SNS COLLEGE OF TECHNOLOGY

An Autonomous Institution

Coimbatore-35

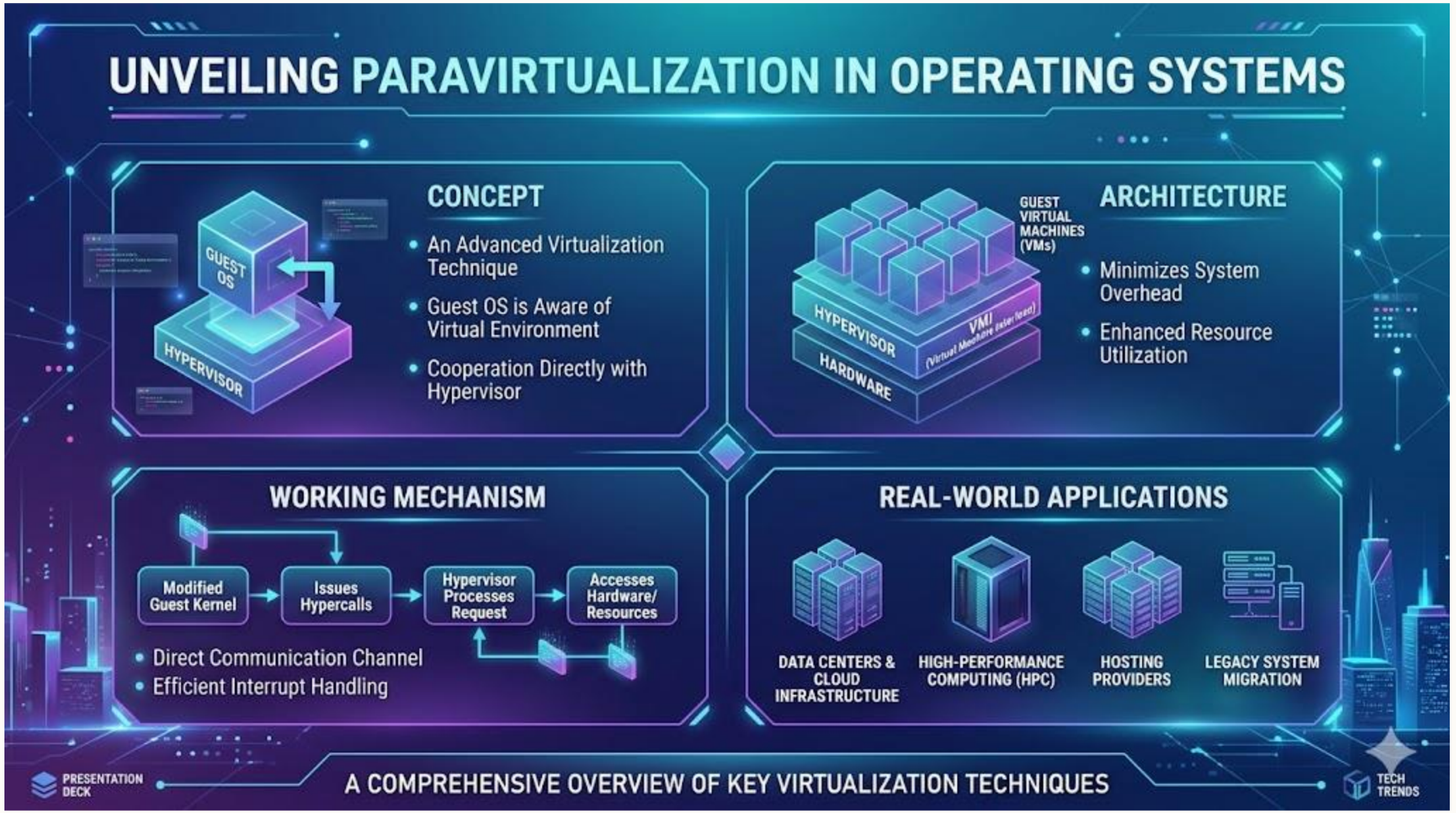
Department of Computer Science and Engineering

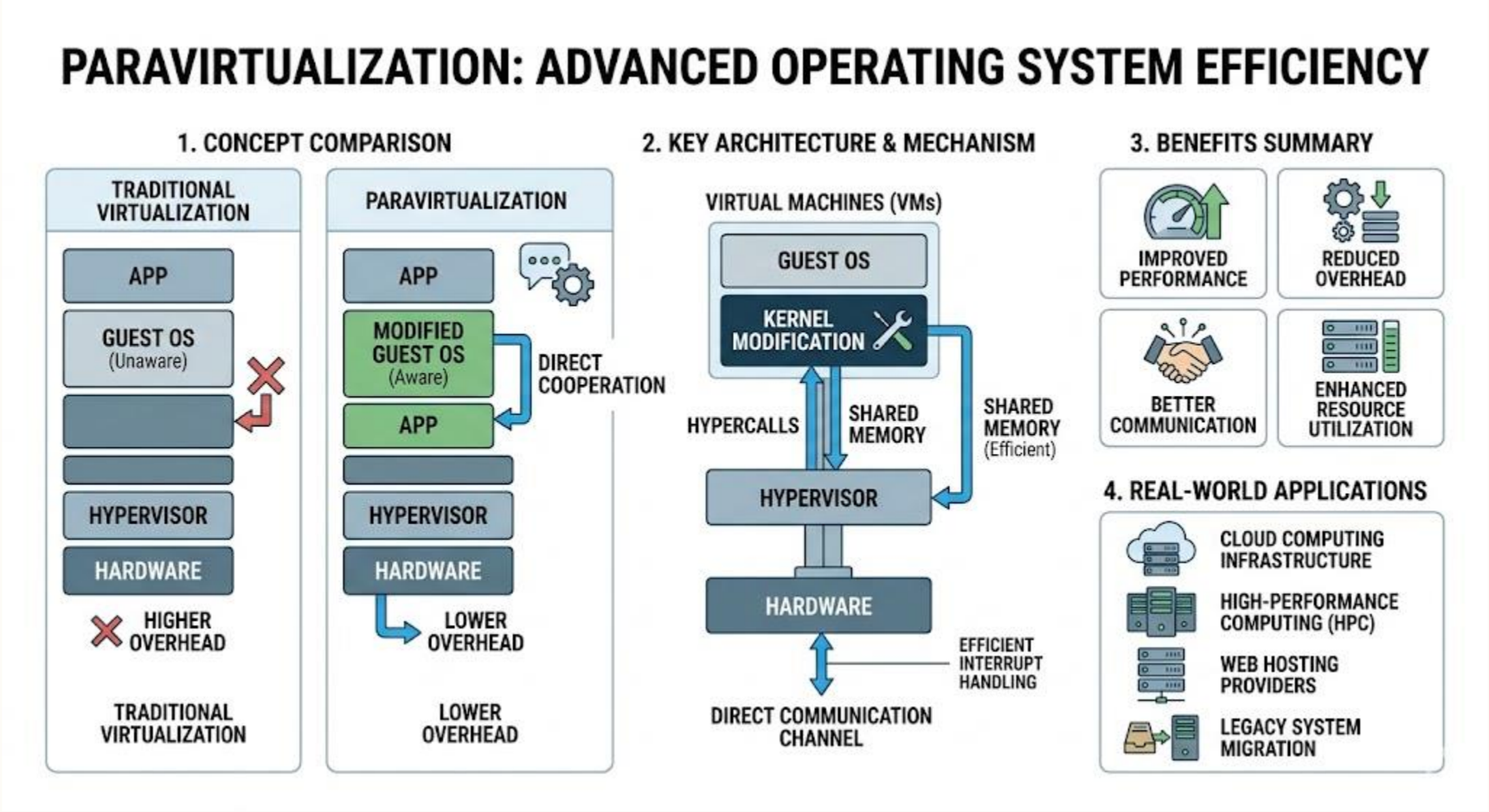
23CST206–OPERATING SYSTEMS AND VIRTUALIZATION

