# L03a: Introduction to Virtualization


## Introduction:

- Virtualization allows for the existence of entire operating systems on top of the same HW.
- It facilitates sharing the same HW resources across multiple applications running on distinct OSs.

## Hypervisor:

- A Hypervisor (Virtual Machine Monitor - VMM) facilitates sharing the HW resources between the virtual machines (guest Oss).
  - Native (bare metal) Hypervisor: Running directly on top of the HW.
  - Hosted Hypervisor: Running as an application process on top of a host OS.

## Full virtualization:

- Guest OSs running on top of the Hypervisor will be full unchanged binaries.
- When a guest OS tries to execute a privileged instruction, a trap will be generated and passed to the Hypervisor, which in turn will emulate the intended functionality of the OS (Trap & Emulate Strategy).
- This makes each guest OS think it's running alone on the HW.
- In some architectures, some privileged instructions may fail silently. This is why the Hypervisor will use a Binary Translation Strategy. The Hypervisor looks into each guest OS binary for the specific instructions that might fail silently and edits the binary to ensure careful handling of these instructions.

## Para virtualization:

- The binaries of the guest OSs will be modified to avoid problematic instructions and utilize optimizations.
- The Hypervisor will change only less than 2% of the guest OS code.
- For the user point of view, the OS is not changed.
- Ex.: Xen Hypervisor.

## What needs to be done?

- Virtualize HW:
  - Memory hierarchy.
  - CPU.
  - HW devices.
- Facilitate data transfer between the Hypervisor and the guest OS.