



# **Software Licensing in Virtual Environments**

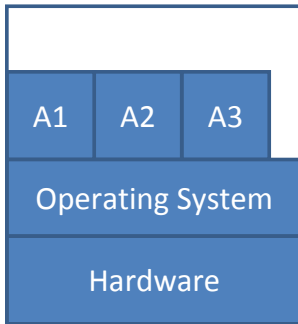
***Managing the Terms of Software Use  
in Virtualized Systems***

## Introduction

While virtualization has numerous IT infrastructure benefits, it can be a concern for software publishers. Virtualization can potentially be used to circumvent software licensing. However, software vendors need to maintain customer satisfaction and avoid prohibiting legitimate use of their applications by honest users.

This white paper describes the potential for software misuse through virtualization and addresses potential solutions for software vendors to maintain a measure of control over the applications, while maximizing customer satisfaction.

## Virtualization Defined

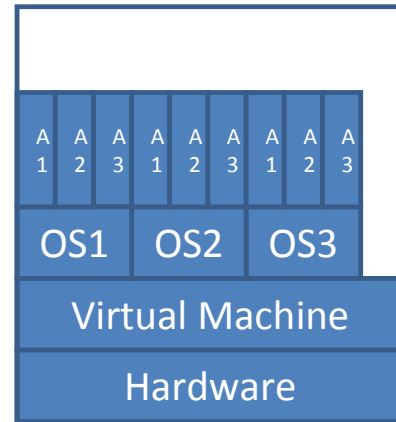


Without virtualization, multiple applications run on only one operating system, which runs on only one piece of hardware.

In computing, virtualization is a broad term that refers to the abstraction of computing resources. Virtualization abstracts the physical characteristics of computing resources. The virtualization is transparent to all users - applications as well as end users.

Virtualization technology is an abstraction layer that decouples the physical hardware from the virtual operating system. This includes making a single physical resource such as a server, an operating system, an application, or storage device appear to function as multiple virtual resources.

Virtualization allows multiple virtual machines (VMs), with heterogeneous operating systems to run in isolation, side-by-side on the same physical machine. Each VM has its own set of virtual hardware (e.g., RAM, CPU, NIC, etc.) upon which an operating system and applications are loaded. The operating system sees a consistent, normalized set of hardware regardless of the actual physical hardware components.



Virtualization makes it possible to run multiple operating systems and multiple applications on the same computer simultaneously.

## Benefits of Virtualization

### Lower Total Cost of Ownership

- Virtualization saves organizations money by enabling one computer to do the job of multiple computers.
- This enables consolidation of server resources, in order to save physical server space, as well as conserves energy.

### Increased Availability

- Virtualization increases availability by enhancing uptime through server redundancy. If one virtual machine goes down, the next image can be deployed instantly.
- In the event of a malicious attack, virtualization can prevent a virus from spreading. The infected virtual machine can be isolated in order to resolve the problem offline without the need to bring down the entire enterprise.

## Types of Virtual Environments

There are three common types of Virtualization technology:

1. Full Virtualization - Provides a complete virtual environment of all the computing resources of an operating system. Examples: VMWare.
2. Hypervisor or Hardware-assisted Virtualization - Provides a more efficient full virtualization using the assistance of native hardware to virtualize the computing resources. Example: Xen
3. Hosted Virtualization: Provides a partial virtualization that allows an application to run in a virtual environment. Example: Java Virtual Machine, Citrix

## Same Action, Different Intention

Virtualization is unique in that the same technology may be deployed by honest users for legitimate reasons as well as by dishonest users attempting to violate licensing terms. Unlike crackers who work to remove licensing protection from executable files, virtualization has many legitimate uses in an organization. It is for this reason that a software vendor must strive to find a balance between simply allowing and denying all use of their applications in virtual environments.

