CS 436
Cloud Computing Applications

07.03.2024

&

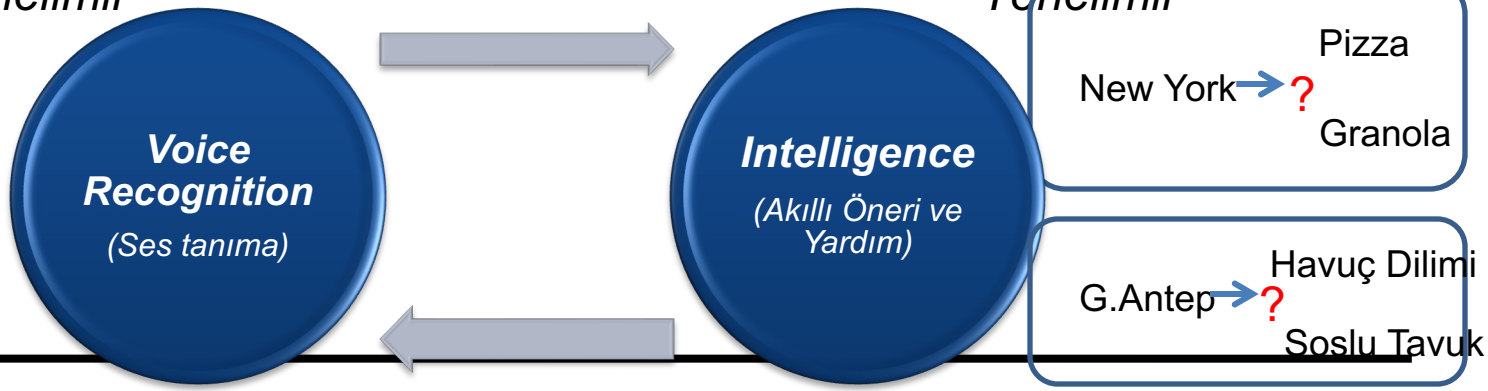
14.03.2024

Akıllı Kişisel Yardımcı Uygulamaları

(Intelligent Personal Assistant Software)

"Voice Search" Yönelimli

Kişisel Yardımcı Yönelimli



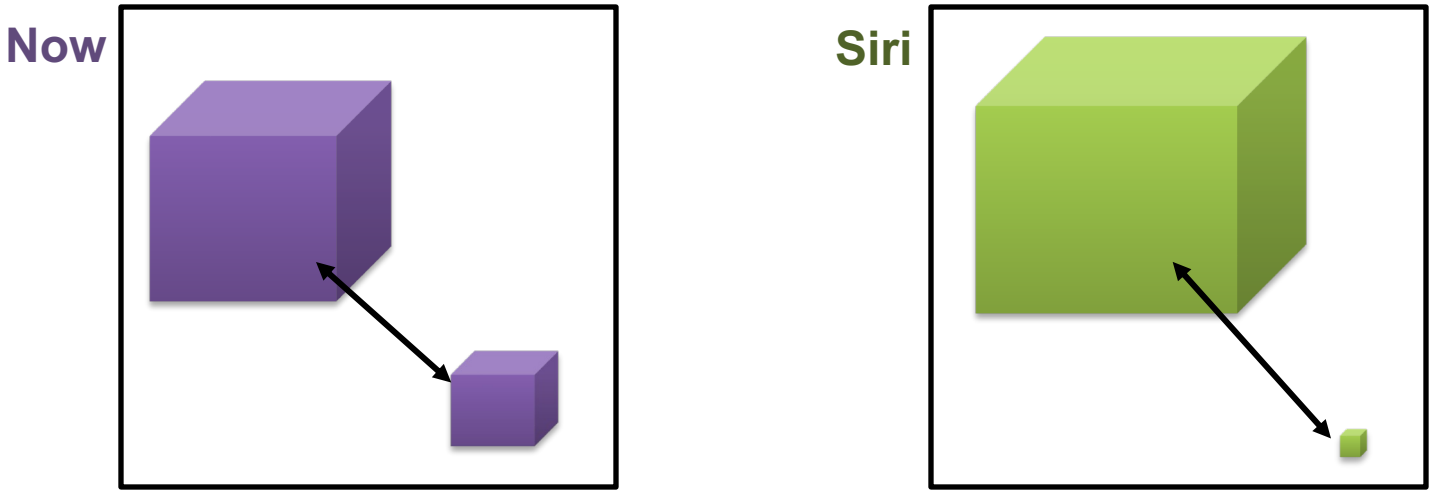
Functinality & Network Communication

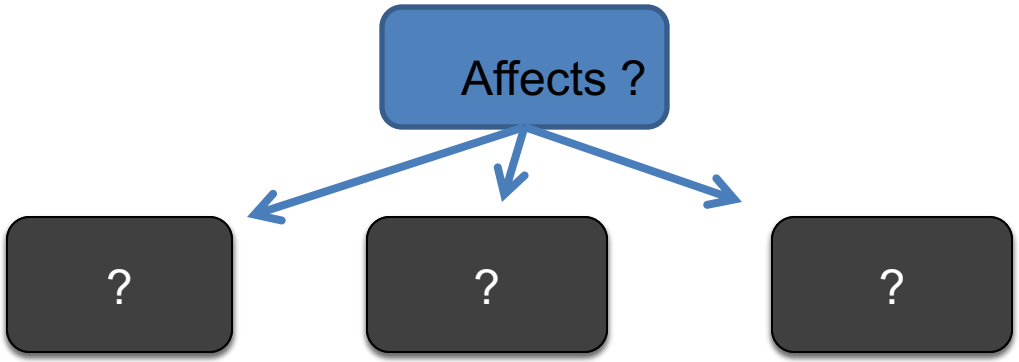
- Speech processing
  - Analog-Digital conversion
  - Noise removal
- Speech to text
- Text interpretation
- Answer / reply
- Presenting it to the user

How much of it runs on handset ? How much of it runs on cloud?