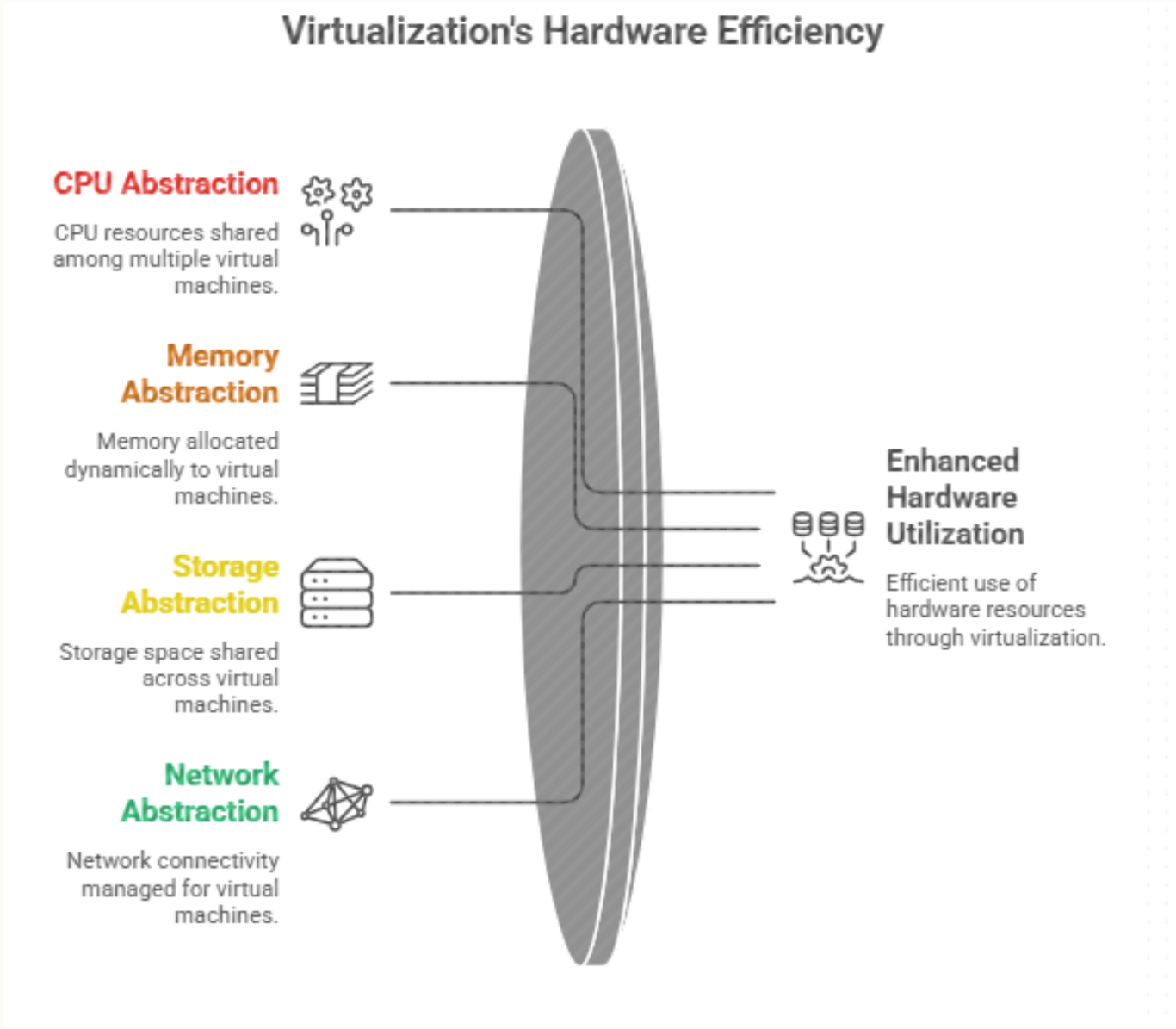
B.E- CSE /IV SEMESTER

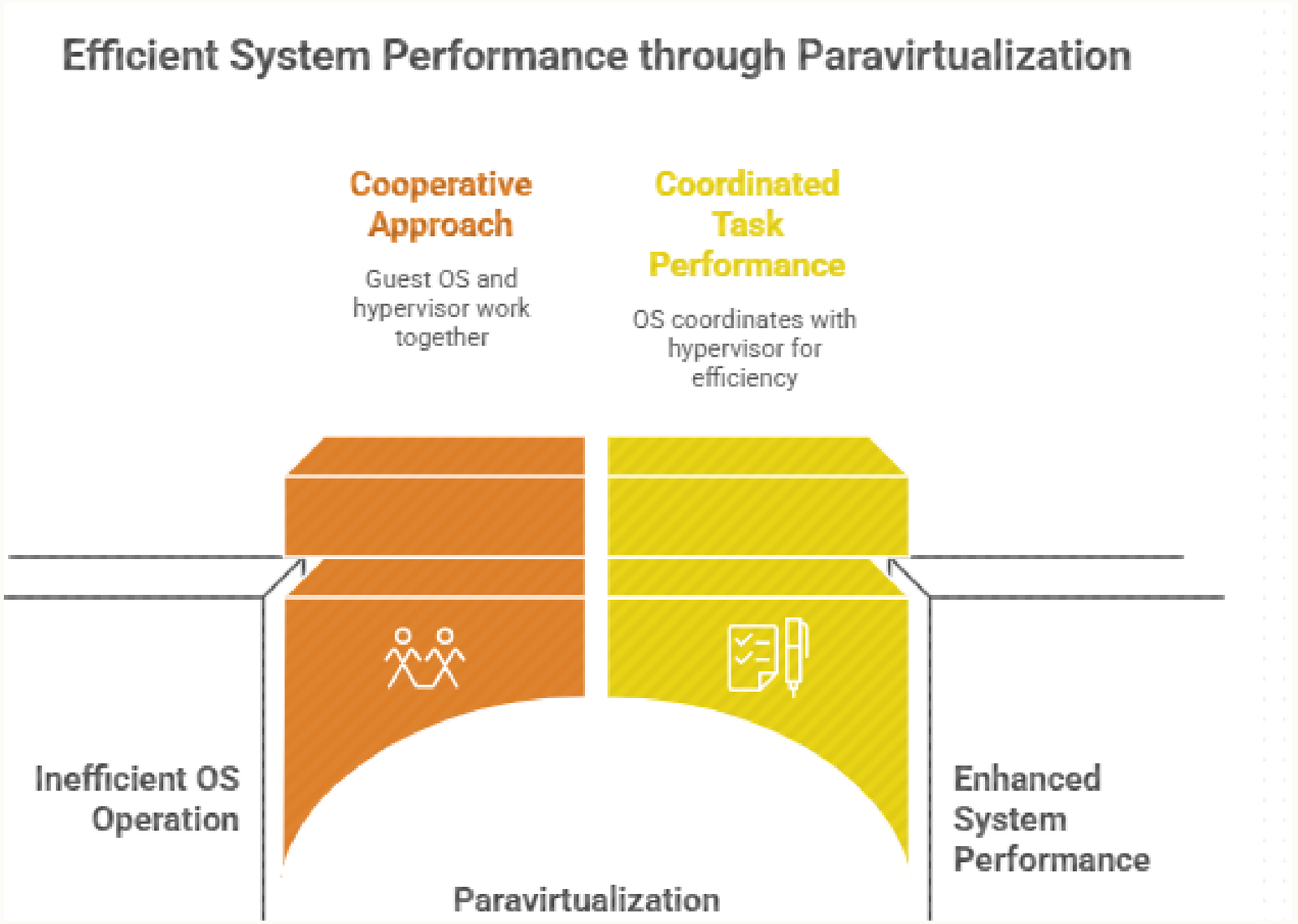
UNIT – V HYPERVISOR

Topic 4:Paravirtualization


UNVEILING PARAVIRTUALIZATION IN OPERATING SYSTEMS
CONCEPT
GUEST OS
HYPERVISOR
An Advanced Virtualization Technique
Guest OS is Aware of Virtual Environment
Cooperation Directly with Hypervisor
ARCHITECTURE
GUEST VIRTUAL MACHINES (VMs)
HYPERVISOR
VMI
(Virtual Machine interface)
HARDWARE
Minimizes System Overhead
Enhanced Resource Utilization
WORKING MECHANISM
Modified Guest Kernel
Issues Hypercalls
Hypervisor Processes Request
Accesses Hardware/ Resources
Direct Communication Channel
Efficient Interrupt Handling
REAL-WORLD APPLICATIONS
DATA CENTERS & CLOUD INFRASTRUCTURE
HIGH-PERFORMANCE COMPUTING (HPC)
HOSTING PROVIDERS
LEGACY SYSTEM MIGRATION
PRESENTATION DECK
A COMPREHENSIVE OVERVIEW OF KEY VIRTUALIZATION TECHNIQUES
TECH TRENDS


