Chapter 10

Approaches to Virtualization

Virtual Machines (VM)

- Virtualization technology enables a single PC or server to simultaneously run multiple operating systems or multiple sessions of a single OS
- A machine with virtualization software can host numerous applications, including those that run on different operating systems, on a single platform
- The host operating system can support a number of virtual machines, each of which has the characteristics of a particular OS
- The solution that enables virtualization is a ***virtual machine monitor (VMM),*** or ***hypervisor***

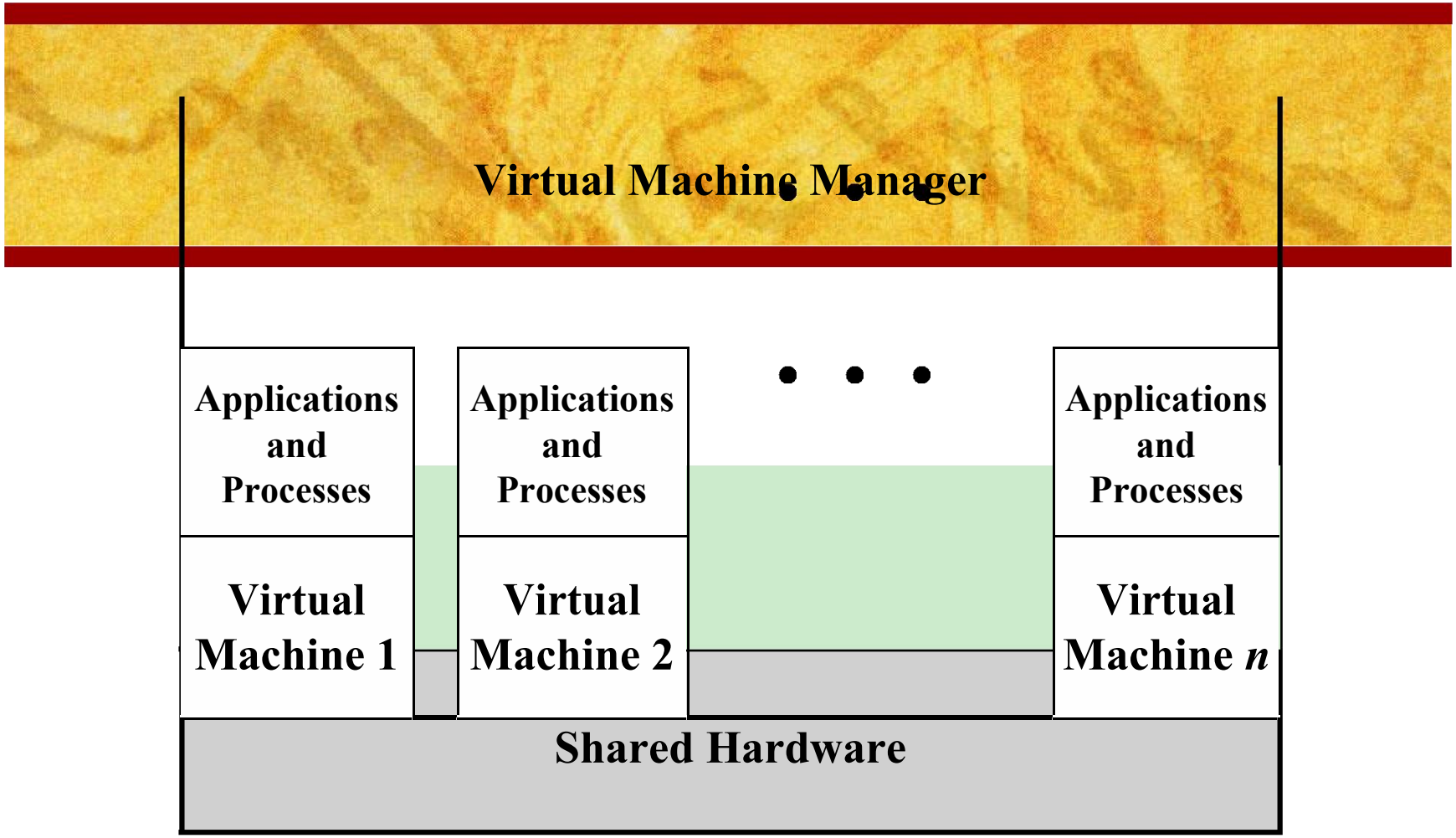

Figure 14.1 Virtual Machine Concept

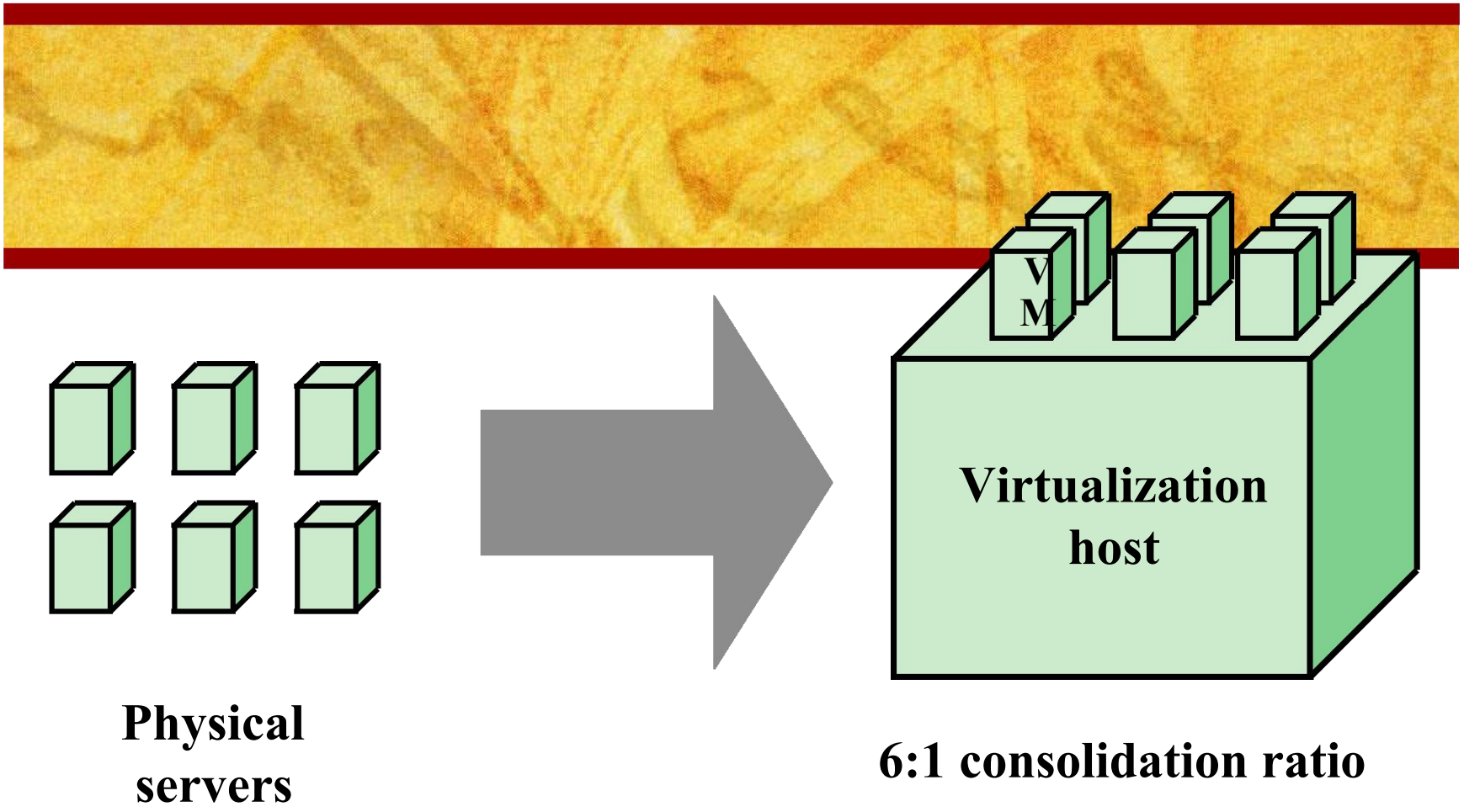

Figure 14.2 Virtual Machine Consolidation

Approaches to Virtualization

A Virtual Machine is a software construct that mimics the characteristics of a physical server

- it is configured with some number of processors, some amount of RAM, storage resources, and connectivity through the network ports
- once the VM is created it can be powered on like a physical server, loaded with an operating system and software solutions, and utilized in the manner of a physical server
- unlike a physical server, this virtual server only sees the resources it has been configured with, not all of the resources of the physical host itself
- the hypervisor facilitates the translation and I/O from the virtual machine to the physical server devices and back again to the correct virtual machine

Virtual Machine Files

Virtual machines are made up of files:

- configuration file describes the attributes of the virtual machine
- it contains the server definition, how many virtual processors (vCPUs) are allocated to this virtual machine, how much RAM is allocated, which I/O devices the VM has access to, how many network interface cards (NICs) are in the virtual server, and more
- it also describes the storage that the VM can access
- when a virtual machine is powered on, or instantiated, additional files are created for logging, for memory paging, and other functions
