Teknologi Virtualisasi: Virtualisasi Hardware

Henry Saptono, S.Si, M.Kom
Sekolah Tinggi Teknologi Terpadu Nurul Fikri
November, 2020

Mengapa Virtualisasi ?

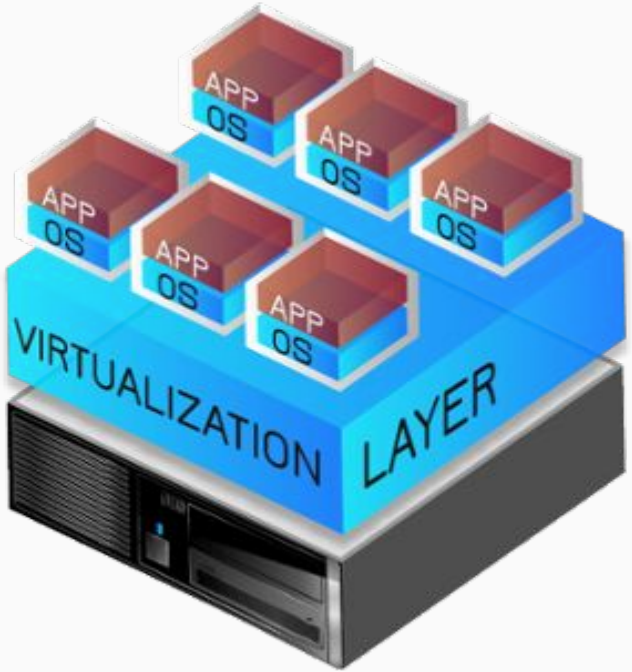
- Memberikan abstraksi yang lebih baik bagi pengguna sumber daya bersama
- Pemanfaatan sumber daya yang lebih tinggi.

Apa itu Virtualisasi Hardware ?

- Dikenal juga dengan sebagai virtualisasi mesin atau virtualisasi server
- Sistem atau komponen komputasi sebelumnya yang berjalan di lingkungan nyata kini berjalan di lingkungan virtual.
- Istilah virtualisasi perangkat-keras (harware) mengacu kepada upaya **menciptakan mesin virtual** yang bekerja layaknya sebuah komputer lengkap dengan sistem operasi.

