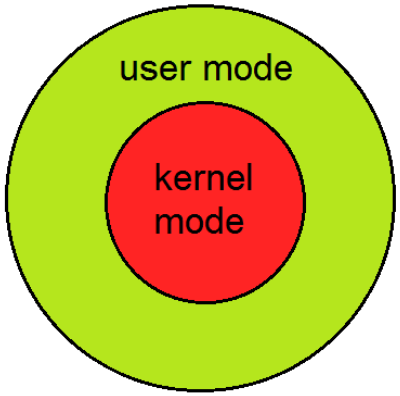
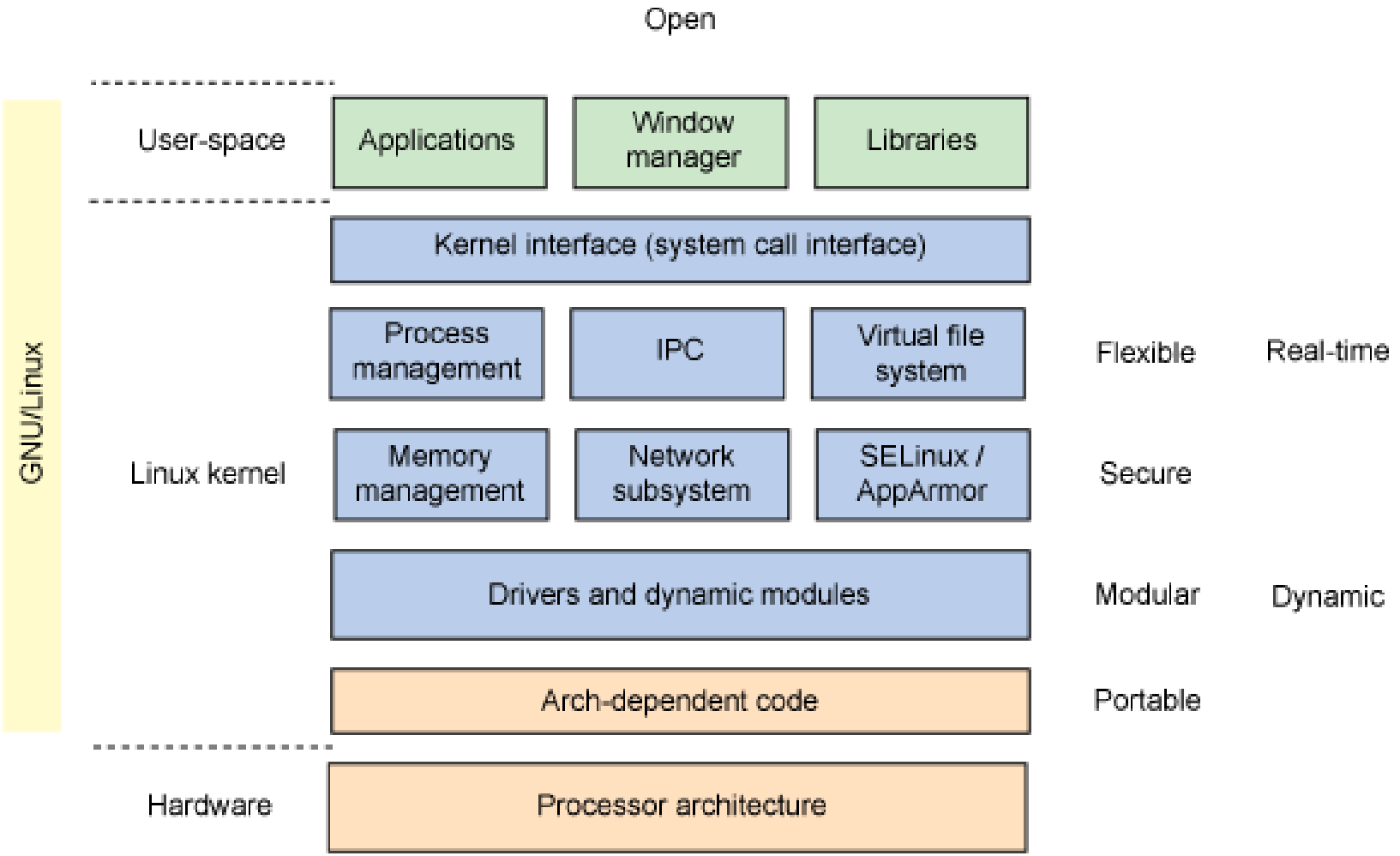
Operating System Virtualization

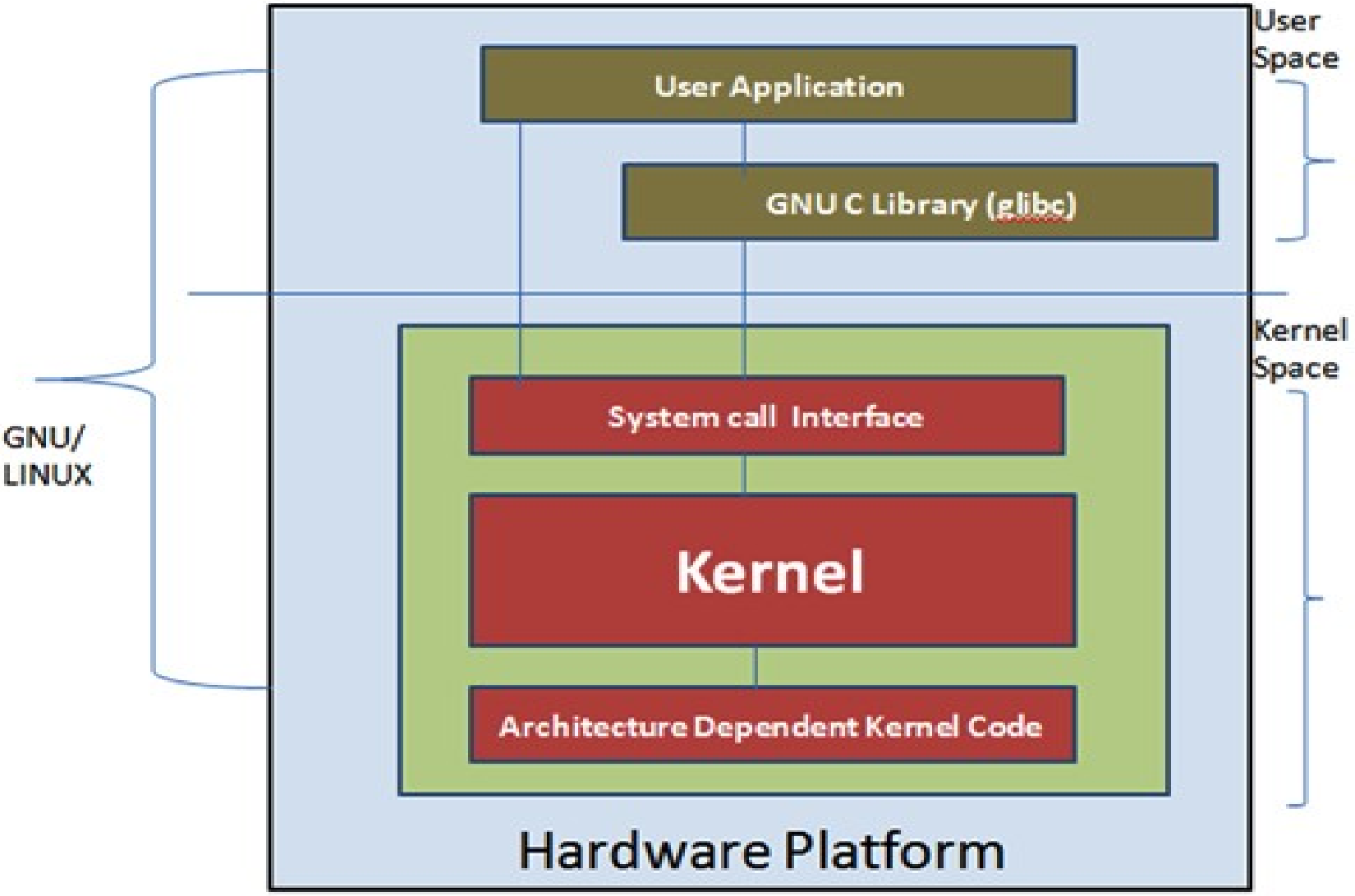
Dr. Bibhudatta Sahoo
Communication & Computing Group
Department of CSE, NIT Rourkela
Email: bdsahu@nitrkl.ac.in, 9937324437, 9337938766,2462358