- since VMs are already files, copying them produces not only a backup of the data but also a copy of the entire server, including the operating system, applications, and the hardware configuration itself

Paravirtualization

- A software assisted virtualization technique that uses specialized APIs to link virtual machines with the hypervisor to optimize their performance
- The operating system in the virtual machine, Linux or Microsoft Windows, has specialized paravirtualization support as part of the kernel, as well as specific paravirtualization drivers that allow the OS and hypervisor to work together more efficiently with the overhead of the hypervisor translations
- Support has been offered as part of many of the general Linux distributions since 2008

Processor Issues

- In a virtual environment there are two main strategies for providing processor resources:
  - Emulate a chip as software and provide access to that resource
    - examples of this method are QEMU and the Android Emulator in the Android SDK

  - Provide segments of processing time on the physical processors (pCPUs) of the virtualization host to the virtual processors of the virtual machines hosted on the physical server
    - this is how most of the virtualization hypervisors offer processor resources to their guests

Processor Allocation

- when applications are migrated to virtual environments, the number of virtual processors allocated to their virtual machines needs to be determined

The number of processors a server has is one of the more important metrics when sizing a server

Moore's law provides processors that would be four times faster than those on the original physical server

If the consolidation estimate utility cannot be run, there are a number of good practices in place

There are tools available that will monitor resource (processor, memory, network, and storage I/O) usage on the physical server and then make recommendations for the optimum VM sizing

