Virtualization & Cloud Computing

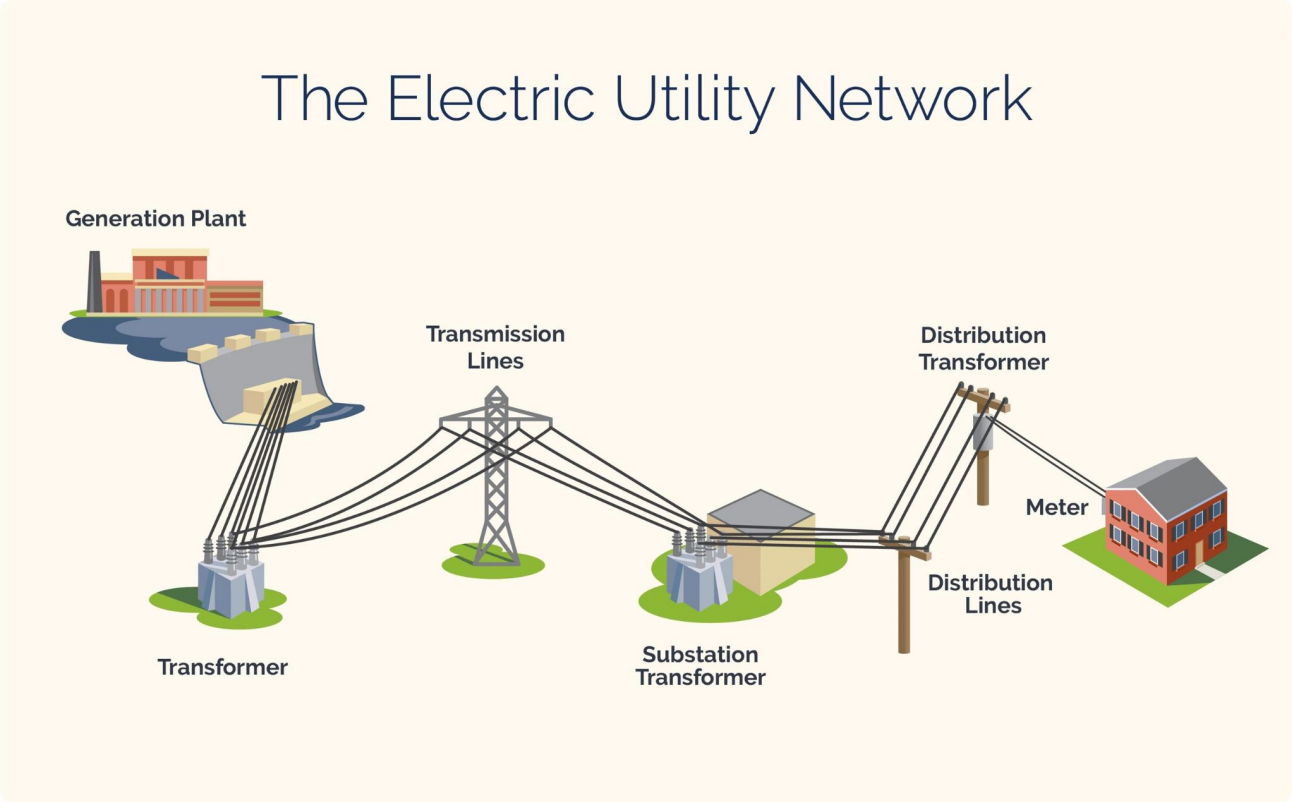
Umesh Bellur
IIT Bombay

Outline

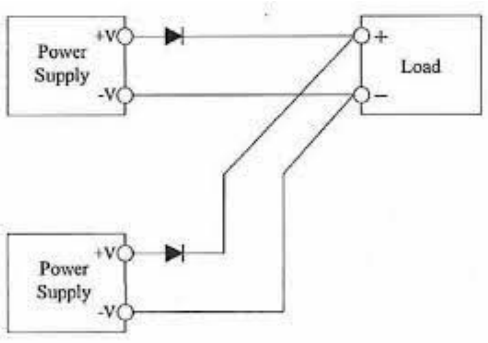
1. Computing as a ***utility***
2. Cloud computing requirements
3. Virtualization fundamentals
4. Beyond virtualization - managing Cloud data centers

Computing as a Utility

What's a ***utility***?