Arsitektur Server tradisional & Server virtual

Traditional Server Architecture



Virtualized Server Architecture

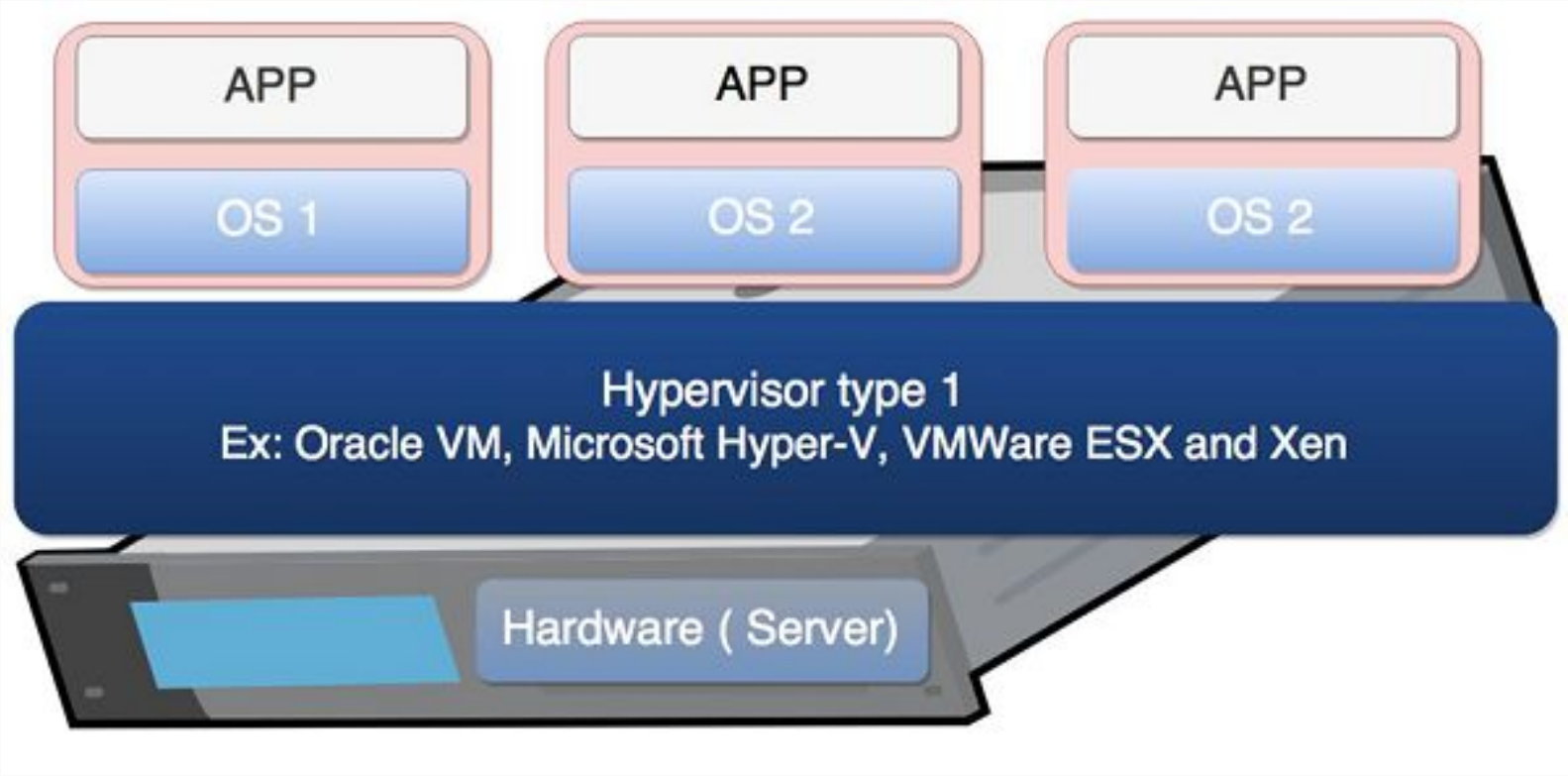
Beberapa istilah dalam Virtualisasi Hardaware

- Istilah mesin tuan-rumah (host) mengacu kepada mesin tempat virtualisasi berjalan
- Istilah mesin tamu (guest) mengacu kepada virtual mesin itu sendiri.
- Istilah hypervisor mengacu kepada perangkat-lunak atau firmware yang membuat/menciptakan mesin virtual.
- Istilah hypervisor juga dikenal dengan isilah virtual machine monitor (VMM)