- one basic rule during VM creation is to begin with one vCPU and monitor the application's performance
- another good practice is not to overallocate the number of vCPUs in a VM

Ring O

- Native operating systems manage hardware by acting as the intermediary between application code requests and the hardware
- One key function of the operating system is to help prevent malicious or accidental system calls from disrupting the applications or the operating system itself
- Protection rings describe level of access or privilege inside of a computer system and many operating systems and processor architectures take advantage of this security model
- The most trusted layer is often called Ring 0 (zero) and is where the operating system kernel works and can interact directly with hardware
- Hypervisors run in Ring 0 controlling hardware access for the virtual machines they host

Memory Management

- Since hypervisor manages page sharing, the virtual machine operating systems are unaware of what is happening in the physical system
- Ballooning
  - the hypervisor activates a balloon driver that (virtually) inflates and presses the guest operating system to flush pages to disk
  - once the pages are cleared, the balloon driver deflates and the hypervisor can use the physical memory for other VMs
- Memory overcommit
  - the capability to allocate more memory than physical exists on a host

I/O Management

- An advantage of virtualizing the workload's I/O path enables hardware independence by abstracting vendor-specific drivers to more generalized versions that run on the hypervisor
- This abstraction enables:
  - live migration, which is one of virtualization's greatest availability strengths
  - the sharing of aggregate resources, such as network paths
- The memory overcommit capability is another benefit of virtualizing the I/O of a VM
- The trade-off for this is that the hypervisor is managing all the traffic and requires processor overhead
  - this was an issue in the early days of virtualization but now faster multicore processors and sophisticated hypervisors have addressed this concern

VMware ESXi

- A commercially available hypervisor from VMware that provides users a Type-1, or bare-metal, hypervisor to host virtual machines on their servers
- VMware developed their initial x86-based solutions in the late 1990s and were the first to deliver a commercial product to the marketplace
- This first-to-market timing, coupled with continuous innovations, has kept VMware firmly on top in market share

