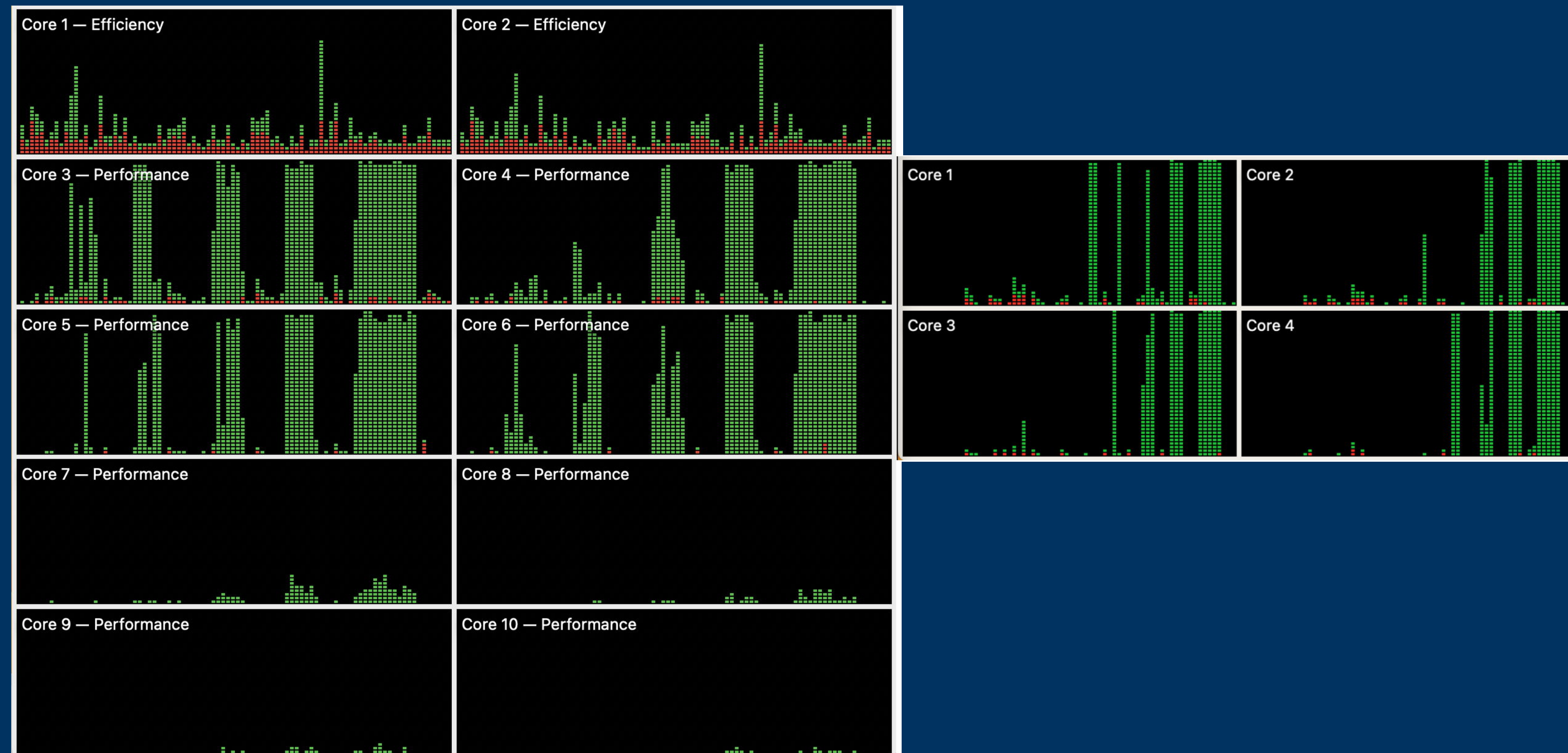
Virtio drivers

Device Support in macOS Virtualisation

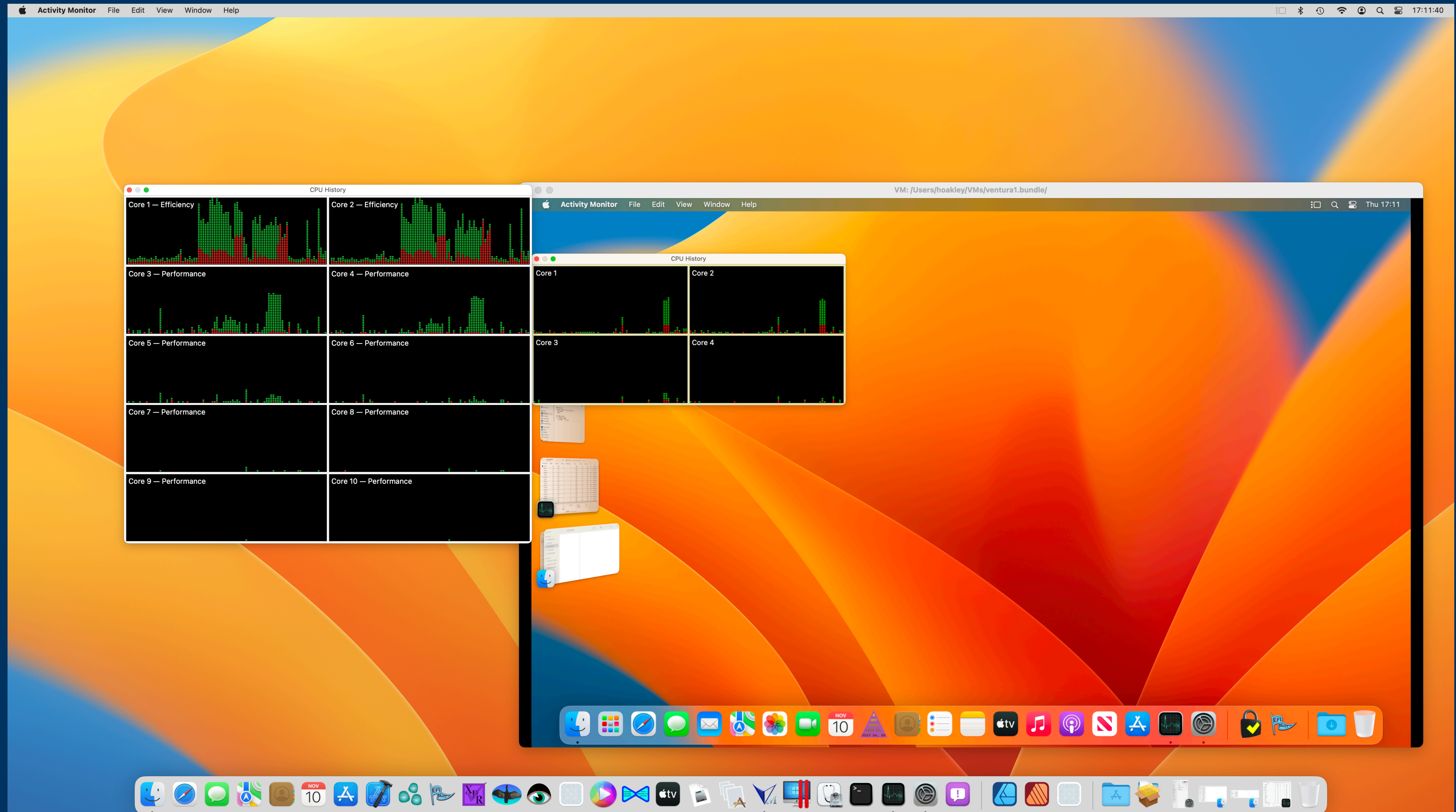


CPU and GPU performance

- each vCPU mapped as a thread on the host
- native performance
- no QoS control
- no Linux GPU



vCPU allocation



Virtual storage

- Virtio Block Device is a Disk Image
- on native 8.1/7.3 GB/s SSD, falls to 4.4/0.7 GB/s in macOS VM
- shared folder support requires Ventura host and guest for virtiofs

Networking

- NAT networking (easy) performs NAT on guest packets for routing through host
- Bridged networking sends and receives on host interface, using different network layer
- no Handoff support