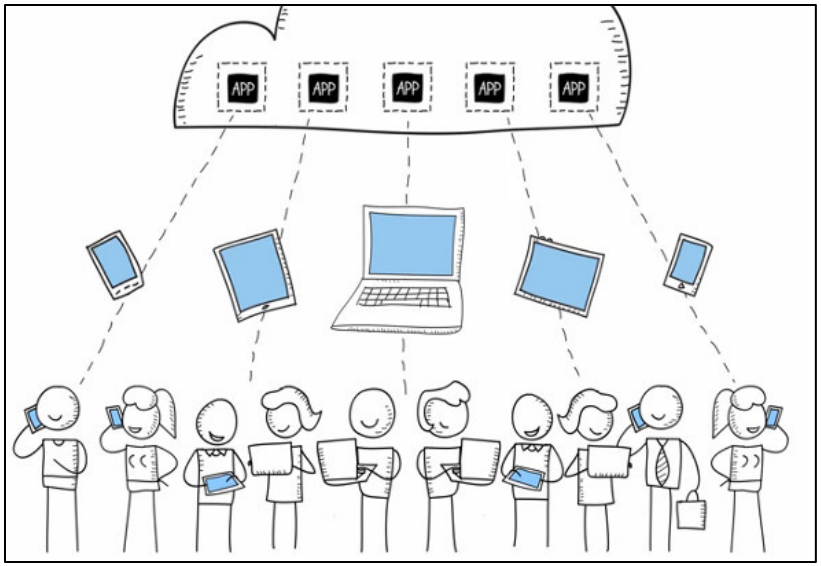
Now ve Siri Hesaplama Yükleri

(Now vs. Siri in terms of Computation Load)





Case Study: Amazon Appstream

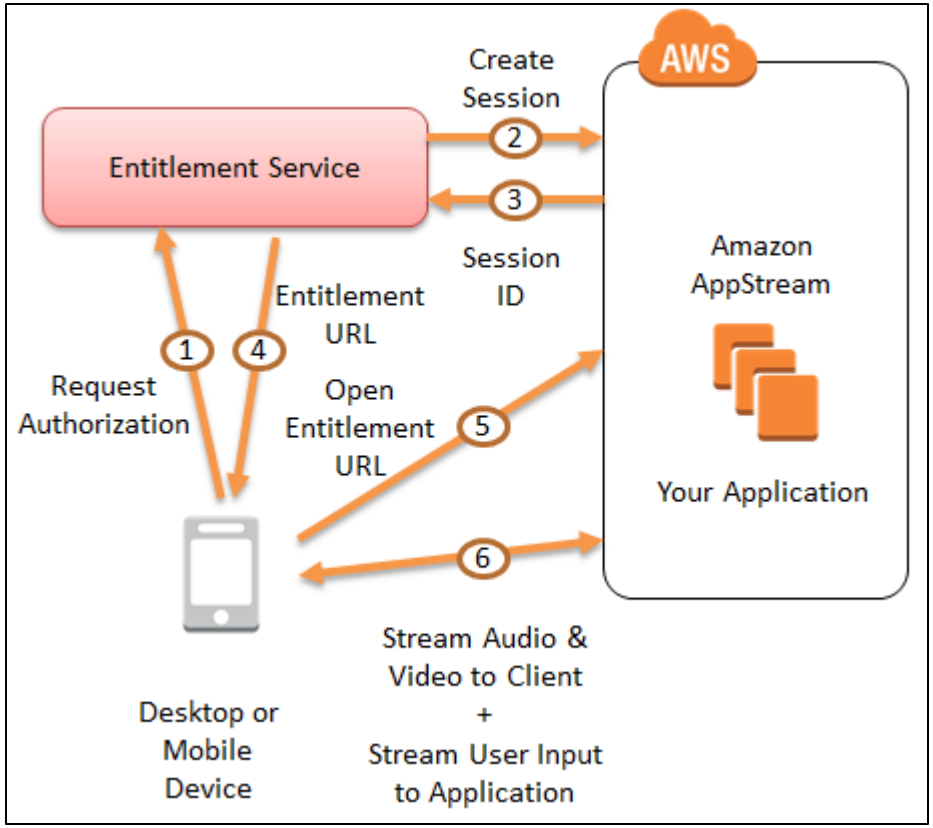


AIM: Stream resource intensive applications (games) from the cloud

Run the application on cloud infrastructure

LATENCY !

Case Study: Amazon Appstream



Use the Amazon AWS infrastructure

Propritery STX protocol allows dynamic adjustment of encoding.