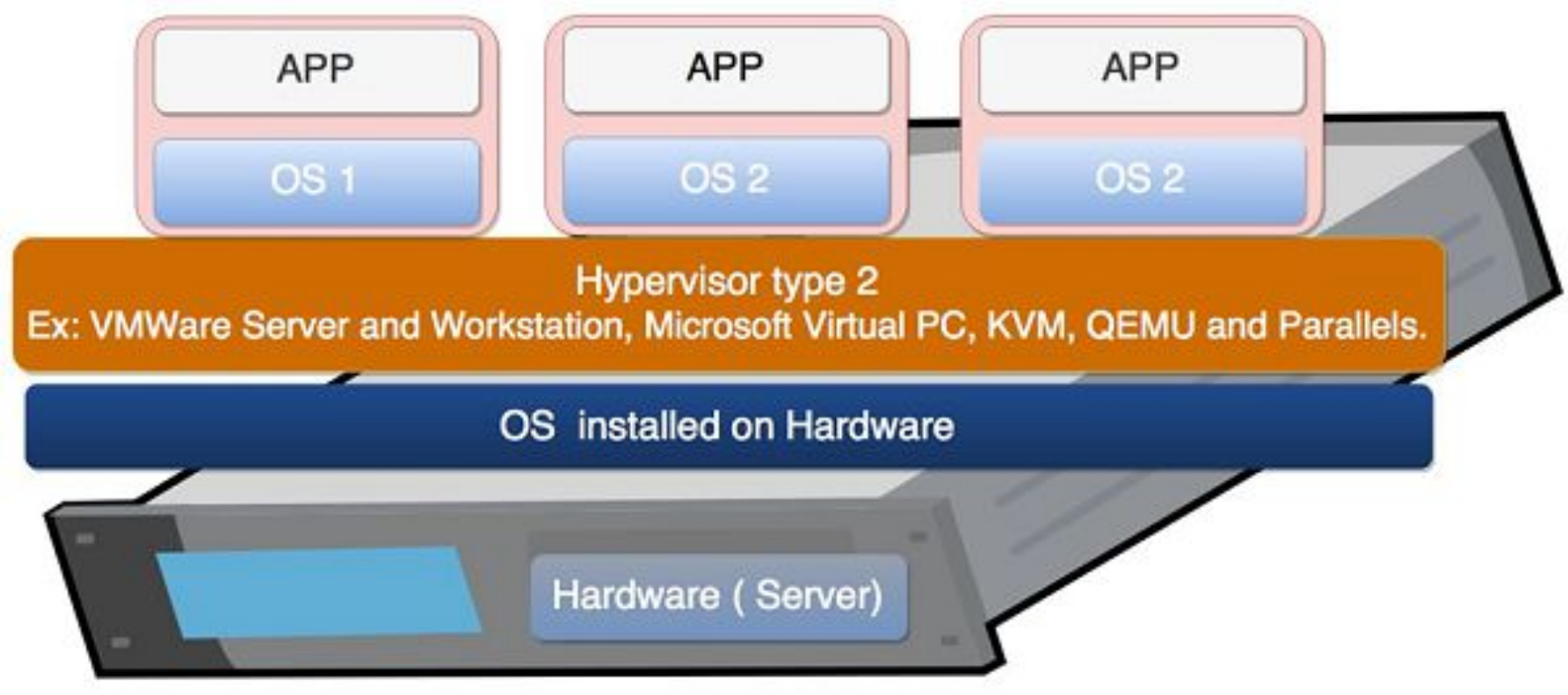
Taxonomy Hypervisor

- Type 1 (bare metal, native): supports multiple virtual machines and runs directly on the hardware
- Type 2 (hosted) VM - runs under a host operating system (e.g.,user-mode Linux)