PARAVIRTUALIZATION: ADVANCED OPERATING SYSTEM EFFICIENCY
1. CONCEPT COMPARISON
TRADITIONAL VIRTUALIZATION
APP
GUEST OS (Unaware)
HYPERVISOR
HARDWARE
HIGHER OVERHEAD
TRADITIONAL VIRTUALIZATION
PARAVIRTUALIZATION
APP
MODIFIED GUEST OS (Aware)
APP
DIRECT COOPERATION
HYPERVISOR
HARDWARE
LOWER OVERHEAD
LOWER OVERHEAD
2. KEY ARCHITECTURE & MECHANISM
VIRTUAL MACHINES (VMs)
GUEST OS
KERNEL MODIFICATION
HYPERCALLS
SHARED MEMORY
SHARED MEMORY (Efficient)
HYPERVISOR
HARDWARE
EFFICIENT INTERRUPT HANDLING
DIRECT COMMUNICATION CHANNEL
3. BENEFITS SUMMARY
IMPROVED PERFORMANCE
REDUCED OVERHEAD
BETTER COMMUNICATION
ENHANCED RESOURCE UTILIZATION
4. REAL-WORLD APPLICATIONS
CLOUD COMPUTING INFRASTRUCTURE
HIGH-PERFORMANCE COMPUTING (HPC)
WEB HOSTING PROVIDERS
LEGACY SYSTEM MIGRATION


Virtualization's Hardware Efficiency
CPU Abstraction
CPU resources shared among multiple virtual machines.
Memory Abstraction
Memory allocated dynamically to virtual machines.
Storage Abstraction
Storage space shared across virtual machines.
Network Abstraction
Network connectivity managed for virtual machines.
Enhanced Hardware Utilization
Efficient use of hardware resources through virtualization.


Efficient System Performance through Paravirtualization
Cooperative Approach
Guest OS and hypervisor work together
Coordinated Task Performance
OS coordinates with hypervisor for efficiency
Inefficient OS Operation
Enhanced System Performance
Paravirtualization

Which virtualization approach offers better performance?

Full Virtualization

Increased CPU usage due to hypervisor emulation.

Hypervisor must intercept and emulate privileged instructions.

Slower performance due to emulation overhead.

Every privileged instruction execution is intercepted.

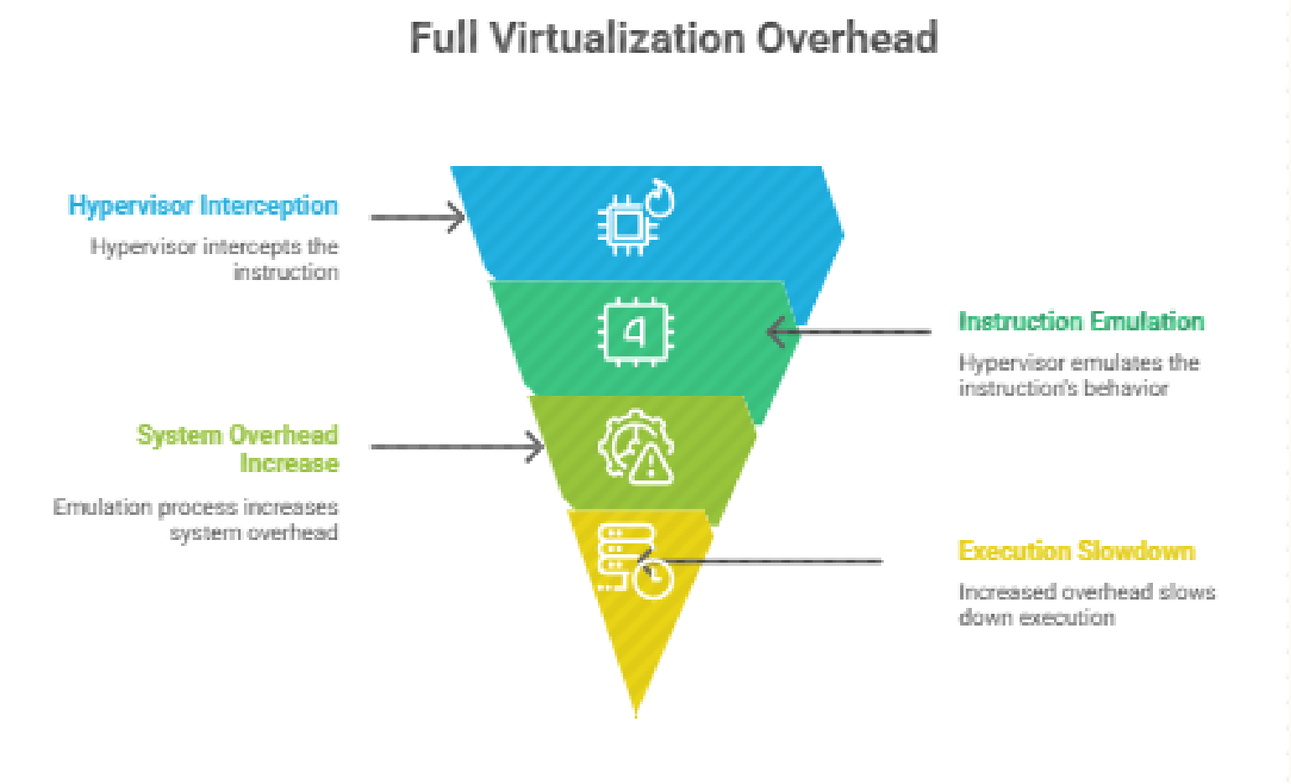
Paravirtualization

Reduced CPU usage by eliminating emulation.

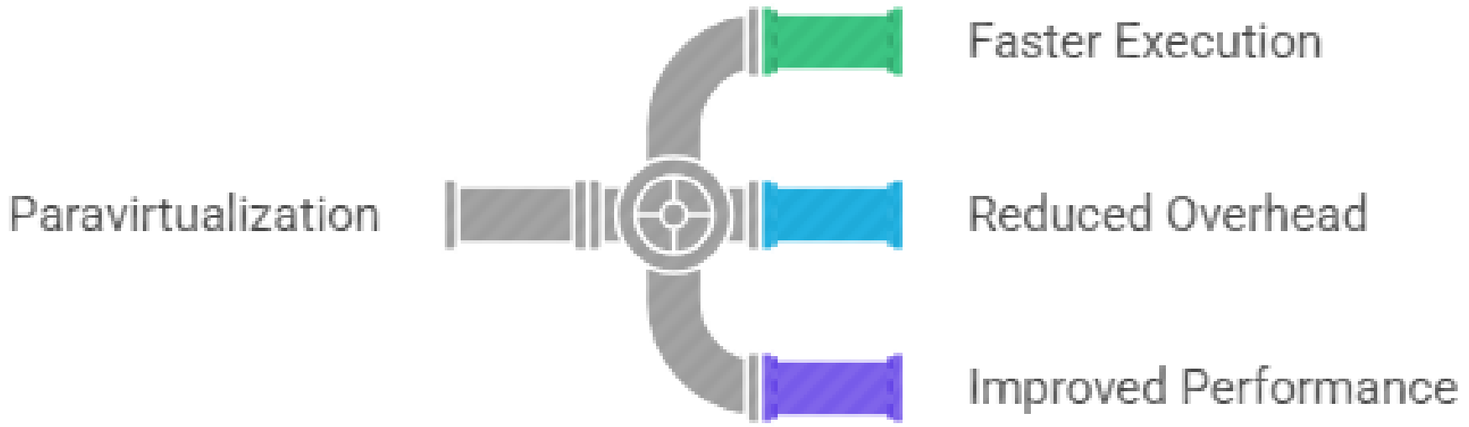
Enables direct communication between guest OS and hypervisor.

Faster performance by eliminating inefficiencies.

Direct communication bypasses emulation bottlenecks.