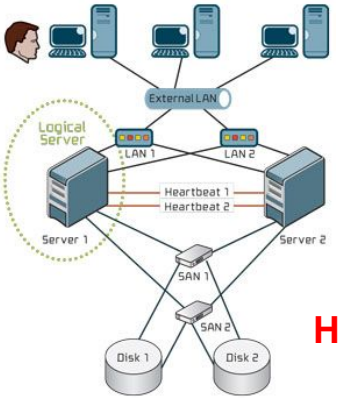


The headaches of running a private DC

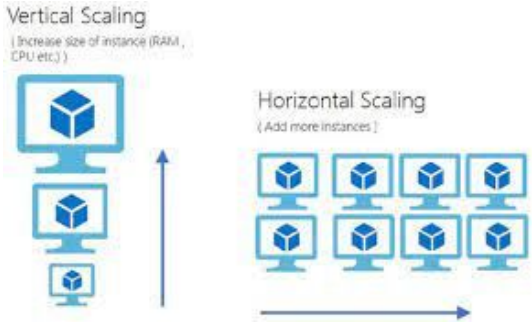


Uninterrupted Power

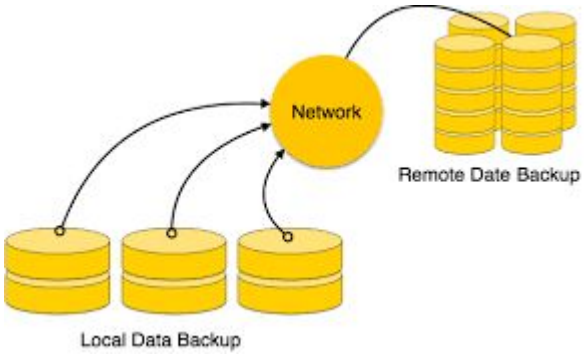
Adequate Cooling



High Availability



Elasticity



Data Governance

Continuum of Utilities

Colocation

Shared Power, Facilities…

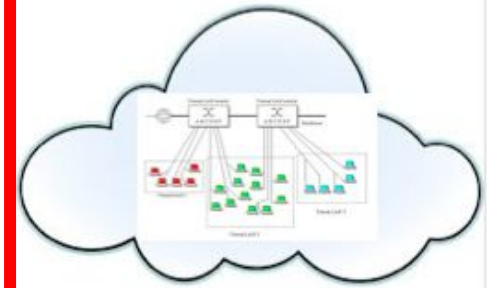
Dedicated Cages

Hosting

Shared Network, Security…

Dedicated Computers

Utility Computing

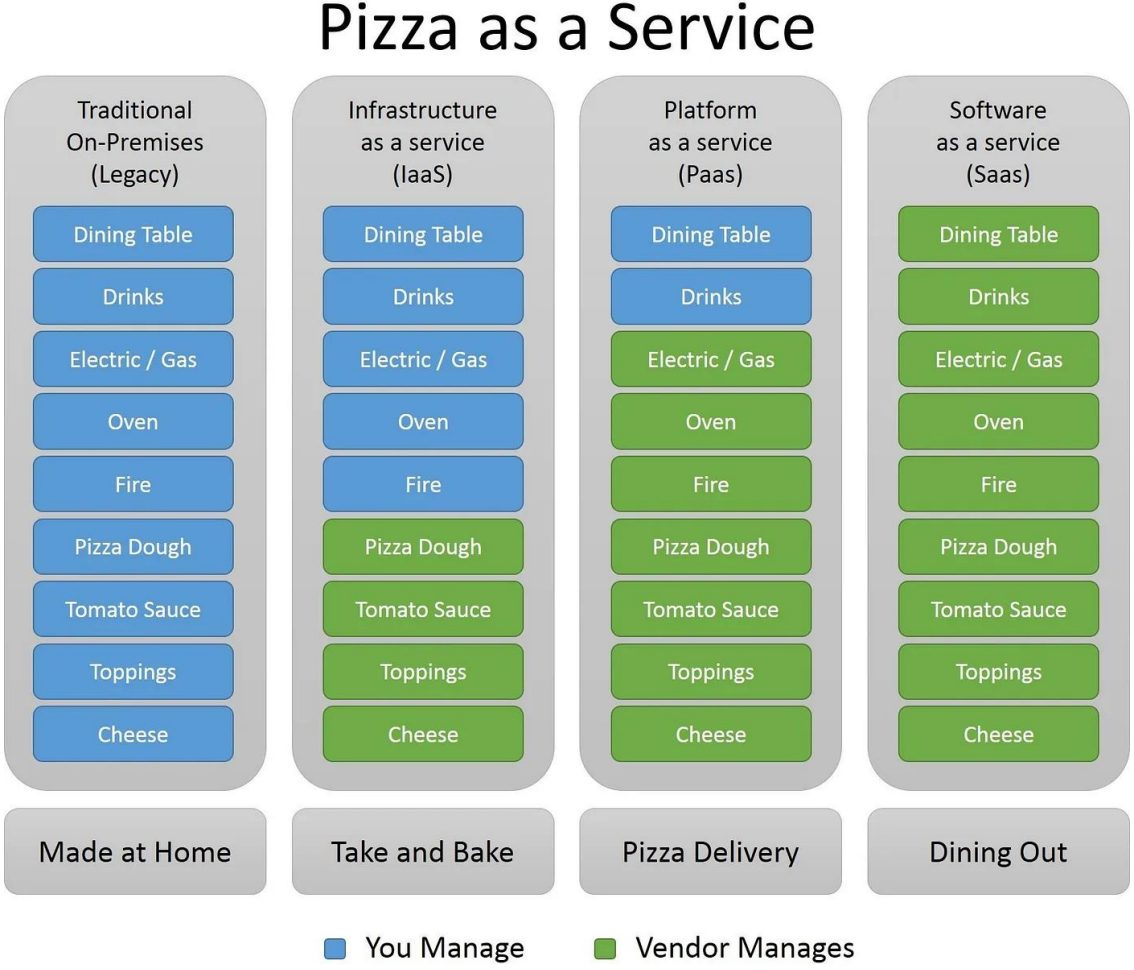


Shared pool of Computers

Isolation via Software

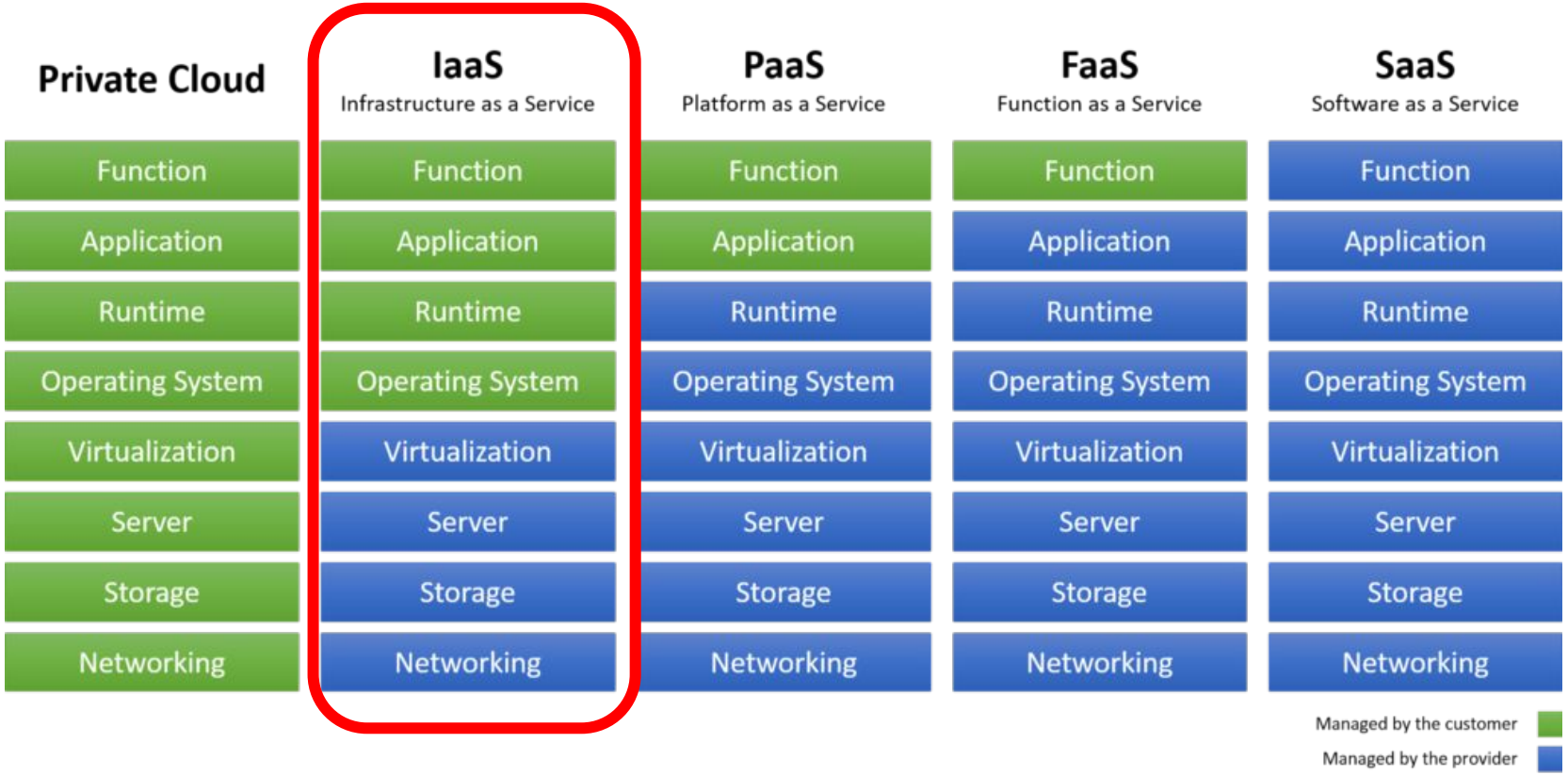
Cloud Computing Models & Requirements

Pizza as a service - variants



Albert Barron, 2014.
https://www.linkedin.com/pulse/20140730172610-9679881-pizza-as-a-service/

The Cloud Continuum



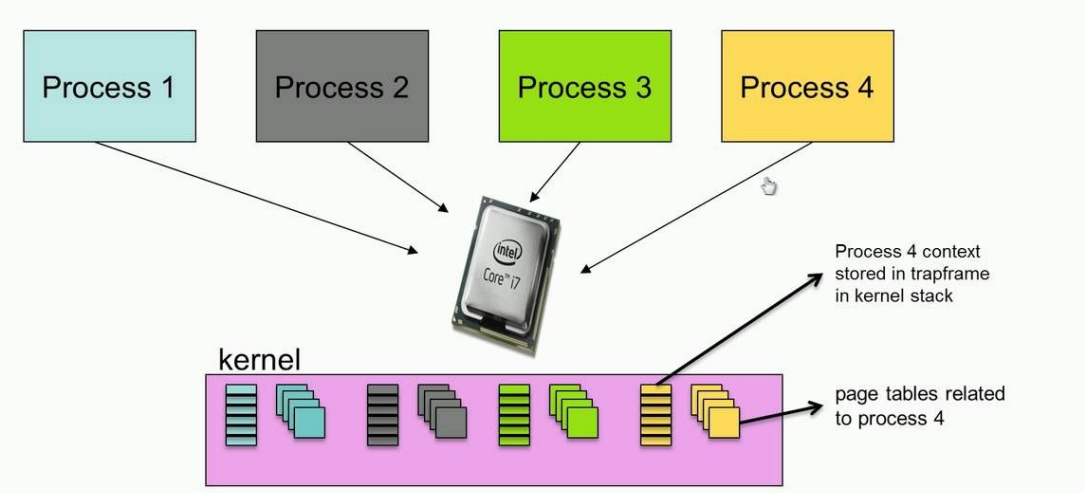
Requirements of IaaS Clouds

- ***Provision*** and manage infrastructure ***over a network***
- ***Elastic*** - can scale both up and down ***horizontally***
- ***Variable sized*** (virtual) machines
- Ability to dynamically resize VMs - ***scale up and down vertically***.
- ***Security*** - only the user should be able to access their infrastructure
- Always on and ***highly available*** infrastructure
- ***Use/Time based metering*** and billing

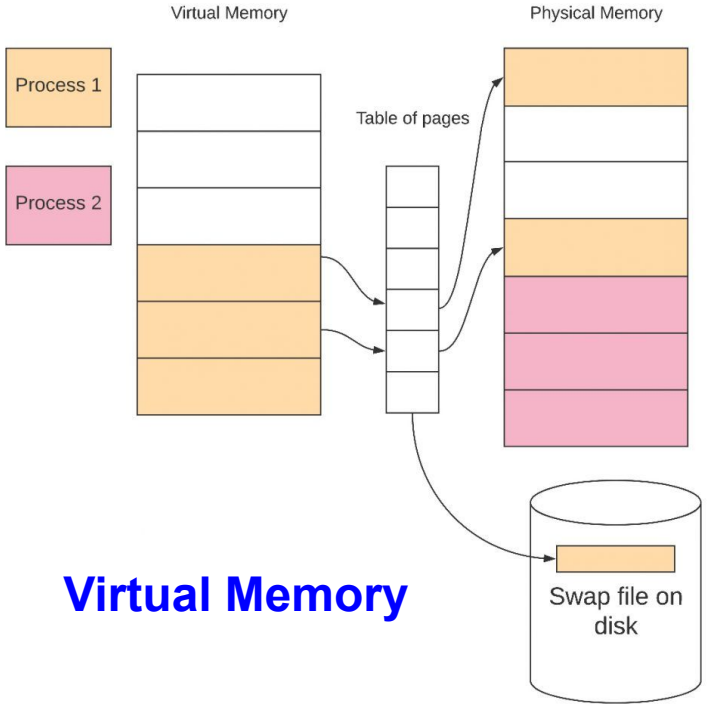
The key to meeting these requirements is virtualization