Display

- macOS Retina support best using HiDPI options in Displays settings
- needs to be set up in each VM
- occasional surprise and magnifying glass

iCloud, Apple ID, App Store

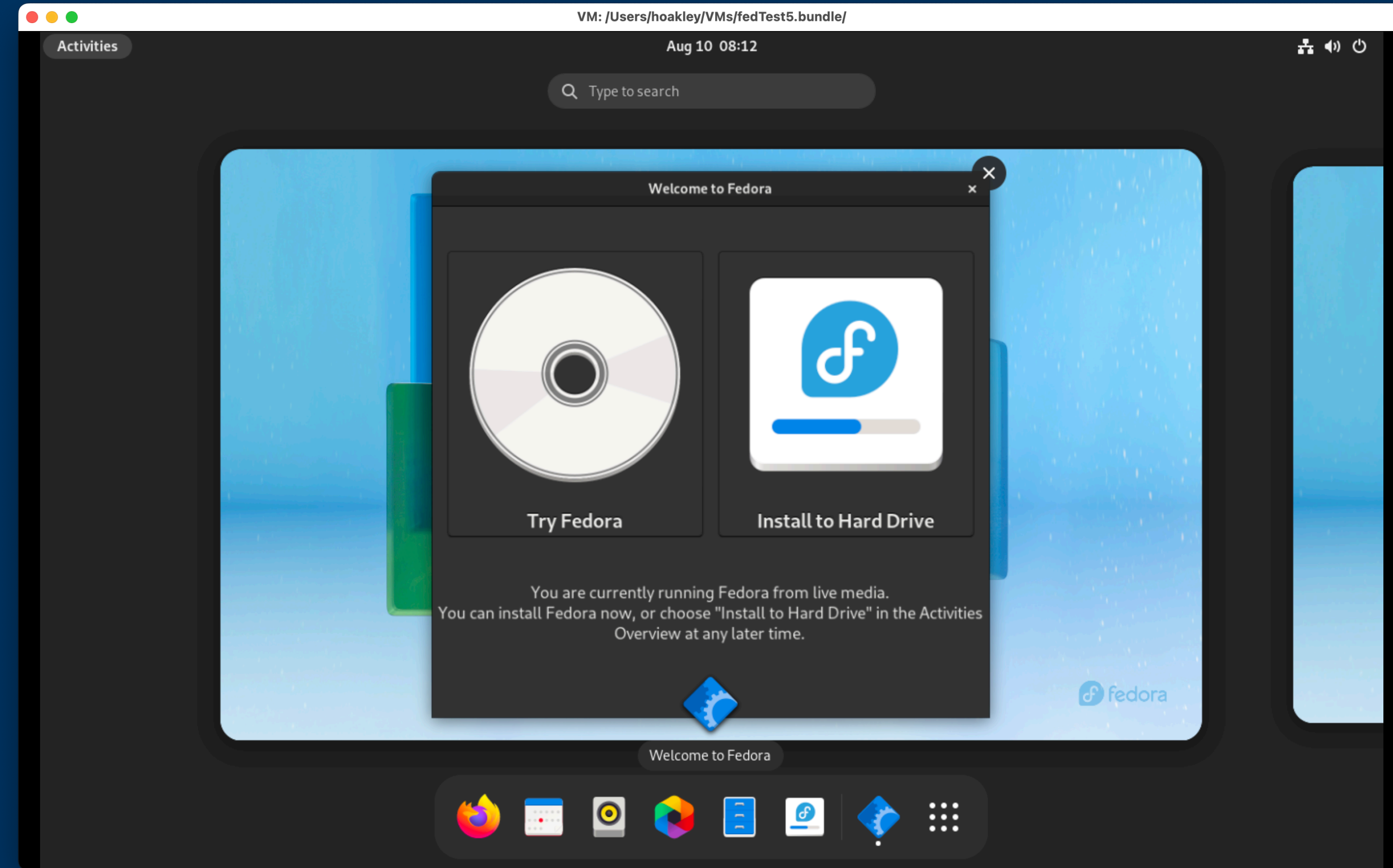
- not supported (yet)
 - no iCloud sign-in
 - no Apple ID
 - no App Store, although Apple's free apps still work
 - no AirDrop etc.