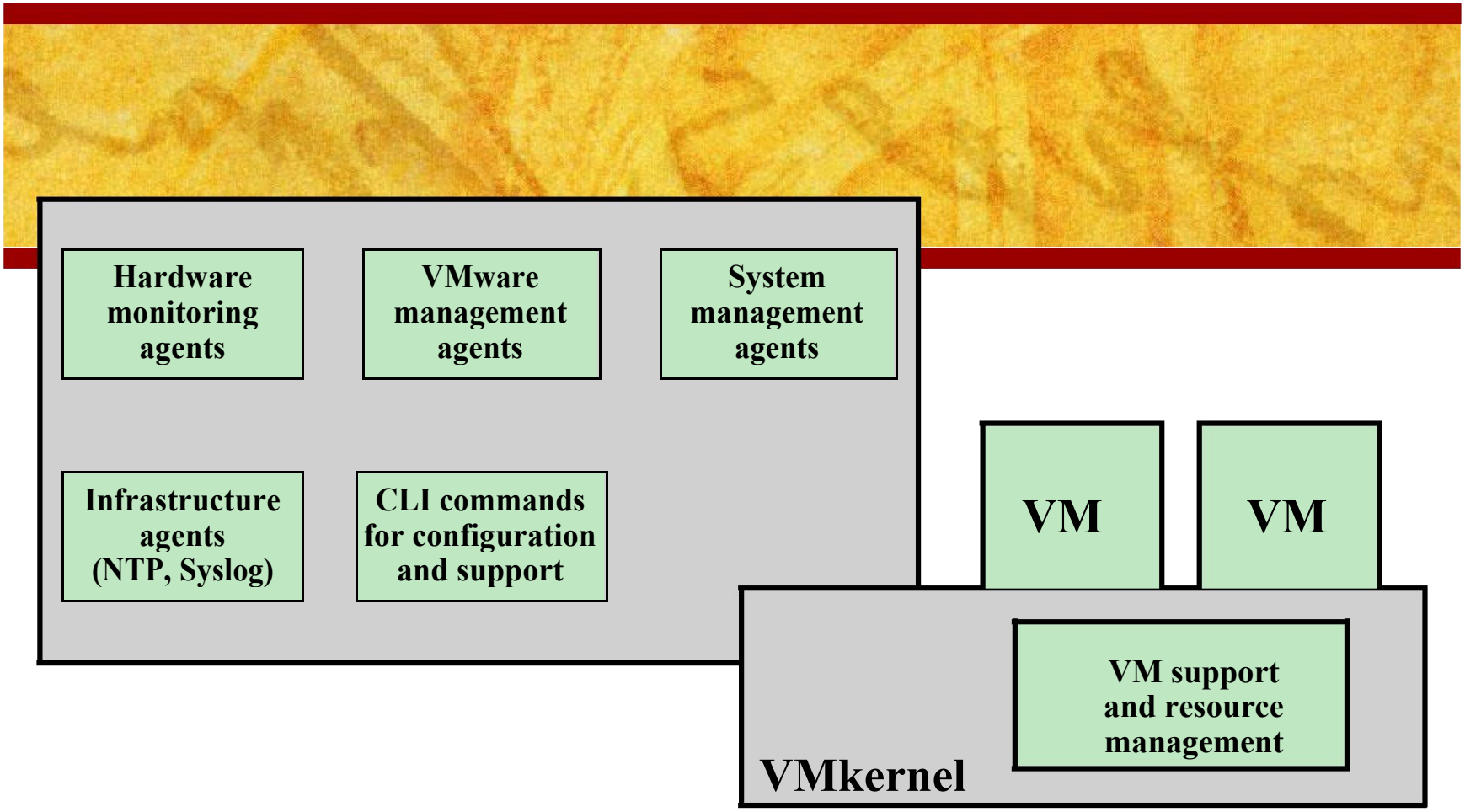

Figure 14.6 ESX

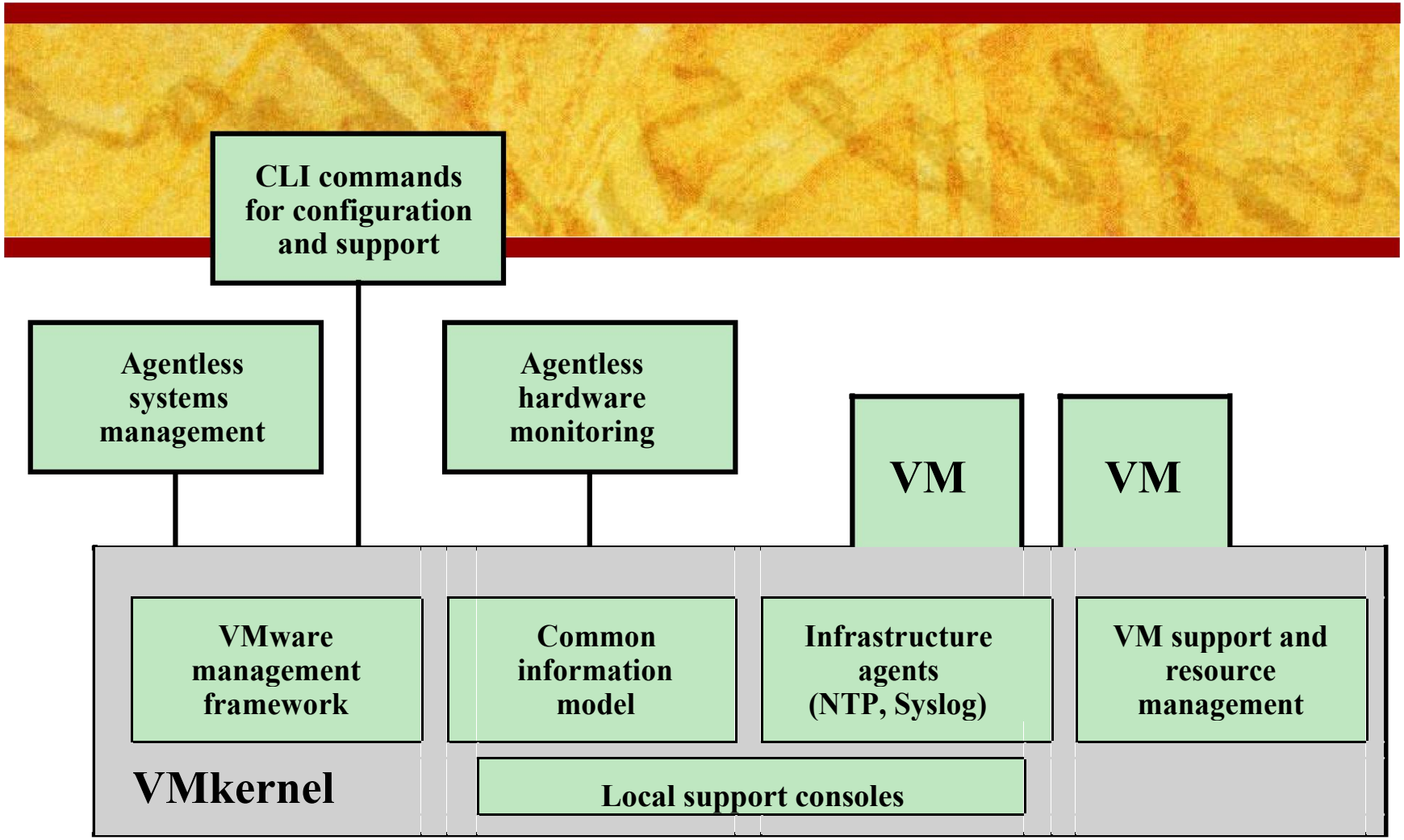

Figure 14.7 ESXi

VMware ESXi Features

Storage VMotion	Permits the relocation of the data files that compose a virtual machine, while that virtual machine is in use
Fault Tolerance	Creates a lockstep copy of a virtual machine on a different host --- if the original host suffers a failure, the virtual machine's connections get shifted to the copy without interrupting users or the application they are using
Site Recovery Manager	Uses various replication technologies to copy selected virtual machines to a secondary site in the case of a data center disaster
Storage and Network I/O Control	Allows an administrator to allocate network bandwidth in a virtual network in a very granular manner
Distributed Resource Scheduler (DRS)	Intelligently places virtual machines on hosts for startup and can automatically balance the workloads via VMotion based on business policies and resource usage