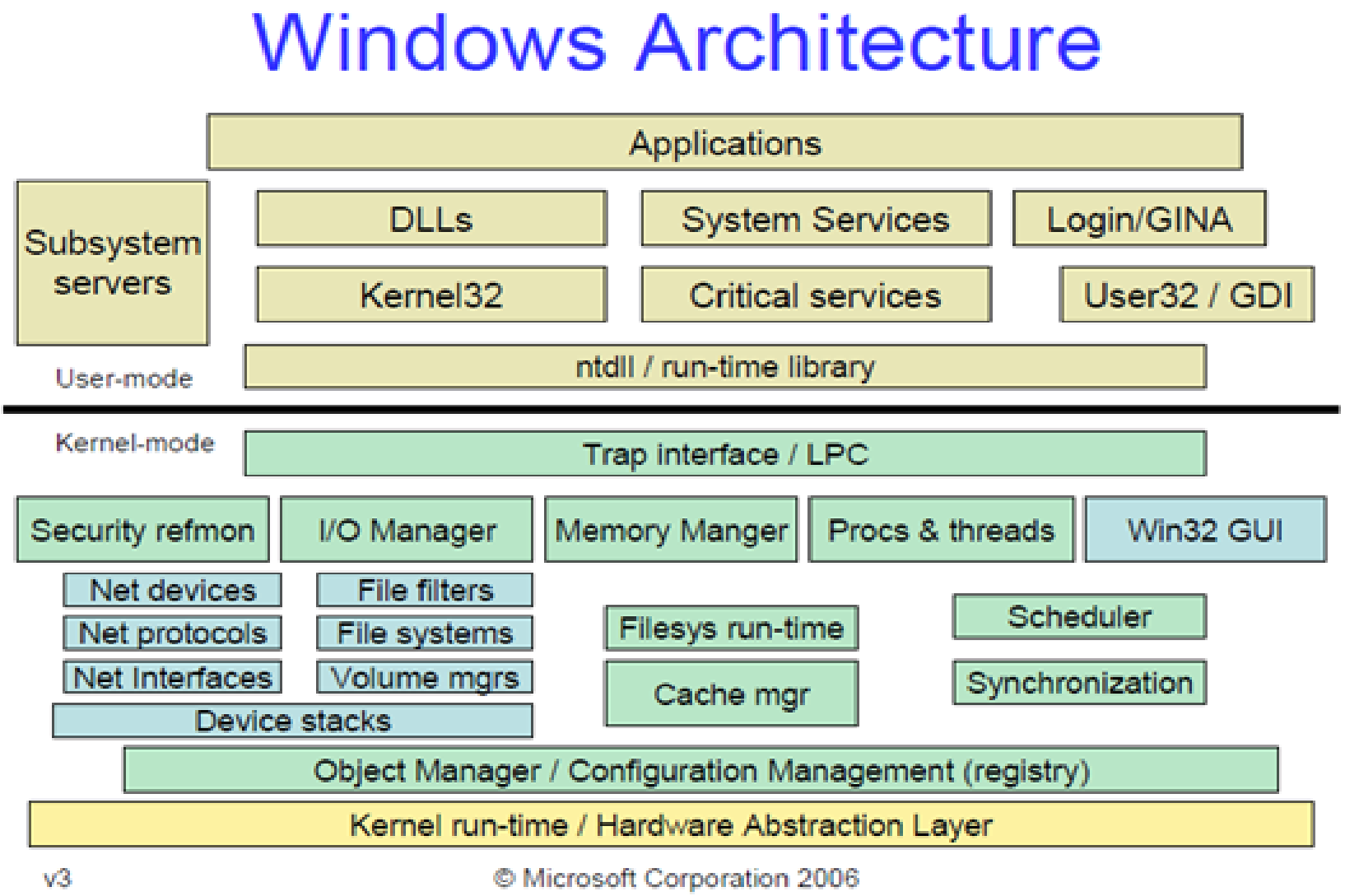
GNU/Linux Stack : User space & Kernel space



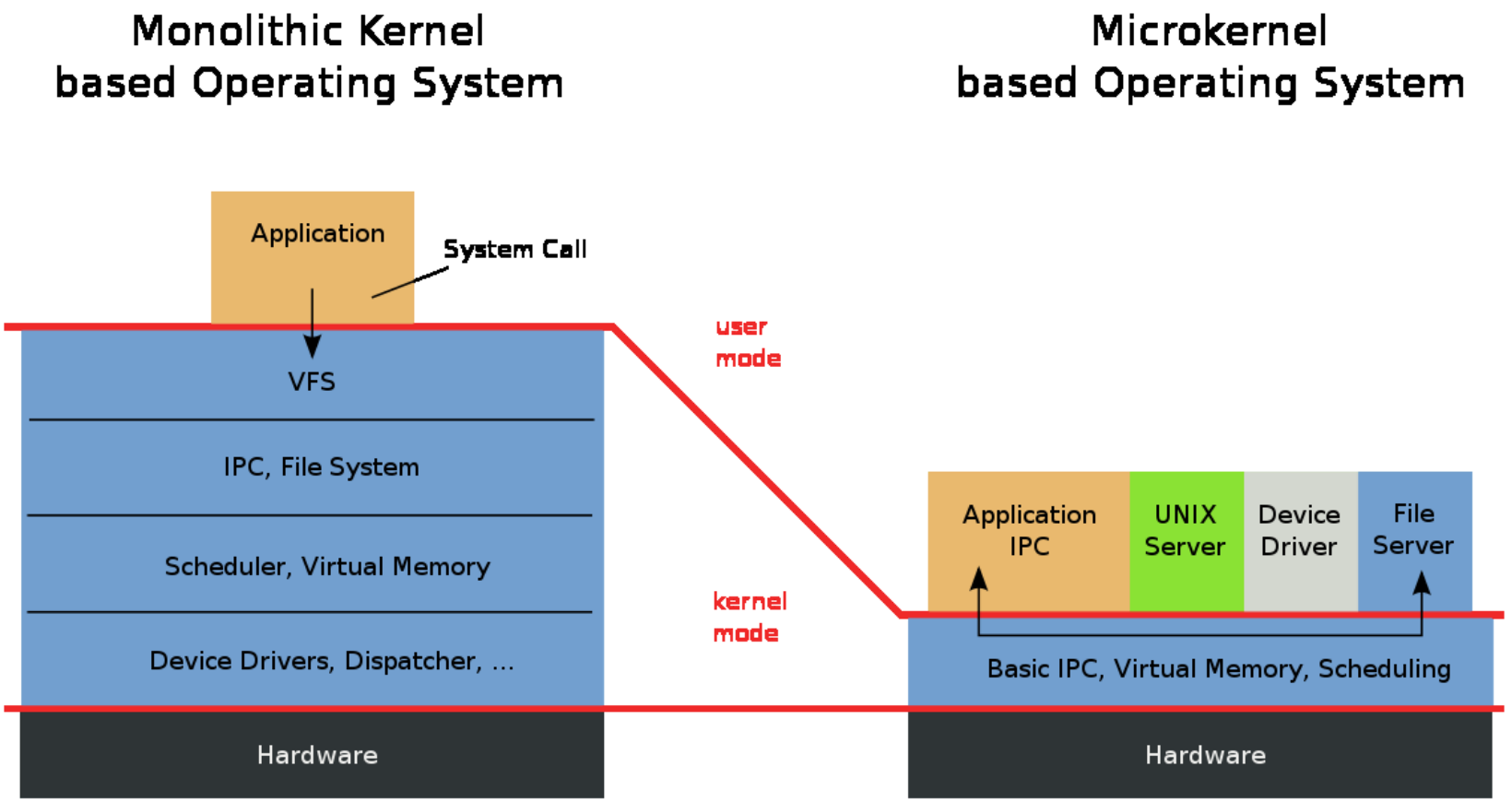
User space & Kernel space



User Mode & Kernel Mode



Monolithic Kernel s Microkernel



Kernel Mode vs. User Mode

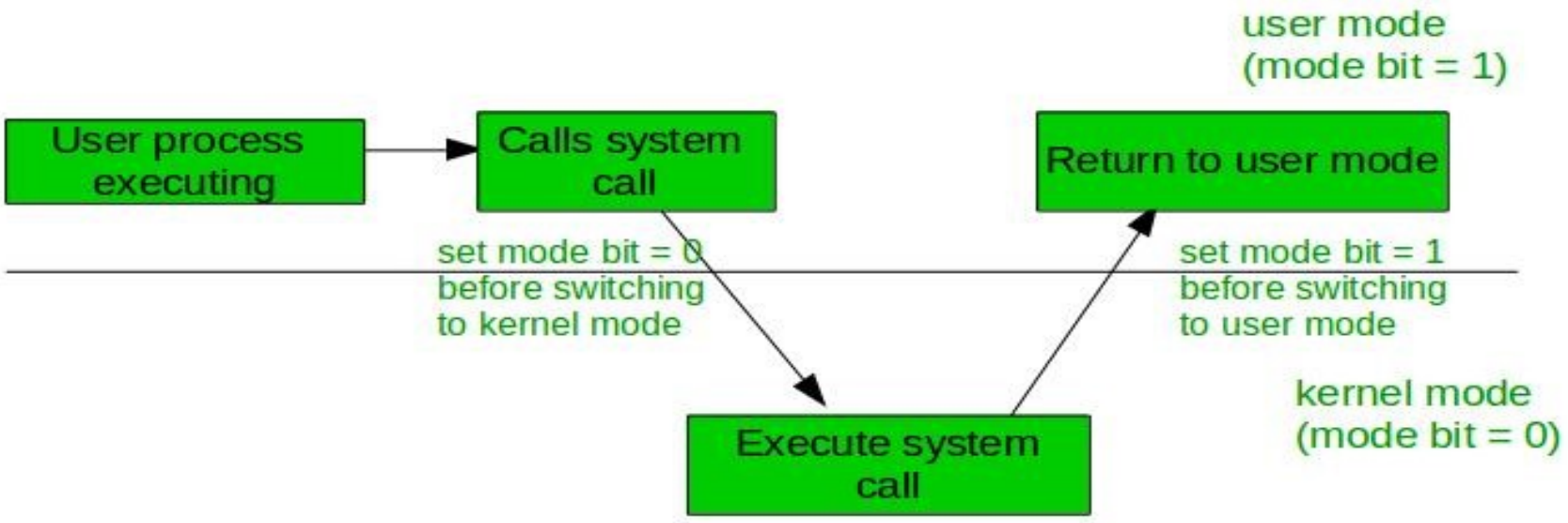
- **The kernel** executes at the highest level (also called supervisor mode), where everything is allowed, whereas **applications** execute in the lowest level (the so-called user mode), where the processor regulates direct access to hardware and unauthorized access to memory.
- In a 32-bit architecture, CPU can generate up to 4GB of virtual memory. In which 1GB is reserved for kernel space and 3GB is reserved for Userspace.

