

Antitrust in the Age of Virtualization: A First Principles Approach

Andrew Chin*

University of North Carolina School of Law
chin@unc.edu

Preliminary Draft

Please do not quote or redistribute without permission.

Please contact author for updates.

I.	Introduction: <i>Microsoft Redux</i>	2
II.	An Introduction to Virtualization Technology	5
	A. VMM Software: Power Strips for Your Microchips	5
	B. Uses for VMM Software.....	6
	C. Native and Hosted VMM Software	7
	D. Characteristics of Efficient VMM Software	8
	E. Hardware Support for Virtualization	9
	F. Trapping and Emulation	10
	G. Virtualization Overhead.....	11
III.	Microsoft and the Market for Virtualization Software Products	12
	A. The x86 Virtualization Industry.....	12
	1. Size.....	12
	2. Participants.....	12
	3. Competitive Variables	14
	B. Microsoft’s Virtualization Strategy	15
	1. Support for Legacy VMs	15
	2. Licensing Windows Server By Running Instance	15
	3. Virtualization of Windows Server	16
	4. “Integration” of Virtualization into Windows	19
	5. PC Virtualization	21
IV.	Do Microsoft’s Plans Constitute an Illegal Tying Arrangement?	21
	A. Software Products and Provisioning.....	22
	B. The Tying and Tied Products in Windows Server “Longhorn”	24
	C. Tying Defined	27
	D. Power in the Tying Product Market.....	29
	E. The Separate Products Inquiry.....	29
	F. Existence of a Tying Condition	33

* Associate Professor of Law, University of North Carolina at Chapel Hill. I would like to thank Greg Vetter for helpful suggestions, and Adam Steadman for excellent research assistance.

1. The Limited Scope of Microsoft’s Copyright.....	34
2. The Applicability of “Technological Tying” Doctrines	38
G. Foreclosure and Anticompetitive Effect in the Tied Product Market.....	39
H. Possible Justifications	42
V. Conclusions.....	42

I. INTRODUCTION: *MICROSOFT REDUX*

Consider the following fact pattern:

Computer scientists develop a new software product that makes it easier for a Windows PC user to run software applications that were not written specifically for the Windows platform. A startup company attains a dominant position in the emerging market for this product, and then is acquired by a Fortune 500 company. As the installed base of this product rapidly grows, so grows the potential market for applications software vendors who develop products for non-Windows platforms.

Microsoft responds by developing its own version of the product based on software developed by a smaller software company that Microsoft has recently acquired. In an apparent competitive effort to grow their respective installed bases, Microsoft and the startup company both begin to offer their products for free.

Microsoft subsequently decides to develop the new product jointly with Windows, thereby producing an “integrated” software package in which the same shared code is used to support enhanced features both in the new product and in the Windows platform. Microsoft distributes the new software package widely, first as an add-on update to previous releases of Windows, and later as a new, “integrated” version of Windows. Microsoft’s combination deters many computer manufacturers and PC users from installing and using other vendors’ versions of the new product.

The above account will be familiar to students of antitrust law as a description of the conduct at the center of *United States v. Microsoft Corp.*: the inclusion of Internet Explorer in Windows 98. But it is an equally apt description of recent developments in the software industry, and Microsoft’s announced plans, in the strategically important field of virtualization technology.

Virtualization is an approach to computing in which certain computing resources are presented for use in a form that differs from their actual configuration in the system. Virtualization technology allows a single computer (a *real machine*) to exhibit the same

behavior as would be exhibited by multiple computers (*virtual machines*).¹ Each virtual machine has its own installed software; the virtual machines on a given real machine can be running the same or different operating systems. For example, with virtualization, a Windows user who wants to use a Linux application does not need to buy a second computer dedicated to running Linux. She simply needs to configure her computer to support two virtual machines, one running Windows and one running Linux.

By reducing the cost of switching from Windows to non-Windows applications, virtualization may threaten an important entry barrier protecting Microsoft's market power in the workgroup server operating systems market. It might be in Microsoft's best interests to try to impede the move toward virtualization, but attractive virtualization solutions from major hardware and software vendors will soon be available with or without Microsoft's cooperation. Instead, Microsoft has decided to build its own virtualization technology into a forthcoming release of Windows. In doing so, Microsoft will likely gain control over the virtualization technology that is installed on most workgroup servers, and may thereby be in a position to blunt virtualization's threat to the Windows monopoly.

The purpose of this Article is to identify potential anticompetitive concerns arising from Microsoft's virtualization plans, and to assess the prospects for an antitrust response to these concerns. Specifically, I will argue that if Microsoft carries out its current plans to market Windows Server with "integrated" virtualization technology, its conduct may be subject to challenge under the Sherman Act for illegal tying and monopoly maintenance.

Given the parallels I have already identified between Microsoft's virtualization plans and Microsoft's operating system/browser combination in Windows 98, the 1998-2004 *Microsoft* proceedings provide an obvious frame of reference for the analysis in this Article. There are, however, significant points of departure.

Since the tying claim in *Microsoft* was never finally adjudicated, the D.C. Circuit's 2001 *en banc* opinion provides incomplete guidance for the review of a tying claim based on Microsoft's next-generation software products. As I have argued in two companion articles,² the district court had already made sufficient findings of fact by 1999 to support a conclusion of tying liability, even under the uniquely lenient rule of reason/limited factual record standard that was prescribed by the *en banc* opinion for the 2001-02 remand proceedings. By taking a "first principles" approach to identify the operating system and Web browser software products in Windows 98 and the markets in

¹ The term "virtualization" can be used broadly to refer to a variety of concepts. "Process virtualization" refers to the use of software to carry out the instructions of a running process that was designed to run on a different platform than the real machine on which the process is running. See SMITH & NAIR, *supra*, at 9-17. The Java Virtual Machine is an example of software designed to support process virtualization. See *id.* at 17. "System virtualization" refers to the use of software to "provide a complete system environment in which many processes, possibly belonging to multiple users, can coexist" and run. See *id.* Unlike the Java Virtual Machine, a system virtual machine is designed to provide the equivalent of a complete hardware platform. See *id.* at 21. Examples of system virtualization products are described in Section III.A. For simplicity, throughout this Article, the term "virtualization" will refer to system virtualization.

² Andrew Chin, *Antitrust Analysis in Software Product Markets*, 18 HARV. J. L. & TECH. 1 (2004) (hereinafter "Antitrust Analysis"); Andrew Chin, *Decoding Microsoft: A First Principles Approach*, 40 Wake Forest L. Rev. 1 (2005) (hereinafter "Decoding *Microsoft*").

which they compete, I was able to complete the judicial review of the tying claim that had been abandoned by the plaintiffs on remand. In this Article, I apply the same “first principles” approach to identify a basis for tying liability based on Microsoft’s current virtualization plans.

The monopoly maintenance claim in *Microsoft* was based in large part on the same facts underlying the tying claim, under the theory that Microsoft was seeking to prevent the widespread distribution and installation of “middleware” (i.e., Netscape Navigator and Java Virtual Machine software) that could compete with Windows as a platform for software application development. In contrast, a claim for monopoly maintenance based on Microsoft’s virtualization plans would argue that Microsoft could use its control over installed virtualization technology to degrade or even prevent the operation of rival operating systems.

The scope of this Article is limited to Microsoft’s virtualization plans for its Windows Server operating system. This is not to dismiss or trivialize other antitrust concerns in connection with Microsoft’s forthcoming releases of Windows. Indeed, Microsoft’s development of new features for its next-generation desktop operating system, Windows Vista, has recently raised other significant tying concerns, most notably in the markets for Web search engines³ and security software products.⁴ Microsoft’s virtualization plans for Windows Server are particularly problematic, however, because of the potential harms to competition in both tying and tied product markets.

To date, virtualization technology development by Microsoft and other vendors has been primarily directed to servers. Recently, Microsoft announced that it also has virtualization plans for the forthcoming PC version of Windows, Windows Vista. Because these plans are only in a preliminary stage, this Article will discuss them only briefly, and will focus instead on the effects Microsoft’s more developed server virtualization plans will have on its competition with other vendors.

The remainder of this Article is organized as follows. Part II provides a basic introduction to virtualization technology. Part III reviews the current state of the market for virtualization software products and Microsoft’s announced virtualization plans. Part IV assesses the prospects for an antitrust challenge to Microsoft’s virtualization plans based on a tying claim. Part V concludes.

³ Cf. Joint Status Report on Microsoft’s Compliance With the Final Judgments, *United States v. Microsoft Corp.*, Civ. No. 98-1232, at 14 (filed May 12, 2006) <<http://www.usdoj.gov/atr/cases/f216100/216127.pdf>> (explaining the Justice Department’s decision to take no action in response to Google’s complaints regarding the new search feature in Internet Explorer 7, which will be bundled with Windows Vista); Ted Bridis, *Judge Agrees to Extend Microsoft Oversight*, WASH. POST, May 17, 2006 (reporting that Judge Colleen Kollar-Kotelly “pressed” Justice Department lawyers to defend their decision).

⁴ In a note that won the ABA Antitrust Section’s 2005 student writing competition, Columbia Law School student Heather Schneider addressed what she and other commentators believed would be “Microsoft’s next antitrust battle”: an alleged tie of its anti-spyware software to Windows. See Heather Schneider, Note, *An Antitrust Tying Analysis of Microsoft’s Security Software Products*, 7 COLUM. SCI. & TECH. L. REV. art. 3, at 1 (2005); Joris Evers, *Spyware-Killing Vista Could Take Out Rivals*, CNET NEWS.COM, March 17, 2006, available at http://news.com.com/2102-1029_3-6050733.html (quoting analysts’ opinions that the bundling of Windows Defender into Windows Vista will foreclose almost all of the aftermarket for anti-spyware software).

II. AN INTRODUCTION TO VIRTUALIZATION TECHNOLOGY

A. VMM Software: Power Strips for Your Microchips

It's a familiar problem around the home entertainment center and workstation: too many appliances, too few wall outlets. Fortunately, an equally familiar solution exists in the affordable, easy-to-use power strip. A power strip does not supply its own electricity, but by providing several outlets that can be powered via a single plug, it can provide the same effect to the user as having the same number of live wall outlets. Some power strips also incorporate other useful features into their designs, such as surge protectors, line noise filters, and automatic switches.