Rosetta 2 x86 translation

- available in VM (macOS and Linux)
- not available to guests (no x86 guests)
- fiddly to set up in Linux

Linux distros

- built for ARM
- distributed as ISO
- containing bootable installer
- CLI Linux requires kernel and RAM disk images

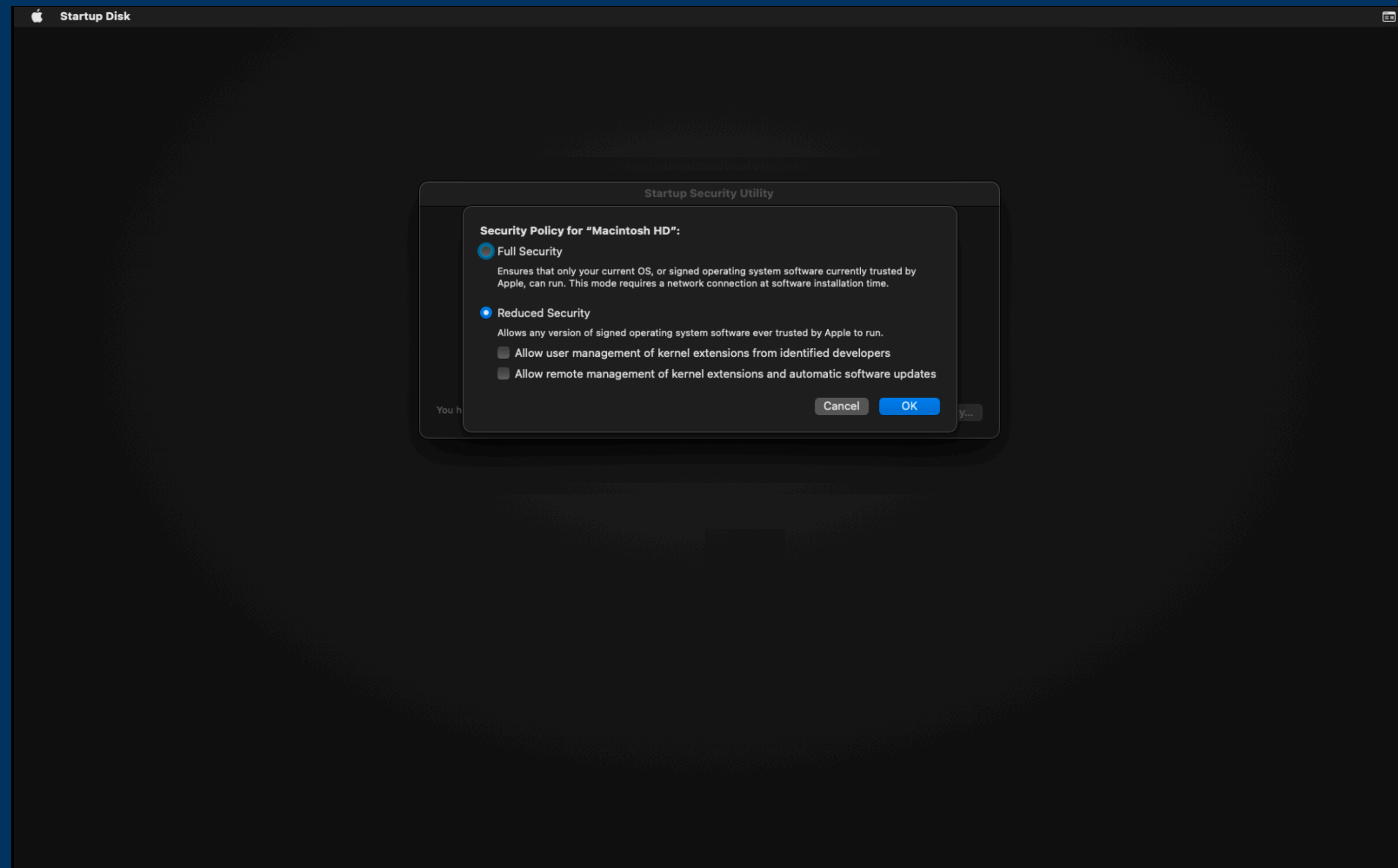


Creating a macOS VM

- from IPSW image, similar to DFU restore
- creates VM bundle containing
 - MachineIdentifier
 - HardwareModel
 - AuxiliaryStorage: iBoot Stage 1, NVRAM, etc.
 - Disk.img: boot disk image with 3 containers, etc.