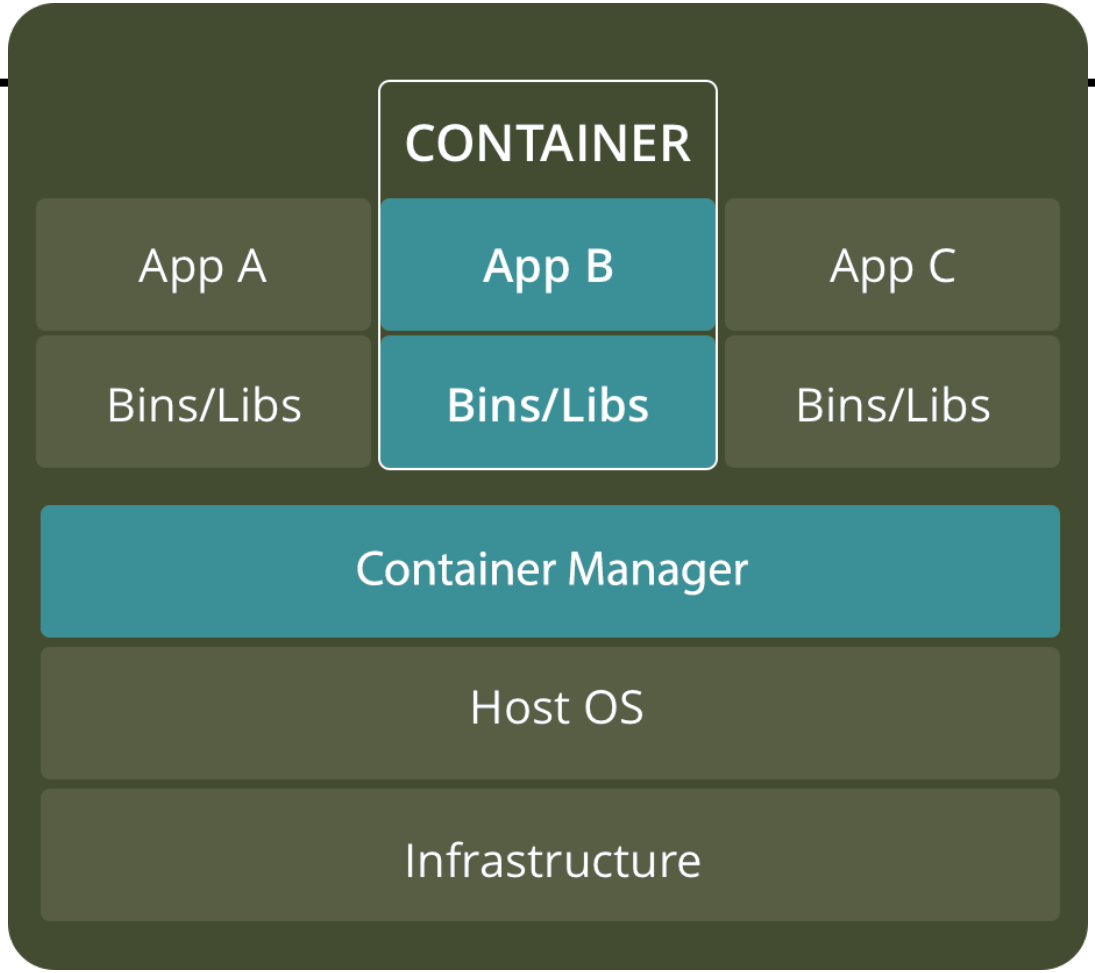
https://www.dropbox.com/scl/fi/8g8sobj5jxo6wmyfs400d/Introduction-to-Amazon-AppStream.mp4?rlkey=bbefe42wq1bba7h5t7au99n0c&dl=0

https://www.dropbox.com/scl/fi/ttm7mkh2cdt78z5599ly4/CCP-Games-Uses-Amazon-AppStream-to-Enable-EVE-Online-Character-Generator.mp4?rlkey=mj5tjifynpx9u526giem7c41g&dl=0

Containers as Virtual Environments

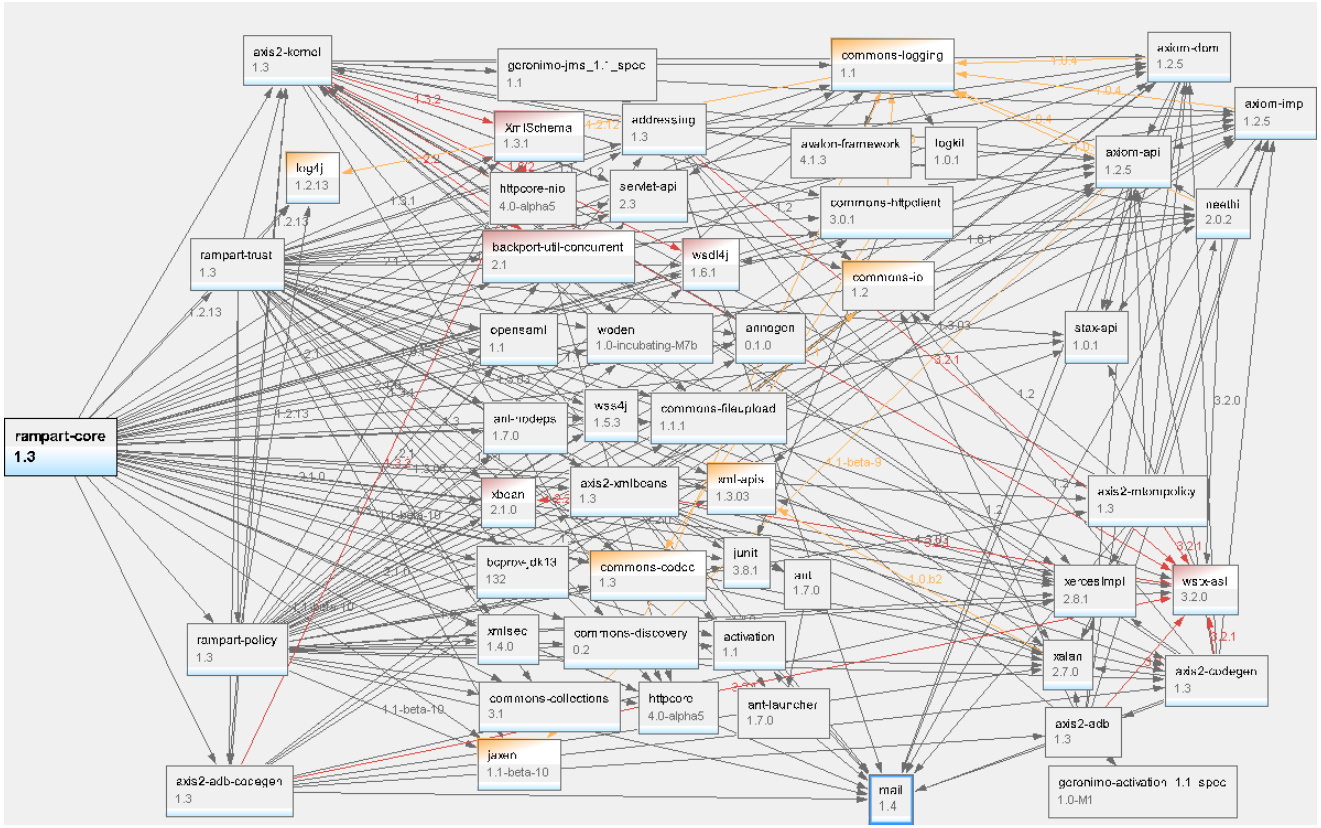
Why ?

