COLLEGE of COMPUTER STUDIES

Platform Architecture

Layers of a Computing Platform

- **Hardware**
  - Physical components (CPU, memory, storage, peripherals)
- **Firmware**
  - BIOS/UEFI, device firmware, initialization processes
- **Operating System (OS)**
  - Manages hardware, runs applications, enforces security
- **Applications**
  - Software that provides user-level services

Mobile vs. Desktop Platforms

- **Mobile Platforms**
  - Optimized for portability & battery life
  - Tight integration of hardware & software
  - App sandboxing and stricter permissions (Android/iOS)
- **Desktop Platforms**
  - Higher performance & flexibility
  - Broader software ecosystem
  - More customization, but larger attack surface

Cloud & Virtualization as Platforms

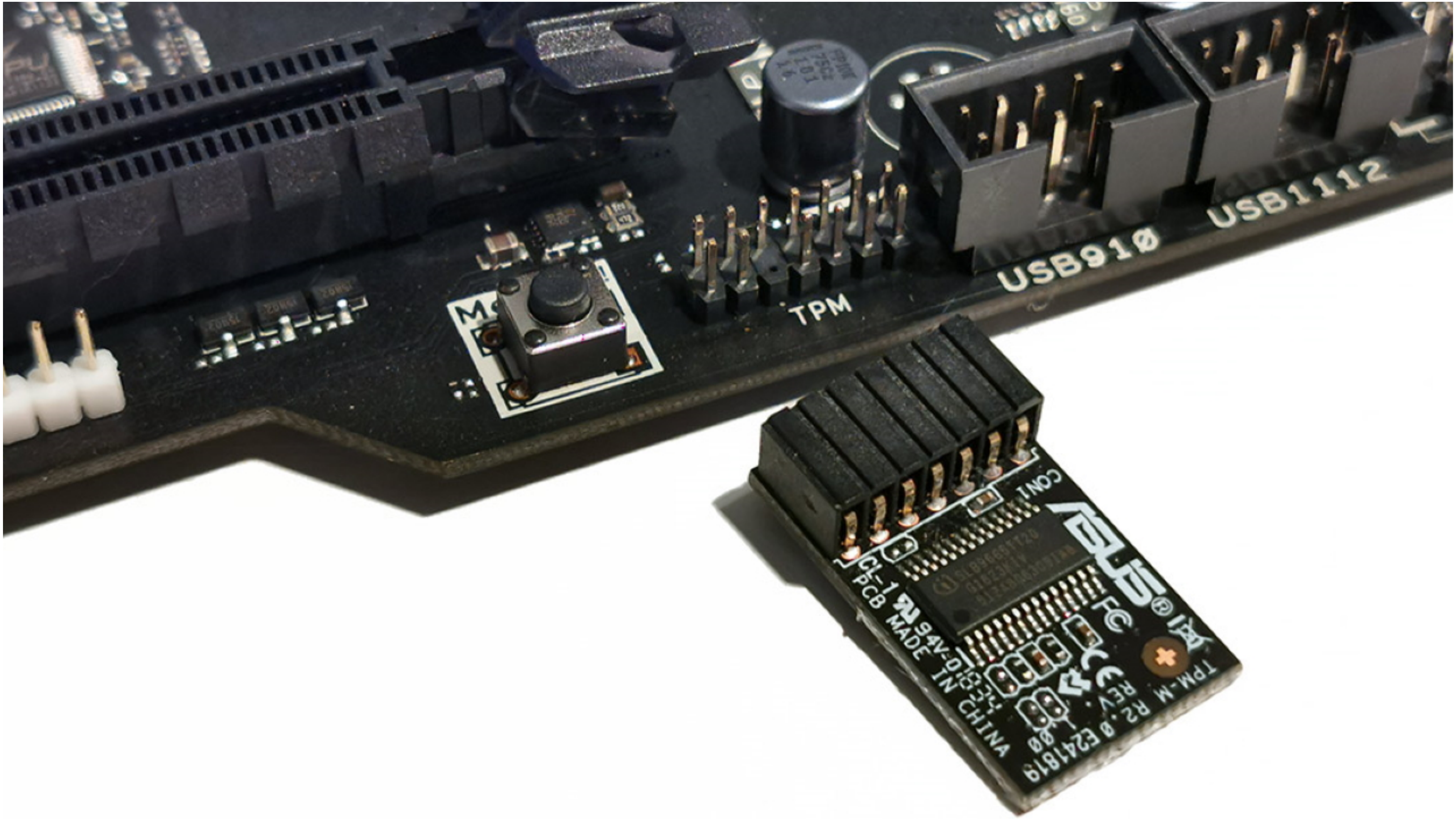
- **Cloud Platforms**
  - Abstract physical resources, provide services on demand (AWS, Azure)
  - Security relies on shared responsibility model
- **Virtualization Platforms**
  - Hypervisors enable multiple VMs on one physical machine
  - Increases resource efficiency but adds risks (VM escape, misconfigurations)