Kernel Space: This is a protected memory space that has full access to the hardware and system state.

- Kernel Space contains kernel code, core data structures identical to all process.
- In kernel space, most of the memory is directly mapped to physical memory at the fixed offset.
- **User Space:** The normal application executes in Userspace and they can see the only subset of the machine's available resources and can perform certain system functions.
- User Space contains process code, data, and memory-mapped files.
- In user space, memory mapping differs from one address space to another.

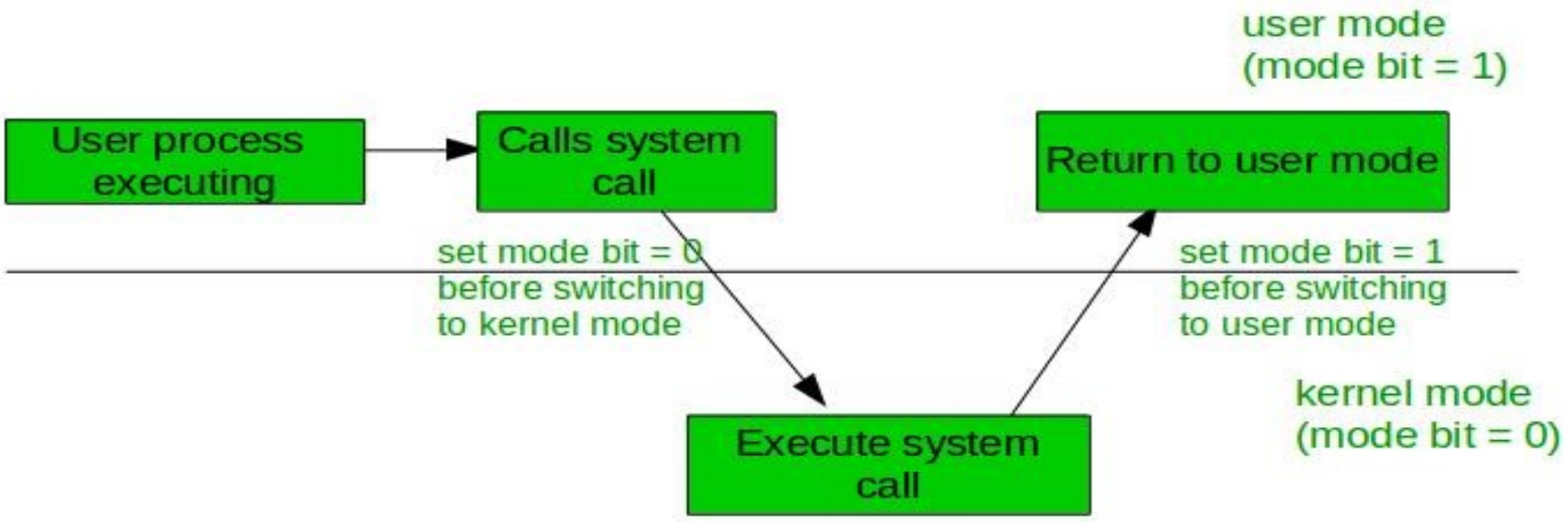
What happen when an interrupt occurs:[User Mode]

- When the computer system run user applications like creating a text document or using any application program, then the system is in the user mode.
- When the user application requests for a service from the operating system or an **interrupt** occurs or **system call**, then there will be a transition from user to kernel mode to fulfill the requests.



What happen when an interrupt occurs: [Kernel Mode]

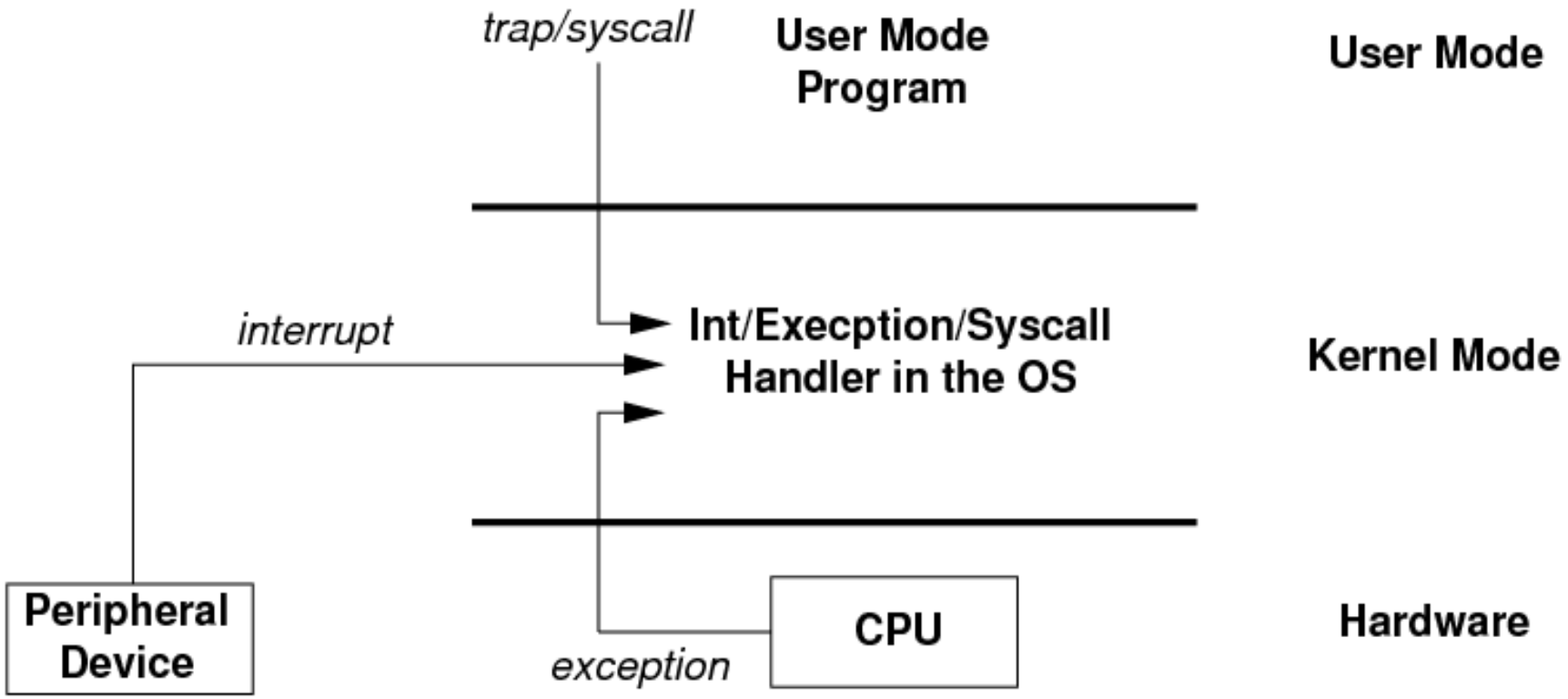
- When the system boots, hardware starts in kernel mode and when operating system is loaded, it start user application in user mode.
- To provide protection to the hardware, we have privileged instructions which execute only in kernel mode. If user attempt to run privileged instruction in user mode then it will treat instruction as illegal and traps to OS.



Introduction of System Call

- In computing, a **system call** is the programmatic way in which a computer program requests a service from the kernel of the operating system it is executed on.
- A system call is a way for programs to **interact with the operating system**.
- A computer program makes a system call when it makes a request to the operating system's kernel.
- System call **provides** the services of the operating system to the user programs via Application Program Interface(API). It provides an interface between a process and operating system to allow user-level processes to request services of the operating system.
- System calls are the only entry points into the kernel system.
- All programs needing resources must use system calls.