Recovery mode

- boot option in Ventura
- access to Startup Security Utility, for kernel extensions and SIP



Coding a virtualiser

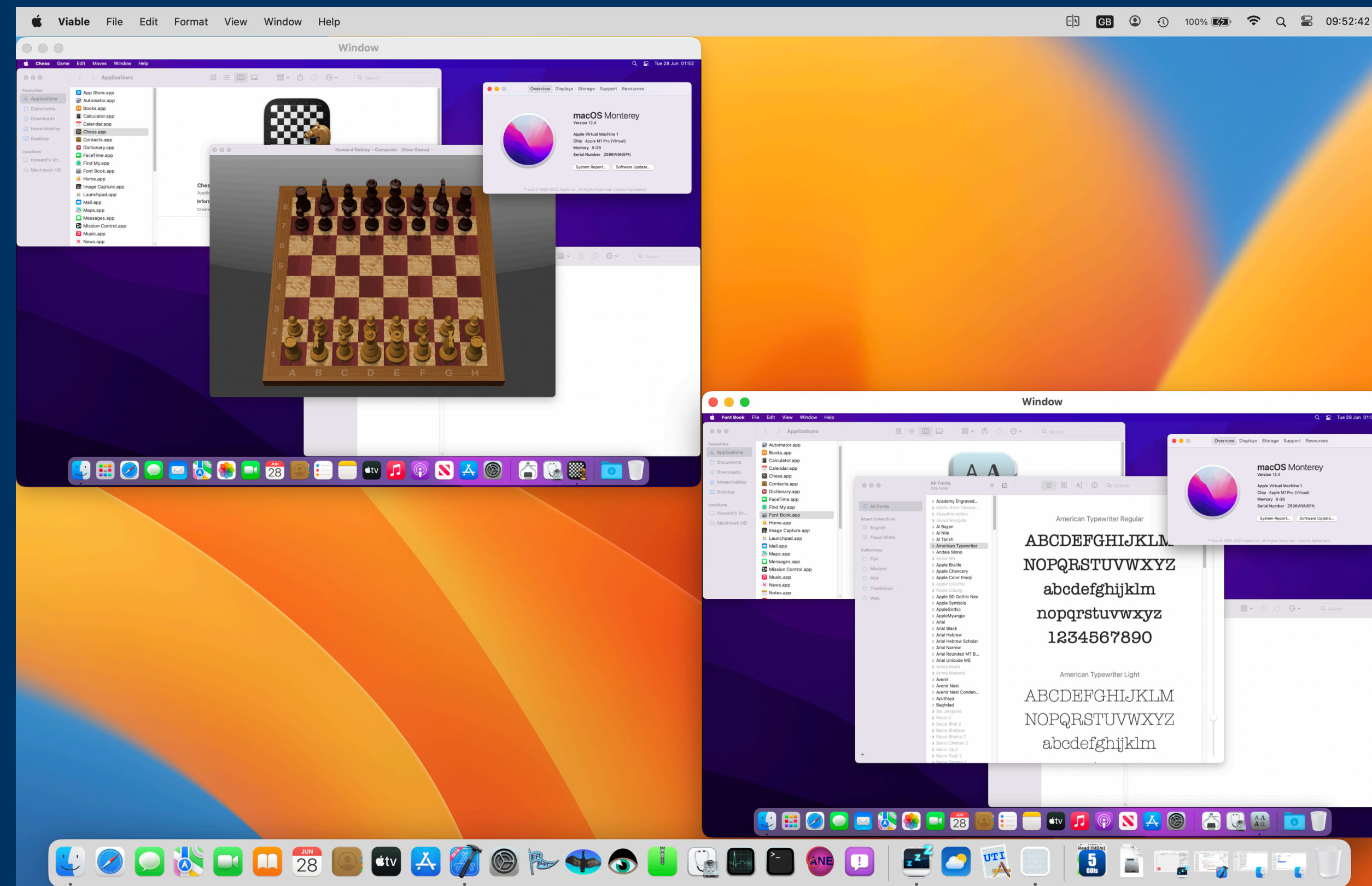
- remarkably simple
- excellent documentation
- excellent example code
- simple API calls

Limitations

- don't run excessive vCPUS, or they will deadlock
- forget iCloud, Apple ID, App Store apps, Handoff, etc.
- limited Linux graphics performance
- limited macOS (12, 13 only) and Linux distro support
- CLI Linux is still a hack