Following are several scenarios in which users may require the use of an application in virtual environments without violating their licensing agreements:

- OS independent floating licenses
- Platform independent server farm
- Control over data and reliable access all over the world

## Electronic Software Licensing

Electronic licensing technologies that protect software have typically done so by binding the technology to some physical characteristic of a PC. This is known as “machine fingerprinting” or “node locking.” In order to work seamlessly on various hardware configurations these bindings have relied on the operating system providing the licensing technology with information about the hardware on which it runs. Potential fingerprinting resources include:

- Hardware serial number
- Volume Disk ID
- Ethernet MAC Address
- Internet Protocol Address (IP)
- Hostname
- USB key
- GUID
- Serial number
- BIOS ID
- Etc.

There are many fingerprinting resources that a developer should consider when locking a license down to a computer. The benefit of using hardware IDs and MAC addresses is their relative ubiquity on most PCs. This allows software publishers to assure a consistent and seamless user experience.

## Defeating Licensing Through Virtualization

Like all applications and processes in a virtual environment, licensing systems typically behave as if the virtual environment is a separate and independent physical machine. In a virtual environment, hardware criteria used for binding a license are virtualized and thereby also easily duplicated.

Licensing systems that lock to a machine fingerprint will compute a fingerprint for that virtual environment. Locking licenses to a hardware ID is defeated in a virtual environment by creating a virtual machine and then, after installing the necessary software on it, making copies of that virtual machine. Each running instance of that virtual machine would then have the same emulated hardware IDs.

Because physical hardware is decoupled and emulated by the virtual software, any hardware ID can be manipulated and changed to match another virtual environment. This invalidates the security of machine fingerprinting. Mirroring another virtual environment for a license server will allow a user to double the quantity of licenses available with each instance of virtualization

### **Software-based Licensing in a Virtual Environment**

Well designed software licensing can continue to operate in a virtual environment exactly in the same way it would in a native environment. Instead of binding to a hardware ID it is possible to bind to the virtual hardware ID. In a virtual environment, numerous hardware criteria continue to be available for use to bind a license.

### **Managing Use of Applications Running in a Virtual Environment**

Software vendors must decide if they want to allow their application to run in a virtual environment. If so, they must determine what level of security they require. Ideally, the level of software protection can vary as needed, providing the vendor the freedom to increase security in certain geographic regions, for example.



Vendors have a spectrum of options for managing software licensing agreements in a virtualized world. First, in order to maximize transparency and usability, vendors may choose to allow all virtualization use of their application. In order to determine the best scenario for their business, software vendors can rely on the professional services of software licensing experts. Through the assistance of experienced consultants, software vendors can determine how best to manage their licensing agreements in a virtualized world.

### **Allow, but Inform**

For optimal usability the software vendor can allow use in virtual environments but focus on making it evident that a license authorization is being violated. Techniques can be used to detect if an application is running in a virtual environment. Similar to a “nag screen” on a free software trial, developers could choose to simply return alerts to inform users. Alerts could be communicated only to system administrators, in order to maximize transparency for end users. Such a system may prove beneficial for deployment to organizations concerned with maintaining license compliance and may be the ideal method for vendors to help keep honest users honest

### **Techniques to Detect Virtualization**

1. For full virtualization (ie: VMWare) a complete hardware and operating system must be simulated within the virtual OS. Therefore, full virtualization detection includes:
  - a) Validating the OS in use through hardware drivers installed
    - o If any VMWare drivers, such as the Ethernet network card driver, are used then one can be assured that the OS is being run in a virtual environment.
  - b) Simulating a general segment fault in the underlying application.

- In such event, the application in the virtual environment will continue to run as the general fault will be captured within the host environment.
- c) Scan the system for a backdoor port where the virtual and the host environment communicate with each other.

2. Hardware assisted Virtualization (ie: Xen): to detect hardware virtualization technology one must implement in software the timing mechanism where in native mode the timing should be much faster than within the virtual environment. This is due to the fact that in a hardware assisted environment, access from the virtual computing resources is trapped and passed to the host environment through a hypervisor. This trap that is being done requires some time discrepancies between the virtual and native applications. On a native machine, the application will perform much faster whereas in a virtual environment, the timing will be increased significantly.