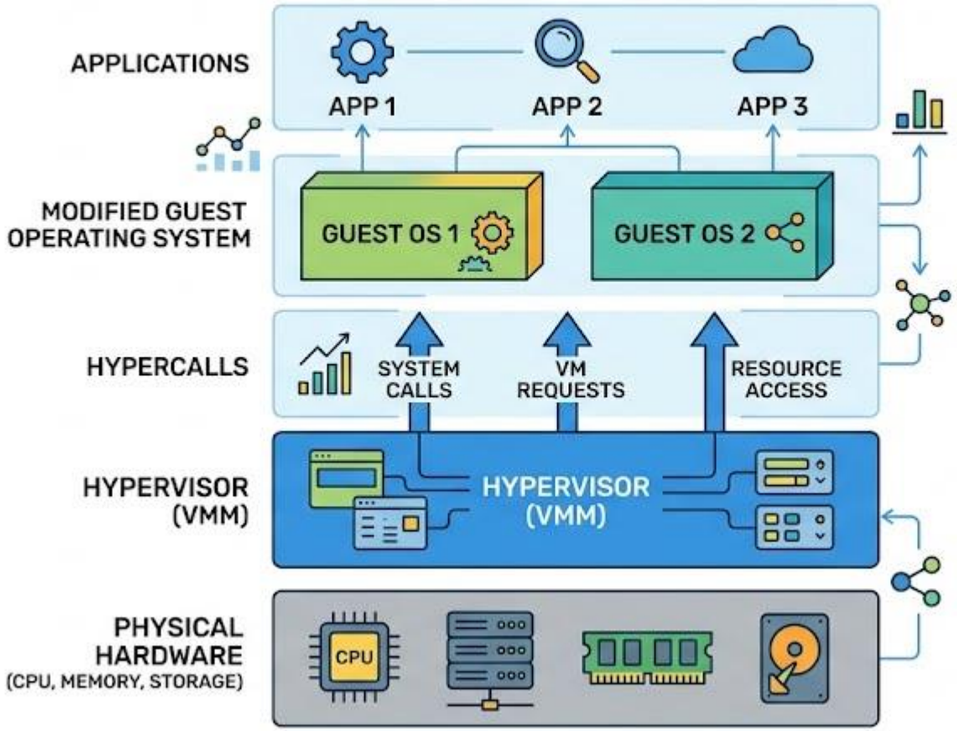

Full Virtualization Overhead
Hypervisor Interception
Hypervisor intercepts the instruction
Instruction Emulation
Hypervisor emulates the instruction's behavior
System Overhead Increase
Emulation process increases system overhead
Execution Slowdown
Increased overhead slows down execution

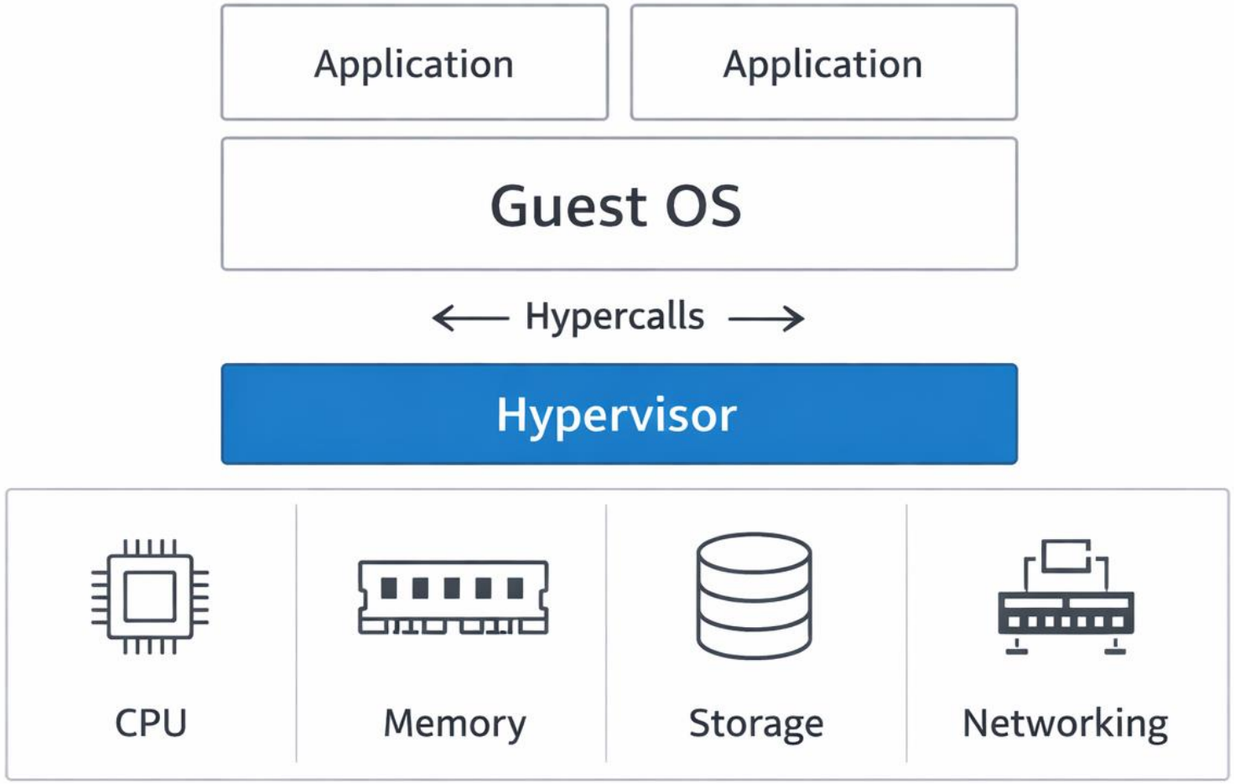
Unveiling the Benefits of Paravirtualization