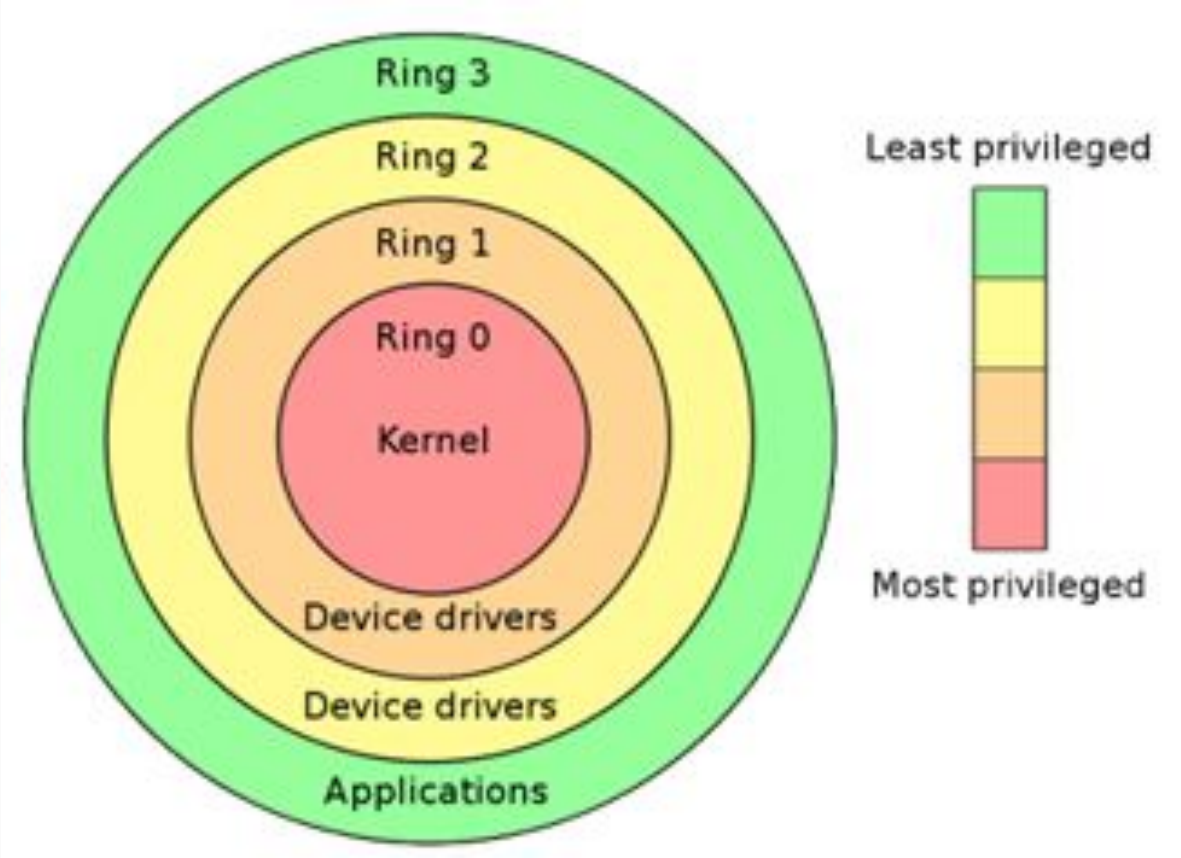
Hypervisor Type 1



Hypervisor Type 2



Protection Ring



Privileged and non-privileged instructions

- Non privileged (Non-hak istimewa) instruction tidak mengubah nilai atau status sumber daya bersama. Sumber daya bersama termasuk prosesor, memori, timer, dan register tujuan khusus. Instruksi non-hak istimewa termasuk instruksi aritmatika, instruksi logika, dan sebagainya.
- Privileged instructions (instruksi yang diistimewakan) digunakan untuk mengakses nilai atau status sumber daya bersama, termasuk mematikan, mengatur timer, mengatur program counter, mengubah nilai register relokasi dan instruksi yang terkait dengan I / O

Tipe Virtualisasi hardware

- Full Virtualization
- Full Virtualization with hardware-assisted
- Para-Virtualization