Kernel Mode vs. User Mode

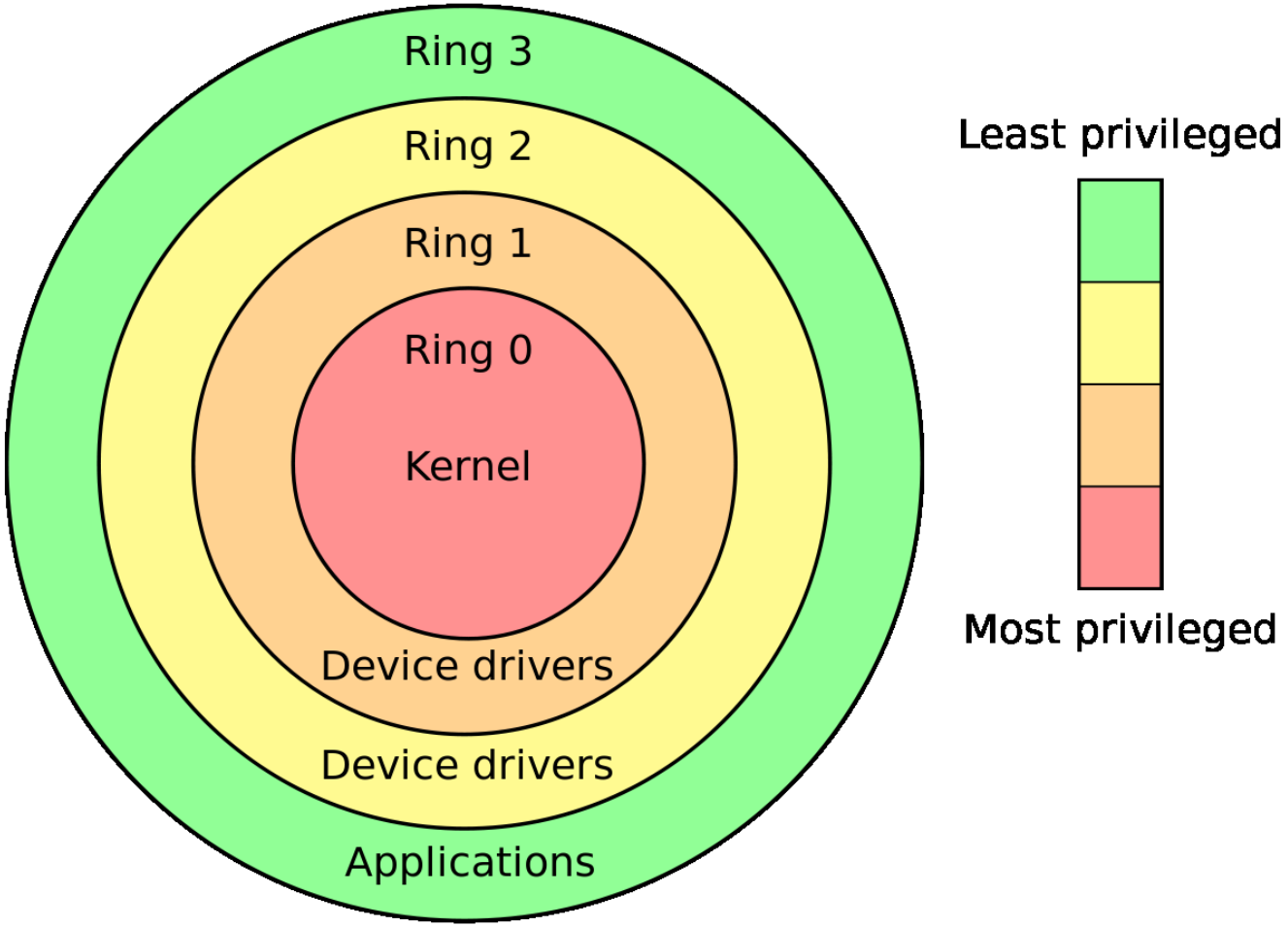


Kernel Mode vs. User Mode

- **Kernel space** is that area of virtual memory where **kernel** processes will run and **user space** is that area of virtual memory where **user** processes will be running. This division is required for memory access protections
- Kernel is a piece of code, which manages your hardware and provide system abstraction. So it needs to have access for all the machine instruction. And it is most trusted piece of software. So i should be executed with the highest privilege. And *Ring level 0* is the most privileged mode. So *Ring Level 0* is also called as **Kernel Mode**.
- User Application are piece of software which comes from any third party vendor, and you can't completely trust them. Someone with malicious intent can write a code to crash your system if he had complete access to all the machine instruction. So application should be provided with access to limited set of instructions. And *Ring Level 3* is the least privileged mode. So all your application run in that mode. Hence that *Ring Level 3* is also called **User Mode**.

Kernel Mode vs. User Mode

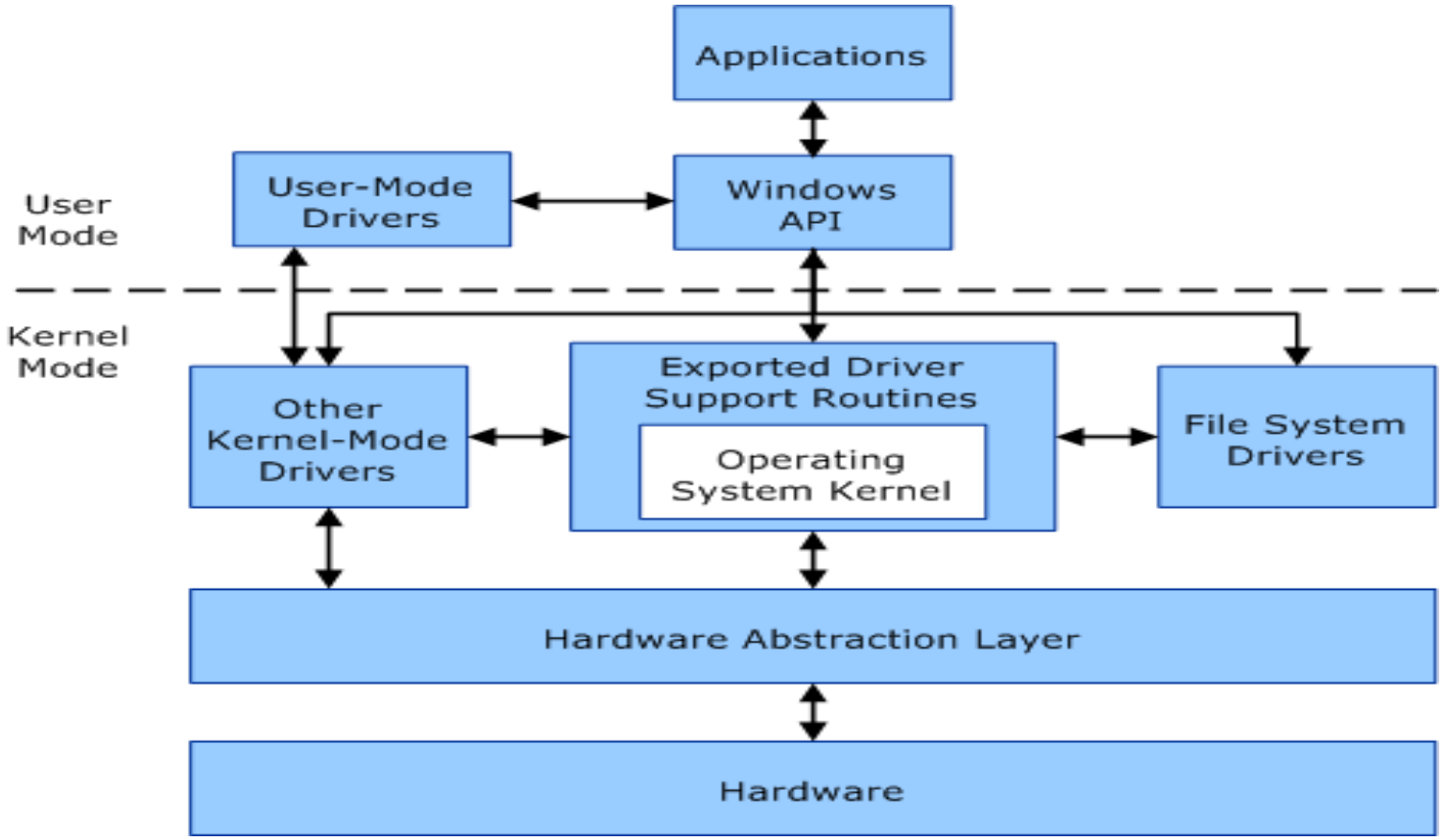
- *Ring Level 0* is also called as **Kernel Mode** .*Ring Level 3* is also called **User Mode**.