- Containers are thinner and lighter weight than traditional VMs
- A portable & rapid way to push application workloads from development, production
- No dependencies on hardware or OS
- Run several workloads on same platform: VM or bare metal.



ref: https://www.backblaze.com/blog/vm-vs-containers/

Bin/Lib Dependencies



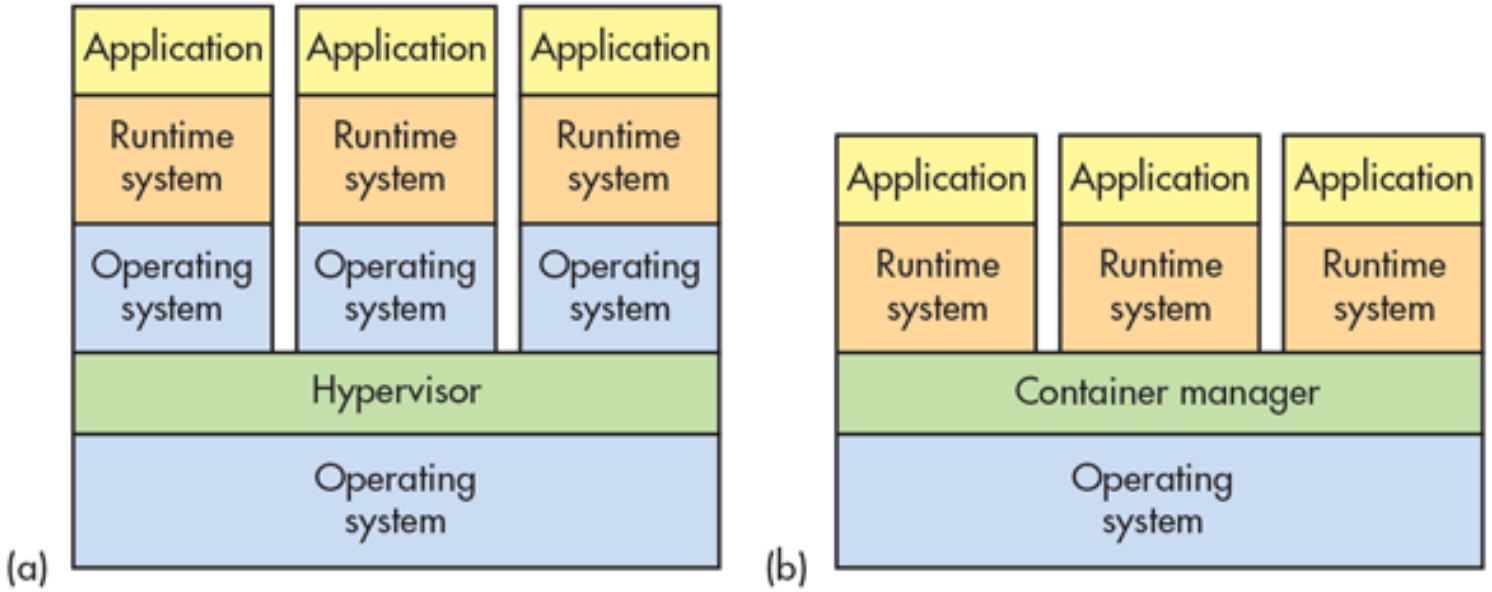
What ?

- Lightweight OS level virtualization method
- Process with isolation,
- Shared resources, and layered filesystems