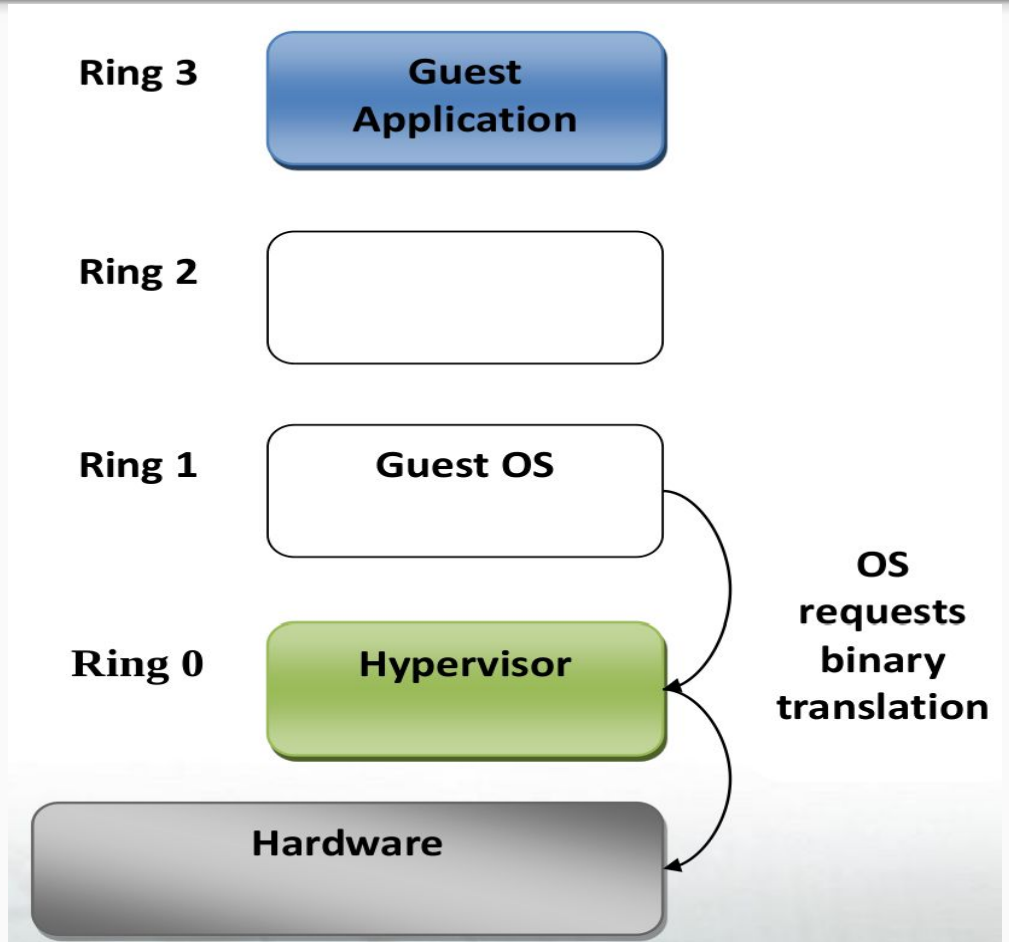
Full Virtualization

- Pada Full Virtualization (Virtualisasi penuh) OS tamu (Guest OS) tidak menyadari sedang divirtualisasi
- Tidak memerlukan OS tamu untuk dimodifikasi dengan cara apa pun
- Dengan demikian, akan bekerja dengan * semua * OS yang ditulis ke set instruksi yang sama
- OS tamu menganggapnya berjalan pada mesin bare-metal

Binary translation

- Full Virtualization menggunakan sejenis teknologi yang disebut Terjemahan Biner (binary tranlation). Ide intinya adalah bahwa hypervisor berjalan di ring 0, yang bertanggung jawab untuk manajemen perangkat keras yang mendasarinya.
- OS Guest berjalan di ring 1, dan ketika mereka memanggil instruksi yang diistimewakan (prosesor, memori, timer, dan register tujuan khusus), VMM (hypervisor) di ring 0 akan menggunakan terjemahan biner untuk menghentikan instruksi ini dan bertanggung jawab untuk pekerjaan instruksi berikut.

Binary translation in Full Virtualization

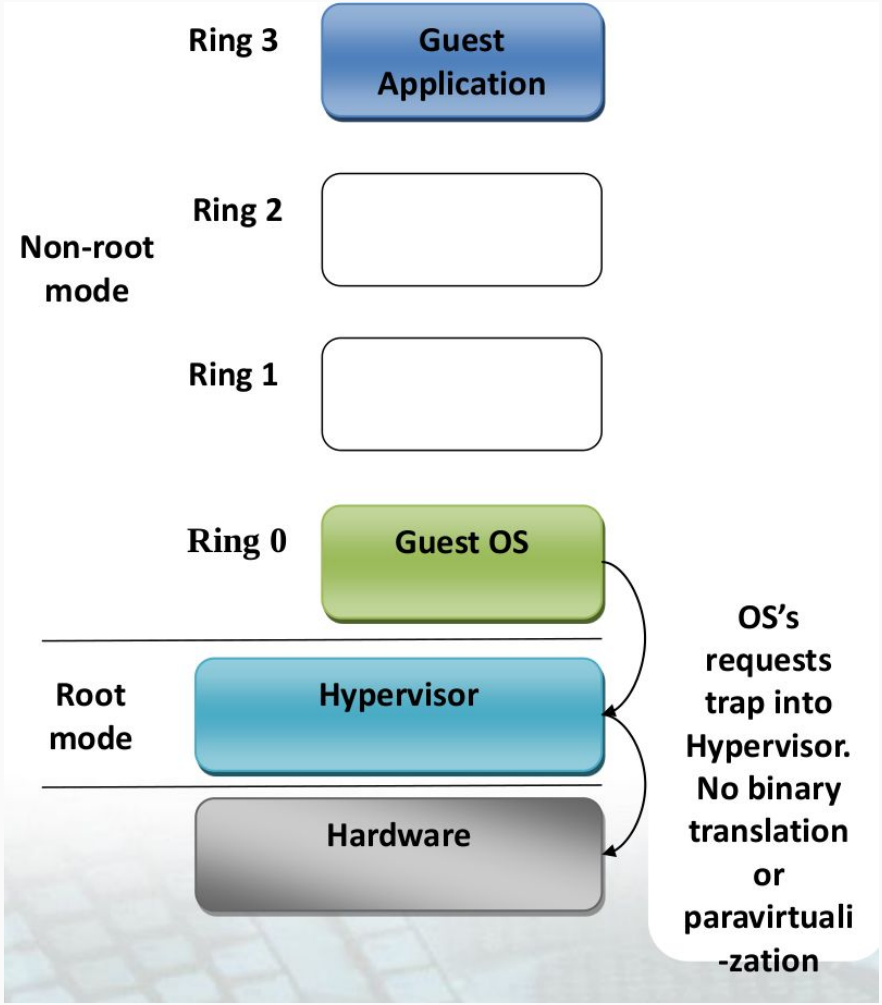


Full Virtualization with hardware assist

Bentuk virtualisasi penuh dengan dukungan atau bantuan perangkat keras:

- teknologi **VT-x** pada prosesor Intel
- **AMD-V** pada prosesor AMD.

