**Xen Project Hypervisor**

(<https://wiki.xenproject.org/wiki/Xen_Project_Beginners_Guide>)

Terms:

* **Virtual Machine Monitor (VMM):** aka hypervisor.
* **Type 1 hypervisor**: aka “**bare-metal” hypervisor**. Hypervisor runs directly on top of the physical machine as opposed to within an operating system.
* **Domain**: Guest virtual machines running on a Xen Project Hypervisor. Composed by dom0 and domUs.
* **dom0:** aka **Control Domain**. It is the domain responsible for controlling the hypervisor and starting other guest operating systems. Privileged domain that can control the hypervisor or start/stop other domains.
* **domUs**: aka **Guest Domain**. It is composed by normal VMs guest operating system. It is an unprivileged domain.
* **Toolstack**: API installed on the dom0 to manage the hypervisor. They are many Toolstack with different functionalities.
* **Enlightened guests**: custom or modified guest operating systems. These operating systems are aware that they are being virtualized and as such don’t require virtual “hardware” devices, instead they make special calls to the hypervisor that allow them to access CPUs, storage and network resources.
* **Paravirtualization (PV)**: virtualization technique to allow Guest VM to directly access unvirtualized devices. It does not require virtualization extensions from the host CPU, but it requires enlightened guests.
* **Hardware virtual machine (HVM**): aka “**full virtualization**”. guests need not be modified as the hypervisor will create a fully virtual set of hardware devices for this machine that resemble a physical x86 computer. It requires special CPU extensions - VT-x for Intel processors and AMD-V for AMD based machines.
* **Paravirtualization on HVM (PVHVM**): it is a HVM domain with Paravirtualized storage, network and other devices. This provides the best of both worlds by reducing expensive emulation but providing hardware accelerated CPU and memory access.

Xen Project:

it is a Type 1 Hypervisor.

Dom0 Functionalities:

1. Used as the interfaces with the hypervisor through special instructions.
2. Used to change the configuration of the hypervisor.
3. Used to create/start/stop/delete Guest VMs.
4. Used to create/start/stop/delete Domains.
5. Used as the primary interface to the hardware. The hypervisor doesn’t contain device drivers.
6. It contains the device drivers needed to address the hardware. The devices are attached to dom0 and you can use standard Linux drivers.
7. It can share devices and resources with Guest OSs through a number of Paravirtualized devices.

Types of virtualization supported by Xen Project:

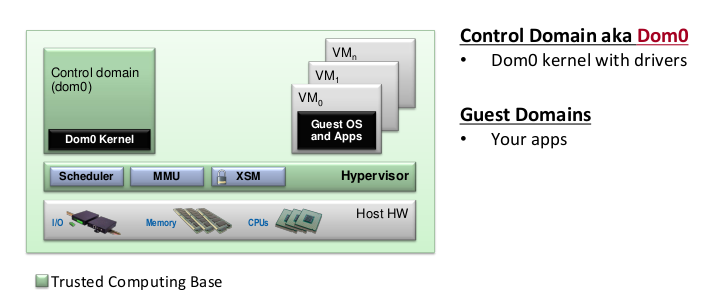
1) Para-virtualization: Hosts enlightened guests.

2) Hardware virtual machine (HVM): aka “full virtualization”.

3) Paravirtualization on HVM (PVHVM):

Xen Project architecture:

1. the hypervisor sits on the bare metal
2. All guest VMs sit on the hypervisor layer.
3. Dom0 which is a guest VM with privileges to start/stop/create guest VMs, configure the hypervisor, and it contains the hardware drivers.
4. Provide device virtualization for Normal Guest OSs and paravirtualization for Enlightened Guest OSs.



Parts of a Paravirtualized Datapath:

1. A “backend” that lives in dom0: provides the paravirtual device.
2. A “frontend” driver within the guest domain: allows the guest OS to access the virtual device.

They used shared memory to communicate.

Common Paravirtualized Datapaths:

* net-back/net-front: paravirtualized networking
* blk-back/blk-front: paravirtualized storage systems