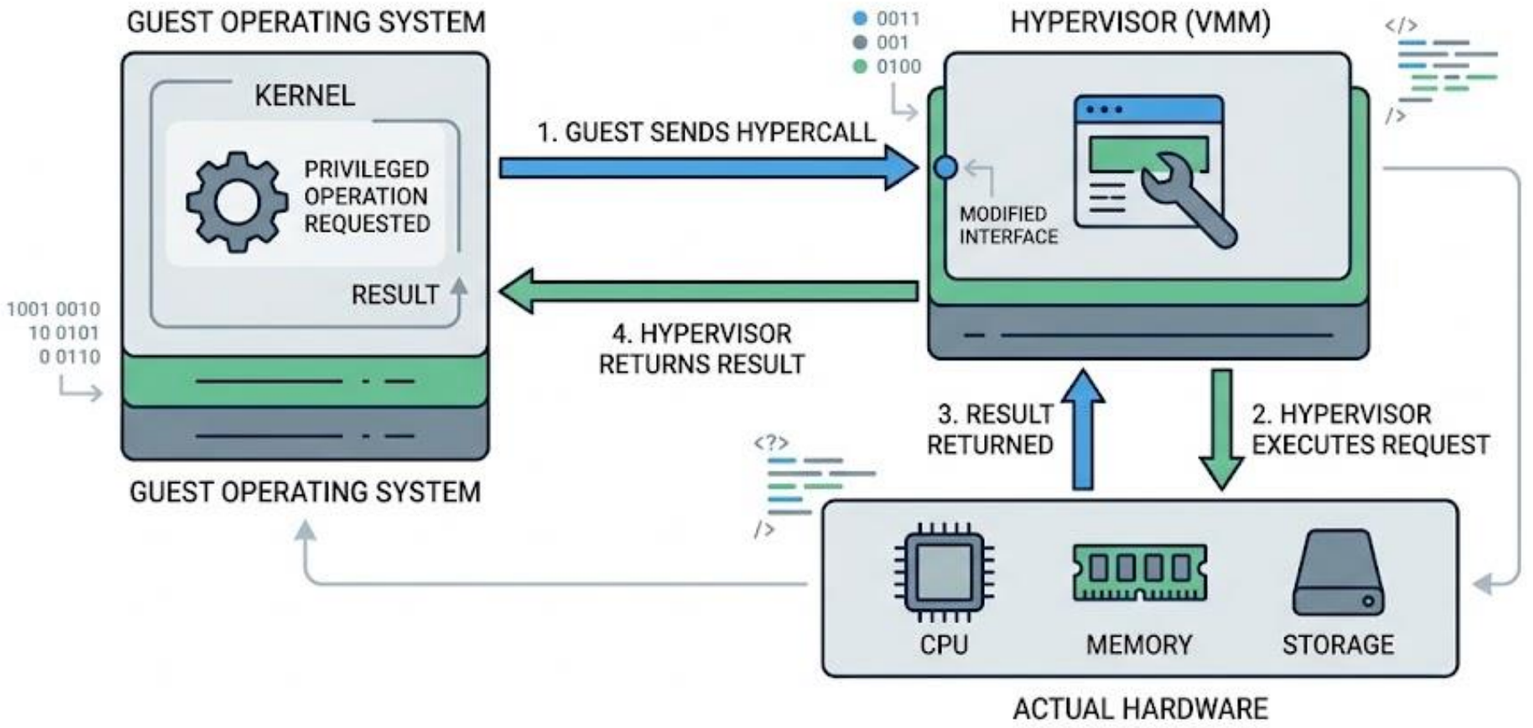
ARCHITECTURE OVERVIEW

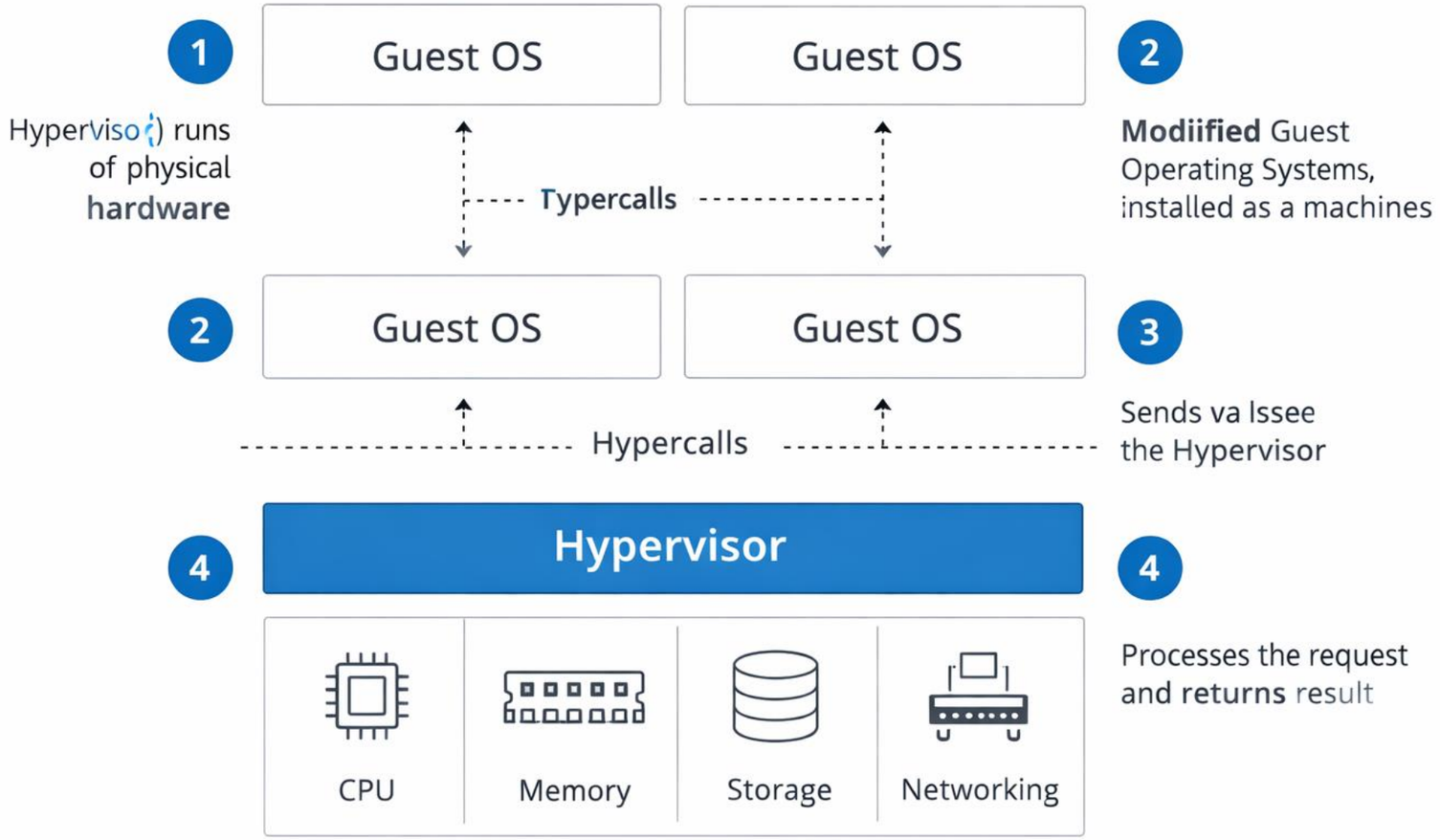
PARAVIRTUALIZED SYSTEM ARCHITECTURE

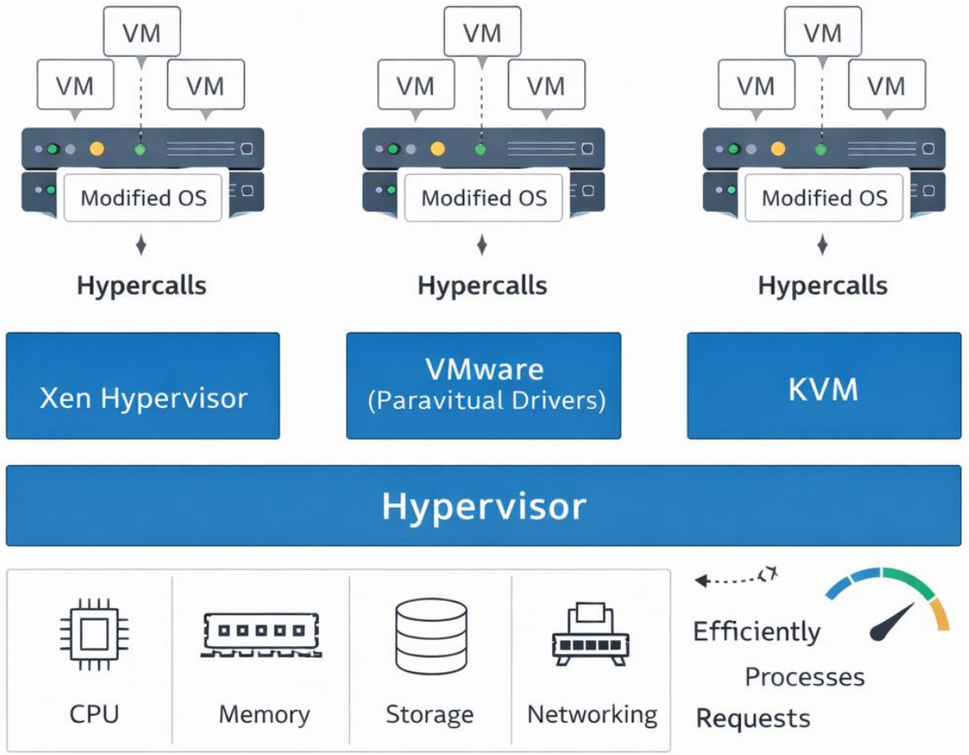


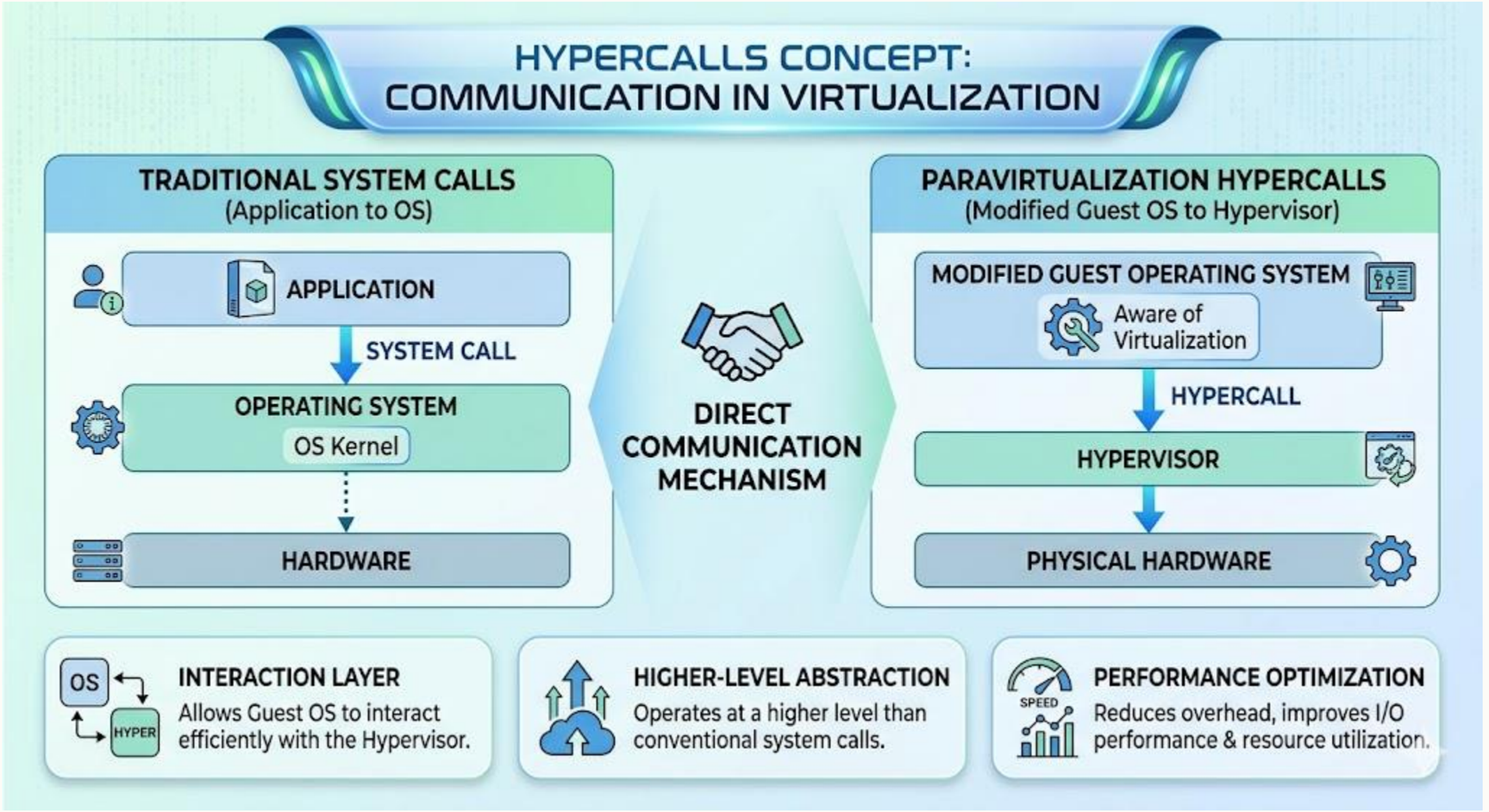

Application
Application
Guest OS
Hypercalls
Hypervisor
CPU
Memory
Storage
Networking

WORKING OF PARAVIRTUALIZATION

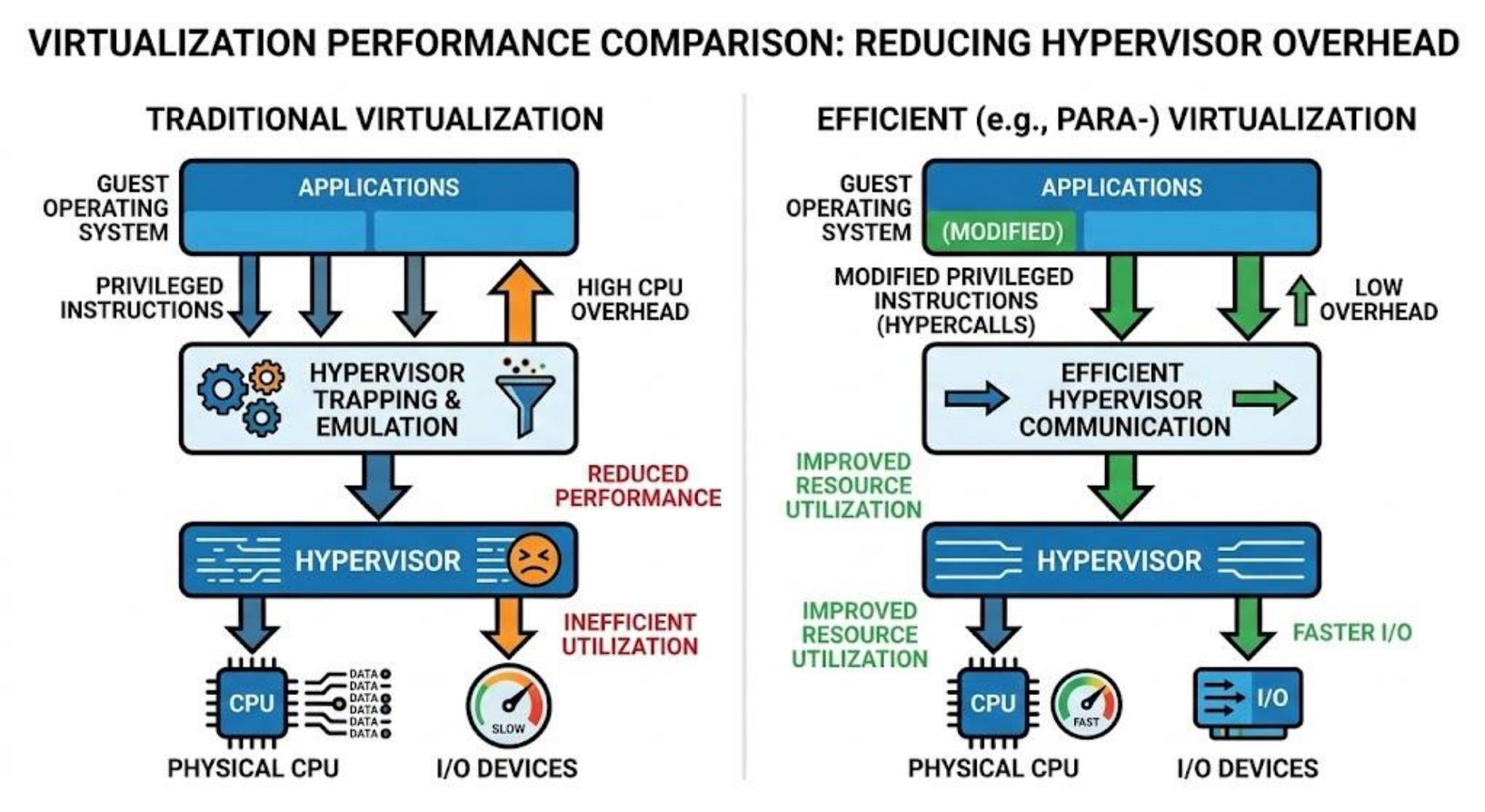



1
Guest OS
Guest OS
2
HyperViso) runs
of physical
hardware
Typercalls