Virtualization Fundamentals

Virtualization - a renaissance of sorts



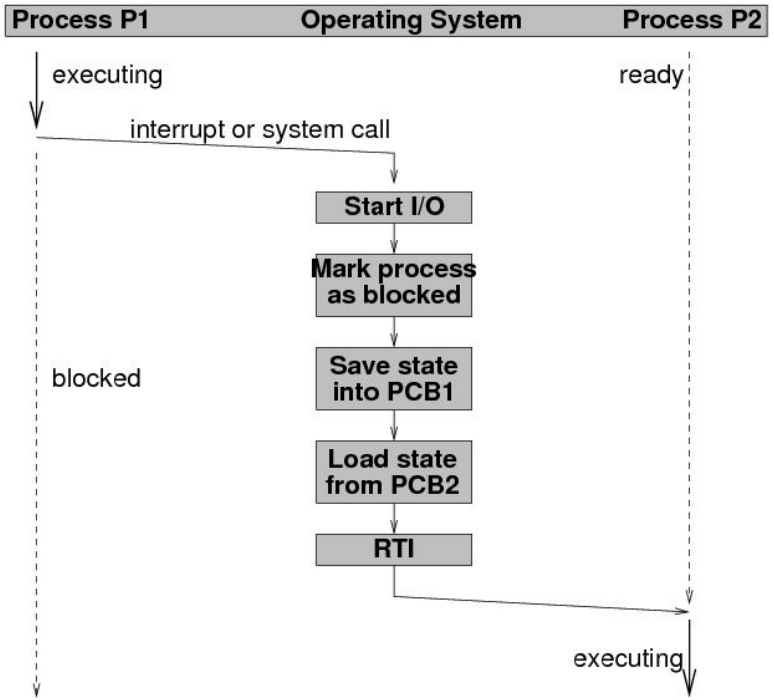
Process Virtualization



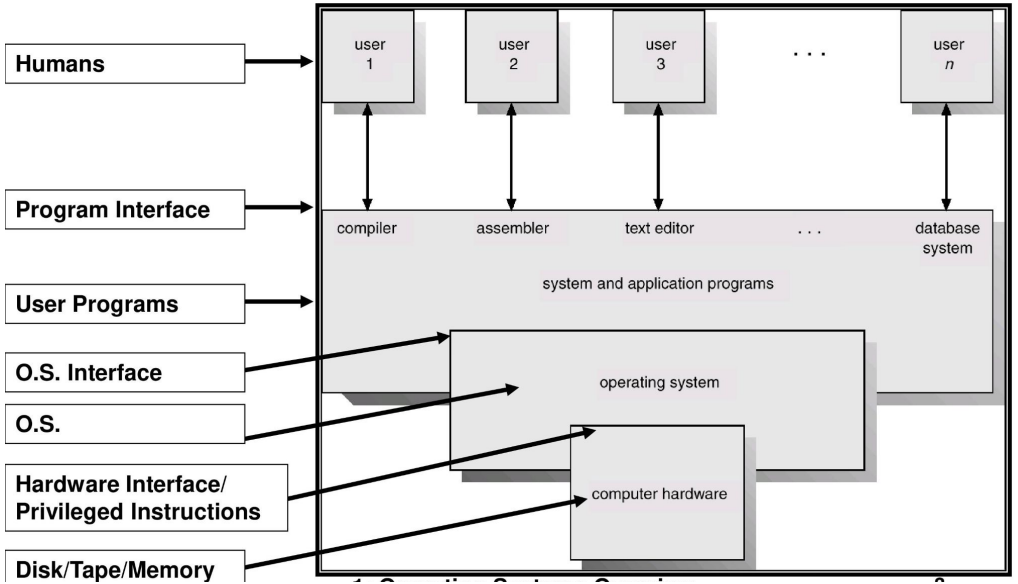
Virtual Memory

Relevant OS Mechanisms

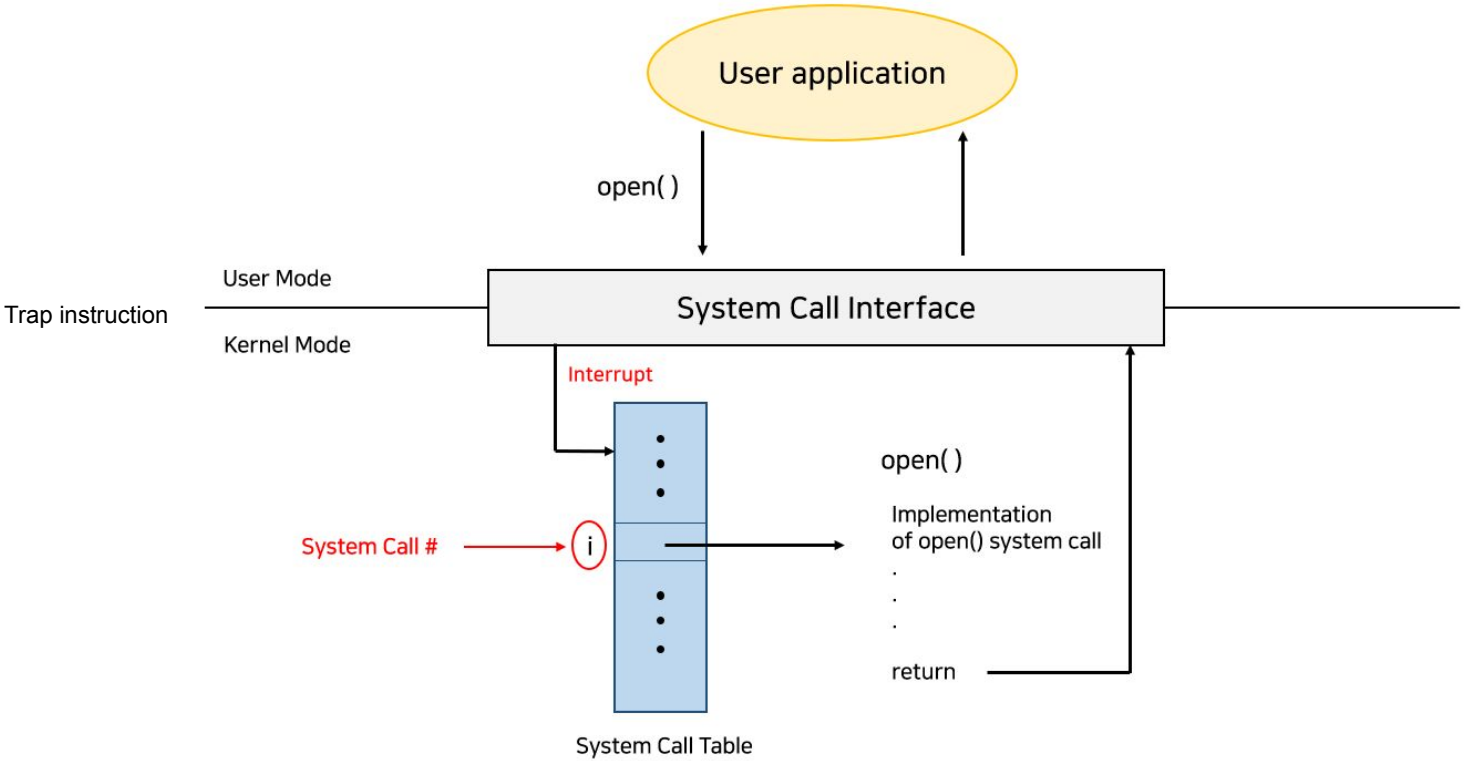
Context Switching



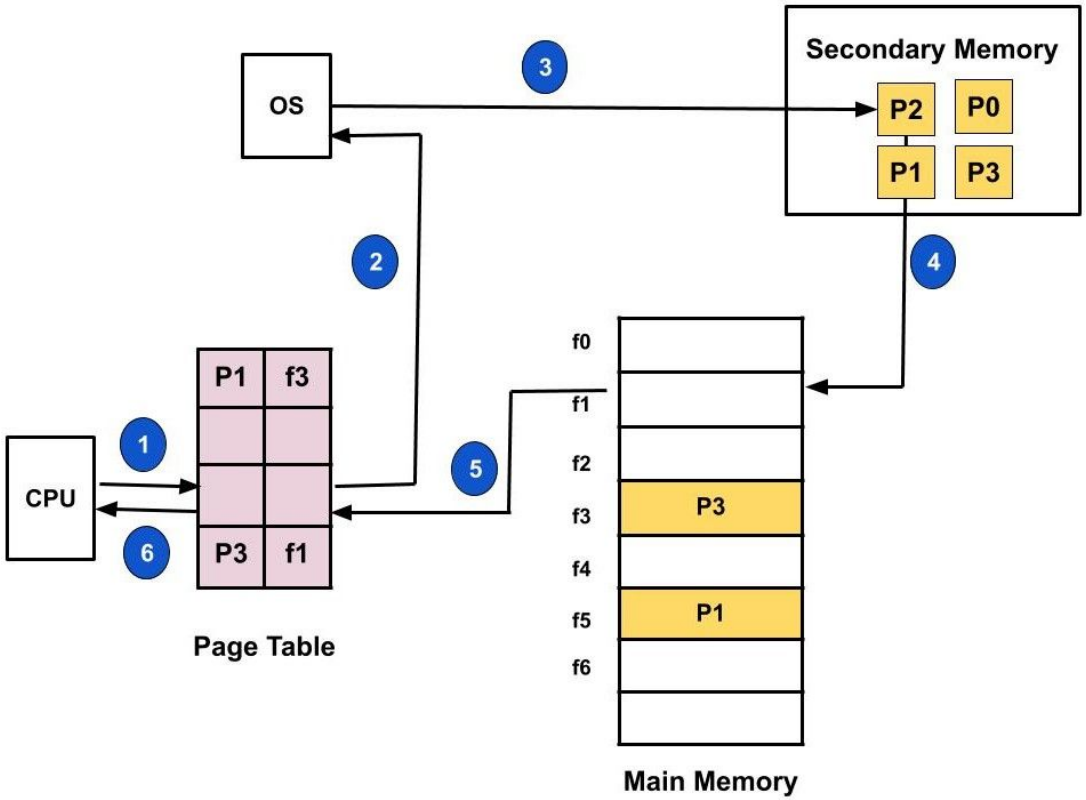
Privileged Instructions



Escalating Privilege - the system call

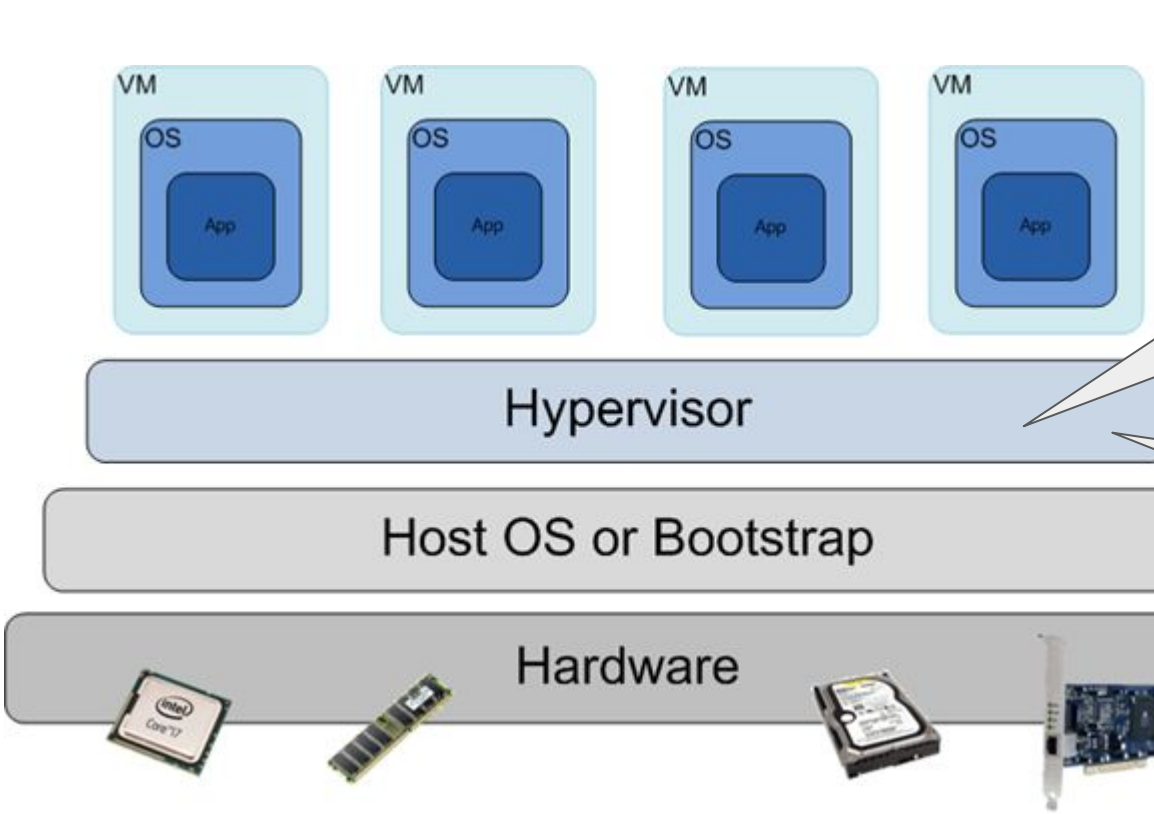


Demand Paging with virtual memory



1. CPU issues a VA
2. If not in the Page Table, the page handler steps in.
3. Fetches the relevant page from secondary memory
4. Inserts into Physical memory (evicting a page from PM if needed)
5. The page table is updated with a Virtual page to Physical page mapping.
6. The contents of the physical address corresponding to the VA is returned to the CPU.

Realizing IaaS - Virtualization as a building block



An OS for other OSs -
manages interaction between the VM and hardware

Machine Switch similar to Context Switch

Design conditions for VMMs (Popek and Goldberg, 1974)

- ***Equivalence***
  - Software on the VMM executes identically to that on the hardware barring performance.
- ***Performance***
  - Non-privileged instructions can be executed on the physical processor with no VMM intervention.
- ***Resource Control***
  - The VMM must have complete control over virtualized resources.

Booting a new OS & Machine Switching

- **Booting** a new OS - limited direct execution
  - Jump to the first instruction of the boot sequence and continue from there.