Computer owners who want to run Windows and Linux applications on the same machine face a similar problem: too many operating systems ("OSes"), too few microprocessors (also known as "central processing units" or "CPUs"). Since computer systems are still relatively expensive compared to operating system software products, most computer users are effectively limited to a choice of only one OS to run all of the applications they use. Ensuing OS-specific investments can lock consumers into an initial choice of OS⁵ (and subsequent versions thereof⁶), even when potentially superior OS options become available. As the courts in *Microsoft* found, this lock-in effect plays a significant part in an "applications barrier to entry" that benefits Windows at the expense of other OS software products and thereby confers monopoly power on Microsoft.⁷

There is a special kind of software that can do for the CPU what the power strip did for the wall outlet. A computer owner can install a thin layer of software, known as a "virtual machine monitor" or "VMM," which provides the same effect to the user as having several CPUs. In a computer system so configured, multiple OSes can be installed on the same VMM. Instructions that an OS would ordinarily send to the CPU are sent instead to the VMM, which is then responsible for passing them along to the CPU at an appropriate time and in an appropriate manner so that the system behaves as if the OS were running directly on the CPU. To achieve this result, the VMM must maintain detailed information about the system's state and behavior with respect to each of the OSes running on it. In effect, the VMM serves both as a traffic controller and as an interpreter, managing the sharing of system resources among the various OSes, scheduling software instructions from the various OSes for processing on the CPU, and rewriting instructions as necessary so that time-sensitive operations and other complications arising from the sharing of system resources are correctly resolved.

⁵ See *United States v. Microsoft Corp.*, 84 F. Supp. 2d 9, 18, at ¶ 30 ("[C]onsumers who already use one Intel-compatible PC operating system are even less likely than first-time buyers to choose a newcomer to the field, for switching to a new system would require these users to scrap the investment they have made in applications, training, and certain hardware.").

⁶ See *id.* at 21-22, at ¶ 44 (finding that third-party developers tend to prefer writing applications for new versions of Windows rather than other OS products because of Microsoft's policy of maintaining downward compatibility).

⁷ See *id.* at 19, at ¶ 33.

VMM software does not supply its own computational power, but draws on the power of a single CPU to provide the outward appearance of being several different computer systems (also known as “virtual machines” or “VMs”) at once. By accessing computer resources only in their virtual form, a human user or computational process can perform a task without regard to the location or configuration of the resources that are ultimately responsible for implementing the computation. Arranging a computer system in this way to support multiple virtual machines, each running its own operating system and applications simultaneously, is referred to as *virtualization*.⁸ More formally, virtualization is the construction of a one-to-one, structure-preserving mapping (also known as an “isomorphism”) between the resources of one computer system (the “guest”) and another (the “host”), so that each part of the guest system (i.e., the VM) plays the same role in the operation of the guest system as the corresponding part of the host system (i.e., the real machine) plays in the operation of the host system.⁹

B. Uses for VMM Software

The ability to run different operating systems simultaneously on the same computer can be especially useful when the computer in question is a server running diverse applications for multiple users. Information technology departments may find that they are able to provide the same services to users with dramatically fewer physical servers. A user who prefers to use applications that run on a different OS can work on the same hardware platform as the rest of the organization. An old version of an OS can be installed on one VM to run legacy applications, while a newly updated version of an OS can be installed on another VM to allow technical support personnel to test and integrate it before making it available for general use. Software developers may also find this kind of installation useful, as they can easily check whether their programs run correctly on different versions of the same OS, or on different OSes.

To provide the outward appearance of many different machines, the VMM software must ensure that each VM operates independently of the others. This isolation between the behaviors of individual VMs is another useful feature of virtualization software. Errors, failures, overloads and other abnormal system conditions can be confined to one VM, allowing other computations to proceed normally. User-installed or other untrusted applications that might introduce bugs, viruses or spyware into a computer system can be detected,¹⁰ analyzed¹¹ and quarantined,¹² and managed

⁸ See JAMES E. SMITH & RAVI NAIR, VIRTUAL MACHINES 369 (2005) (“A system VM [virtual machine] environment is capable of supporting multiple system images simultaneously, each running its own operating system and associated application programs.”).

System virtualization is one variety of a more general concept in computing known as *virtualization*. A computer system makes various computing resources available for use, either by a human user or by a process running on the computer.

⁹ Id. at 3-4.

¹⁰ See, e.g., Tal Garfinkel & Mendel Rosenblum, *A Virtual Machine Introspection Based Architecture for Intrusion Detection*, in PROC. 2003 NETWORK & DISTRIBUTED SECURITY SYMP., Feb. 2003, available at <http://www.isoc.org/isoc/conferences/ndss/03/proceedings/papers/13.pdf>.

applications that access sensitive data can be specifically secured, e.g., through data encryption.¹³ New users of a system can be trained on a VM that replicates the behavior of the machine they will be using, but which can be discarded afterward without affecting other users.¹⁴ Technical support personnel can use a VM to duplicate the state and configuration of a client’s machine so that they can decide in advance which modifications are necessary, thereby minimizing disruption to the client’s work.¹⁵

Some VMM software can encapsulate the entire state of a VM in the form of a (very large) file for easy transmission and storage. This portability of VMs can aid in automating important system maintenance procedures. VMM software can periodically archive backup (or “failover”) copies of running VMs, then activate them if necessary to recover from a failure.¹⁶ VMs can also be moved as users move from one machine to another¹⁷ or as may be necessary to balance loads among different machines.¹⁸

C. Native and Hosted VMM Software

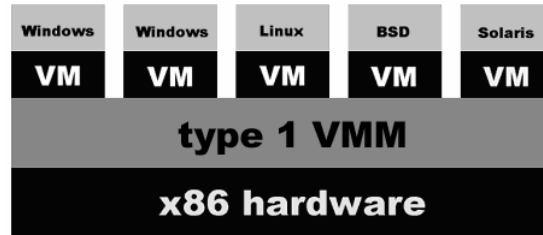


Figure 1. Use of type-1 VMM software to support multiple operating systems.

VMM software has traditionally been one of two basic types. As shown in Figure 1, *native* (or “Type-1”) VMM software is installed directly on the computer system hardware, before any operating system or other software is installed. Instructions from native VMM software are directed to the computer system hardware.

¹¹ See, e.g., George W. Dunlap et al., *ReVirt: Enabling Intrusion Analysis Through Virtual-Machine Logging and Replay*, in PROC. 2002 SYMP. ON OPERATING SYSTEMS DESIGN & IMPLEMENTATION, Dec. 2002, available at <http://www.eecs.umich.edu/~kingst/revirt.pdf>.

¹² Smith & Nair, *supra*, at 371.

¹³ See, e.g., VMware, *Virtual Machine Encryption Basics*, available at http://www.vmware.com/pdf/ace_encrypt_bg.pdf.

¹⁴ Smith & Nair, *supra*, at 372.

¹⁵ *Id.*

¹⁶ See, e.g., Thomas C. Bressaud & Fred B. Schneider, *Hypervisor-Based Fault Tolerance*, 14 ACM TRANSACTIONS ON COMPUTER SYSTEMS 80 (1996).

¹⁷ See, e.g., Constantine P. Sapuntzakis et al., *Optimizing the Migration of Virtual Computers*, in Proc. 2002 Symp. on Operating Systems Design and Implementation, Dec. 2002, available at <http://suif.stanford.edu/collective/osdi02-optimize-migrate-computer.pdf>

¹⁸ See generally JAMES E. SMITH & RAVI NAIR, VIRTUAL MACHINES 533 (2005).

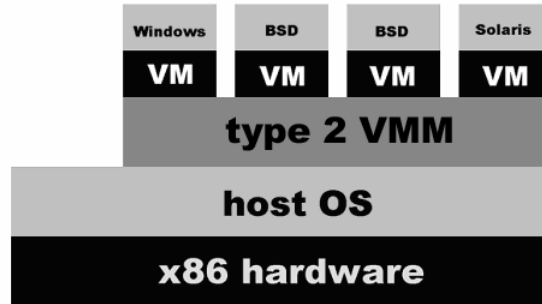


Figure 2. Use of type-2 VMM software to support multiple operating systems.

As shown in Figure 2, *hosted* (or “Type-2”) VMM software is specifically designed to run on a particular OS (the “host OS”), and presumes that the host OS software be preinstalled on the computer system. Instructions from hosted VMM software are directed to the applications programming interface (“API”) of the host OS, which in turn implements the API by sending its own instructions to the computer system hardware.

D. Characteristics of Efficient VMM Software

A classic paper by Popek and Goldberg¹⁹ identifies three properties that VMM software must satisfy in order to support efficient virtualization: efficiency, resource control, and equivalence. Efficiency requires that the VMM allow all “innocuous” VM instructions to be executed immediately and directly by the hardware.²⁰ An instruction is “innocuous” if both (a) it does not attempt to change the configuration of the system resources and (b) its behavior does not depend on the configuration of the system resources.²¹ Resource control requires that the software running on a VM be prohibited from directly changing the configuration of any of the system resources available to the VM.²² The VMM must maintain overall control over the allocation of system resources.²³ Equivalence requires that any guest software must behave identically to the way it would have behaved when running directly on the hardware, except for reductions in performance due to the VMM’s need to rewrite certain instructions, the need to share system resources among different VMs, and the need to alter timing relationships to accommodate the rewriting of code and the sharing of system resources.²⁴

While strict adherence to the Popek-Goldberg properties can yield important benefits for the design of virtualizable computer architectures,²⁵ some developers have departed from them in order to meet other design objectives. For example, one

¹⁹ G.J. Popek & R.P. Goldberg, *Formal Requirements for Virtualizable Third-Generation Architectures*, 17 COMMUNICATIONS OF THE ACM 412 (1974).

²⁰ Smith and Nair, *supra*, at 387.