Kernel Mode vs. User Mode

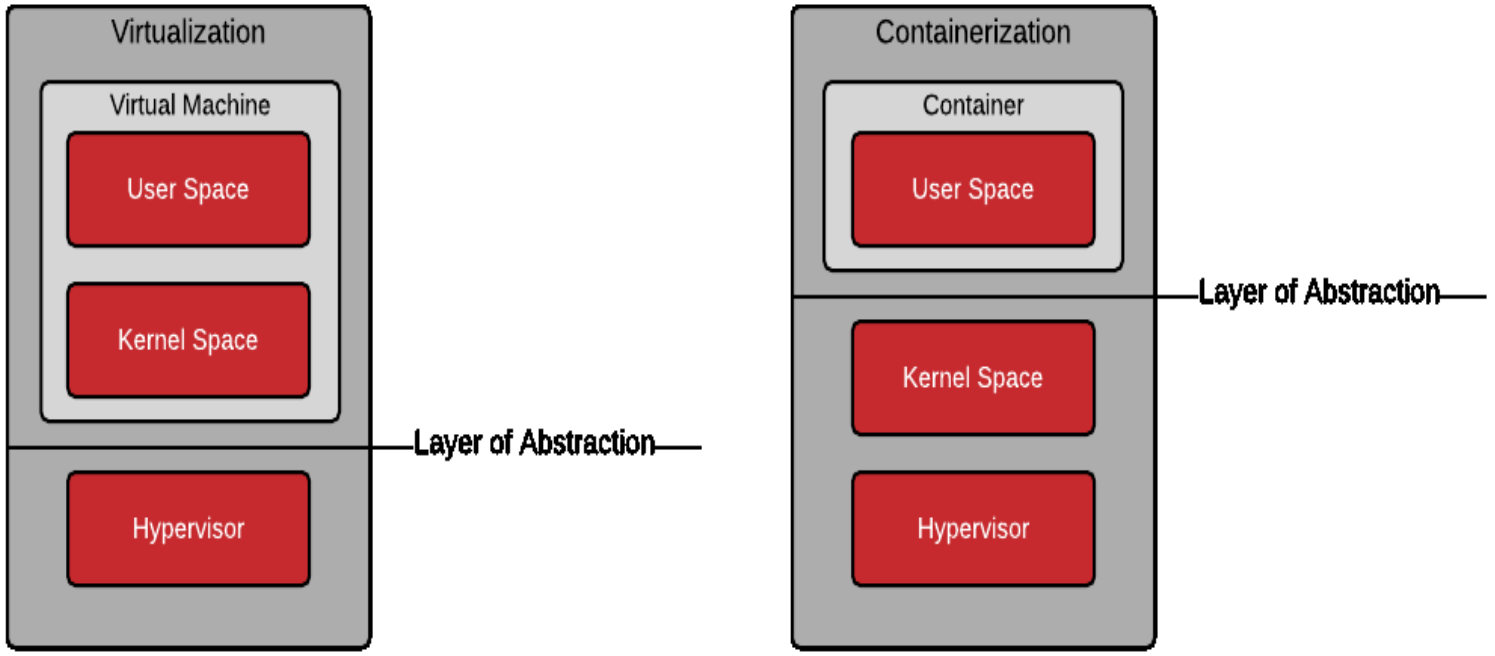
- Kernel is a piece of code, which manages your hardware and provide system abstraction. So it needs to have access for all the machine instruction. And it is most trusted piece of software. So i should be executed with the highest privilege. And *Ring level 0* is the most privileged mode. So *Ring Level 0* is also called as **Kernel Mode**.
- User Application are piece of software which comes from any third party vendor, and you can't completely trust them. Someone with malicious intent can write a code to crash your system if he had complete access to all the machine instruction. So application should be provided with access to limited set of instructions. And *Ring Level 3* is the least privileged mode. So all your application run in that mode. Hence that *Ring Level 3* is also called **User Mode**.

Kernel Mode vs. User Mode



Virtual Machines vs. Containers

- A virtual machine is a convenient way of packaging up virtual hardware, a kernel, and a user space. A container, on the other hand, packages up only the user space; there is no kernel or virtual hardware.

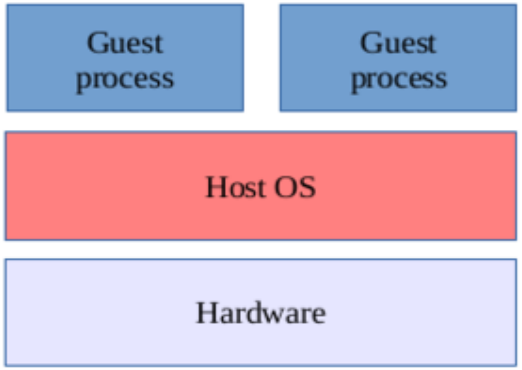


Operating System Virtualization

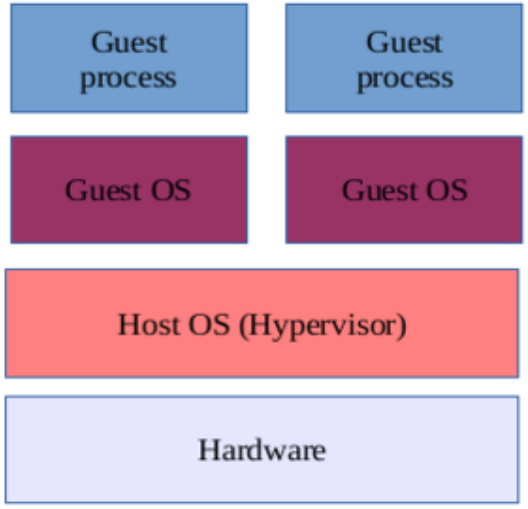
- Operating System **Virtualization** (**OS Virtualization**) is the last types of **Virtualization in Cloud Computing**.
- Operating system **virtualization** is a part of **virtualization** technology and is a type of server **virtualization**.
- Operating system virtualizations includes a modified form than a normal operating system so that different users can operate its end use different applications. This whole process shall perform on a single computer at a time.
- In OS virtualizations, the virtual eyes environment accepts command from any of the user operating it and performs different task on the same machine by running different applications.

Operating System Virtualization

- In ***operating system virtualizations*** when the application does not interfere with another one even though they are functioning in the same computer.
- The kernel of an operating system allows more than one isolated user-space instance to exist. These instances call as *software containers*, which are virtualizations engines.