- **Machine switching** - what is the challenge?
  - The Guest OS expects to have full access to hardware and will execute privileged instructions, unlike user processes - example: updating the TLB after a miss.
  - But one guest alone must not get full access since there are other guests at the same level as it.
  - The VMM must therefore intercept attempts to perform “privileged” operations and thus retain control of the machine.

How to execute a system call on a virtualized OS?

- Similar “trap” instruction - except that the VMM has installed the trap handler that will execute in the kernel mode
- But…. the VMM does not know anything about HOW to execute the system call - that is in the Guest OS.
- But the VMM knows where the OS’s trap handler is - it got this information when the OS was installing its trap handlers at boot time (that is privileged as well).
- The VMM therefore jumps to the Guest OS’s trap handler
- When the OS is done it again executes a privileged instruction to “return” from the trap handler and so jumps back into the VMM
- The VMM then executes the real trap return.

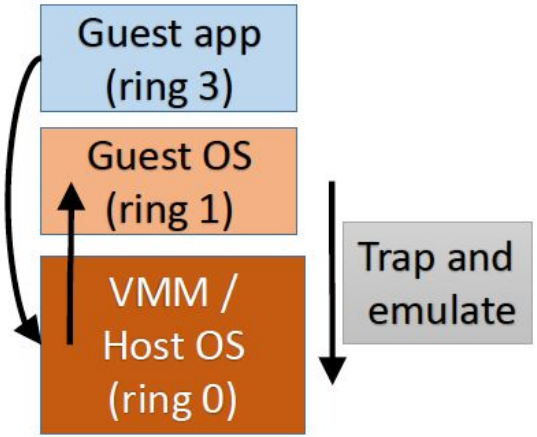


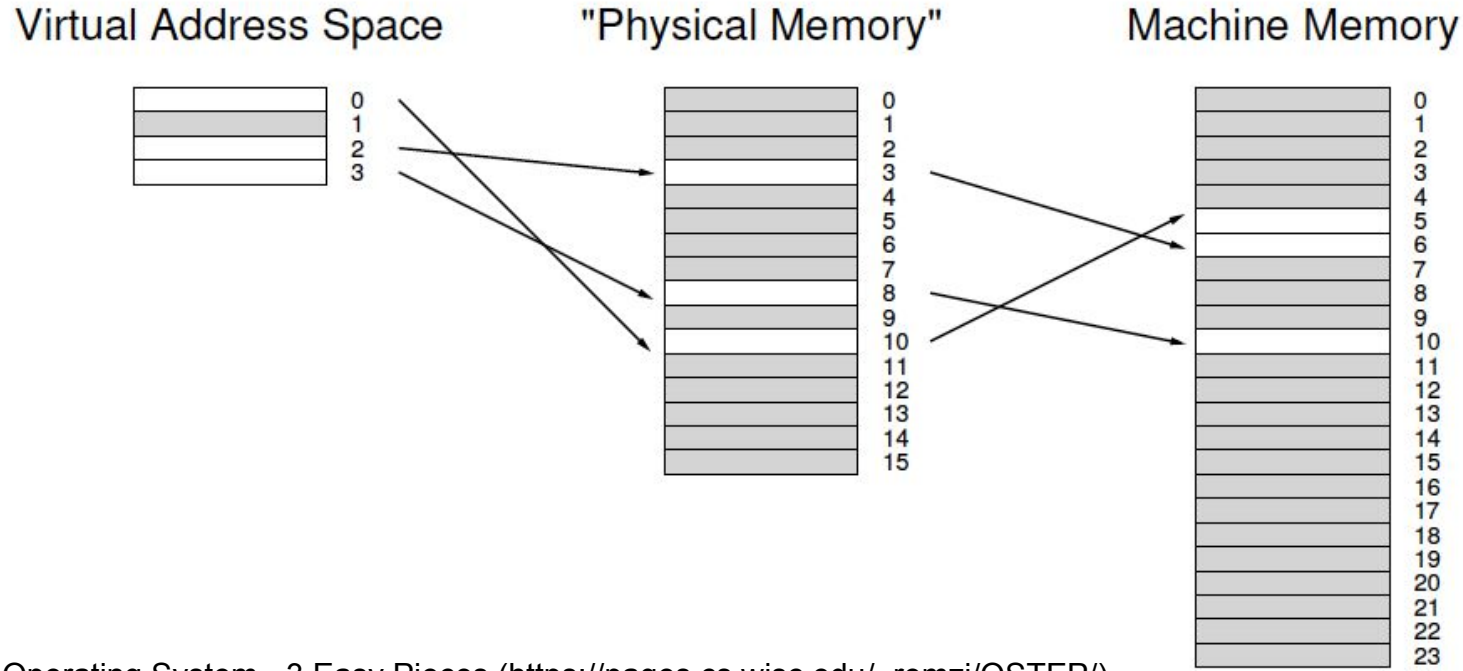
Figure courtesy Prof. Mythili

Process	Operating System	VMM
1. System call: Trap to OS		
		2. Process trapped: Call OS trap handler (at reduced privilege)
	3. OS trap handler: Decode trap and execute syscall; When done: issue return-from-trap	
		4. OS tried return from trap: Do real return from trap
5. Resume execution (@PC after trap)		