²¹ *Id.* at 385.

²² *Id.* at 387.

²³ *Id.* at 377.

²⁴ *Id.* at 387.

²⁵ One important benefit is the ability to support recursive virtualization, i.e., the ability to run VMM software in user mode on a copy of itself. See SMITH & NAIR, *supra*, at 390-91.

implication of the equivalence property is that any guest systems running in a VM environment will be unaware that they are not running on a real machine.²⁶ In some design approaches, however, guest systems violate the equivalence property by using the knowledge that they are actually running on a particular VMM to adapt the guest software at runtime for more efficient execution by the VMM.²⁷

E. Hardware Support for Virtualization

Modern computer architectures, including the x86 hardware platform that operates most PCs and workgroup servers, are designed to provide multiple “privilege levels” that regulate the ability of software to access system resources. While the operating system is permitted to access, change, and allocate system resources, software applications are normally allowed more restricted access.

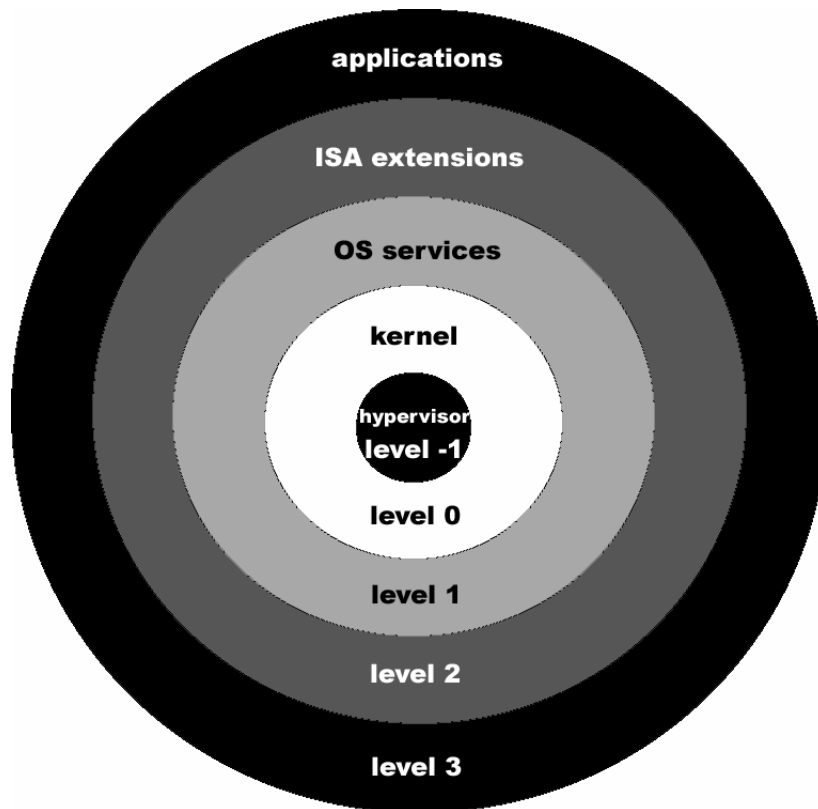


Figure 3. Privilege levels in the Intel VT x86-64 architecture.

²⁶ See *id.* at 419. Some researchers have developed special techniques by which a software application could try to detect whether the underlying operating system is running on a virtual, rather than a physical machine, but these techniques are experimental in nature and can generate both false positives and false negatives. See Ninad Ghodke & Renato Figueiredo, *On the Implications of Machine Virtualization for DRM and Fair Use: A Case Study of a Virtual Audio Device Driver*, Proc. 4th ACM Workshop on Digital Rights Management 91 (2004), available at <http://portal.acm.org/citation.cfm?id=1029162> (visited May 7, 2006).

²⁷ For a more detailed discussion of one such approach, see *infra* Section III.B.3.

Intel's 32-bit x86 (x86 for short) Pentium architecture, the longtime market leader, supports four "privilege levels" which permit the execution of increasingly restricted sets of x86 instructions, indicated respectively in Figure 3 by levels 0 (kernel), 1 (OS services), 2 (ISA extensions) and 3 (applications). To date, Windows and Linux have utilized only levels 0 and 3, with the operating system running in "kernel mode" at level 0 and user applications running in "user mode" at level 3. This design approach has helped prevent applications running at the same time from interfering with each other's use of system resources, but it has somewhat impeded efforts to build a native VMM for the x86 architecture that efficiently runs Windows and Linux-based VMs. This is because of the resource control requirement for efficient virtualization, wherein only the VMM, and not any of the guest OSes, should enjoy unrestricted access to system resources by operating at level 0.²⁸ Designers of some VMMs have coped with this problem by a method known as "ring depriving," in which the VM supports only levels 1, 2 and 3, and the guest OS runs at level 1 or level 3 instead of level 0.²⁹ Because this technique presents the guest OS with a VM architecture that is similar to, but not identical to, the underlying native hardware, it does not qualify as a true virtualization³⁰ and is often referred to as a "paravirtualization" technique.³¹ AMD's x86 Athlon architecture presents similar issues for virtualization.

Intel and AMD have responded to the recent growth in the virtualization software market by introducing "virtualization-ready" versions of their microprocessors that provide an additional privilege level below level 0, which is often referred to as "level - 1" for lack of a better name. The new privilege level is operative only when a VMM has been installed, and it assumes responsibility for controlling accesses to system resources, leaving levels 0-3 available for guest systems.³² In this way, the guest software is presented with a four-level VM architecture that matches the underlying native hardware, even while the VMM retains overall resource control.

These developments are roughly coinciding with an industry-wide migration to 64-bit x86 (x86-64 for short) processors, Intel's Itanium and AMD's Opteron. Intel's virtualization technology is being marketed under the names "VT-x" (for x86/Pentium) and "VT-I" (for x86-64/Itanium),³³ while AMD's virtualization technology for its x86-64/Opteron architecture has been given the brand name "Pacifica."³⁴

F. Trapping and Emulation

A corollary of the Popek-Goldberg resource control requirement is that the VMM must intercept, or "trap," any instructions from guest systems that might change the

²⁸ See *supra* text accompanying notes 22-23.

²⁹ See Rich Uhlig et al., *Intel Virtualization Technology*, COMPUTER, May 2005, at 48, 49-50.

³⁰ See *supra* text accompanying note 9.

³¹ See Smith & Nair, *supra*, at 422.

³² See Uhlig, *supra* note 29, at 52-54.

³³ See *id.*

³⁴ See Advanced Micro Devices, Inc., Press Release, *AMD Launches Industry's First X86-Based 64-Bit Server Virtualization Solution*, Feb. 15, 2005 (visited May 15, 2006) <http://www.amd.com/us-en/Weblets/0,,7832_8366_7595~94608,00.html>.

configuration of system resources (referred to as “control-sensitive” instructions).³⁵ Similarly, to satisfy the equivalence property, the VMM must trap any instructions whose behavior or results depend on the configuration of system resources (referred to as “behavior-sensitive” instructions).³⁶ An instruction that is either control-sensitive or behavior-sensitive (or both) may also be simply called “sensitive.”³⁷ (It may be noted here that “sensitive” is an exact antonym of “innocuous.”)³⁸

In addition to trapping sensitive instructions (and allowing the immediate execution of innocuous instructions), the VMM performs two other functions, “allocation” and “interpretation.”³⁹ The VMM performs its allocation function by assigning sufficient system resources to satisfy the requirements of each control-sensitive instruction, consistent with the VMM’s policies governing the overall allocation of system resources and resolution of conflicts.⁴⁰ The VMM performs its interpretation function by emulating each behavior-sensitive instruction: i.e., by performing a routine designed to produce the same behavior and result as if the instruction had been executed on a real machine.⁴¹ The actual implementation of the allocation and interpretation functions is itself an interesting technical achievement,⁴² but the details are immaterial for purposes of this Article and well beyond its scope.

G. Virtualization Overhead

Asking computer hardware to support system virtualization can be a tall order. Not only must the hardware do the work expected of all of the VMs it supports, but it must also perform additional operations as necessary to support the VMM’s allocation and interpretation functions. There is a further level of complexity in the case of a hosted VMM. Since a hosted VMM is designed to run on a host OS, its operations cannot be performed directly by the hardware, but must be implemented through the host OS API.⁴³

The additional load imposed by virtualization on system resources is referred to as *virtualization overhead*. While virtualization overhead can slow a computer system down, it may be a price worth paying for the ability to utilize multiple VMs and the many other useful features of virtualization software.⁴⁴ Because of the existence of virtualization overhead, however, virtualization is not a preferred approach to implementing highly-intensive and performance-critical applications that may require all of the resources of a dedicated physical server.

³⁵ See Smith & Nair, *supra*, at 384-85.

³⁶ See *id.*

³⁷ See *id.* at 385.

³⁸ See *id.*

³⁹ See *id.* at 386.

⁴⁰ See *id.*

⁴¹ See *id.* at 386-87.

⁴² See *id.* at 27-82, 329-68.

⁴³ See *supra* section II.C.

⁴⁴ See *supra* section II.B.

III. MICROSOFT AND THE MARKET FOR VIRTUALIZATION SOFTWARE PRODUCTS

A. The x86 Virtualization Industry

A detailed and critical examination of the rapidly changing virtualization software industry for purposes of antitrust enforcement should look beyond the public announcements of the current industry participants themselves, and would therefore require substantial investigatory resources. While it is necessary here to rely primarily on public statements, some additional third party research is also available in the form of a recent white paper published by Thomas Weisel Partners, a San Francisco-based institutional brokerage firm with equity research expertise in the software and computer industries. The paper, authored by Tim Klasell and Jeffrey Peck,⁴⁵ appears to provide an accurate and comprehensive review of the industry as of February 2006.