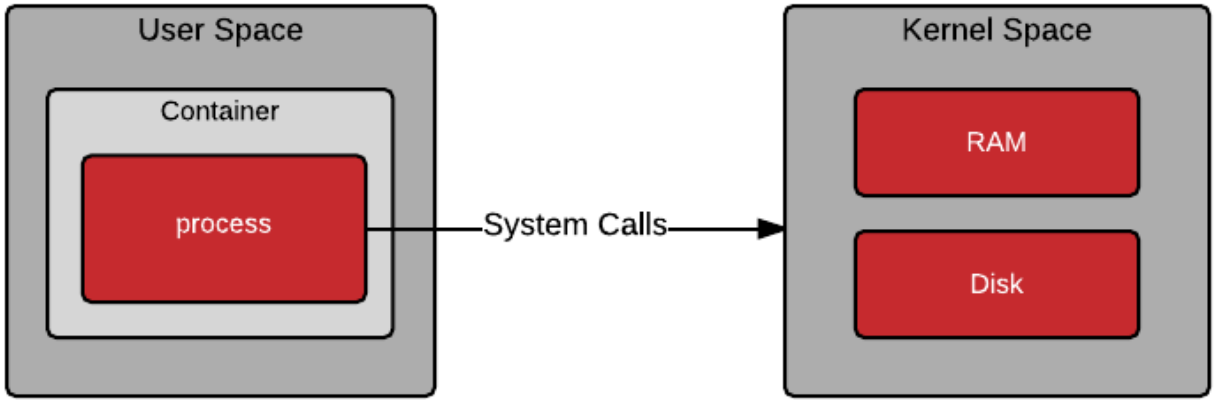
OS-level virtualization



Hardware virtualization

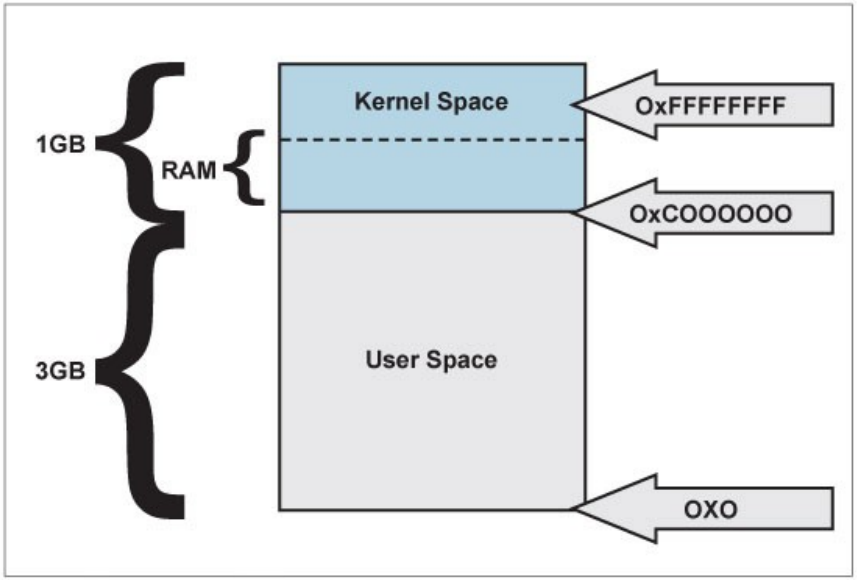
Execution of system call

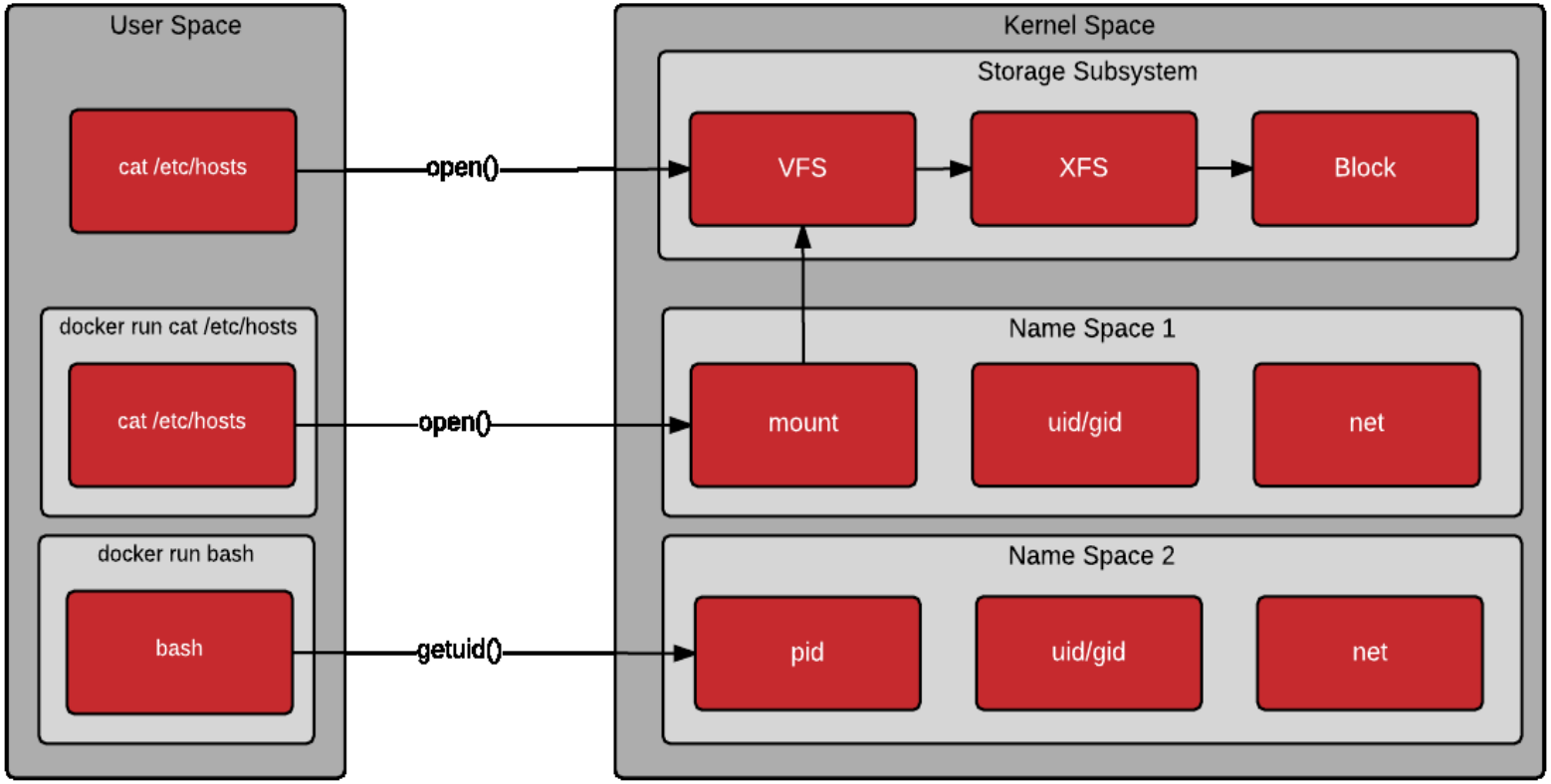
- **Kernel space** is where the **kernel** (i.e., the core of the **operating system**) runs and provides its services. Its something that the **user** is not allowed to interfere with.
- User **space** is that set of memory locations in which user processes (i.e., everything other than the **kernel**) run.
- **User space** is that portion of system memory in which **user** processes run.
- A process is an executing instance of a program.



Example: Linux

- User applications cannot directly communicate with hardware because Linux does not allow. Linux divides RAM memory into two regions: ***kernel space*** and ***user space***.
- The kernel space is where Linux runs and provide their services and where device drivers reside. User space is the area of memory where the user processes are executed. The kernel space can be accessed by user processes only through the use of **system calls**.




User Space
cat /etc/hosts
docker run cat /etc/hosts
cat /etc/hosts
docker run bash
bash
open()
open()
getuid()
Kernel Space
Storage Subsystem
VFS
XFS
Block
Name Space 1
mount
uid/gid
net
Name Space 2
pid
uid/gid
net

How does OS Virtualization works?

Components needed for using OS Virtualization in the infrastructure are given below:

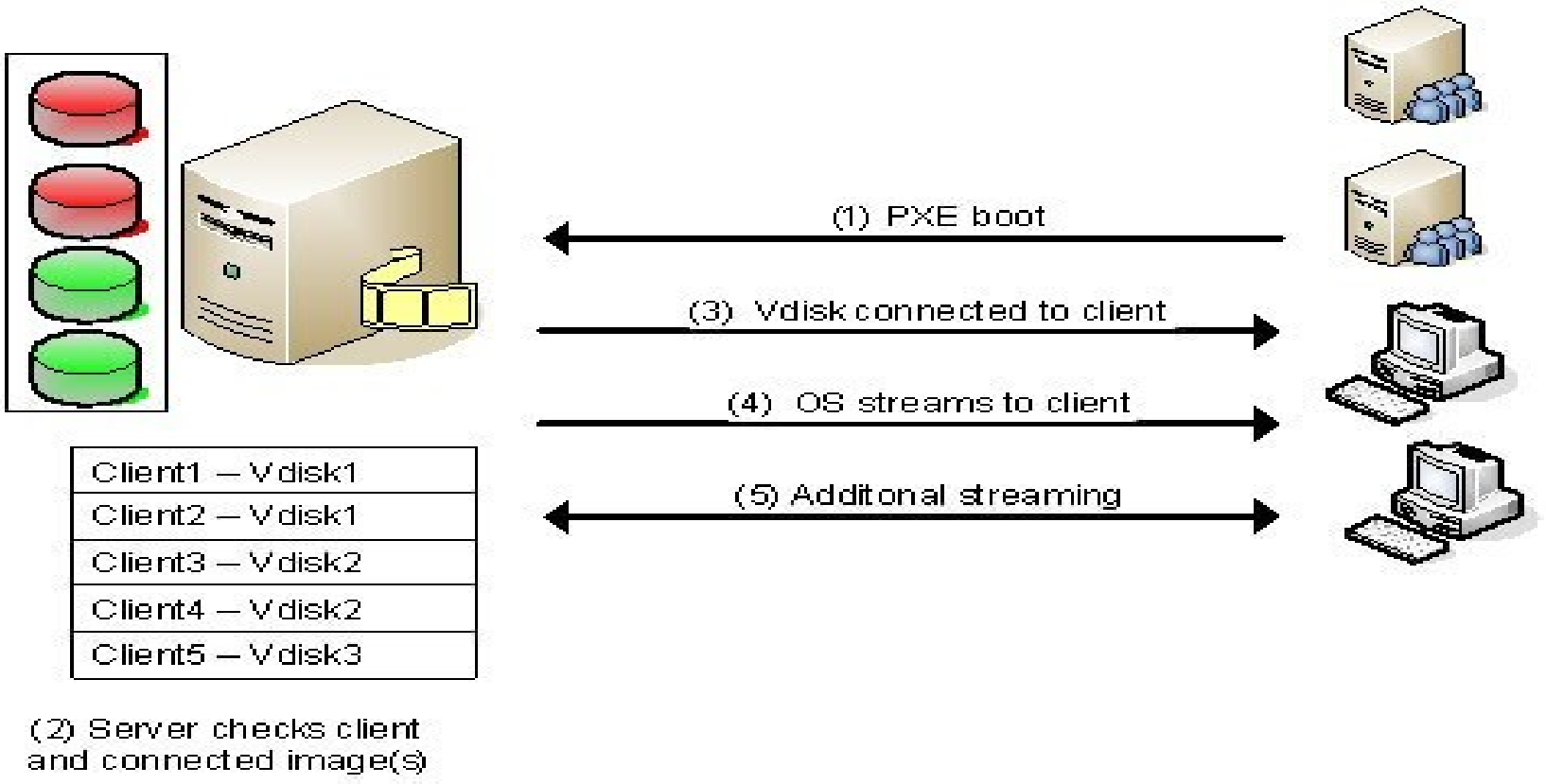
- The first component is the **OS Virtualization server**. This server is the center point in the OS Virtualization infrastructure.
- The server manages the streaming of the information on the virtual disks for the client and also determines which client will be connected to which virtual disk (using a database, this information is stored).
- The server can host the storage for the virtual disk locally or the server is connected to the virtual disks via a SAN (Storage Area Network).
- In high availability environments there can be more OS Virtualization servers to create no redundancy and load balancing. The server also ensures that the client will be unique within the infrastructure.

How does OS Virtualization works?

- Secondly, there is **a client** which will contact **the server** to get connected to the virtual disk and asks for components stored on the virtual disk for running the operating system.
- The available supporting components are database for storing the configuration and settings for the server, a streaming service for the virtual disk content, a (optional) Trivial File Transfer Protocol (TFTP) service and a (also optional) Preboot eXecution Environment (PXE) boot service for connecting the client to the OS Virtualization servers.
- As it is already mentioned that the virtual disk contains an image of a physical disk from the system that will reflect to the configuration and the settings of those systems which will be using the virtual disk. When the virtual disk is created then that disk needs to be assigned to the client that will be using this disk for starting. The connection between the client and the disk is made through the administrative tool and saved within the database. When a client has a assigned disk, the machine can be started with the virtual disk using the following as displayed in the following figure.

How does OS Virtualization works?

- Preboot eXecution Environment (PXE) boot service for connecting the client to the OS Virtualization servers.



How does OS Virtualization works?

[1] Connecting to the OS Virtualization server: First we start the machine and set up the connection with the OS Virtualization server. Most of the products offer several possible methods to connect with the server.