Sample Platform Attack Surface

- **Hardware:** Side-channel attacks, physical tampering
- **Firmware:** BIOS/UEFI rootkits, firmware backdoors
- **OS:** Privilege escalation, kernel exploits
- **Applications:** Malware, unpatched software vulnerabilities

COLLEGE of COMPUTER STUDIES

Trusted Platform Module (TPM)

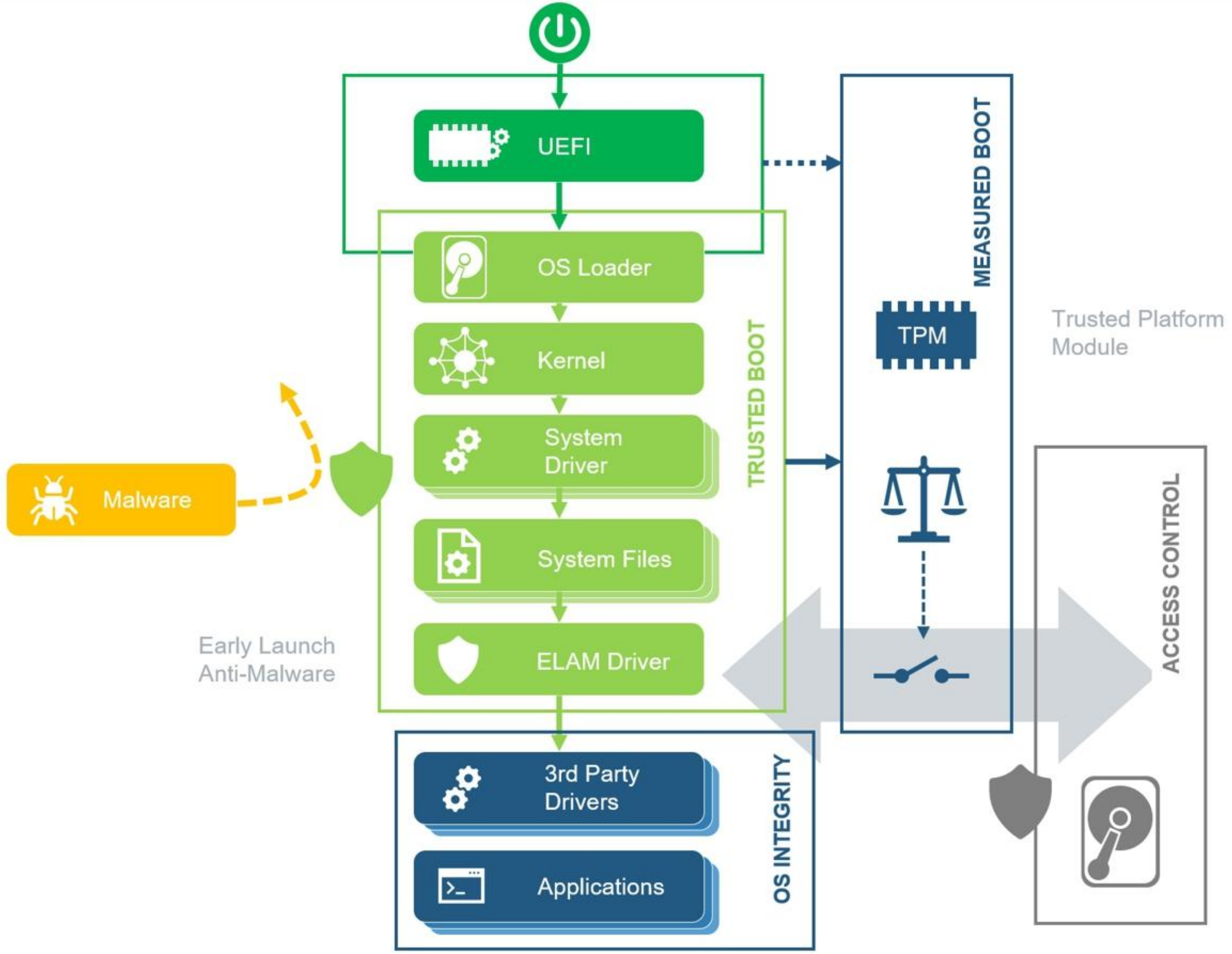

TPM
USB910
USB1112
CON1
TPM-M
MADE IN CHINA

Trusted Platform Module

- TPM is a hardware-based security chip
- Provides a Root of Trust for a computing platform
- Functions independently of the main CPU and OS
- Ensures secure storage, encryption, and attestation