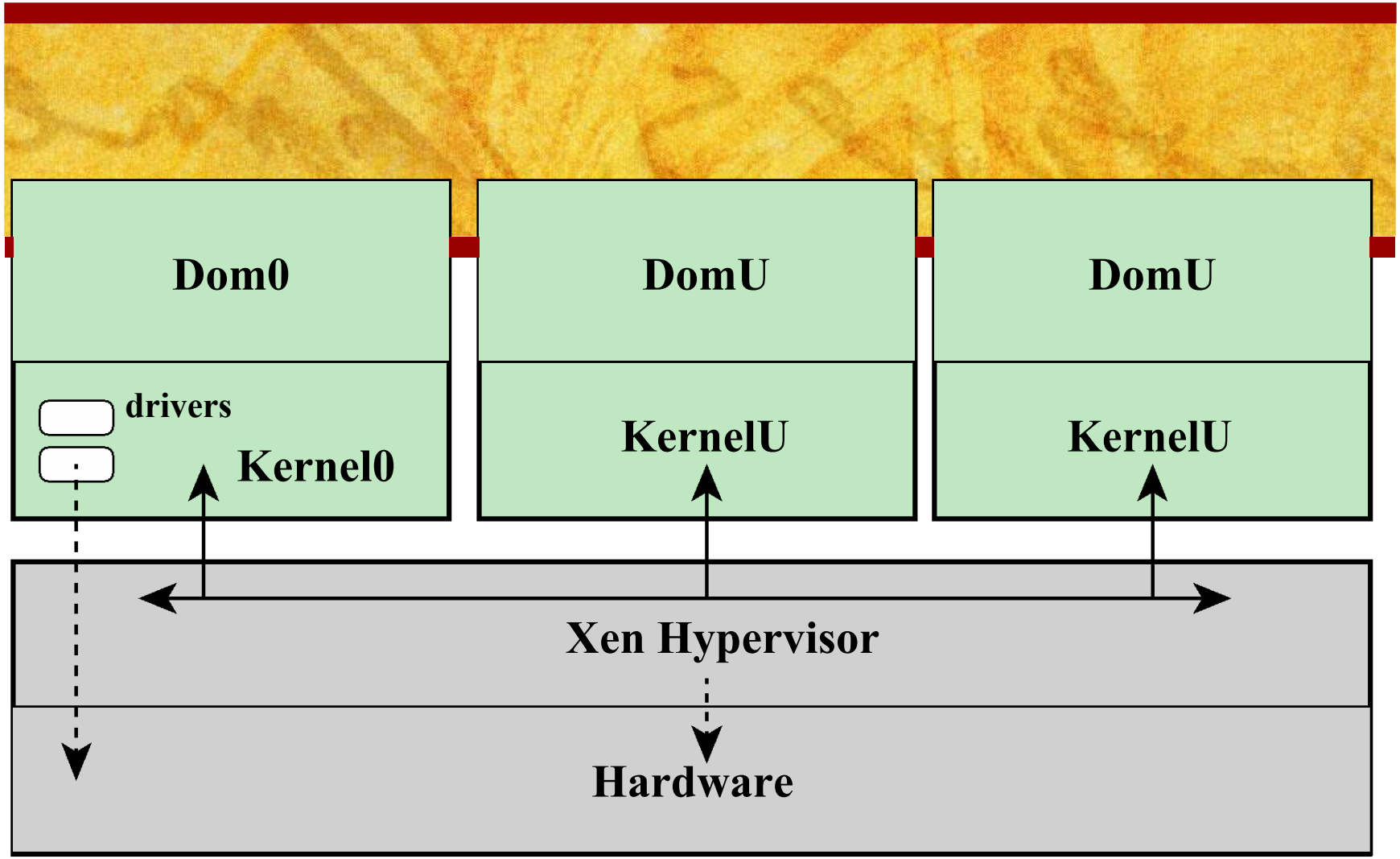

Figure 14.8 Xen

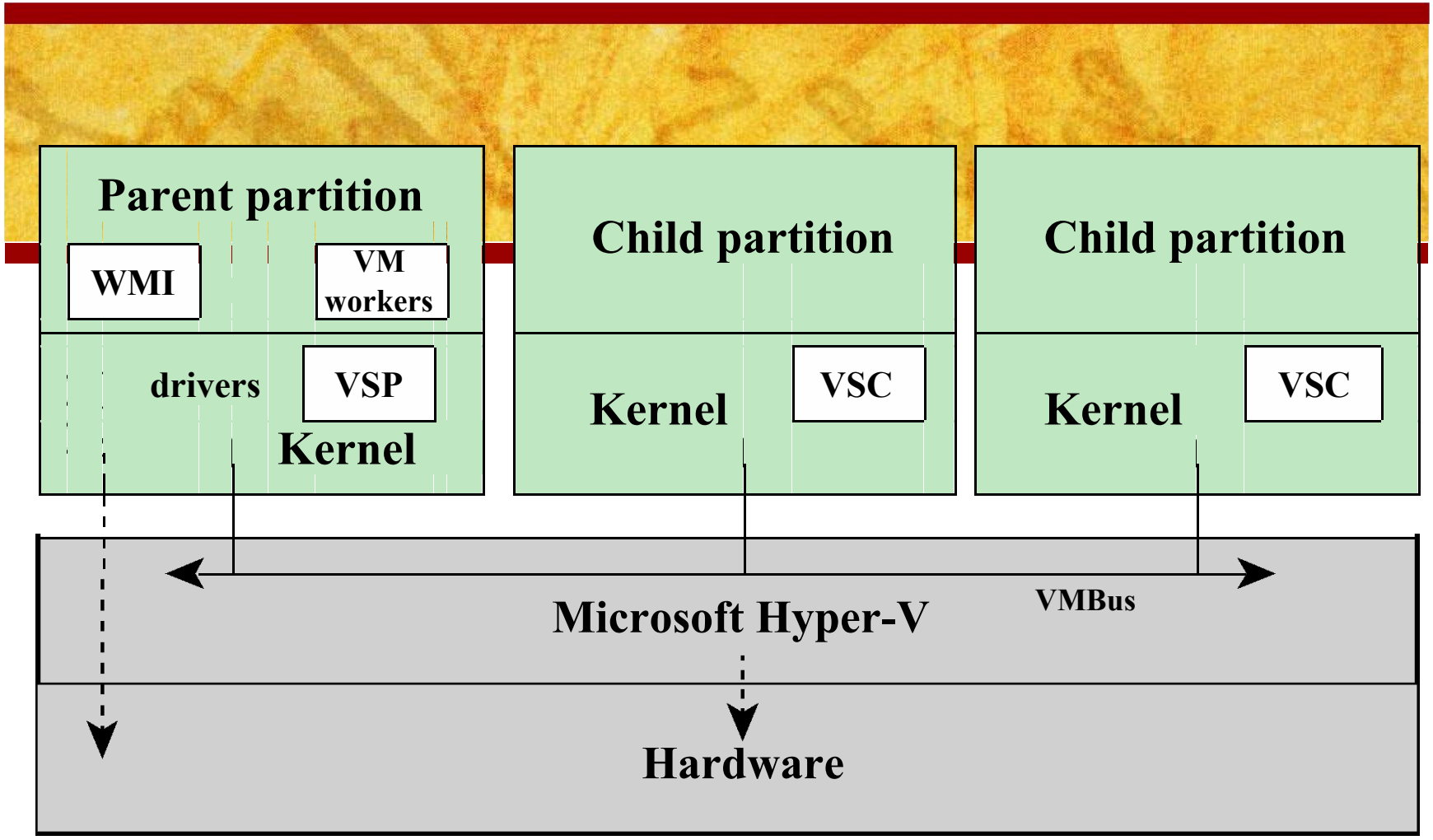

Figure 14.9 Hyper-V

Java VM

- The goal of a Java Virtual Machine (JVM) is to provide a runtime space for a set of Java code to run on any operating system staged on any hardware platform without needing to make code changes to accommodate the different operating systems or hardware
- The JVM can support multiple threads
- Promises “Write Once, Run Anywhere”
- The JVM is described as being an abstract computing machine consisting of:
  - an instruction set
  - a program counter register
  - a stack to hold variables and results
  - a heap for runtime data and garage collection
  - a method area for code and constants

Linux VServer

- Linux VServer is an open-source, fast, lightweight approach to implementing virtual machines on a Linux server
- Only a single copy of the Linux kernel is involved
- VServer consists of a relatively modest modification to the kernel plus a small set of OS userland tools
- The VServer Linux kernel supports a number of separate virtual servers
- The kernel manages all system resources and tasks, including process scheduling, memory, disk space, and processor time

Architecture

- Each virtual server is isolated from the others using Linux kernel capabilities
- The isolation involves four elements:
  - chroot
    - a UNIX or Linux command to make the root directory (/) become something other than its default for the lifetime of the current process
    - this command provides file system isolation
  - chcontext
    - Linux utility that allocates a new security context and executes commands in that context
    - each virtual server has its own execution context that provides process isolation
  - chbind
    - executes a command and locks the resulting process and its children into using a specific IP address
    - system call provides network isolation
  - capablities
    - refers to a partitioning of the privilege available to a root user
    - each virtual server can be assigned a limited subset of the root user's privileges which provides root isolation