Modiified Guest
Operating Systems,
installed as a machines
2
Guest OS
Guest OS
3
Hypercalls
Sends va Issee
the Hypervisor
4
Hypervisor
4
CPU
Memory
Storage
Networking
Processes the request
and returns result

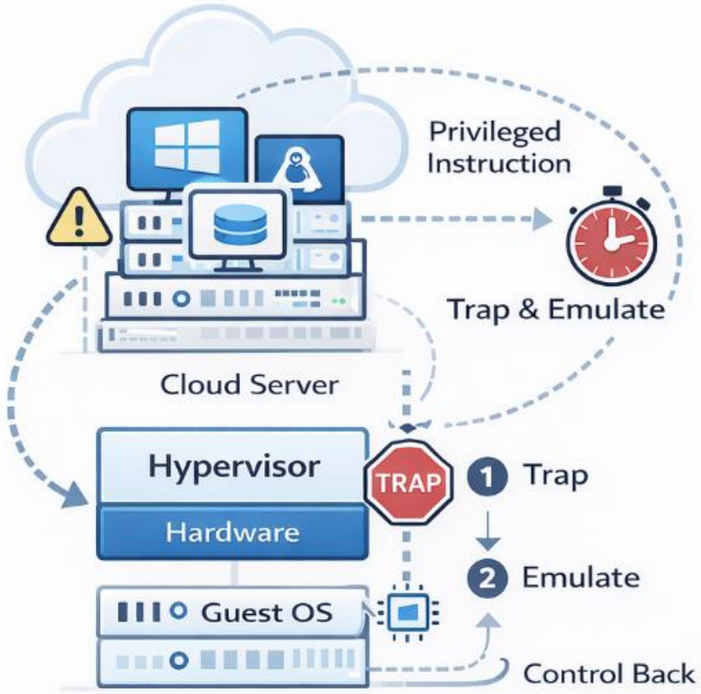

VM
VM
VM
Modified OS
Hypercalls
VM
VM
VM
Modified OS
Hypercalls
VM
VM
VM
Modified OS
Hypercalls
Xen Hypervisor
VMware
(Paravitual Drivers)
KVM
Hypervisor
CPU
Memory
Storage
Networking
Efficiently
Processes
Requests