1. Size

The licensing market for x86 server virtualization software was around \$301 million in 2005 and was “one of the fastest-growing software markets,” with the possibility of growth to \$1 billion by 2008.⁴⁶ Growth in the market is driven by the sustained popularity of the x86 platform for increasingly larger working environments,⁴⁷ and a growing appreciation of the valuable functionalities and features that virtualization software can offer to computer users and the information technology professionals who serve them.⁴⁸

2. Participants

VMware. Around 1997, a team led by Stanford computer science professor Mendel Rosenblum began developing VMs to run different commodity operating systems, including Windows, simultaneously on the same computer system.⁴⁹ This work led to the founding of VMware, Inc. in 1998 by Rosenblum and his wife, Diane Greene, who are now chief scientist and president of the company, respectively.⁵⁰ Since its founding, VMware has been the leading vendor of x86 virtualization software products. VMware was acquired by EMC in 2003 for \$635 million.⁵¹

⁴⁵ Tim Klasell & Jeffrey Peck, *The Rise of the Virtual Machine and the Real Impact It Will Have: A White Paper on the Infrastructure Software Industry*, Feb. 3, 2006 (visited May 15, 2006) <http://www.xensource.com/files/VM_Whitepaper_TWP.pdf>.

⁴⁶ See *id.* at 3.

⁴⁷ See *id.*

⁴⁸ See *id.* at 6-9.

⁴⁹ See Edouard Bugnion et al., *Disco: Running Commodity Operating Systems on Scalable Multiprocessors*, PROC. 16TH ACM SYMP. ON OPERATING SYSTEM PRINCIPLES, at 143 (1997).

⁵⁰ See VMware, *VMware Leadership* (visited May 15, 2006) <<http://www.vmware.com/company/leadership.html>>.

⁵¹ See VMware, News Release, *EMC to Acquire VMware, Advancing Convergence of Server and Storage Virtualization*, Dec. 15, 2003 (visited May 15, 2006) <<http://www.vmware.com/news/releases/emc.html>>.

Since 2001, VMware has licensed ESX Server, a type-1 VMM software product for x86 hardware, and GSX Server, a type-2 VMM software product for Windows and Linux server OSes.⁵² In February 2006, VMware introduced VMware Server, a product described as a successor to GSX Server that will be licensed royalty-free.⁵³ As of April 2006, VMware had 20,000 customers for its \$3,750 ESX Server software product.⁵⁴ For the desktop, VMware offers VMware Workstation, a type-2 VMM software product for Windows and Linux desktop OSes,⁵⁵ and VMware Player, a royalty-free platform product for running single-application virtual machines in the form of “software appliances.”⁵⁶

Open Source Projects. Xen, the leading open-source virtualization software product, was produced by a team at the University of Cambridge Computer Laboratory. Xen is a type-1 VMM for x86 hardware. In 2003, the Cambridge team conducted a series of benchmarking tests in which it found Xen to outperform VMware Workstation 3.2 with respect to various performance measures, including overhead, performance isolation, and scalability.⁵⁷

Other open-source hardware virtualization projects include Bochs, CoLinux, FAUmachine, OpenVZ, and QEMU.

Red Hat. Red Hat has recently announced an “Integrated Virtualization” strategy, wherein the company “will provide seamless integration and support for the Xen virtualization technology” in its Red Hat Enterprise Linux Version 5 product scheduled for release by the end of 2006.⁵⁸ According to Red Hat, “[t]he resulting product will deliver the performance, robustness and security that enterprises demand.”⁵⁹

Microsoft. In 2003, Microsoft Corporation acquired Connectix, a developer of x86 hardware emulators for the Windows PC, Windows Server, and Macintosh PC operating systems.⁶⁰ Microsoft used Connectix’s technology to develop and release its

⁵²

⁵³ See VMware, News Release, *VMware Introduces Free VMware Server*, Feb. 6, 2006 (visited May 15, 2006) <http://www.vmware.com/news/releases/server_beta.html>.

⁵⁴ See Darrell Dunn, *Virtualization Vendors Jockey for Advantage*, INFORMATION WEEK, April 3, 2006, available at <http://www.informationweek.com/showArticle.jhtml?articleID=184417548> (quoting Microsoft product manager Jim Ni).

⁵⁵ See VMware, News Release, *VMware Announces Availability of New Feature-Packed Release of Its Powerful Desktop Virtualization Software*, Nov. 29, 2005 (visited May 15, 2006) <<http://www.vmware.com/news/releases/ws55.html>>.

⁵⁶ See VMware, News Release, *VMware Delivers VMware Player*, Dec. 12, 2005 (visited May 15, 2006) <<http://www.vmware.com/news/releases/player.html>>.

⁵⁷ See Paul Berham et al., *Xen and the Art of Virtualization*, Proc. ACM Symp. on Operating System Principles (2003), available at <http://www.cl.cam.ac.uk/Research/SRG/netos/papers/2003-xensosp.pdf>.

⁵⁸ Red Hat Corporation, *Red Hat Formally Announces Integrated Virtualization*, Press Release, March 14, 2006, available at <http://www.redhat.com/about/news/prarchive/2006/virtualization.html>.

⁵⁹ Id.; see also Peter Galli, *Red Hat “Integrated Virtualization” Strategy Thinks Big*, EWEEK, March 14, 2006 (discussing AMD’s and Intel’s responses to Red Hat’s virtualization plans).

⁶⁰ See Microsoft Corp., Press Release, *Microsoft Acquires Connectix Virtual Machine Technology*, Feb. 19, 2003 (visited May 15, 2006) <<http://www.microsoft.com/presspass/press/2003/Feb03/02-19PartitionPR.mspx>>.

first two virtualization products: Virtual Server 2005⁶¹ and Virtual PC 2004,⁶² which are type-2 VMM software products for the Windows Server and PC operating systems, respectively. Microsoft began royalty-free licensing of Virtual Server 2005 R2 in April 2006.⁶³ As of April 2006, Microsoft had 5,000 customers for Virtual Server 2005 R2.⁶⁴

3. Competitive Variables

On the other hand, the broad range of useful functionalities and features that virtualization can deliver, the heterogeneity of industry approaches to virtualization, and the complexity of designing and implementing virtualization solutions have all resulted in a high degree of differentiation in the market for virtualization software products. Thus, even though the competitive price for x86 virtualization software products may turn out to be zero,⁶⁵ there are many competitive variables that can serve as the basis for a robust competition among vendors seeking to extend the installed base of their virtualization software solutions.

Competitive Variable	Measurable Criteria
Management/Flexibility	Ease of use (in the face of new requirements and opportunities for real-time control and redeployment of complex system resources)
Reliability	Error prevention; fault detection and removal; user confidence
Enterprise OS Class Support	Industry-standard technical support services; certifications from software and hardware vendor partners
Security	Small, validated VMM code base, providing small “target footprint” and effective isolation from attacks against VMs
Performance	Low virtualization overhead; resource isolation between VMs; scalability

Table 1. Nonprice competitive variables in the market for virtualization software products.

⁶¹ See Microsoft Corp., Press Release, *Microsoft Announces General Availability of Virtual Server 2005*, Sept. 13, 2004 <<http://www.microsoft.com/presspass/press/2004/sep04/09-13availabilityvs2005pr.msp>>

⁶² See Microsoft Corp., Press Release, *Microsoft Releases Microsoft Virtual PC 2004 to Manufacturing To Ease Customer Migration to Windows XP*, Nov. 10, 2003 (visited May 15, 2006) <<http://www.microsoft.com/presspass/press/2003/nov03/11-10VPC2004RTMPR.msp>>.

⁶³ See Microsoft Corp., Press Release, *Q&A: Bringing the Benefits of Virtualization Technology to the Mainstream*, April 3, 2006 (visited May 15, 2006) <<http://www.microsoft.com/presspass/features/2006/apr06/04-03virtualizationqa.msp>>.

⁶⁴ See Dunn, *supra* note 54.

⁶⁵ See, e.g., Dunn, *supra* note 54 (quoting Virtual Iron president John Thibault) (“We believe that over time the hypervisor and virtualization stack is going to be commoditized, and the real value is going to be in the management as we build a very viable alternative to VMware.”).

Even if the competitive price is nonzero, Klasell and Peck suggest that price differentials will often be immaterial to a user’s choice of products. See Klasell & Peck, *supra* note 45, at 15 (“Given the overall cost of labor and the other [nonprice] concerns above, we think the current \$1,000 to \$2,500 is less important than the value provided above, as noted by the historical growth of the industry.”).

Table 1 itemizes five nonprice variables identified by Klasell and Peck in their analysis of the x86 virtualization software product market⁶⁶ and identifies some of the criteria by which each of these variables might be measured in an economic analysis of the market's structure and performance.

B. Microsoft's Virtualization Strategy

Microsoft has made a series of public statements that its current line of standalone type-2 VMM software products is only a transitional strategy, and that the company's plan is for a type-1 VMM ultimately to be "integrated" into the Windows software code base. This section will describe, in somewhat more detail, how Microsoft is currently implementing this plan.

1. Support for Legacy VMs

The new VMM will be backward compatible, supporting VMs that were created using Microsoft's Virtual Server 2005 and Virtual PC 2004 products.⁶⁷ Thus, Microsoft is laying the groundwork for widespread adoption of its type-1 VMM by encouraging Windows users to generate an installed base of VMs that will be compatible with Microsoft's next-generation virtualization products.

2. Licensing Windows Server By Running Instance

Microsoft is also encouraging early adoption of its server virtualization technology by moving to a licensing model that benefits certain VM-intensive uses of its existing products. In December 2005, Microsoft introduced several changes to its licensing terms for its Windows Server family of products, including two changes that directly affect current users of Microsoft's virtualization products.⁶⁸

First, licensees of Windows Server 2003 R2 Enterprise Edition are now permitted to run up to four virtual instances (i.e., copies) of the OS on one physical server at no additional cost.⁶⁹ This change will allow enterprises to experience some of the benefits of virtualization on a limited basis, while retaining for Microsoft the opportunity to pursue additional licensing revenues from more intensive users of the technology.

⁶⁶ See Klasell & Peck, *supra* note 45, at 15.