Containers vs VMs

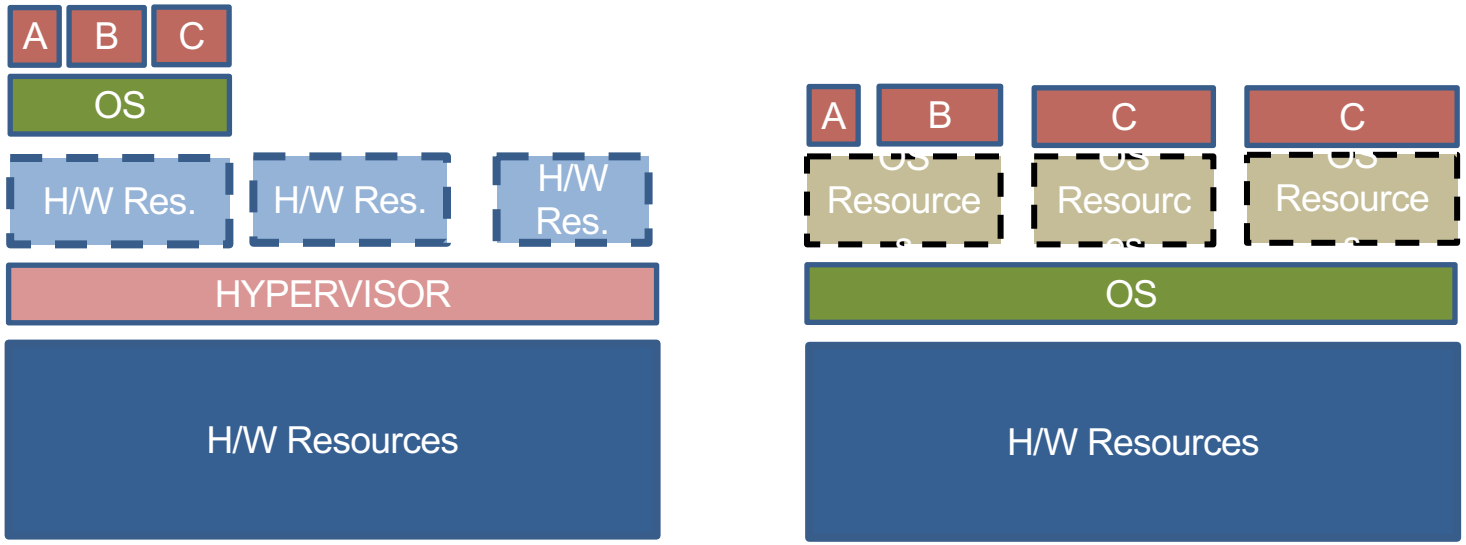
- isolated machines vs isolated applications (processes)
- Applications running in a container environment share an underlying operating system
- Typically a VM will host multiple applications whose mix may change over time versus a container that will normally have a single application. However, it’s possible to have a fixed set of applications in a single container.

Containers vs VMs

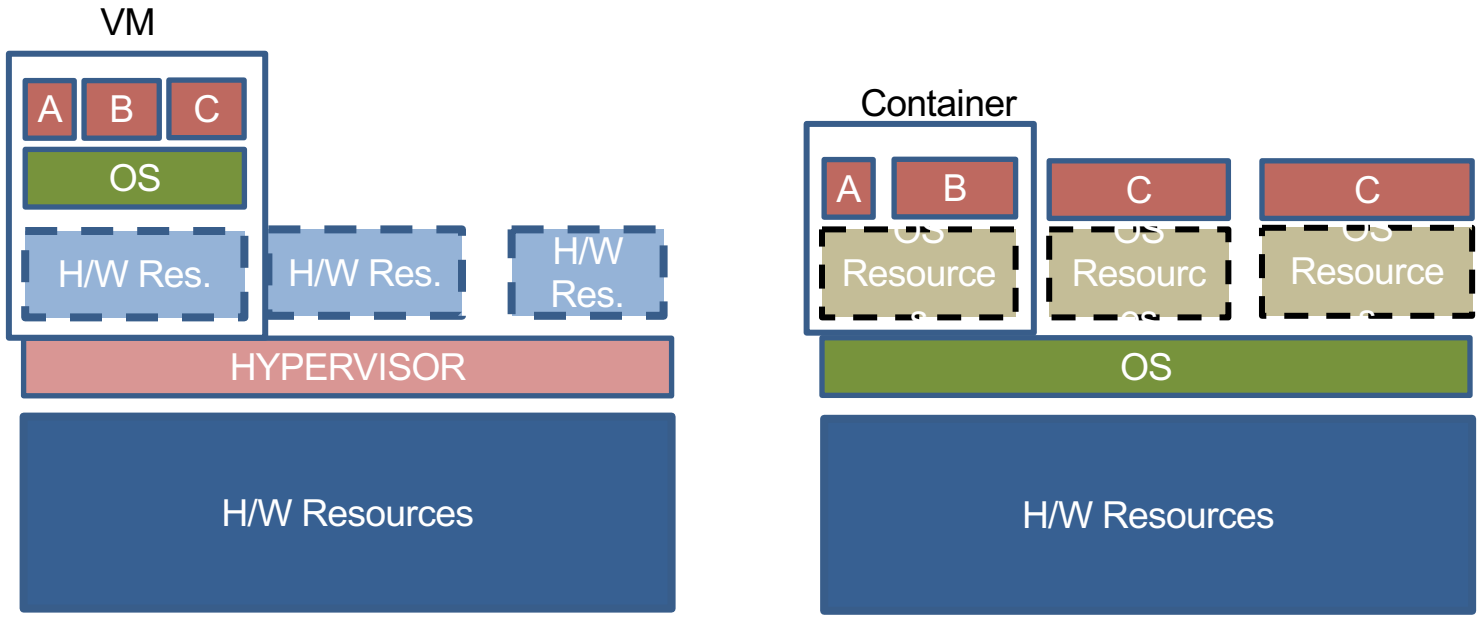


https://www.electronicdesign.com/dev-tools/what-s-difference-between-containers-and-virtual-machines

H/W Level vs OS Level Virtaulization



H/W Level vs OS Level Virtaulization



How ?