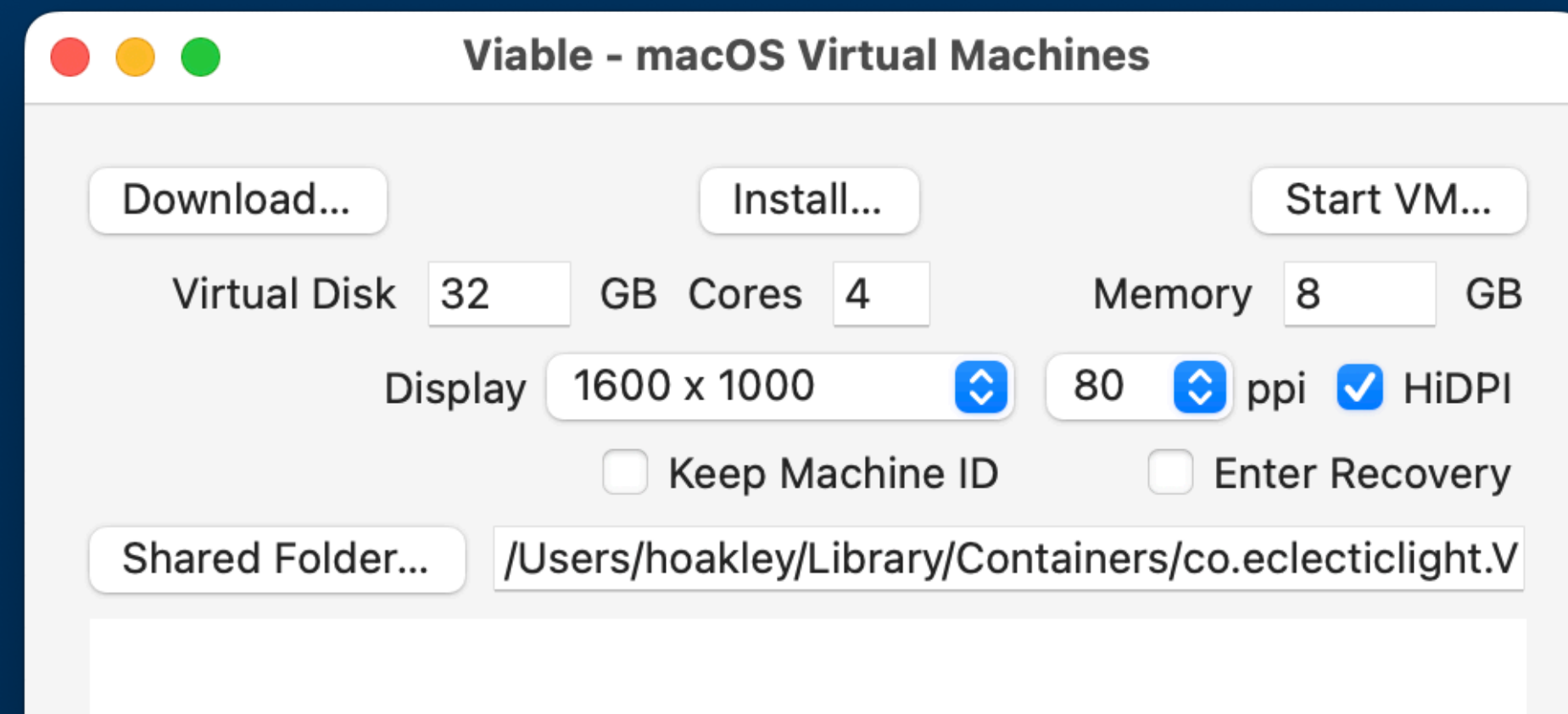
macOS licence limits

- maximum 2 VMs running concurrently
- purposes:
 - software development
 - testing during software development
 - using macOS Server
 - personal, non-commercial use



Machine Identifiers

- each VM has a unique Machine Identifier (MID)
- generated when the VM is first created
- for macOS guests, determines Serial Number, Hardware UUID, Provisioning UUID
- need control over MID for duplicate VMs



Why lightweight virtualisation?

- future support for x86 code on Apple silicon Macs – run Rosetta 2 in macOS 13 VM
- Linux required by some users
- Docker-like containers
- Windows?

Thank you for your attention ...

... and happy virtualising

<https://eclecticlight.co>