Memory Virtualization in the presence of Guest OSs

OS Page Table
VPN 0 to PFN 10
VPN 2 to PFN 03
VPN 3 to PFN 08

VMM Page Table
PFN 03 to MFN 06
PFN 08 to MFN 10
PFN 10 to MFN 05



Handling a TLB Miss

Process	Operating System	Virtual Machine Monitor
1. Load from mem TLB miss: Trap		
		2. VMM TLB miss handler: Call into OS TLB handler (reducing privilege)
	3. OS TLB miss handler: Extract VPN from VA; Do page table lookup; If present and valid, get PFN, update TLB	
		4. Trap handler: Unprivileged code trying to update the TLB; OS is trying to install VPN-to-PFN mapping; Update TLB instead with VPN-to-MFN (privileged); Jump back to OS (reducing privilege)
	5. Return from trap	
		6. Trap handler: Unprivileged code trying to return from a trap; Return from trap
7. Resume execution (@PC of instruction); Instruction is retried; Results in TLB hit		

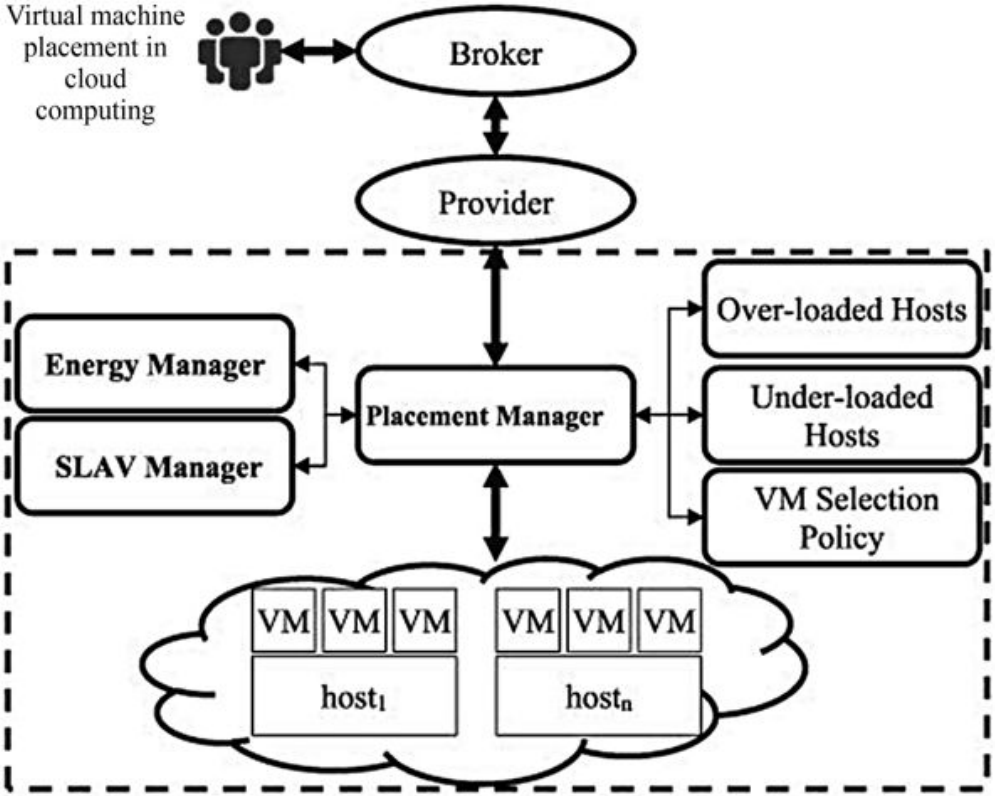
Operating System - 3 Easy Pieces (https://pages.cs.wisc.edu/~remzi/OSTEP/)

Beyond Virtualization: Provider Concerns

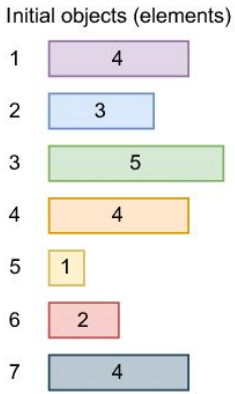
Important considerations in a Cloud Data Center

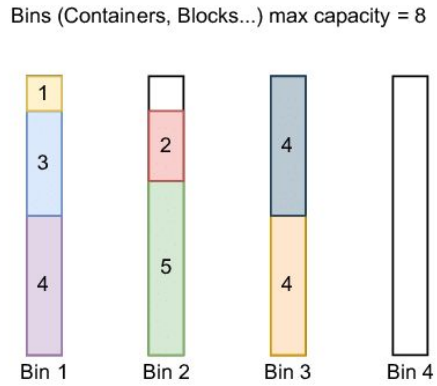
- Cloud provider **concerns**
  - Efficient management of resources
  - Minimize power and cooling costs
  - Meet customer SLAs via VM Performance.
- **Strategies** to achieve these goals
  - Server consolidation
  - Managed over provisioning
  - Hot spot mitigation
- **Mechanisms** to implement these strategies
  - VM Placement
  - Memory Ballooning
  - VM Migration

VM Placement

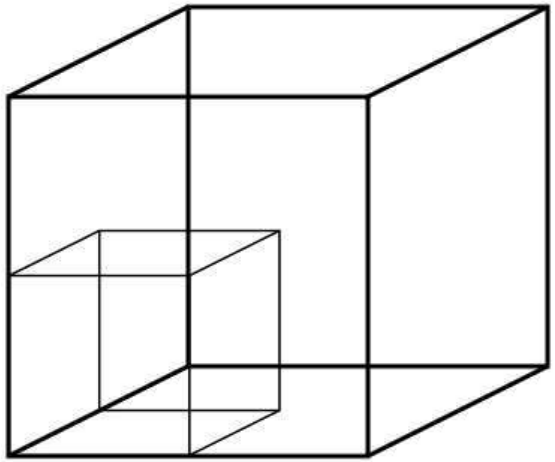


The abstraction - bin packing





Vector bin packing



$$\text{minimize}: \sum_j y_j \quad \text{s.t.} \tag{1}$$

$$\sum_j x_{ij} = 1 \qquad 1 \le i \le n \tag{2}$$

$$\sum_i p_i^k . x_{ij} \le 1 \qquad 1 \le j \le m, 1 \le k \le d \tag{3}$$

$$y_j \ge x_{ij} \qquad 1 \le i \le n, 1 \le j \le m \tag{4}$$

$$x_{ij} \in \{0, 1\} \qquad 1 \le i \le n, 1 \le j \le m \tag{5}$$

- Constraint (2) states that every vector is packed in a bin.
- Constraint (3) ensures that the packed vectors do not exceed the bin dimensions.
- Constraint (4) tells whether a bin is used or not.
- Constraint (5) ensures that a vector is either packed entirely in a bin or not.

Constraint (5) can be relaxed as follows to obtain a linear program (LP).

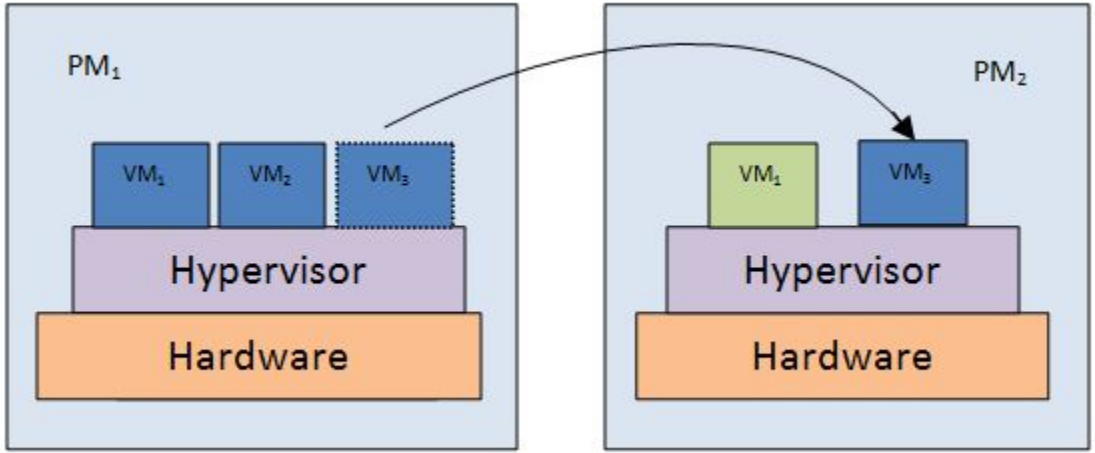
$$x_{ij} \ge 0 \qquad 1 \le i \le n, \quad 1 \le j \le m \tag{5a}$$

Bellur, Umesh, Chetan S. Rao, and Madhu Kumar SD. "Optimal placement algorithms for virtual machines." *arXiv preprint arXiv:1011.5064* (2010).