- Namespaces
- cgroups
- Layered file systems

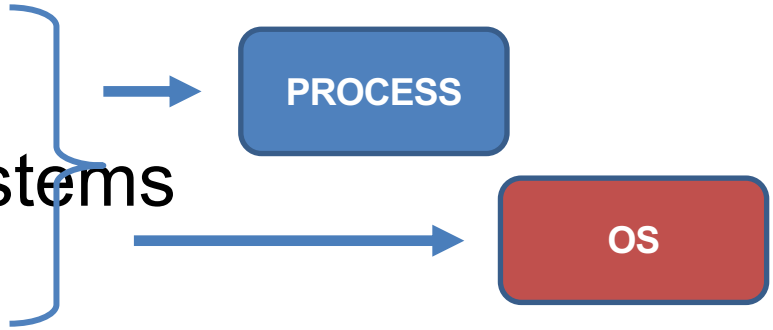
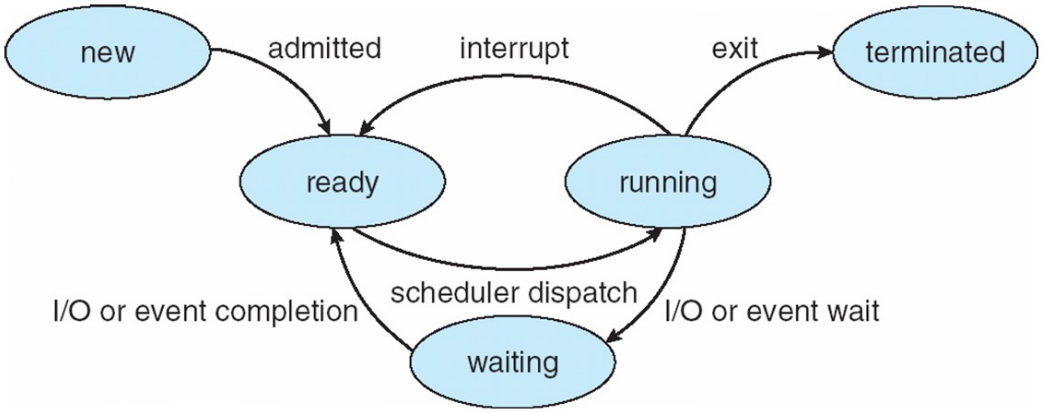
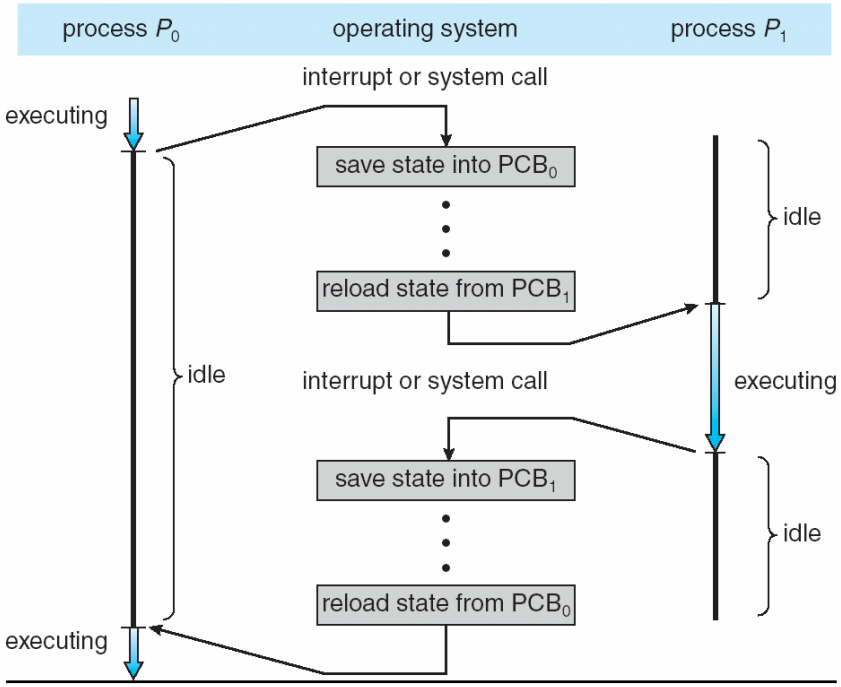


Diagram of Process State



CPU Switch From Process to Process

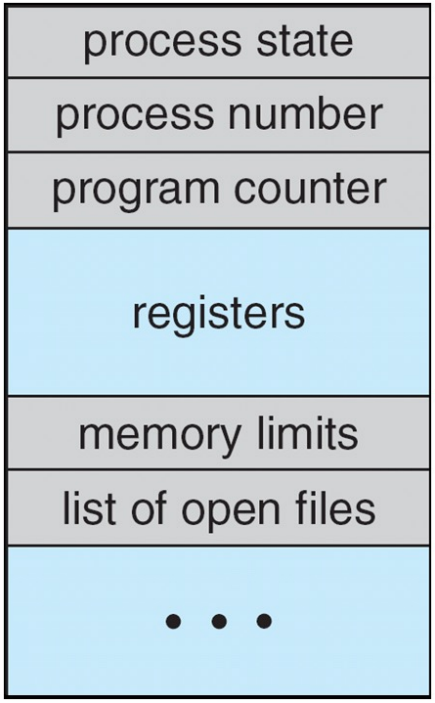


Process Control Block (PCB)