Why TPM Matters

- **Keeps secrets safe:**
  TPM stores important “digital keys,” certificates, and passwords in a secure chip, so hackers can’t easily steal them. (Think of it like a tiny safe built into your computer.)
- **Helps your computer start safely (Secure Boot):**
  When you turn on your computer, TPM checks that the system hasn’t been tampered with before letting it load.
- **Blocks hidden attacks in startup (firmware/boot-level attacks):**
  Hackers sometimes try to sneak malware in the system before Windows or Linux even loads. TPM helps stop these early attacks


UEFI
OS Loader
Kernel
System Driver
System Files
ELAM Driver
TRUSTED BOOT
Malware
Early Launch Anti-Malware
MEASURED BOOT
TPM
Trusted Platform Module
ACCESS CONTROL
3rd Party Drivers
Applications
OS INTEGRITY

Unified Extensible Firmware Interface (UEFI)

- It's basically the modern replacement for BIOS (Basic Input/Output System), which is the traditional firmware that starts your computer when you power it on.
- When you press the power button, UEFI is the first program that runs.
- It initializes the hardware (CPU, RAM, drives, etc.) and then hands control over to the operating system (Windows, Linux, etc.).
- Think of UEFI as the bridge between the hardware and the OS.

Chain of Trust

- A process where each stage of the boot process checks the integrity of the next stage before handing control.

1. **Root of Trust (TPM/firmware key)** – trust anchor
2. **Firmware checks bootloader** (must be signed/verified)
3. **Bootloader checks OS kernel**
4. **OS verifies drivers/applications**

- If any step fails, boot process is stopped or alerts are raised

TPM Versions

- **TPM 1.2:** Basic cryptographic support (SHA-1, RSA)
- **TPM 2.0:** Stronger crypto algorithms (SHA-256, ECC), flexible authorization

TPM 2.0 is required for:

- Windows 11
- Modern enterprise-grade security solutions

TPM Security Benefits

- Protects against:
  - Rootkits & bootkits
  - Key theft
  - Unauthorized firmware changes
- Provides **hardware-enforced trust** instead of relying on software-only security

TPM Limitations

- Requires hardware support (not all devices have it)
- If TPM fails, access to encrypted data may be lost
- Cannot stop OS-level malware once system is booted
- Physical attacks on the chip (advanced threat) still possible

COLLEGE of COMPUTER STUDIES

ITA 216 Platform Security

Virtualization

Ms. Kezia Abegail T. Velasco

SY 2024-2025

Virtualization

Virtualization is a powerful technology that enables multiple virtual machines (VMs) to run concurrently on a single physical server.

Consolidated Hardware

- One physical server hosts many virtual environments.