⁶⁷ See Microsoft Corp., Press Release, *Microsoft Announces Comprehensive Virtualization Strategy to Enable Self-Managing Dynamic Systems*, Oct. 10, 2005 (visited May 15, 2006) <<http://www.microsoft.com/presspass/press/2005/oct05/10-10VirtualizationStrategyPR.msp>> (hereinafter "Microsoft Virtualization Strategy"); Microsoft Corp., Volume Licensing White Paper, *Licensing Microsoft Server Products with Microsoft Virtual Server R2 and Other Virtual Machine Technologies*, Dec. 2005 (visited May 15, 2006) <http://download.microsoft.com/download/6/8/9/68964284-864d-4a6d-aed9-f2c1f8f23e14/virtualization_whitepaper.doc> (hereinafter "Microsoft Virtualization White Paper").

⁶⁸ See Microsoft Virtualization Strategy.

⁶⁹ See *id.*

Second, all products in the Windows Server family are now licensed under terms that permit the creation and storage of an unlimited number of software instances, and limit only the number of software instances that can be *running* at any given time.⁷⁰ Microsoft has noted that this change will allow data centers to maintain a library of inactive backup copies of running VMs that will be ready to deploy in a recovery from a system failure.⁷¹

Present and future users of non-Microsoft virtualization software products do not appear to be in a position to benefit from Microsoft's new licensing policies. Windows includes a product activation feature, in which the operating system creates a numerical key based on the hardware in the machine in which it is installed.⁷² When Windows is installed on a third-party VM, this numerical key is specific to the virtual hardware configuration of the VM. Thus, for example, VMware users are advised that they will need a separate OS license for each installation of a guest OS inside a VMware VM,⁷³ and that all aspects of the VM configuration should be set before performing the Windows product activation.⁷⁴ By requiring users of third-party VMMs to verify their licenses at install time rather than run time, Microsoft's product activation feature may prevent such users from exercising the full scope of their rights under Microsoft's new Windows Server licensing terms.⁷⁵

3. Virtualization of Windows Server

Microsoft's type-1 VMM software, referred to as "Windows Server Virtualization," is currently under development. A beta version of the software was demonstrated at Microsoft's annual Windows Hardware Engineering Conference (WinHEC) in May 2006, where Microsoft officials also announced the timeframe for the software's release and described plans for its design and implementation.⁷⁶ Microsoft's rollout of Windows Server Virtualization will coincide roughly with the release of the

⁷⁰ See *id.*

⁷¹ See *id.*

⁷² According to a Microsoft sales presentation, the installation of a

⁷³ See VMware, *Guest Operating System Installation Guide* at 16-17 (visited May 15, 2006) <<http://www.vmware.com/support/pubs/>>. The guide states that "[a]s with physical computers, a separate operating system license is required for each virtual machine you run." *Id.* at 17. To install an instance of an OS on a VMware VM, however, it is necessary first to "boot up," or run, the VM. See *id.* Thus, the requirement of a separate OS license is effective at the time the OS is installed on a new VM.

⁷⁴ See *id.* at 22.

⁷⁵ Microsoft's press release and white paper on its licensing changes for virtualization products nowhere discuss Windows Product Activation, and in particular, do not address how Windows Product Activation may or may not account for a licensing model based on the number of running instances. See Microsoft Virtualization Strategy, *supra*; Microsoft Virtualization White Paper, *supra*. A February 2005 PowerPoint presentation by Microsoft's Rich McBride states that Virtual Server and Virtual PC support the creation of VMs with "[n]o 'hardware detection step'" and "[n]o Product Activation step," but it is unclear whether these statements apply to all Microsoft software that may be installed within the VMs. Rich McBride, *Virtual Server 2005 Overview* (visited May 15, 2006) <<http://www.snenug.org/Presentations.htm>>. It is possible that a separate, proprietary digital rights management mechanism has been built into Microsoft's Virtual Server and Virtual PC (and will be built into Windows Hypervisor) to accommodate unlimited software installations while limiting running software instances of Windows Server products.

⁷⁶ See Microsoft Corp., *WinHEC 2006 Conference Sessions: System Fundamentals – Virtualization* (visited May 25, 2006) <<http://www.microsoft.com/whdc/winhec/trackdetail06.msp?track=3>>.

next generation of Microsoft’s Windows Server, codenamed “Longhorn.” Microsoft plans to distribute Windows Server Virtualization initially as a “service pack” update to the first release of Windows Server “Longhorn,” and ultimately as an “integrated” part of the second release of Windows Server “Longhorn.”

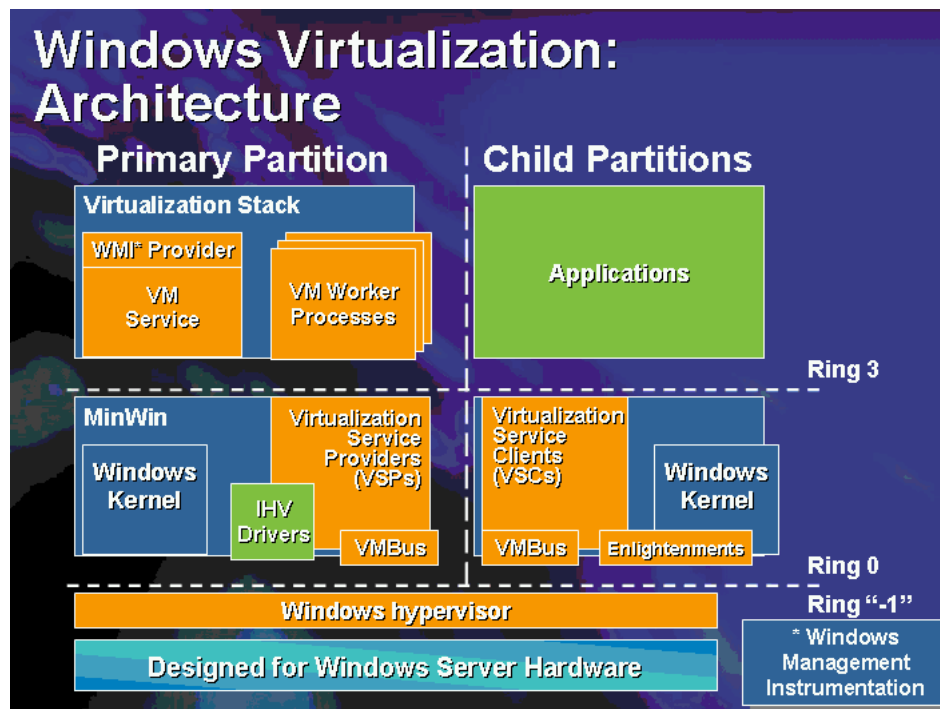


Figure 4. Screenshot from Microsoft presentation on planned Windows VMM architecture.⁷⁷

In a November 2005 Webcast, Microsoft’s Jeff Woolsey described the architecture of the planned combination of the operating system and VMM. As shown in Figure 4, the VMM will be made up of several components. First, a streamlined, “microkernelized” component, known informally as “Windows hypervisor,”⁷⁸ will be installed first on the server hardware, under all operating systems. By installing the Windows hypervisor in “Ring –1,” Microsoft’s design will rely on the new privilege level that Intel and AMD are building into their next-generation microprocessors to support virtualization.⁷⁹ The Windows hypervisor will consist of a minimal code base that contains no third-party or untrusted code,⁸⁰ with the aims of maximizing security and

⁷⁷ Jeff Woolsey, Understanding Windows Hypervisor and Virtualization in Windows Server Codenamed “Longhorn,” Nov. 11, 2005 (visited May 25, 2006) <<http://msevents.microsoft.com/CUI/WebCastEventDetails.aspx?EventID=1032284286>> (screenshot at 51 minutes; registration required).

⁷⁸ “Hypervisor” is a registered trademark of International Business Machines Corp., see U.S. Patent & Trademark Office, Trademark Electronic Search System (visited May 15, 2006) <<http://tess2.uspto.gov/bin/showfield?f=doc&state=c81n0e.4.1>>, but Microsoft is continuing to use the term generically.

⁷⁹ See *supra* text accompanying note 32-34.

⁸⁰ See Woolsey, *supra* note 77, at minute 49.

stability.⁸¹ The hypervisor will be support a “simple partitioning functionality”⁸² and will also be responsible for controlling access to the CPU, memory, and hardware devices.⁸³

Microsoft’s design envisions the partitioning of a server’s system resources into a “primary partition” and some number (possibly zero) of “child partitions” that are started, managed, and shut down by their respective parents.⁸⁴ Child partitions can have children of their own, but all child partitions are ultimately subject to the management of the primary partition through a chain of parent-child relationships. For this reason, child partitions are also referred to as “secondary partitions.”

In Microsoft’s design, the first instance of the Windows Server “Longhorn” OS on any server will be installed into the server’s “primary partition,” with a minimal version of Windows to run at privilege level 0, and a virtualization stack — responsible for implementing additional VMM functionality — to run at privilege level 3. Hardware device drivers, which are made available to secondary partitions via a client-provider protocol, are also installed in the primary partition to run at privilege level 0.

Guest OSes and their applications may be installed into secondary partitions to run at the same privilege levels as they would run on native hardware. Figure 4 illustrates the case where the guest OS is a further instance of Windows Server “Longhorn.” Windows Server “Longhorn” will include special software components called “enlightenments” that will inform the OS that it is running as a guest in a secondary partition, so that the OS can reduce the VMM software’s virtualization overhead when performing memory management and other functions in the Windows API.⁸⁵ These performance optimizations will be supported not only by enlightenments in the guest OS, but also by host-side modifications to the Windows hypervisor and virtualization stack.⁸⁶ Because of the latter modifications, even though Microsoft describes child partitions as equivalent to VMs,⁸⁷ the interface they present to a guest OS is actually different from the underlying hardware.