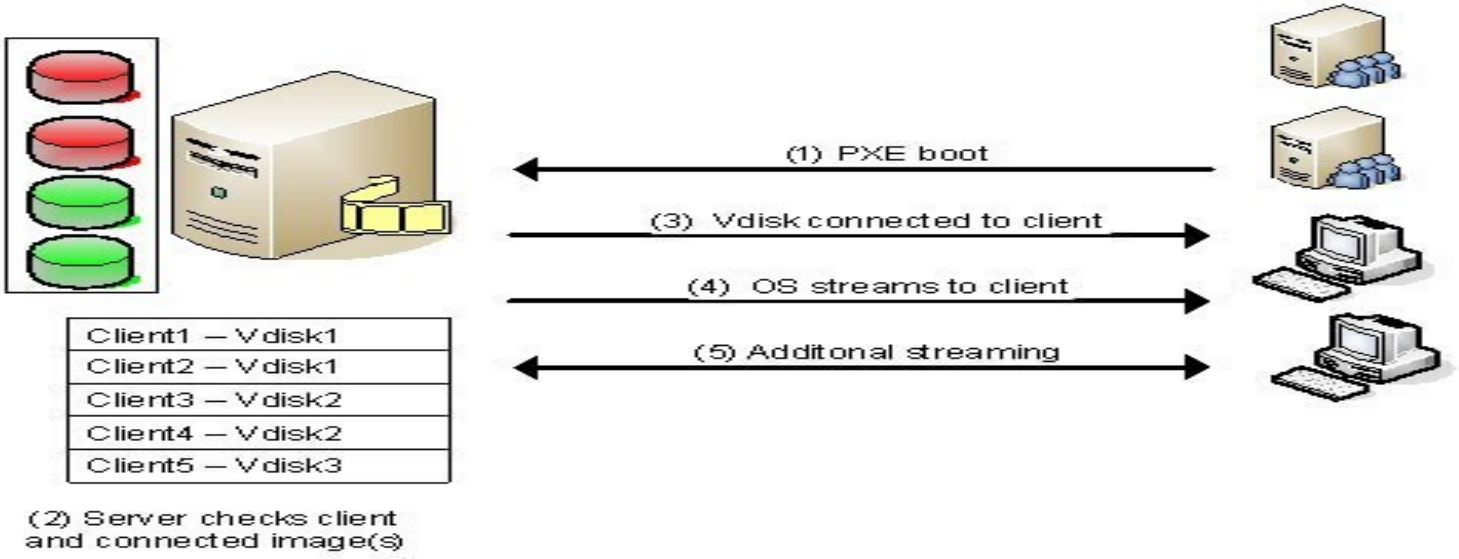
One of the most popular and used methods is using a Preboot eXecution Environment (PXE) boot service for connecting the client to the OS Virtualization servers.

PXE service, but also a boot strap is used a lot (because of the disadvantages of the PXE service). Although each method initializes the network interface card (NIC), receiving a (DHCP-based) IP address and a connection to the server.

How does OS Virtualization works?

[2] Connecting the Virtual Disk: When the connection is established between the client and the server, the server will look into its database for checking the client is known or unknown and which virtual disk is assigned to the client.

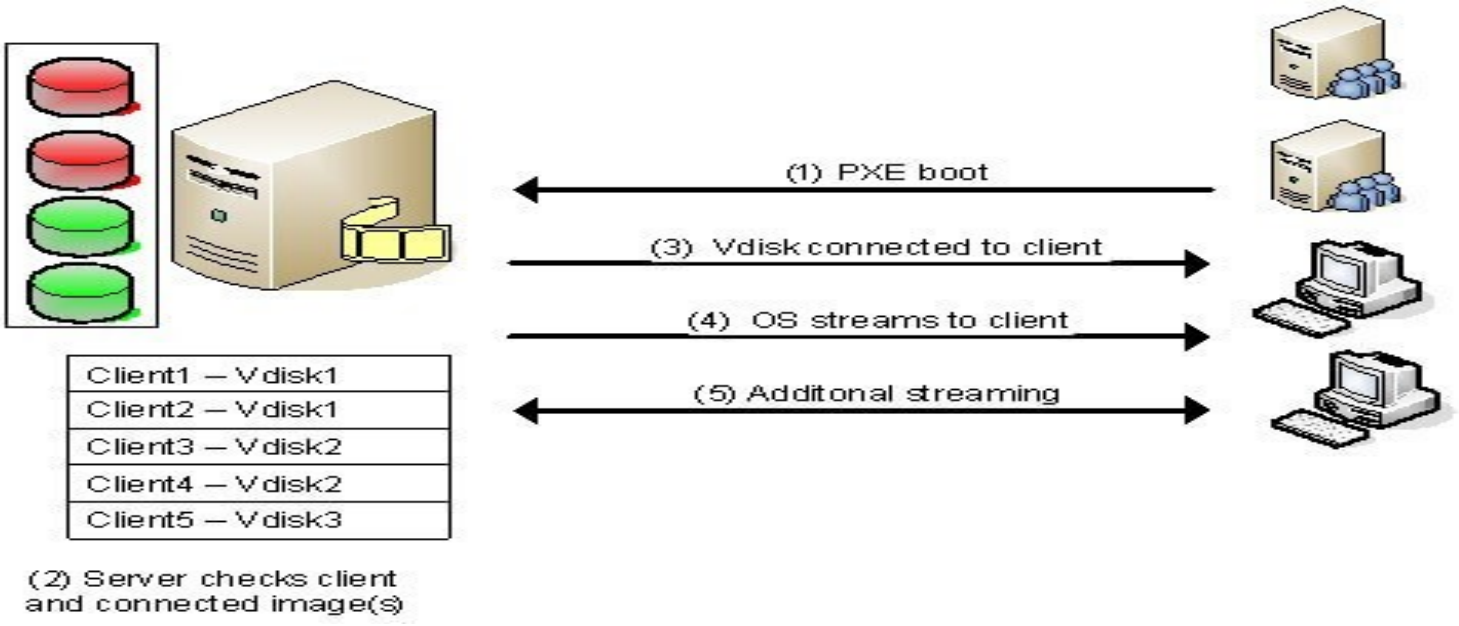
When more than one virtual disk are connected then a boot menu will be displayed on the client side. If only one disk is assigned, that disk will be connected to the client which is mentioned in step number 3.



How does OS Virtualization works?

[3] **VDisk connected to the client**: After the desired virtual disk is selected by the client, that virtual disk is connected through the OS Virtualization server .

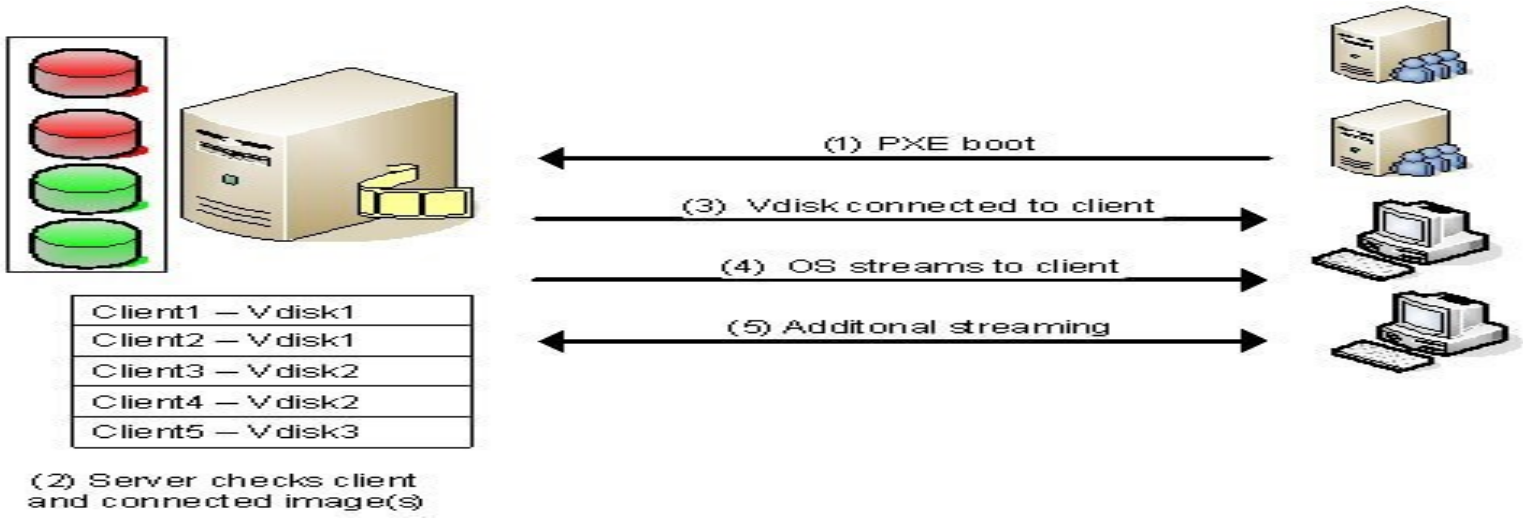
At the back-end, the OS Virtualization server makes sure that the client will be unique (for example computer name and identifier) within the infrastructure.



How does OS Virtualization works?