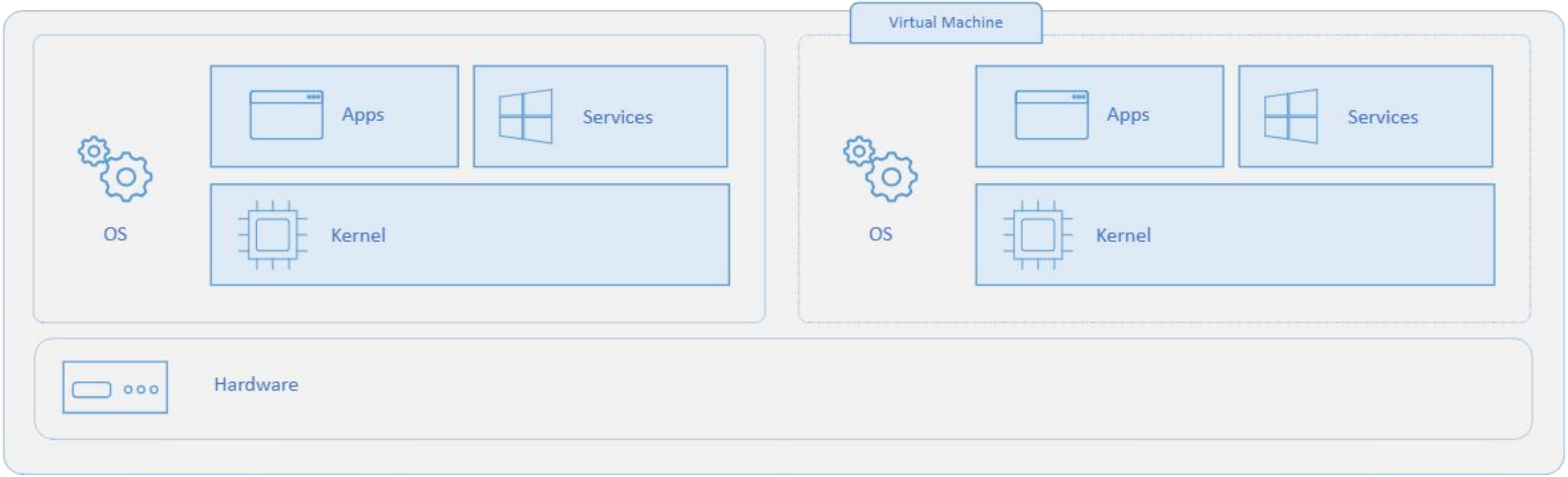
Isolated Environments

- Each VM operates independently with its own OS and applications.

Maximized Utilization

- Optimizes the use of underlying hardware resources.

Virtual Machine



Types of Virtualization

1. Server Virtualization
- Multiple virtual servers on one physical server
- Most common type

2. Desktop Virtualization
- Virtual desktop infrastructure (VDI)
- Remote desktop access

3. Network Virtualization
- Virtual networks independent of physical hardware

4. Storage Virtualization
- Pool storage from multiple devices

5. Application Virtualization
- Applications run in isolated environments

Types of Hypervisor

The core of virtualization is the **hypervisor**, a software layer that manages and allocates physical hardware resources to virtual machines, ensuring their isolation and efficient operation.

Type 1: Bare-Metal

- Runs directly on hardware, offering high performance and security (e.g., VMware ESXi, Microsoft Hyper-V).

Type 2: Hosted

- Runs on top of a host operating system (e.g., VirtualBox, VMware Workstation).

Benefits of Virtualizations

Virtualization offers significant advantages for modern businesses:

1. **Cost Reduction**
2. **Increased Agility**
3. **Enhanced Disaster Recovery**
4. **Improved Management**
5. **Testing & Compatibility**

Popular Virtualization Platforms

Enterprise Solutions:

- VMware vSphere
- Microsoft Hyper-V
- Citrix XenServer
- Red Hat Virtualization

Desktop/Development:

- VMware Workstation/Fusion

Security Risks of Virtualization

- Hyperjacking (compromised hypervisor)
- Escaping from VM to host
- Misconfiguration vulnerabilities

Hyperjacking (Compromised Hypervisor)

A cyberattack where the attacker gains control over the hypervisor itself, effectively taking over all hosted virtual machines.

- **Why it’s dangerous:** The hypervisor has the highest privilege level — if compromised, attackers can monitor, manipulate, or shut down all VMs.
- **Real-world example:** A malicious hypervisor installed underneath an existing OS (a “blue pill” attack) to control the system invisibly.

Countermeasures:

1. Apply regular hypervisor patching.
2. Limit admin/root access with MFA and strict role separation.

Escaping from VM to Host (VM Escape)

An attack where malicious code running inside a virtual machine breaks isolation and gains access to the hypervisor or host system.

Why it's dangerous:

- Once the attacker reaches the host, they can control all other VMs.
- This violates the core promise of virtualization — isolation.

Misconfiguration Vulnerabilities

Weaknesses introduced not by flaws in the hypervisor software, but by incorrect or insecure configuration by administrators.

Examples of risky misconfigurations:

1. Assigning too many privileges to VM users (e.g., unrestricted root/admin rights).
2. Improperly configured virtual networks (e.g., flat networks without VLANs or segmentation).
3. Weak or default management console credentials.
4. Overcommitting resources (CPU, RAM) leading to denial-of-service (DoS) attacks.

VM Isolation Techniques

1. Strong Separation: Each VM Must Act as an Independent System
2. Resource Allocation Controls: CPU, RAM, Storage Quotas
3. Access Control: Prevent VM-to-VM Unauthorized Access
4. Snapshots Monitoring: Detect Rollback Attacks

Secure VM Networking

- Virtual switches: Control VM traffic
- Segmentation: VLANs for separating workloads
- Firewall rules: Per-VM or per-network segment
- IDS/IPS integration: Detect malicious VM traffic

VM Snapshots and Rollback Security

- **Snapshots:** Save system state for recovery/testing
- **Risks:**
  - Rollback to vulnerable versions
  - Exposure of sensitive data in snapshots
- **Best Practices:**
  - Encrypt snapshots
  - Monitor and control snapshot creation
  - Regularly patch and update after rollback

Best Practices for Virtualization Security

- Keep hypervisor updated and patched
- Enforce strict access controls
- Use security baselines: CIS (Center of Internet Security), NIST – (National Institute of Standards and Technology)
- Monitor VM behavior with logs & alerts
- Encrypt VM images and storage