⁸¹ Microsoft’s new microkernelized hypervisor design appears to be in part a response to critics who find the large Windows OS code base too unstable and insecure. Microsoft had planned to implement its VMM as a type-2 hypervisor until recently. See Charlie Demerjian, *Microsoft is Trapped in a Virtual Bind*, INQUIRER, June 20, 2005 (visited May 1, 2006) <<http://www.theinquirer.net/?article=24060>>. At the time, commentators noted one drawback of this approach: the VMM would only be as secure and stable as the Windows OS it was running on. See *id.* In particular, the ability of a Windows-hosted VMM to switch to a stored VM in a failover situation would be compromised if Windows itself failed. As one commentator put it, “If Windows flakes out and goes BSOD [blue screen of death] on you, your hosted OS dies as well.” *Id.*

⁸² Woolsey, *supra* note 77, at minute 49.

⁸³ Microsoft Corporation, *Microsoft Machine Virtualization Roadmap Chat*, Aug. 26, 2005 <http://www.microsoft.com/technet/community/chats/trans/windowsnet/05_0826_tn_muglia.msp> (visited March 20, 2006).

⁸⁴ See *id.* at minute 47.

⁸⁵ See *id.* at minute 54.

⁸⁶ See George Ou, *Microsoft Reveals Virtualization Strategy and Architecture*, TECHREPUBLIC.COM, May 28, 2006 <<http://blogs.techrepublic.com.com/Ou/?p=233>>.

⁸⁷ See *id.* The child partitions will provide support for both x86 and x86-64 guests. See *id.* at minute 55.

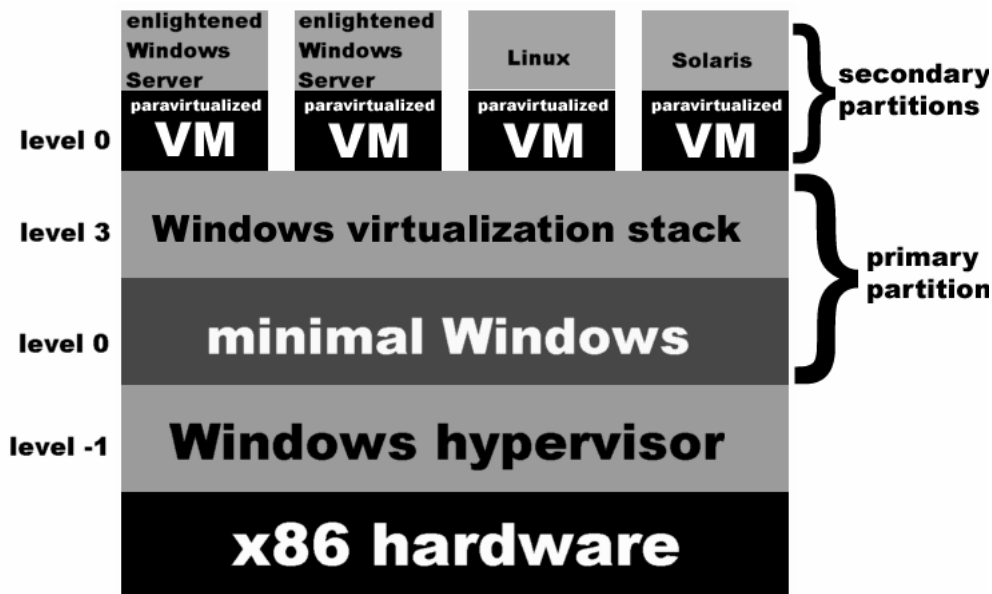


Figure 5. Platform dependency relationships in the planned Windows VMM architecture.

Figure 5 provides a different view of the Windows VMM architecture presented in Figure 4, highlighting the dependency relationships among the various components of Windows Server “Longhorn,” Windows Server Virtualization, and the various VMs that may be installed on the server.

At the 2006 WinHEC conference, Microsoft officials discussed the company’s plans to license the software modifications used to implement the Windows “enlightenments.”⁸⁸ According to technology reporter George Ou, Microsoft plans to license the guest-side modifications to OS vendors royalty-free, but will license the host-side modifications to hypervisor vendors only at a charge.⁸⁹

4. “Integration” of Virtualization into Windows

While Microsoft plans to distribute Windows Server Virtualization initially as a “service pack” update six months after Windows Server “Longhorn” begins shipping in 2007,⁹⁰ the company’s ultimate plan is to offer an “integrated” Windows Server product with the Windows Server Virtualization software already built into the operating system software.⁹¹ Current plans are to introduce the integrated product as Release 2 of Windows Server “Longhorn,” expected in 2009 or 2010.⁹²

⁸⁸ See id.

⁸⁹ See id.

⁹⁰ See Scott Bekker, *Microsoft Maps Out Plan to Integrate Virtualization Into Windows*, REDMONDMAG.COM, Oct. 2005 (visited May 15, 2006) (reporting Bob Muglia’s statement that “[Hypervisor support in Longhorn] will ship after Longhorn Server” and noting that “Longhorn Server is promised for 2007.”) (bracketed text in original).

⁹¹ See Microsoft Corporation, *Microsoft Machine Virtualization Roadmap Chat*, Aug. 26, 2005 <http://www.microsoft.com/technet/community/chats/trans/windowsnet/05_0826_tn_muglia.mspx> (visited March 20, 2006) (reporting comments of senior vice president Bob Muglia) (“[A]fter we ship

By that time, Microsoft officials say they expect virtualization to be preinstalled as a system default on almost all servers shipped.⁹³ Microsoft's integration plans will ensure that for Windows Users, the preinstalled virtualization technology will be Windows Server Virtualization. At the May 2006 WinHEC conference, Microsoft's Jeff Woolsey argued that this "standardization" will deliver benefits even to users of non-Microsoft OSes:

[W]hy is Microsoft supporting non-Microsoft operating systems in their virtualization solutions? The answers are, interoperability, and standardization. Our customers have told us, they want to standardize on a single platform for virtualization, they don't want to use one virtualization technology for Windows, and another virtualization technology for other operating systems. Since the majority of operating systems being virtualized are Windows, it only makes sense that we provide our customers the best platform for virtualization, so that they can standardize on Windows Server virtualization, and Longhorn Server.⁹⁴

Still, acknowledging that some users have reason not to use virtualization when running "really large applications," Microsoft officials have indicated that the company will continue to provide those users the option of not using virtualization.⁹⁵

Longhorn we will be building a hypervisor and virtualization capabilities directly into the Windows Server operating system. So we're actively working on building a new set of technologies that are from the ground up, designing a very modern hypervisor as a part of Windows Server to allow for virtualized sessions.") (hereinafter "Virtualization Roadmap Chat"); Bob Muglia, Remarks at Microsoft Professional Developers Conference 2005, Sept. 15, 2005 (visited May 15, 2006) <<http://www.microsoft.com/presspass/exec/bobmuglia/09-15PDC2005.msp>>; Bekker, *supra* note 90 <<http://redmondmag.com/reports/article.asp?EditorialsID=188>> (reporting Muglia's statement that "The future of virtualization is in thin hypervisors with a virtualization stack built into the OS. That is what we're doing.").

⁹² See Paula Rooney, *Xen Pushes Virtual Software Battle Upstream*, CRN, Jan. 6, 2006 (visited May 15, 2006) <<http://www.crn.com/sections/microsoft/microsoft.jhtml?articleId=175801784>> ("[Microsoft] announced plans this year to integrate a virtualization hypervisor directly into Release 2 of its Windows Longhorn Server operating system due in the 2009-2010 time frame.").

⁹³ Bob Muglia, WinHEC 2006, May 23, 2006 (visited May 25, 2006) <<http://www.microsoft.com/presspass/exec/bobmuglia/05-23WinHEC06.msp>> ("In the long-term, we expect virtualization to be standard on all servers shipped and, in fact, eventually become the default mode, that is that all workloads will run in a virtualized environment. I don't expect that to happen for at least three to four years, but I do expect that over the next few years, more and more customers will begin to utilize virtualization for a very large percentage of the work that they do.").

⁹⁴ See Bill Gates, *Advancing the Platform, 2006 WinHEC Keynote Address*, May 23, 2006 (visited May 25, 2006) <<http://www.microsoft.com/billgates/speeches/2006/05-23WinHEC.asp>>.

⁹⁵ In response to a question from an audience member during his Webcast, Microsoft's Jeff Woolsey said: "[I]f you have a large enterprise-class SAP or SQL database, for example, pushing a four-processor server, and it's pushing that pretty hard, that would not be an example of a workload I would virtualize. And that's actually one of the differences between our approach at Microsoft and other virtualization folks. We've heard a lot of people tell me that the other people out there say, go virtualize everything. And that's not necessarily the Microsoft approach. We want to provide a number of solutions, not just provide a one-size-fits all answer to every consolidation effort out there, because we don't think that's prudent and in the customer's best interest."

Woolsey, *supra* note 77, at minute 65.

A likely consequence of standardization on Windows Server Virtualization will be the foreclosure of other virtualization software products. One Microsoft official suggested that Microsoft's virtualization plans are already causing customers to "think[] twice about spending thousands of dollars for a [third-party] virtualization product that will be free in the future and part of the operating system."⁹⁶ One developer has predicted that Microsoft's distribution of system virtualization software at no additional cost to Windows Server users will "eat VMWare's lunch."⁹⁷

5. PC Virtualization

Compared with Microsoft's detailed roadmap for server virtualization, the company's plans for client virtualization are less clear. Microsoft's immediate plan is to provide an update to Virtual PC for use with the forthcoming release of Windows Vista.⁹⁸ The update will provide improved performance and x86-64 support.⁹⁹ Microsoft has made it clear that it plans eventually to incorporate virtualization software into the Windows PC OS code base, but has made no product-related announcement as to the form the combination will take.¹⁰⁰ Given Microsoft's durable monopoly power in the market for PC OS software products, these developments will warrant further scrutiny as more details become available.

IV. DO MICROSOFT'S PLANS CONSTITUTE AN ILLEGAL TYING ARRANGEMENT?

Any analysis of the antitrust implications of changes to the designs of software products must begin from first principles, with an accurate understanding of what a software product is. In a previous article, I identified the legal rights and technical capabilities that comprise a software product, and distinguished them from the software code that accompanies, but is not normally included with, a software product.¹⁰¹ I also showed how to characterize these rights and capabilities at an appropriate level of abstraction for antitrust analysis.¹⁰² Based on this first principles approach, I have argued in a second article that for Microsoft's conduct in combining its Web browser and operating system software products in Windows 98, the company should have been found liable for illegal tying even under the most lenient approaches to tying analysis that were proposed during the *United States v. Microsoft* litigation.¹⁰³

⁹⁶ See Dunn, *supra* note 54.