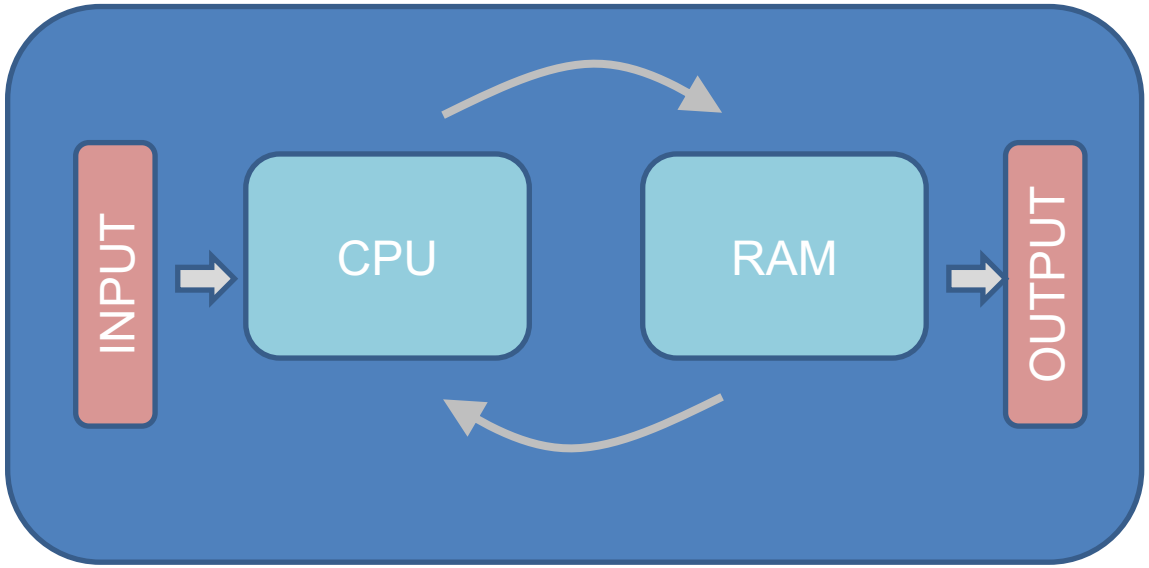
Information associated with each process

(also called **task control block**)

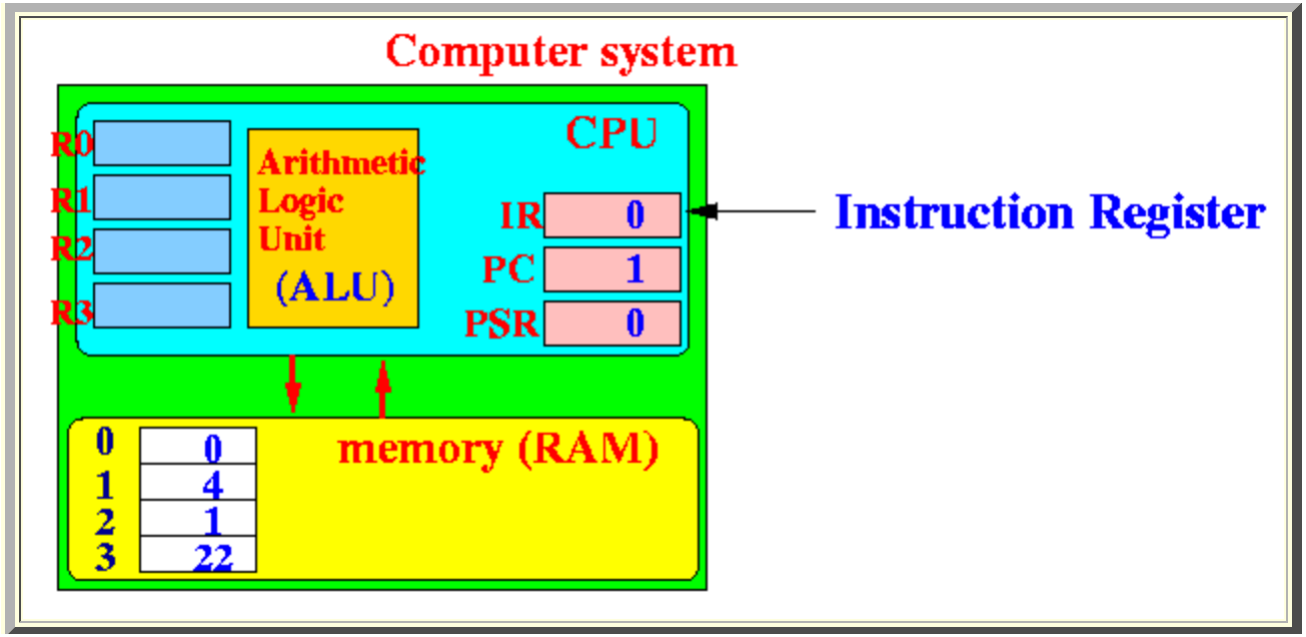
- Process state – running, waiting, etc
- Program counter – location of instruction to next execute
- CPU registers – contents of all process-centric registers
- CPU scheduling information- priorities, scheduling queue pointers
- Memory-management information – memory allocated to the process
- Accounting information – CPU used, clock time elapsed since start, time limits
- I/O status information – I/O devices allocated to process, list of open files