Dukungan dari prosesor terhadap teknologi virtualisasi ini adalah pengambilalihan tugas menghadang perintah-perintah yang mengakses memori atau perangkat keras secara langsung yang dilakukan pada teknologi virtualisasi penuh (full virtualization) dengan implementasi melalui software menjadi salah satu fitur perangkat keras hardware.



CPU perlu mendukung teknologi virtualisasi. Selain ring 0 hingga dering 3, CPU perlu menyediakan ring tambahan hanya untuk Hypervisor yaitu ring -1.

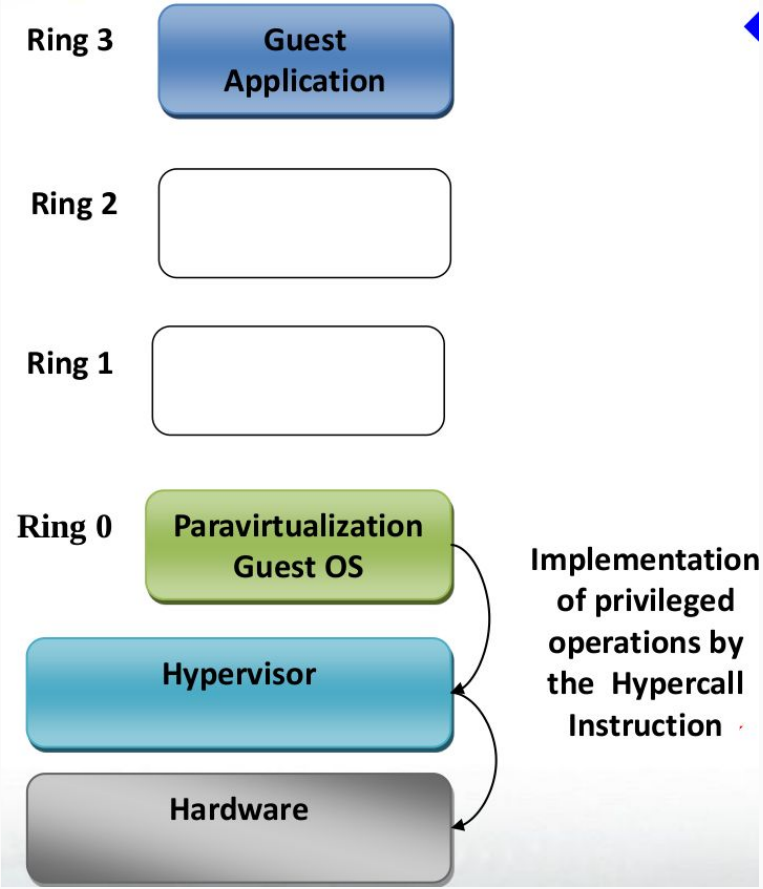
OS Tamu masih berjalan di ring 0, tetapi saat OS memanggil instruksi istimewa, mereka akan dialihkan ke Hypervisor di ring -1 melalui mekanisme perangkat keras, dan Hypervisor mengelola perangkat keras.

Para-Virtualization

OS Guest masih bisa berjalan di ring 0, tapi kita perlu memodifikasi kernel OS. Panggilan untuk instruksi yang memiliki hak istimewa memanggil hypervisor, yang disebut Hypercall. Contoh para-virtualisasi adalah Xen.

Jadi, ketika OS Guest di ring 0 memanggil instruksi yang diistimewakan, itu akan berubah menjadi **Hypercall**, tetapi hypervisor masih mengawasi sumber daya perangkat keras sistem.

Para-Virtualization



Implementasi teknologi

VMM (Hypervisor)

- Virtualization CPU
- Virtualization memory
- Virtualization I/O

Selesai