HYPERCALLS CONCEPT:
COMMUNICATION IN VIRTUALIZATION
TRADITIONAL SYSTEM CALLS
(Application to OS)
APPLICATION
SYSTEM CALL
OPERATING SYSTEM
OS Kernel
HARDWARE
DIRECT COMMUNICATION MECHANISM
PARAVIRTUALIZATION HYPERCALLS
(Modified Guest OS to Hypervisor)
MODIFIED GUEST OPERATING SYSTEM
Aware of Virtualization
HYPERCALL
HYPERVISOR
PHYSICAL HARDWARE
OS
HYPER
INTERACTION LAYER
Allows Guest OS to interact efficiently with the Hypervisor.
HIGHER-LEVEL ABSTRACTION
Operates at a higher level than conventional system calls.
SPEED
PERFORMANCE OPTIMIZATION
Reduces overhead, improves I/O performance & resource utilization.

PROBLEM STATMENET



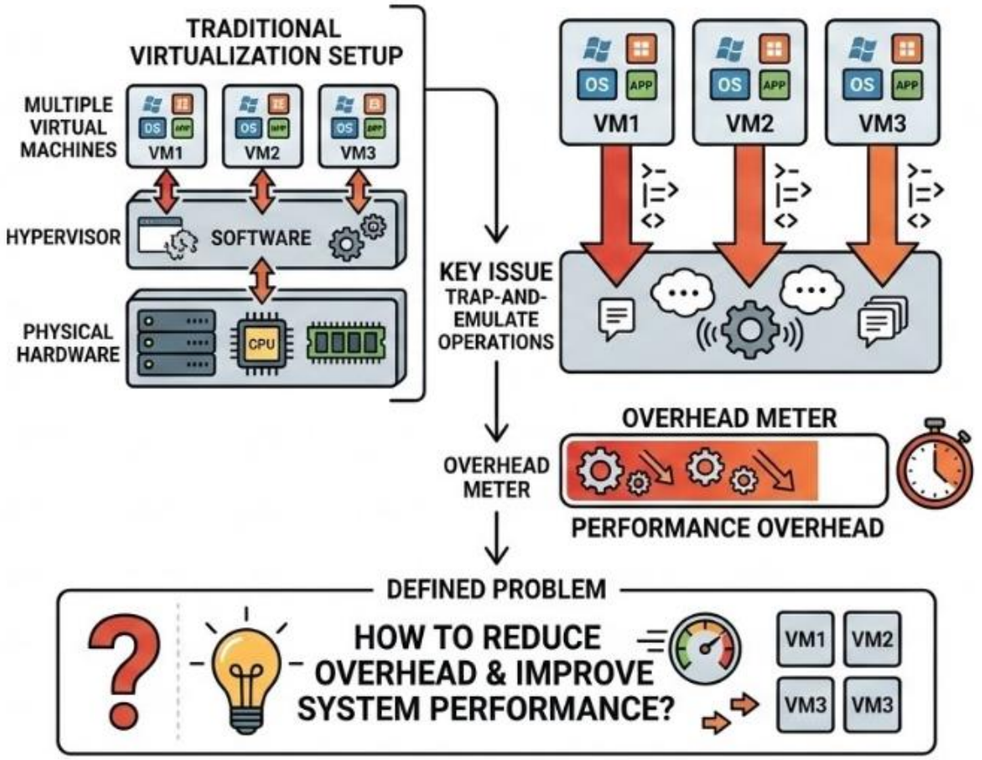
1. Empathize (Understanding the Problem)

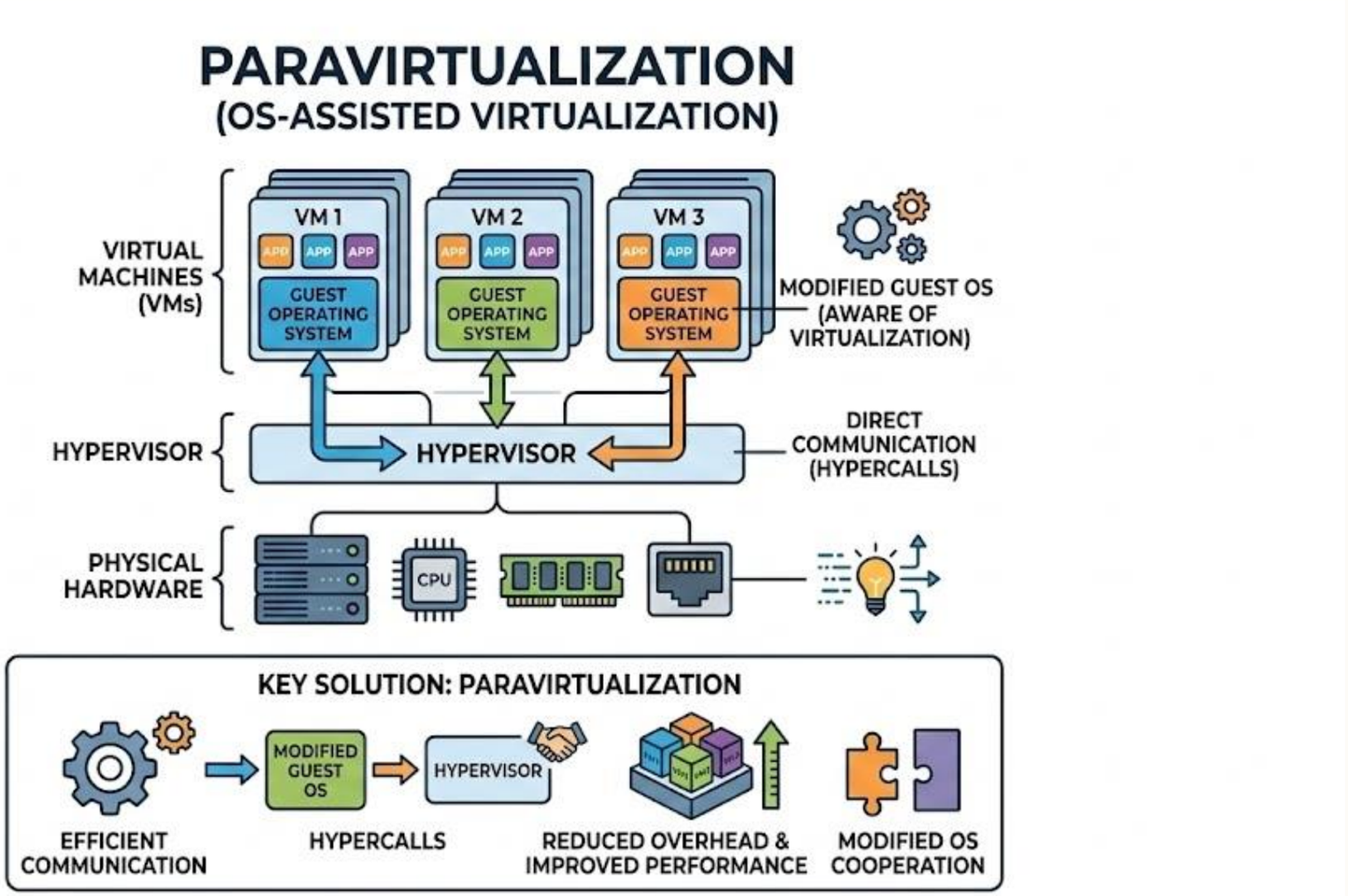


- High Overhead
- Slower I/O Performance
- Increased CPU Utilization
- Reduced Efficiency