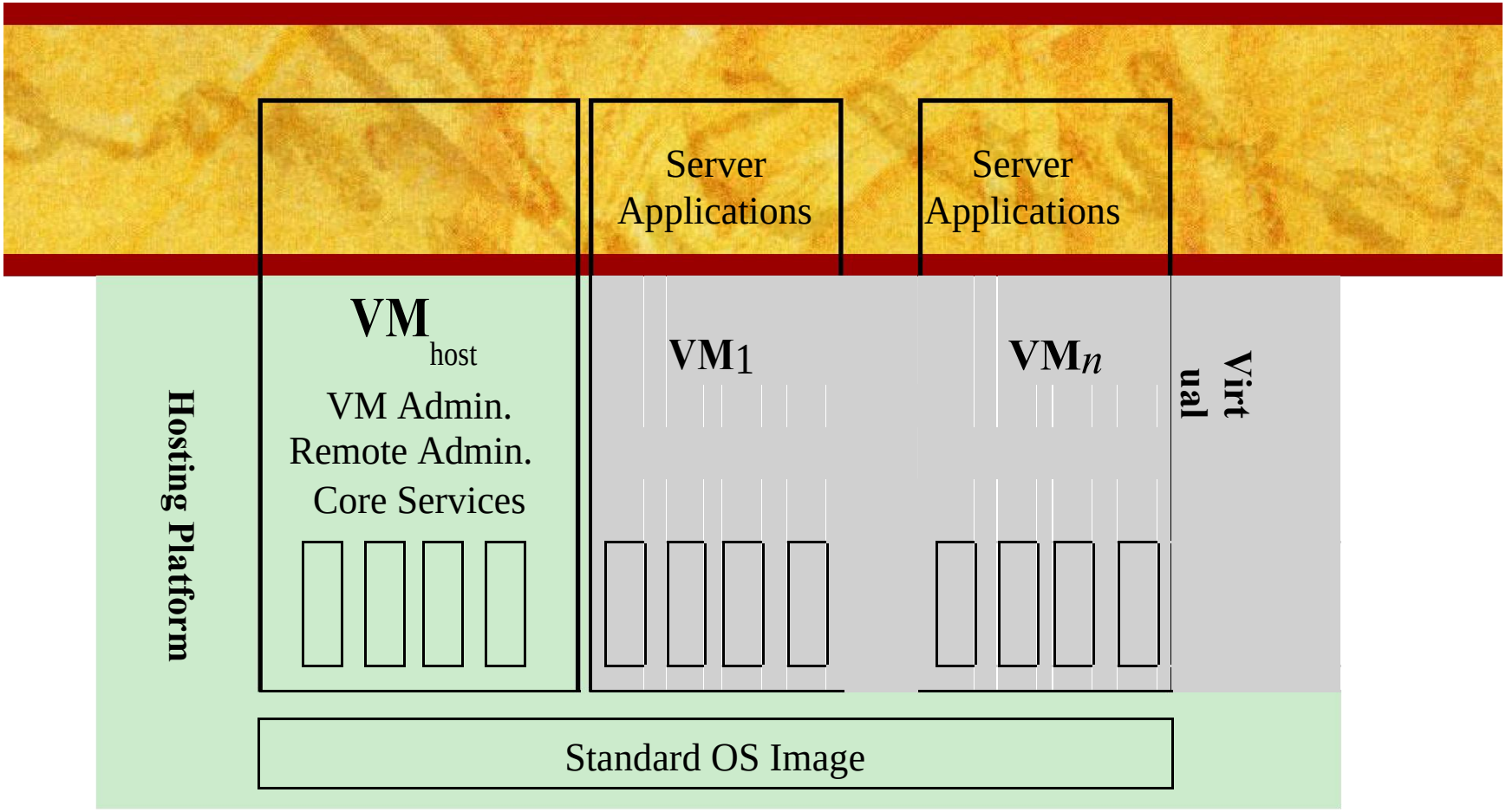

Figure 14.10 Linux VServer Architecture

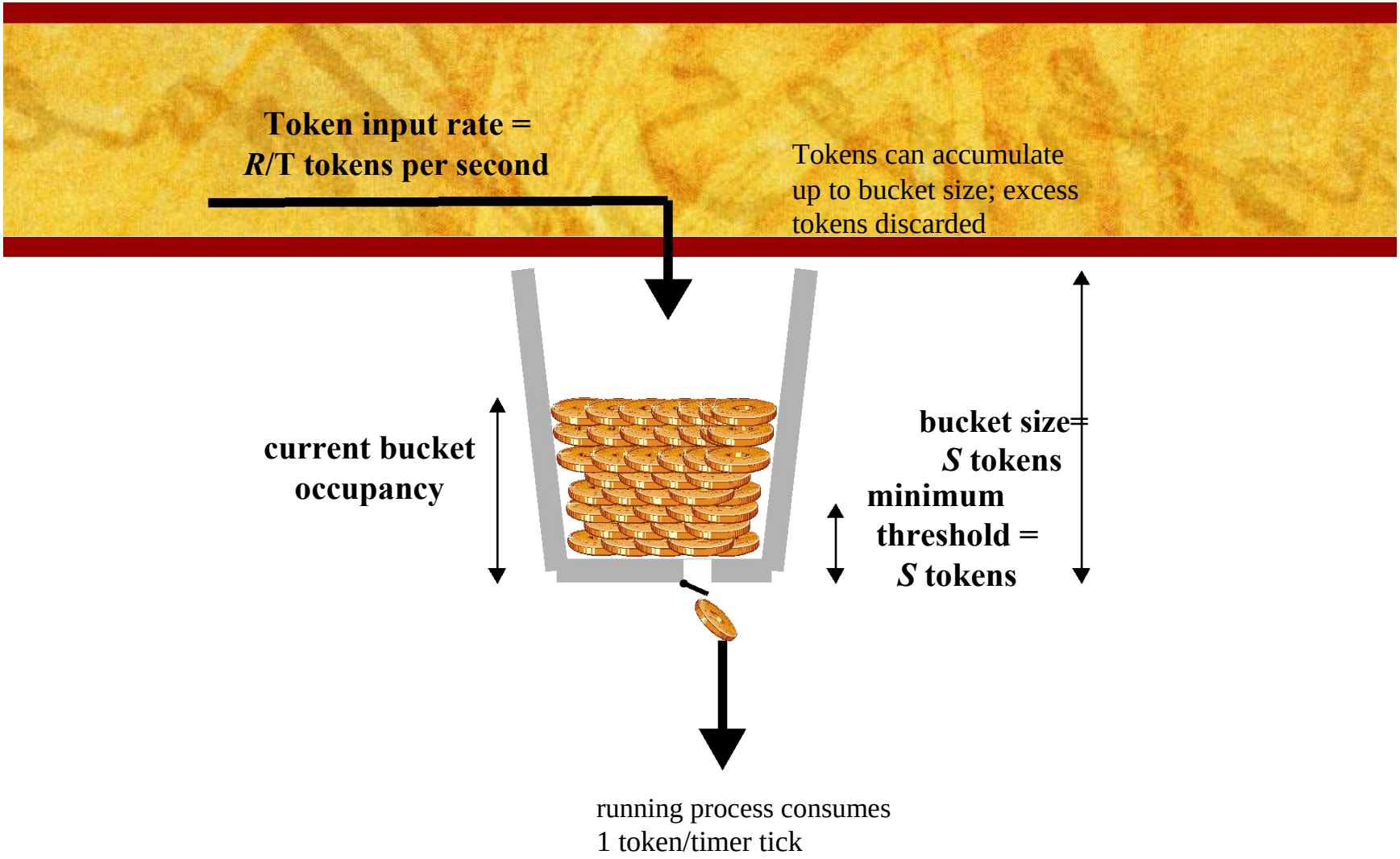

Figure 14.11 Linux VServer Token Bucket Scheme

Android Virtual Machine

Referred to as Dalvik

The Dalvik VM (DVM) executes files in the Dalvik Executable (.dex) format

The Dalvik core class library is intended to provide a familiar development base for those used to programming with Java Standard Edition, but it is geared specifically to the needs of a small mobile device

Each Android application runs in its own process, with its own instance of the Dalvik VM

Dalvik has been written so that a device can run multiple VMs efficiently

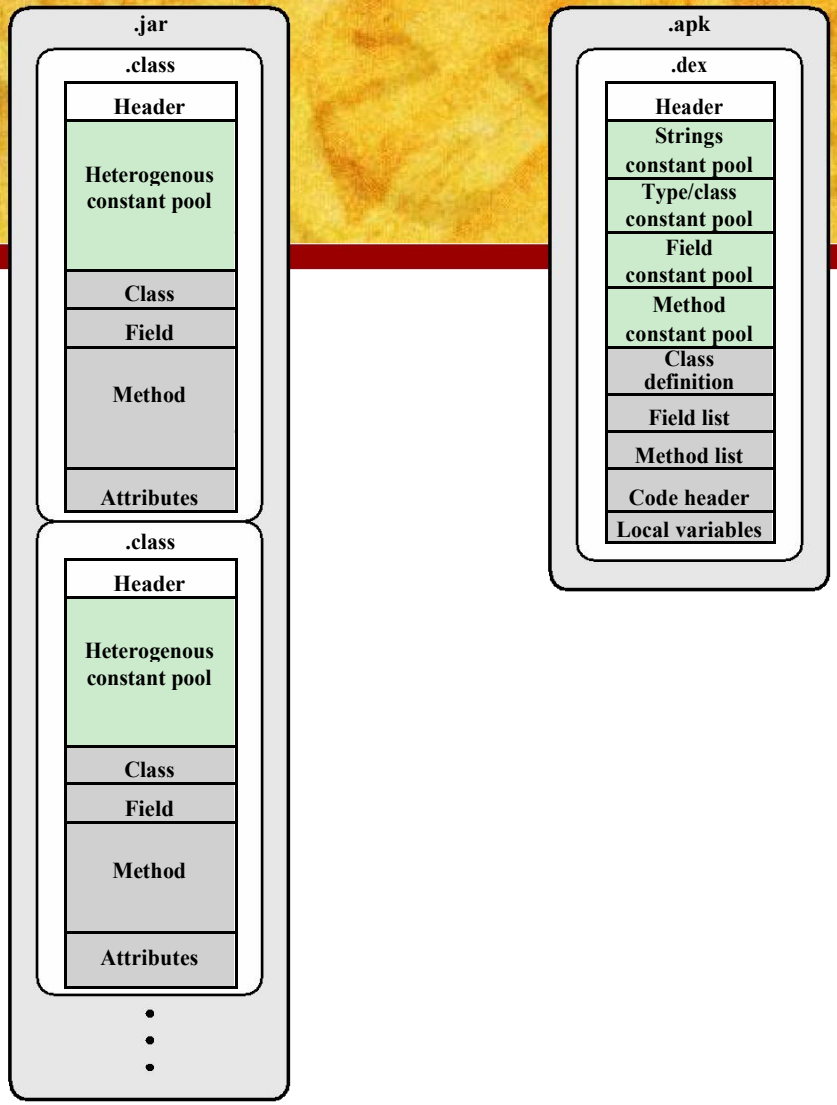

Figure 14.12 Java and Dalvik Formats

Zygote

- A process running on a DVM that is launched at boot time
- Generates a new DVM every time there is a request for a new process
- Intended to minimize the amount of time it takes to generate a new DVM by sharing items in memory to the maximum extent possible
- When first launched it preloads and preinitializes all Java core library classes and resources that an application may potentially need at runtime
- Additional memory need not be allocated for copies of these classes when a new DVM is forked from the Zygote DVM

Summary

- Approaches to virtualization
- Processor issues
- Memory management
- I/O management
- VMware ESXi
- Microsoft hyper-V and Xen variants
- Java VM
- Linux VServer virtual machine architecture
  - architecture
  - process scheduling
- Android virtual machine
  - Dex file format
  - Zygote