A few heuristics

- **First Fit:** It is a greedy approach where the placement mgr considers the PMs sequentially, one by one, and places the VM to the first PM that has enough resources.
- **Next Fit:** This placement method considers the PMs one by one and places the VM to the second PM which has the required resources.
- **Random Fit:** A random physical machine is chosen for placing the VM.
- **Least full first:** The physical machine which is least full and satisfies the resource requirement of the VM selected.
- **Most Full First:** The physical machine which is most full and has the resource requirement of VM is selected.

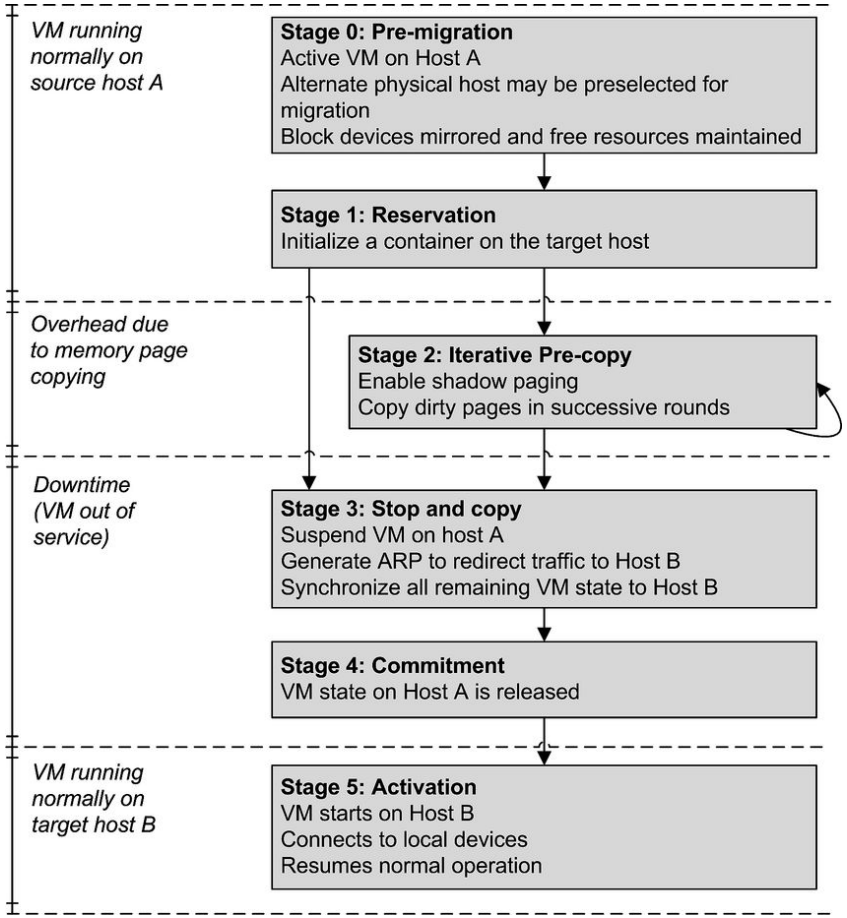
VM (live)Migration



Motivations:

- Server Consolidation
- Hot spot mitigation
- Energy Savings

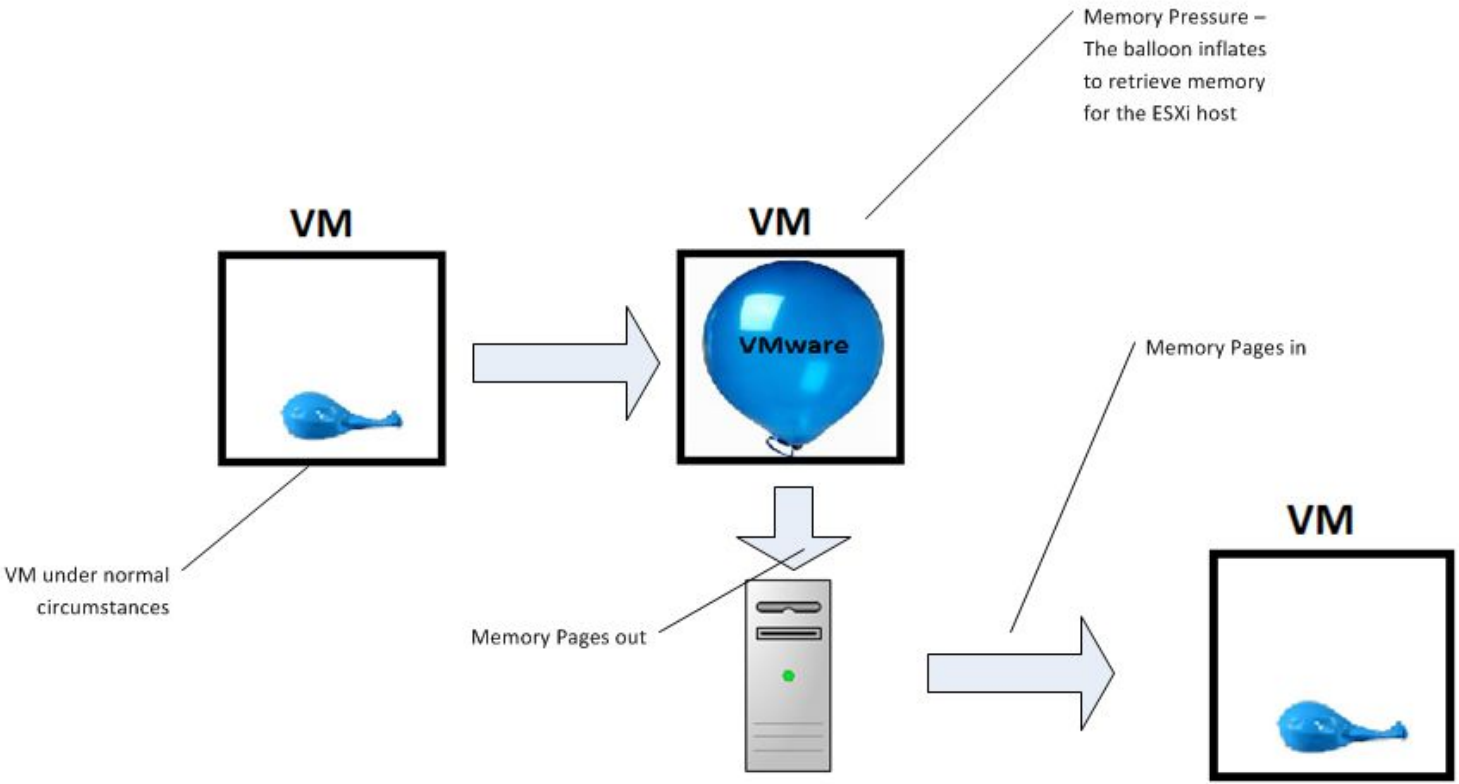
VM Migration - Iterative pre-copy mechanism