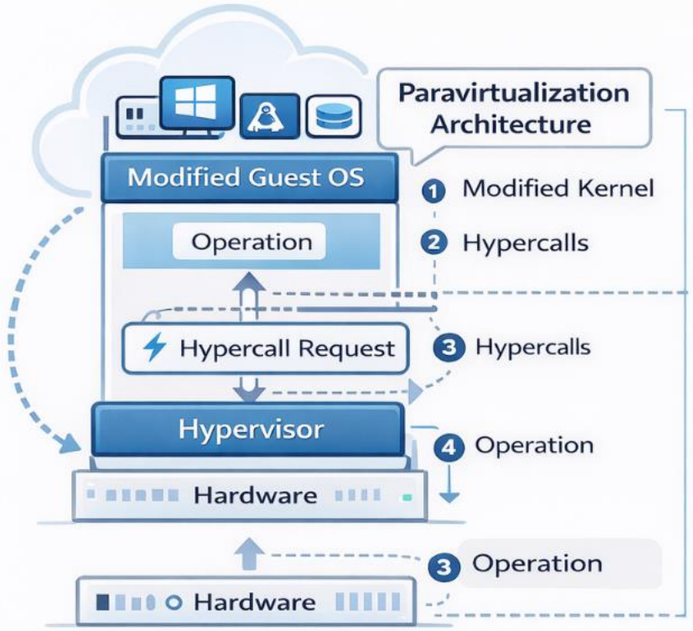
- High Overhead
- Slower I/O Performance
- Increased CPU Utilization
- Reduced Efficiency

2. DEFINE (CLEARLY DEFINING THE PROBLEM)




PARAVIRTUALIZATION
(OS-ASSISTED VIRTUALIZATION)
VIRTUAL MACHINES (VMs)
VM 1
VM 2
VM 3
APP
GUEST OPERATING SYSTEM
MODIFIED GUEST OS (AWARE OF VIRTUALIZATION)
HYPERVISOR
HYPERVISOR
DIRECT COMMUNICATION (HYPERCALLS)
PHYSICAL HARDWARE
CPU
KEY SOLUTION: PARAVIRTUALIZATION
MODIFIED GUEST OS
HYPERVISOR
EFFICIENT COMMUNICATION
HYPERCALLS
REDUCED OVERHEAD & IMPROVED PERFORMANCE
MODIFIED OS COOPERATION

4. Prototype (Developing the Solution)

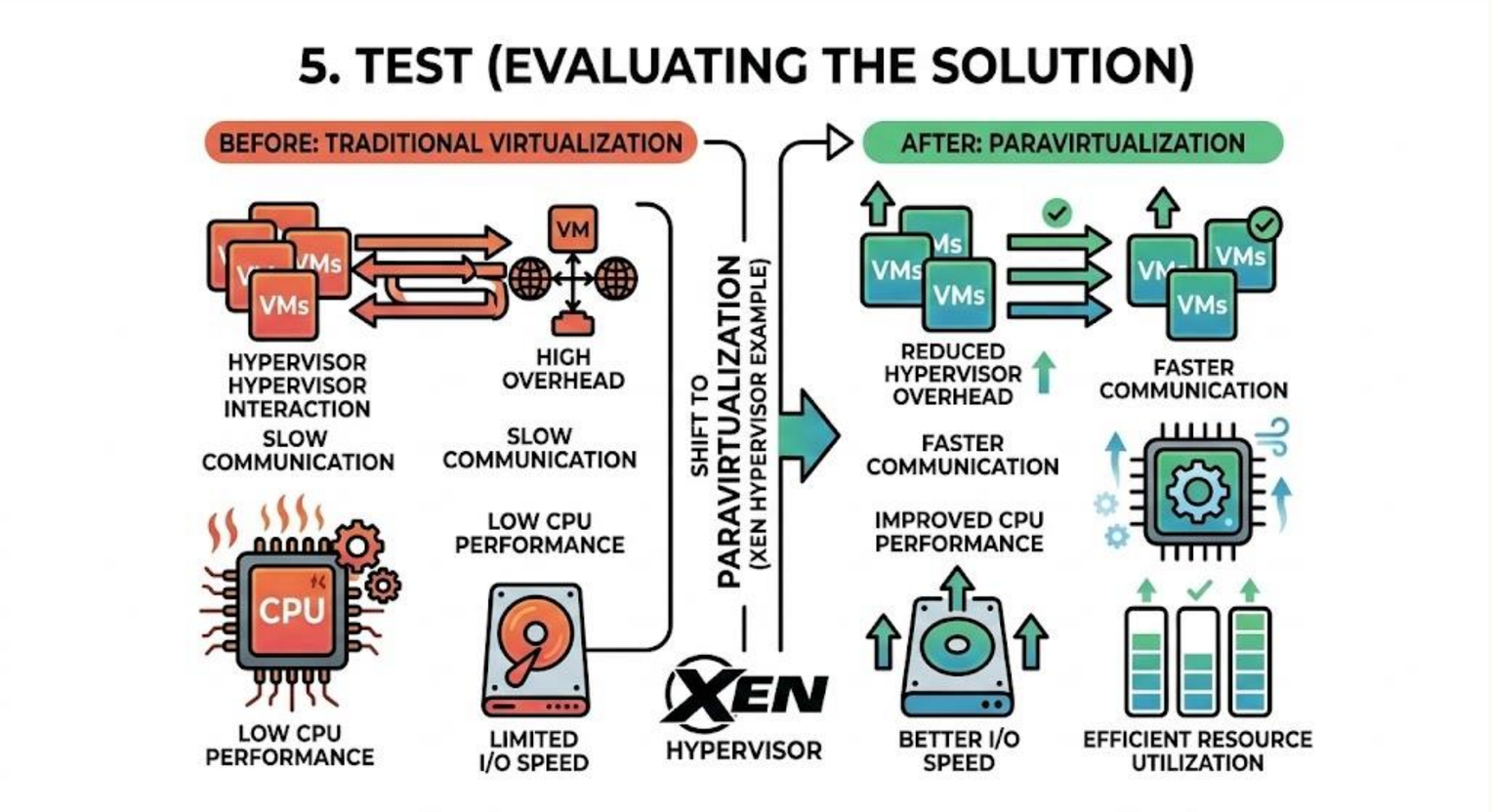


Key Design Features include:

- Modified Kernel
- Hypercalls
- Direct Communication
- Optimized Drivers

Working Process:

1. Modified *Kernel*
2. Hypercall Request
3. Direct Communication
4. Operation Executed


5. TEST (EVALUATING THE SOLUTION)
BEFORE: TRADITIONAL VIRTUALIZATION
AFTER: PARAVIRTUALIZATION
VMs
VM
HYPERVISOR
HYPERVISOR
INTERACTION
SLOW
COMMUNICATION
HIGH
OVERHEAD
SLOW
COMMUNICATION
LOW CPU
PERFORMANCE
CPU
LOW CPU
PERFORMANCE
LIMITED
I/O SPEED
SHIFT TO
PARAVIRTUALIZATION
(XEN HYPERVISOR EXAMPLE)
XEN
HYPERVISOR
REDUCED
HYPERVISOR
OVERHEAD
FASTER
COMMUNICATION
FASTER
COMMUNICATION
IMPROVED CPU
PERFORMANCE
BETTER I/O
SPEED
FASTER
COMMUNICATION
EFFICIENT RESOURCE
UTILIZATION

PARAVIRTUALIZATION IN OPERATING SYSTEMS
INTRODUCTION & DEFINITION
What is it?
Efficient virtualization technique
Key Feature
Modified Guest OS is aware of virtual environment
Cooperation
Guest OS & Hypervisor collaborate
EXAMPLES & IMPACT
Xen Project
Xen Hypervisor
Improved Performance
CPU, I/O
Cloud Environments
ARCHITECTURE & WORKING
Applications
Modified Guest OS
Hypervisor
Physical Hardware
Hypercalls
Direct communication method
Kernel Modification
Replace privileged instructions
Process
Operation -> Hypercall -> Hypervisor executes -> Result
PROBLEMS IN FULL VIRTUALIZATION
STOP
Trap-and-Emulate Mechanism
causes overhead
Guest OS Unaware
Increased intercepts
Performance Limitations
High CPU, slow I/O, delays with multiple VMs
SOLUTION & NEED
PARAVIRTUALIZATION SOLUTION
Modifies Guest OS
Eliminates Trap-and-Emulate
Benefits
Faster execution, reduced overhead, efficiency