### **Locking Licenses to Maximize Security**

In order to add security and maintain electronic controls over the use of software it is necessary to lock to a non-virtualized element. This feature allows a software publisher to create binding criteria of their own so they can tailor the security to their needs to complement what is supported natively.

The license must be bound to some criteria that is not trivially virtualized i.e. not controlled by the native operating system. Protection can be provided by use of a physical asset, such as a dongle or PCI card, as the license key. A high level of protection is realized by using a dongle that is unique to the software.

The dongle must be unique and bound to that software. This renders attempts to replicate the software on multiple machines impossible because only one physical dongle exists that will enable the software to run.

Fortunately, this strategy can also maintain transparency for end users, as the hardware devices can be used on the servers deploying software licenses, as opposed to requiring each individual user to hold their own dongle.

To control virtualization and eliminate the possibility of the abuse of licensing in virtual environments, developers need to consider binding the license to the following:

1. USB Token - only one machine is allowed access to the token at one time. In a multi-virtual environment, only one instance can use the key and the switch needs to happen if the other instance requires the use of the key. In a multi-virtual environment on separate physical hardware it is impossible to duplicate the USB token as the token would not be able to be virtualized.
2. Lock to a unique external ID where it is in a controlled environment such as a web server hosted by the ISV. The application will have to access the ISV for locking information. The ISV will control the number of instances a program can run.

### **Network Licensing and Virtualization**

In many cases, a suitable approach is to enforce the use of a license server on a native machine. The above methods can be utilized to ensure that a license server can not be installed within a virtual environment, or will refuse to serve some or all of its licenses if a virtual environment is detected. Although this still requires a native OS for the license server, it will allow complete virtualization of the licensed (client) applications, and yet ensure that the license enforcement rules are maintained.



## Conclusion

Virtualization is a technological advancement that involves both challenges and opportunities for software vendors. In particular, virtualization is demonstrative of the challenge ISVs continually face in attempting to balance their need to protect their software and customers' need for flexible use. Fortunately, as with other licensing challenges, well-designed software license management systems exist to help vendors easily alter their license configurations and adapt to various markets.



## SafeNet Overview

SafeNet is a global leader in information security. Founded 25 years ago, the company provides complete security utilizing its encryption technologies to protect communications, intellectual property, and digital identities, and offers a full spectrum of products, including hardware, software, and chips. UBS, Nokia, Fujitsu, Hitachi, Bank of America, Adobe, Cisco Systems, Microsoft, Samsung, Texas Instruments, the U.S. Departments of Defense and Homeland Security, the U.S. Internal Revenue Service, and scores of other customers entrust their security needs to SafeNet. SafeNet was taken private by Vector Capital in 2007. For more information, visit [www.safenet-inc.com](http://www.safenet-inc.com).

## Corporate Headquarters

4690 Millennium Drive, Belcamp, Maryland 21017 USA

Tel.: +1 410 931 7500 or 800 533 3958, Fax: +1 410 931 7524,

Email: [info@safenet-inc.com](mailto:info@safenet-inc.com)

## EMEA Headquarters

Tel.: + 44 (0) 1276 608 000

Email: [info.emea@safenet-inc.com](mailto:info.emea@safenet-inc.com)

## APAC Headquarters

Tel: +852 3157 7111

Email: [info.apac@safenet-inc.com](mailto:info.apac@safenet-inc.com)

For all office locations and contact information, please visit [www.safenet-inc.com/company/contact.asp](http://www.safenet-inc.com/company/contact.asp)

©2008 SafeNet, Inc. All rights reserved. SafeNet and SafeNet logo are registered trademarks of SafeNet. All other product names are trademarks of their respective owners.