Computation Hardware



Simple Model of CPU & Memory



namespace

namespace: linux kernel feature that isolates and virtualizes system resources

for a collection of processes and their children

- PID: gives process own view of subset of system processes. ✓
- MNT: gives process mount table and allows process to have own filesystem ✓
- NET: gives process own network stack. (Container can have virtual ethernet pairs to link to host or other containers.)
- UTS: gives process own view of system hostname and domain name ✓
- IPC: isolates inter-process communications (i.e. message queues)
- USER: newest namespace that maps process UIDs to different set of UIDs on host (can map containers root uid to unprivileged UID on host)

cgroups

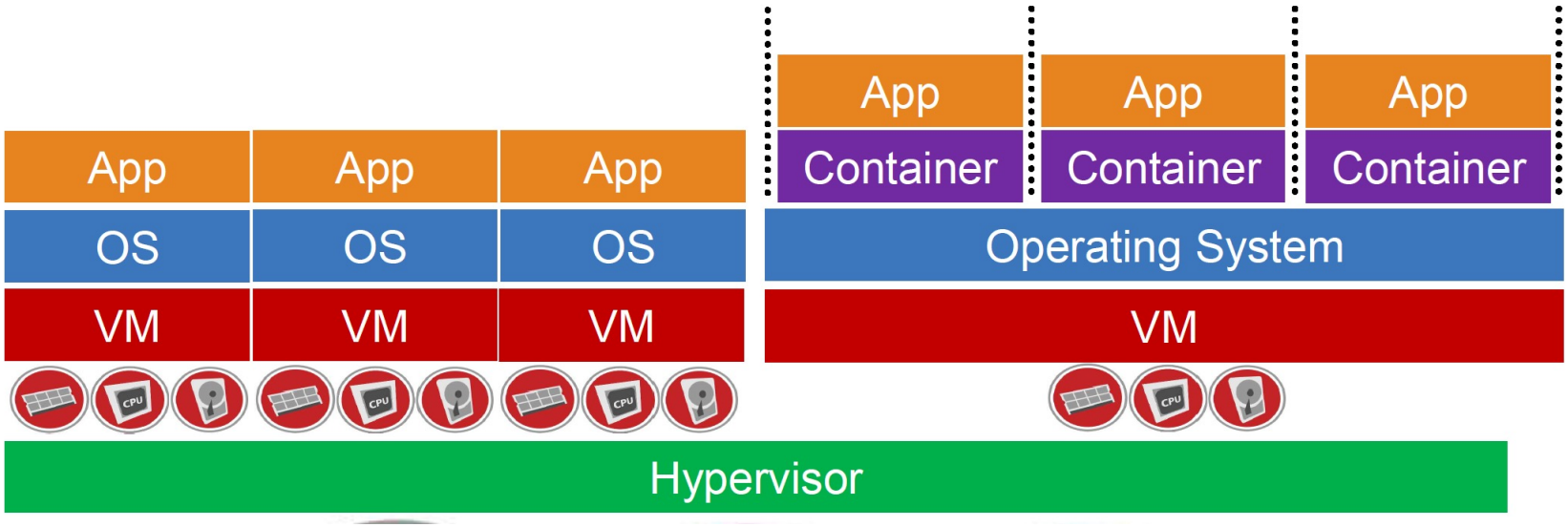
- control groups collect set of process tasks IDS together and apply

limits, such as for resource utilization

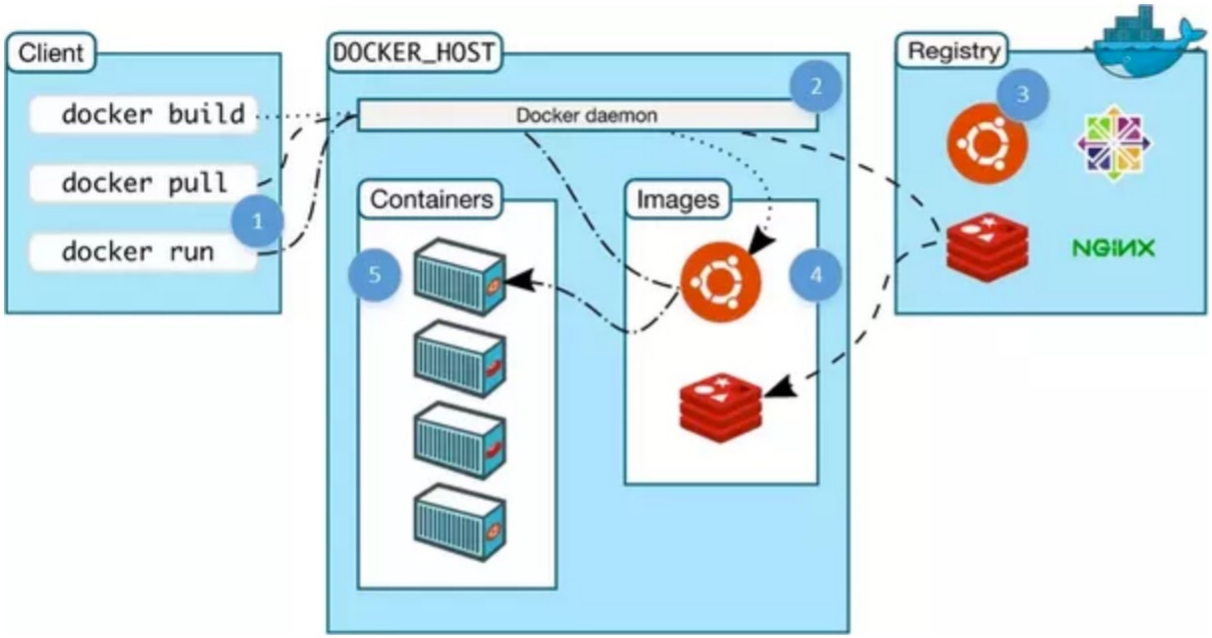
Layered File System

- optimal way to make a copy of root filesystem for each container

Containers + VMs Together



ref: Actual Tech



A command line client (1) tells a process on the machine called the docker daemon (2) what to do. The daemon pulls images from a registry/repository (3). These images are cached (4) on the local machine and can be booted up by the daemon to run containers (5). Image Source: Docker