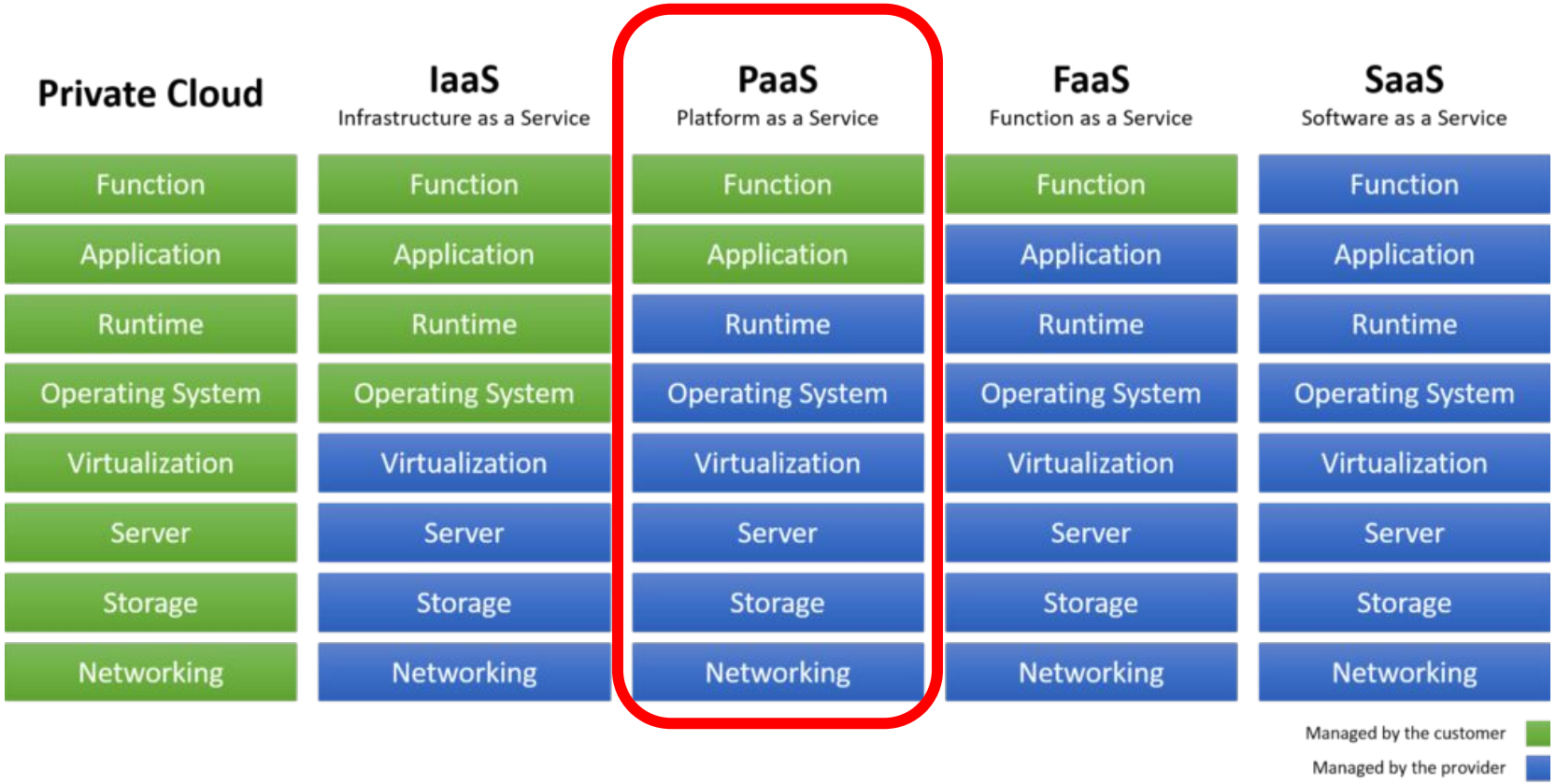
Some questions of interest in VM Migration

- Which VM to migrate?
- Where to migrate it to?
- What is the cost of migration?
  - Downtime
  - CPU at source and destination
  - Network bytes consumed
  - Impact to other applications that are NOT being migrated

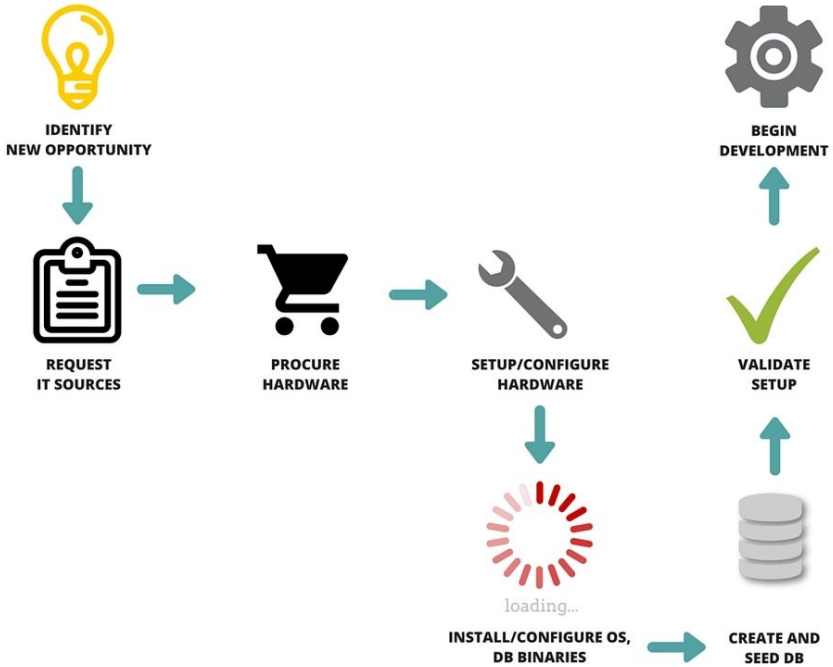
Over provisioning - memory ballooning



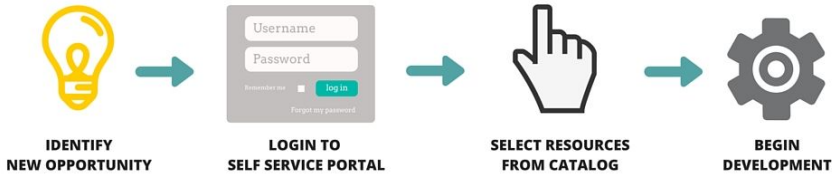
The Cloud Continuum - Up the value chain