⁹⁷ Microsoft Corporation, Virtualization Roadmap Chat, *supra* note 91.

⁹⁸ See *id.*

⁹⁹ See *id.*

¹⁰⁰ See Microsoft Corporation, Virtualization Roadmap Chat, *supra* note 91 ("We will be adding virtualization to Windows Client however no product vehicle has been determined.").

In response to the question "Will [Virtual PC] (or some form of it) be part of the Vista product line?," Microsoft's Mike Neil replied, "We are building virtualization support into Windows." *Id.*

¹⁰¹ See Antitrust Analysis, *supra* note 2, at 25-28.

¹⁰² See *id.* at 28-32.

¹⁰³ See Decoding *Microsoft*, *supra* note 2.

In this section, I will embark on a similar liability analysis with respect to Microsoft’s plans for “integrating” a virtualization software product into Windows Server “Longhorn.” Based on the limited facts currently available, I will try to conduct a fairly thorough per se analysis here [and then venture some preliminary thoughts toward a rule of reason analysis during our discussion]. Because this work serves in part to enrich and generalize the arguments and methodologies I presented earlier, the reader may find it helpful to refer to my two previous “First Principles” articles as a frame of reference for the present study.

A. Software Products and Provisioning

Software is code.¹⁰⁴ A *system* consists of one or more computers, together with associated *hardware resources* and software. Software is *used* by installing and running it on a system, thereby producing *system behavior*. Consumers desire to use software for producing system behavior that supports various *tasks* (which are sometimes also referred to as *functionalities*). System behavior of the kind that supports a task occurs in the form of an *interaction* between a user and the system. Systems are often configured in a network that connects one or more *client* computers with one or more *server* computers. A client-server system supports interactions between each user and her respective client computer by running software that may be on the client itself, or *distributed* between the client and one or more additional computers on the network (e.g., the client and a server).

In response to consumer desires for user-system interactions, producers market *software products*. A software product is defined by reference to accompanying software and *documentation*, and consists essentially of the necessary legal rights, and technological capabilities, to install and run the software on a system according to the documentation; it does not include any of the software or documentation itself, in which the vendor retains copyright.¹⁰⁵ The documentation describes legal and technological *preconditions* for using the software product, and tasks that may be supported by using the software product subject to such preconditions.

¹⁰⁴ Software code may be *source code* or *object code*. Source code is “[h]uman-readable program statements written by a programmer or developer in a high-level or assembly language that are not directly readable by a computer” and “needs to be compiled into object code before it can be executed by a computer.” MICROSOFT CORPORATION, MICROSOFT COMPUTER DICTIONARY 418 (1999). Object code is “[t]he code, generated by a compiler or an assembler, that was translated from the source code of a program,” usually by the software developer or vendor prior to the distribution of the software that accompanies a software product. *Id.* at 317. Throughout this article, the term “code” will generally be used to refer to object code.

¹⁰⁵ For example, in a suit by Microsoft for copyright infringement, a defendant who had legitimately obtained a single user license for Microsoft Office could not validly claim to own the accompanying software code itself. *See, e.g., Microsoft v. Software Wholesale Club, Inc.*, 129 F. Supp. 2d 995, 1007-08 (S.D. Tex. 2000) (holding defendant liable for copyright infringement because, *inter alia*, defendant was unable to show that counterfeit copies had good chain of title).

This distinction between a software product and its accompanying software is, of course, a fundamental one. To preserve it, I will employ the somewhat unwieldy terminology “software that accompanies a software product” throughout this Article. As a side benefit, however, the distinction also permits me to discuss the “sale” or “purchase” of a software product without implying that the accompanying software is sold or purchased.

The use of a software product may require a system to run not only the software that accompanies the software product, but also other software that has previously been installed on the system. It may therefore be a precondition for using one software product that another software product has previously been acquired and its accompanying software preinstalled on the system. In such a case, the two products are recognized as complements, not substitutes;¹⁰⁶ and any required preinstalled software is referred to as *platform software*. Platform software is often installed on the system's hard drive(s) in the form of *library files*.¹⁰⁷

A software product may specify which software is to run on the system when the software product is used, even though not all such software necessarily accompanies the software product.¹⁰⁸ For example, a program may instruct the system to run specific routines in preinstalled platform software by using the conventions, or *calls*, defined in the platform software's *programming interface* (which is part of the documentation accompanying the platform software).¹⁰⁹

A collection of software products is said to *support a task* if it specifies which software is to run on the system in order to produce behavior that supports the task, and (subject to any documented preconditions) confers sufficient legal rights and technological capabilities to run such software.

The collection of software products that must be acquired and installed together to support a particular functionality is referred to as a *software stack*. The software stack for a given functionality may be distributed among multiple computers, as in a client-server system. Software stacks are described as involving multiple *layers*, where the use of one layer requires the previous installation of other layers. The term *middleware* refers to platform software that itself requires other platform software to be preinstalled.

The documented preconditions for use of a software product usually include minimum requirements for available hardware resources to ensure that the accompanying software can be used when it is installed and run on a system according to the documentation. Beyond these documented requirements, a system may require additional hardware resources to improve software performance, to support intensive software use, or to address undocumented software issues. The collection of computers and other

¹⁰⁶ This distinction is especially significant in the context of product market definition. A properly defined relevant market includes goods that are reasonably close substitutes for one another, but not complementary goods. See IIA PHILLIP E. AREEDA ET AL., *ANTITRUST LAW*, ¶ 565a-b, at 329-32.

¹⁰⁷ See *Free On-Line Dictionary of Computing* <<http://www.foldoc.org>> (last visited December 1, 2003) (defining "library" as "[a] collection of subroutines and functions stored in one or more files, usually in compiled form, for linking with other programs").

¹⁰⁸ Specifically, the software that accompanies a software product may make procedure calls to previously installed software, as when an application makes calls to the application programming interfaces of an operating system. See *id.* For a more detailed description of this process, see JOHN R. LEVINE, *LINKERS & LOADERS* 187-227 (2000) (describing linking of code using shared libraries, including Windows dynamically linked libraries). Cf. *United States v. Microsoft Corp.*, 84 F. Supp. 2d 9, 50, at ¶ 162 (describing "knitting" together of different software layers).

¹⁰⁹ See *Free On-Line Dictionary of Computing*, *supra* note 107 (defining "application program interface" as "[t]he interface (calling conventions) by which an application program accesses operating system and other services").

hardware resources that must be acquired and installed together to support a particular functionality is referred to as a *hardware stack*.

Provisioning is the function of ensuring that adequate resources are available to enable a system to support a particular task for a particular user. When a single user wishes to run a single PC software application, provisioning can be as simple as acquiring and plugging in the machine, and (assuming that the necessary OS has been preinstalled) acquiring and installing the necessary software product. The provisioning of both software and hardware for a system on which many users must perform many different tasks, however, can present many complex, time-sensitive problems for an administrator. In response to these problems, some vendors have developed *provisioning software products*, which support some of the provisioning tasks commonly confronting system administrators. For example, there are several software products directed to the problem of managing software licenses: i.e., to ensure, in advance, that the user of a system has the legal right to install and run each layer of the software stack that supports the user's desired functionality.¹¹⁰

B. The Tying and Tied Products in Windows Server “Longhorn”

The plaintiff who advances a theory of liability for illegal tying has the sole right — and the sole duty — to identify the alleged tying and tied products that serve as predicates for the claim. The concepts of a “tying product” and a “tied product” are therefore distinct from that of a “software product,” inasmuch as the legal rights and technological capabilities that constitute a software product are often marketing decisions made by the product's vendor. If tying doctrine supports the plaintiff's allegations, the fact that a vendor markets the tying product and tied products as a unitary item will not preclude a court from finding the item to be a combination of separate tying and tied products.¹¹¹

As described by Microsoft's officials,¹¹² the second release of Windows Server “Longhorn” will be accompanied by software that serves both (1) as an operating system platform for Windows applications, and (2) as a provisioning tool that allows an administrator to construct a virtual hardware stack on which a guest operating system, and software stacks based on that platform, can be installed and run. For purposes of establishing a tying claim based on Microsoft's virtualization plans for Windows Server “Longhorn,” I will therefore characterize Windows Server “Longhorn” as a bundle of two products: an operating system platform (the alleged tying product) and a provisioning tool (the alleged tied product).

The operating system platform consists essentially of a limited, nonexclusive right, and the necessary technological capabilities, to make copies and adaptations of the

¹¹⁰ See *Providers of License-Management Software*, PROCESSOR.COM, March 4, 2005 (visited May 20, 2006) <<http://www.processor.com/articles//P2709/35p09/35p09chart.pdf>>.

¹¹¹ See X AREEDA, *supra* note 106, ¶ 1741a, at 177 (citation omitted) (noting that in tying analysis, “the essence of what constitutes ‘one product’ cannot be resolved by logic, language, or physical considerations”); see also *id.* at 175 (“[J]ust about any two products could be described as mere parts in a more encompassing single product”).

¹¹² See *supra* Section III.B.4.

software code accompanying Windows Server “Longhorn” on a (real or virtual) machine for the purposes of supporting the functions of applications¹¹³ that reference the software’s APIs,¹¹⁴ and controlling the allocation and use of the machine’s resources by these applications.¹¹⁵ The provisioning tool (i.e., a VMM) consists essentially of a limited, nonexclusive right, and the necessary technological capabilities, to make copies and adaptations of the software code accompanying Windows Server “Longhorn” for the purpose of constructing one or more virtual hardware stacks on which a guest operating system platform, and software stacks based on that platform, can be installed and run.