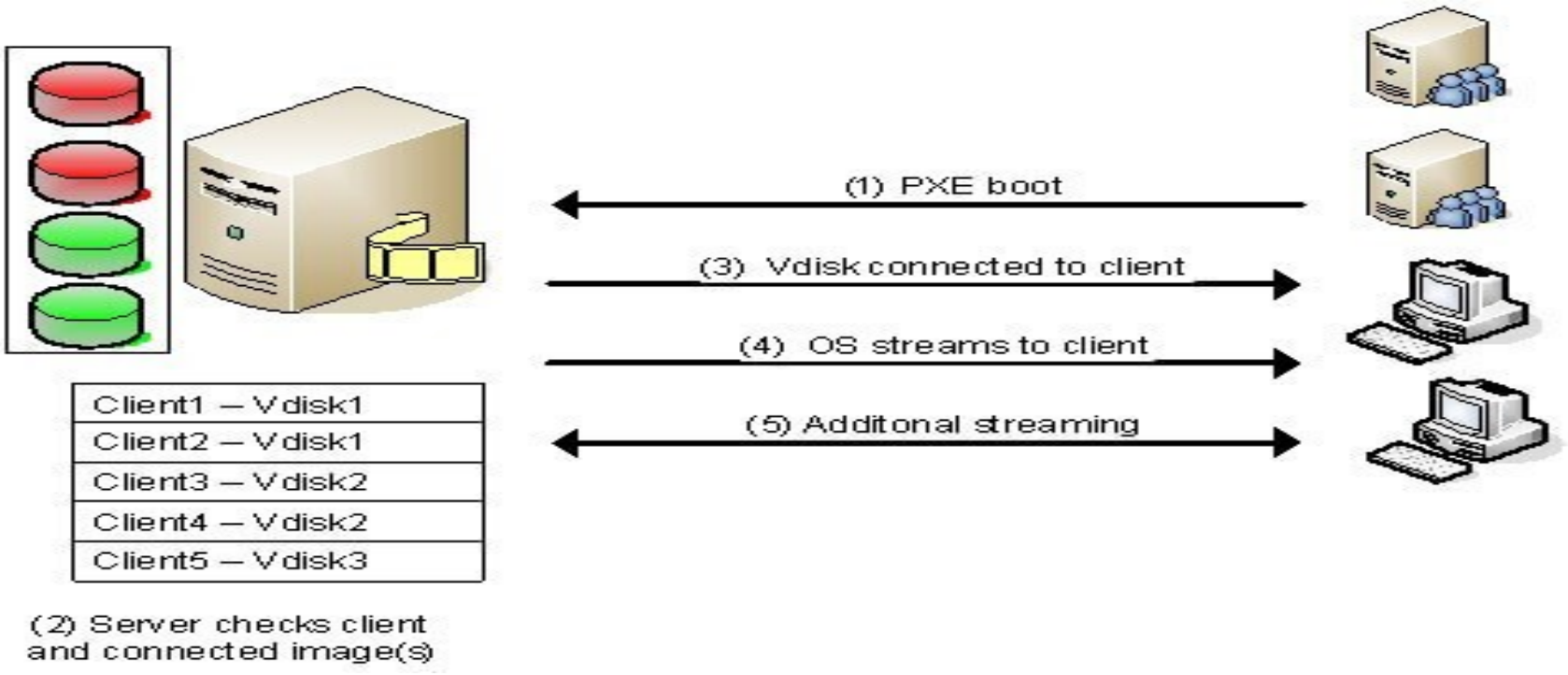
[4] OS is "streamed" to the client: As soon the disk is connected the server starts streaming the content of the virtual disk. The software knows which parts are necessary for starting the operating system smoothly, so that these parts are streamed first.

The information streamed in the system should be stored somewhere (i.e. cached). Most products offer several ways to cache that information. For examples on the client hard disk or on the disk of the OS Virtualization server.



How does OS Virtualization works?

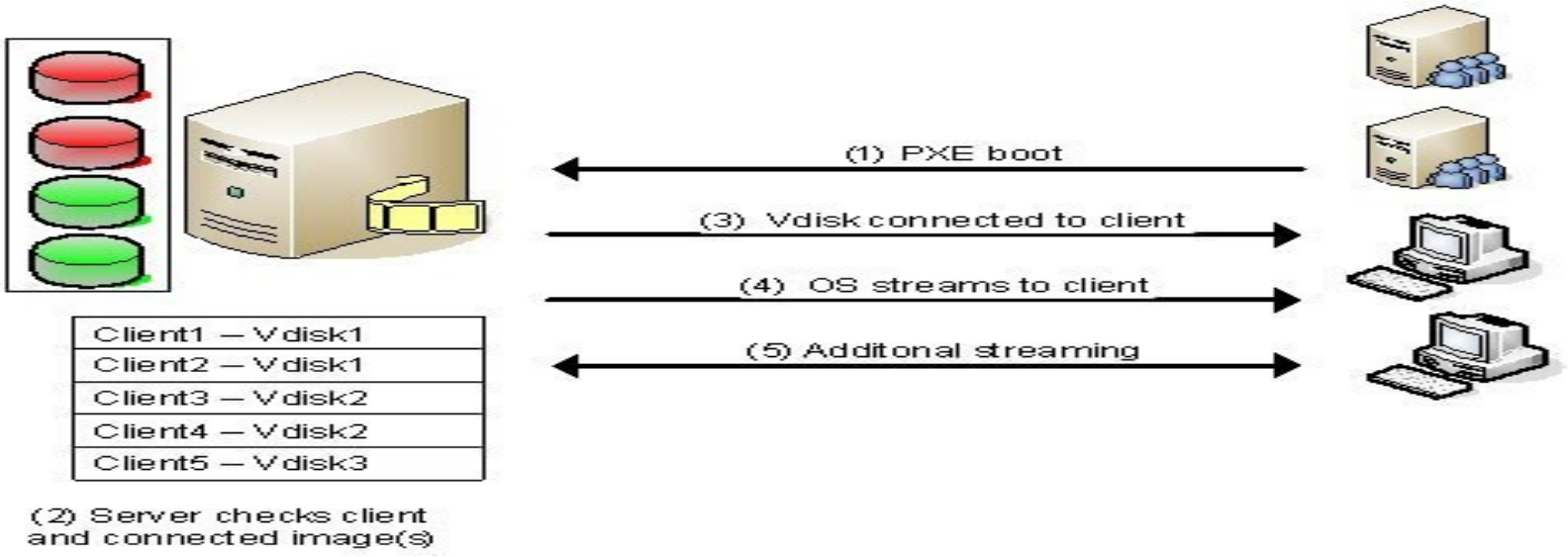
[5] Additional Streaming: After that the first part is streamed then the operating system will start to run as expected. Additional virtual disk data will be streamed when required for running or starting a function called by the user (for example starting an application available within the virtual disk)



How OS Virtualization Works

The steps for how these virtualization works are listed below:

1. Connect to OS Virtualization Server
2. Connect to virtual disk
3. Then connect this virtual disk to the client
4. OS is streamed to the client
5. If further additional streaming is required, it is don



Windows OS virtualization

- Users need to install VMware first to install windows OS virtually.
- After installing such virtualization software the steps to install a new OS are:

1. Click on "create new Virtual Machine"
2. Browse the OS that is to be installed, and click Next
3. Give the product key if required; and click Next
4. In the 'New Virtual Machine Wizard' window; click Next
5. Give it a name & location and click Next
6. The next step will create disk and the user can see the 1st Window screen.
7. Select the version & OS architecture (64 bits or 86 bits)
8. The Installation will be done

Linux OS virtualization

- To virtualized Linux systems, VMware workstation software is used. To install any software virtually, users need VMware software to install first.
- To create a virtual machine for Linux OS the steps to be followed are:

1. Double click the VMware to run
2. Click on "create new Virtual Machine"
3. A window pops up, choose "custom" option and then click Next
4. The next window appears, 'VM Hardware Compatibility window'; Click on Next button
5. In the guest OS window pane - choose ISO image from the disk or any drive. Browse your ISO image & click Next
6. A new window pops up waiting for username, password & confirm password to be feed in and then click Next
7. In the processor configuration information, select your precise number of processor per core. If you want to keep the default setting > just ignore selecting and press Next
8. The next window pops up to let user set memory limits. Put the value and click Next
9. Next window let users set the disk size; then press Next
10. In specify disk file window user can specify the disk file; then click Next
11. In the last pop up window, click Finish.
12. Now user will see a VMware screen then the OS's installation screen. And the installation will get started

Advantages of OS virtualization

- OS virtualization usually imposes little or no overhead.
- OS Virtualization is capable of live migration
- It can also use dynamic load balancing of containers between nodes and a cluster.
- The file level copy-on-write (CoW) mechanism is possible on OS virtualization which makes easier to back up files, more space-efficient and simpler to cache than the block-level copy-on-write schemes

Virtual Disks in OS Virtualization

- The client will be connected via the network to the virtual disk & will boot the OS installed on virtual disk. Two types of virtual disks are there for implementation.

1. **Private Virtual Disk**: is used by one client only like that of a local hard disk. Users can save information on the virtual disk based on the rights assigned. So as the client restart the system, the settings are retained just like working with physical local hard disk.
2. **Shared/Common Virtual Disk**: It is used by multiple clients at the same time. The changes are saved in a special cache & these caches gets cleaned as the user restarts or shutdowns the system. In other words, when a client is booting up, it will use the default configuration available on the virtual disk.

References

- http://techgenix.com/windows-and-hyper-v-containers-windows-server-2016/
- https://www.javatpoint.com/os-virtualization
- https://www.w3schools.in/cloud-virtualization/os-virtualization/
- https://www.redhat.com/ja/blog/architecting-containers-part-2-why-user-space-matters
- https://www.geeksforgeeks.org/introduction-of-system-call/?ref=lbp


Thank you for your
kind attention