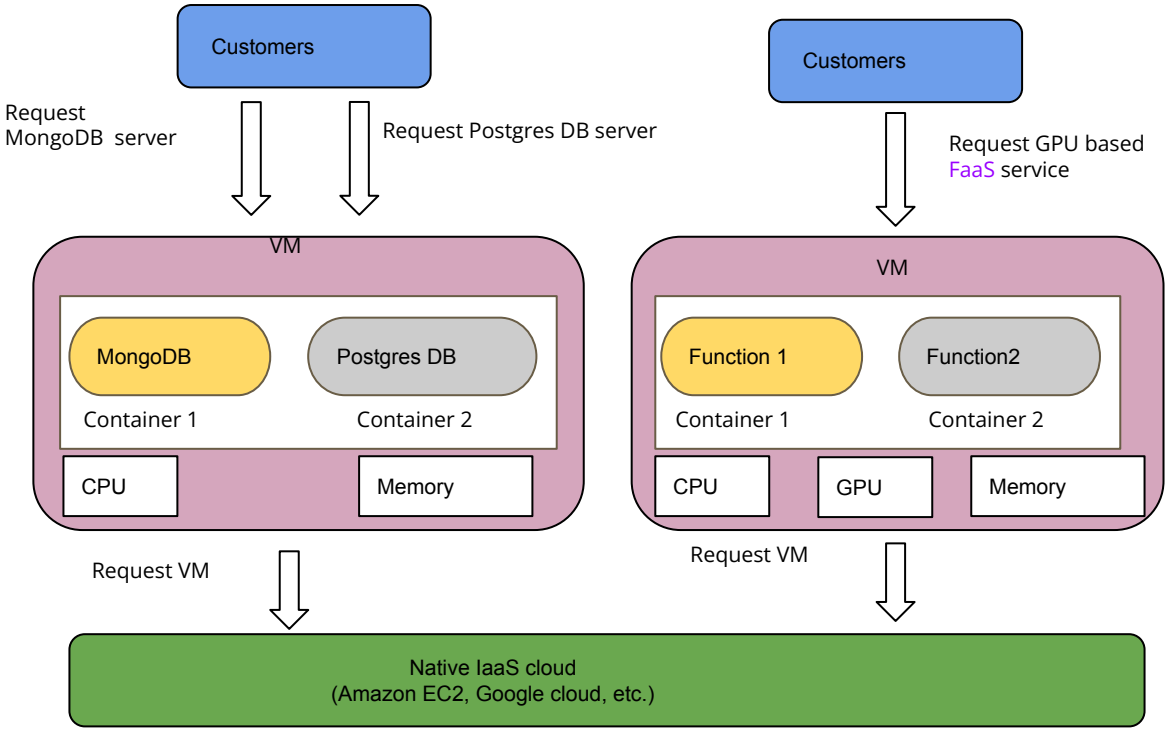
The traditional way



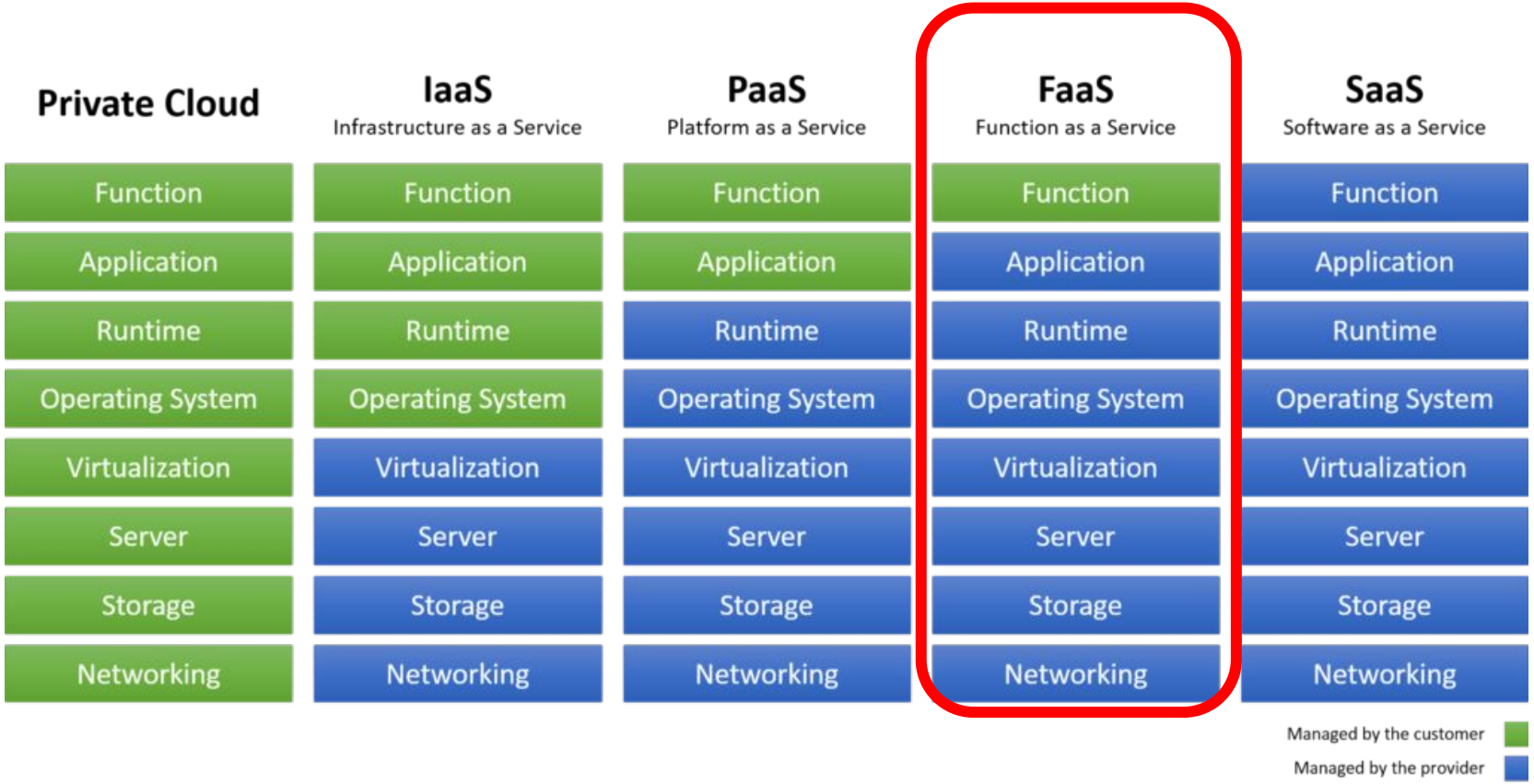
The PaaS way



Implementing PaaS - Derivative clouds with Containerisation

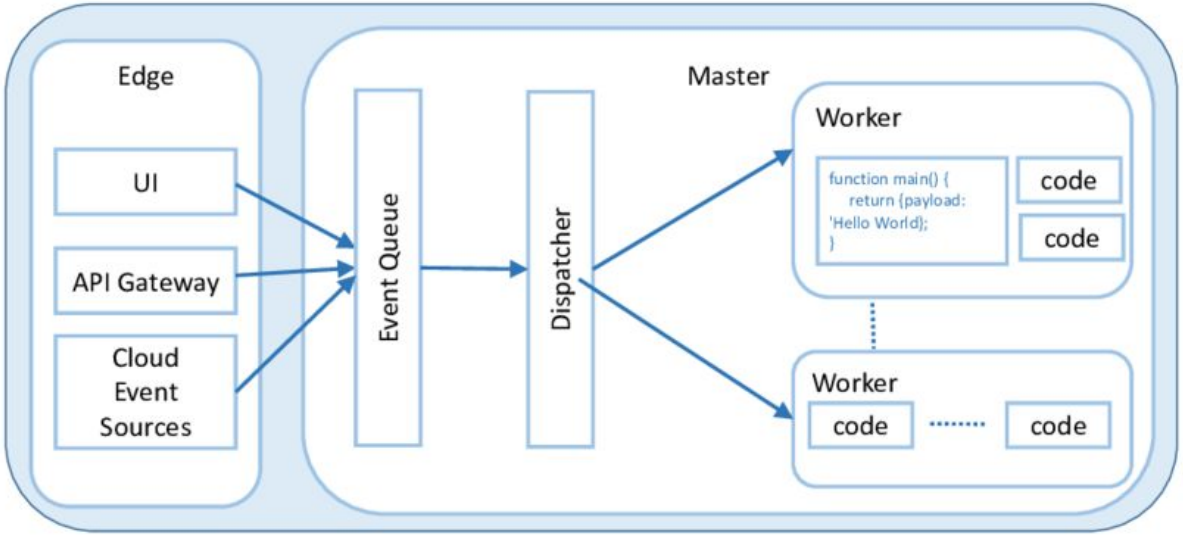


The Cloud Continuum - Up the value chain



FaaS/Serverless - Pay as you use

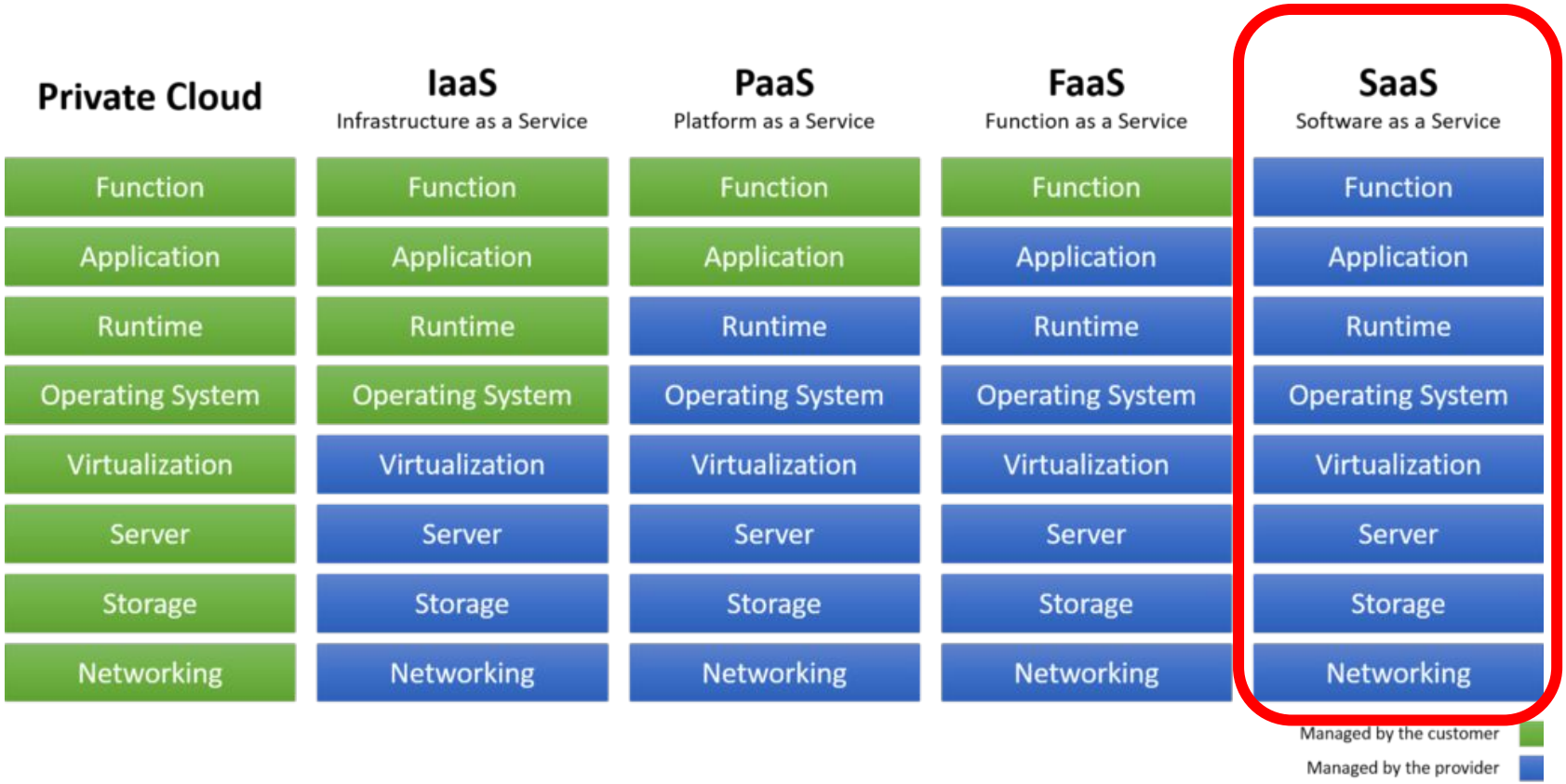
Tech stack for FaaS



Knative

APACHE OpenWhisk

The Cloud Continuum - Up the value chain



Conclusions

- ***Cloud computing is here to stay*** - all version of it appear to be popular and in use.
  - Most enterprises today are operating at least partially out of the cloud
- ***Intersection with new hardware*** capabilities is ongoing
  - SmartNICs
  - GPUs
  - Encryption/Decryption and other accelerators
  - Custom ASICs
- Other research topics include ***Multi-cloud (Cloud Hypervisors)***
- ***Policy*** around the use of data storage/movement and clouds is in the works.