Three ramifications of this characterization may initially seem counterintuitive. First, a precondition for the use of the provisioning tool in Windows Server “Longhorn” will be that the OS platform software be installed. In other words, the alleged tied product is useless without the alleged tying product. This functional dependence is unexceptional and has no bearing on tying liability.¹¹⁶

Second, as a result of Microsoft’s decision to market Windows Server “Longhorn” as a unitary item, the same software will happen to accompany two different software products. While the concept of different software products with the same accompanying software might seem novel, this situation is actually quite common in the software industry. Software licenses often contain terms specifying legal restrictions on use.¹¹⁷ Digital rights management technologies, reinforced by legal prohibitions against circumvention,¹¹⁸ are increasingly being used to give effect to technological restrictions on use. The existence of such use restrictions generally benefits software vendors, who can use them to facilitate price discrimination.¹¹⁹ When different users obtain licenses to install and run the same software pursuant to different use restrictions, they thereby obtain different software products. We defer for now the question of whether such software products are “separate products” for purposes of tying doctrine.¹²⁰

Third, the legal rights that comprise the alleged tying and tied products will be derived from the same copyright. As Judge Jackson noted in the *Microsoft* summary judgment opinion, Judge Jackson noted the IBM cases of the 1960s and 1970s, in which courts generally held that “‘where a court is dealing with what is physically and in fact a single product,’ the antitrust laws do ‘not contemplate judicial dissection of that product into parts and the reconstitution of these parts into a tying agreement.’”¹²¹ My

¹¹³ An application is a “software program[] . . . that perform[s] specific user-oriented tasks.” United States v. Microsoft Corp., 84 F. Supp. 2d 9, 12, at ¶ 2 (D.D.C. 1999).

¹¹⁴ See *supra* text accompanying note 109.

¹¹⁵ An operating system (“OS”) is a “software program that controls the allocation and use of computer resources (such as processing time, main memory space, disk space, and input/output channels).” *Id.*

¹¹⁶ See *Jefferson Parish Hosp. Dist. No. 2 v. Hyde*, 466 U.S. 2, 19 (1984) (“[T]he answer to the question whether one or two products are involved turns not on the functional relation between them, but rather on the character of the demand for the two items.”).

¹¹⁷ See 2 L.J. KUTTEN, COMPUTER SOFTWARE § 8.02[2][f][ii], at 8-34 to 8-36.

¹¹⁸ See 17 U.S.C.A. § 1201.

¹¹⁹ See *id.*

¹²⁰ See *infra* section IV.E.

¹²¹ See *Microsoft*, 1998 WL 614485, at *8 (quoting *Telex Corp. v. IBM*, 367 F. Supp. 258 (N.D. Okla. 1973), *rev’d on other grounds*, 510 F.2d 894 (10th Cir. 1975)); cf. *Kenworth of Boston, Inc., v. Paccar Financial Corp.*, 735 F.2d 622, 624 (1st Cir. 1984) (Breyer, J.) (“We need not treat the concept of a ‘tie’ like a procrustean bed onto which this practice must be squeezed or stretched”). Judge Jackson

identification of distinct operating system and Web browser software products does not involve a dissection of Windows Server “Longhorn”’s code, but arguably does involve a “judicial dissection” of Microsoft’s exclusive rights in the software accompanying Windows Server “Longhorn” under § 106 of the Copyright Act.

Section 106’s grant of enumerated exclusive rights, however, is expressly subject to other statutory provisions and cannot be read in isolation.¹²² Moreover, the Copyright Act does not operate to convert all products derived from a single copyright into what is “physically and in fact a single product” for purposes of antitrust analysis. Instead, the statute contemplates that rights in a single copyrighted work may be licensed so as to enable different licensees to exploit the respective demands for different uses of the work.¹²³ Hovenkamp, Janis and Lemley’s treatise on intellectual property and antitrust clearly illustrates this point:

[T]he fact that two different media are protected by the same intellectual property right does not seem to be decisive one way or another. For example, suppose a theater showing *Titanic* after videos were issued required patrons to purchase a copy of the *Titanic* video cassette as a condition of getting a seat in the theater. Ordinarily, one observes, movie tickets and video cassettes are sold separately, and this combination should be regarded as a tie of separate products.¹²⁴

As I will argue, the “integrated” version of Windows Server “Longhorn” serves in part to exclude competing software developers from using Intel’s VT-x and VT-i virtualization technologies and AMD’s Pacifica technology. Since the Copyright Act provides no warrant for such a restraint, the single copyright on Windows Server “Longhorn”’s code should not be taken to imply that Windows 98 is a single product for purposes of tying law.

distinguished the IBM cases by noting that IBM offered the two bundled products in their unbundled form. See *id.* at *9.

¹²² See 17 U.S.C. § 106 (granting certain exclusive rights to the copyright owner “[s]ubject to sections 107 through 122”).

¹²³ See *New York Times v. Tasini*, 533 U.S. 483, 505 (2001) (holding that a license granting the right to publish a freelance author’s article in a collective work did not extend to authorize the republication of the article in an online database). The Court reasoned:

Essentially, [17 U.S.C.] § 201(c) adjusts a publisher’s copyright in its collective work to accommodate a freelancer’s copyright in her contribution. If there is demand for a freelance article standing alone or in a new collection, the Copyright Act allows the freelancer to benefit from that demand; after authorizing initial publication, the freelancer may also sell the article to others.

Id. at 497.

¹²⁴ HOVENKAMP ET AL., *supra* note 127, at § 22.3, at 22-15.

According to some courts and commentators, no tying liability should be found where the Copyright Act specifically authorizes the bundling of the alleged tying and tied products. See, e.g., *Waldbaum v. Worldvision Enterprises, Inc.*, 1978 WL 956, at *6 (S.D.N.Y. Nov. 21, 1978) (citing 17 U.S.C. §§ 1(a) & (d) (1909)) (holding that a requirement that a licensee of films also purchase prints is “within the statutory copyright monopoly” and does not represent a tie-in); X AREEDA, *supra* note 106, ¶ 1749b, at 259 & n.35 (citing *Waldbaum* for the proposition that “a single product should be found when copyright law itself bundles certain rights, such as a book or film copyright that its owner bundles with hard copies of the copyrighted work”).

C. Tying Defined

Before proceeding to the liability analysis of the alleged tie-in, it is helpful to review the necessary elements for liability under both the generally applicable per se rule and the D.C. Circuit's rule of reason approach

A tying arrangement exists when a seller conditions the sale of one product (the "tying product") on the buyer's purchase of another product (the "tied product"). Where the seller has market power in the tying product, a tying arrangement may have the effect of foreclosing consumers from purchasing the tied product from other sellers and thereby injuring competition in the tied product market.¹²⁵

Tying arrangements may be challenged as unreasonable restraints of trade under Section 1 of the Sherman Act¹²⁶ and, more specifically, as leases or sales on unreasonable conditions under Section 3 of the Clayton Act.¹²⁷ The Federal Trade Commission also has authority to challenge tying arrangements under Section 5 of the FTC Act.¹²⁸ With few exceptions, the substantive standard for illegality is the same under all three statutes.¹²⁹

Certain tying arrangements are subject to condemnation under a per se rule of illegality.¹³⁰ In general, a tying arrangement is per se unlawful if (1) it involves two separate products; (2) the sale of one product (the "tying product") is conditioned on the purchase of the other (the "tied product"); (3) the seller has sufficient market power in the tying product market to enable it to restrain trade in the tied product market; and (4) the arrangement affects a not insubstantial amount of interstate commerce in the tied product market.¹³¹ In addition, the Supreme Court has stated that "as a threshold matter there

¹²⁵ See, e.g., *Northern Pacific Railway v. United States*, 356 U.S. 1, 6 (1958) ("[Tying arrangements] deny competitors free access to the market for the tied product, not because the party imposing the tying requirements has a better product or lower product but because of his power or leverage in another market.")

¹²⁶ 15 U.S.C. § 1.

¹²⁷ 15 U.S.C. § 14 (prohibiting tying and exclusive dealing arrangements involving "goods, wares, merchandise, machinery, supplies, or other commodities . . . where the effect . . . may be to substantially lessen competition or tend to create a monopoly in any line of commerce").

Although tying arrangements involving software products have been held unlawful under Section 3, see, e.g., *Digidyne Corp. v. Data General Corp.*, 734 F.2d 1336 (9th Cir. 1984), the courts have not specifically addressed the question of whether software products come within the scope of Section 3. See, e.g., *In re Data General Corp. Antitrust Litigation*, 490 F. Supp. 1089, 1100 n. 10 (D.C. Cal. 1980) (declining to reach question); see also HOVENKAMP ET AL., *IP AND ANTITRUST: AN ANALYSIS OF ANTITRUST PRINCIPLES APPLIED TO INTELLECTUAL PROPERTY LAW* § 21.1, at 21-6 (2003) (stating that Section 3 by its terms does not apply to "simple [intellectual property] license[s]," but noting that "the presence of copyrights or trademarks has never been regarded as a defense to a tying arrangement").

¹²⁸ 15 U.S.C. § 45.

¹²⁹ See ABA SECTION OF ANTITRUST LAW, *ANTITRUST LAW DEVELOPMENTS* 177 n. 957 (4th ed. 1997) (citing cases); IX AREEDA, *supra* note 106, ¶¶ 1719b-c, at 254-57 (2002); HOVENKAMP ET AL., *supra* note 127, at ¶ 21.1, at 21-6 to 21-7.

¹³⁰ See *Jefferson Parish Hosp. Dist. No. 2 v. Hyde*, 466 U.S. 2, 9 (1984) ("It is far too late in the history of our antitrust jurisprudence to question the proposition that certain tying arrangements pose an unacceptable risk of stifling competition and therefore are unreasonable 'per se.'").

¹³¹ See *Fortner Enterprises, Inc. v. U.S. Steel Corp.*, 394 U.S. 495, 499 (1969) ("[Tying arrangements] are unreasonable in and of themselves whenever a party has sufficient economic power with respect to the

must be a substantial potential for impact on competition in order to justify per se condemnation,”¹³² and many lower courts, though not yet those in the D.C. Circuit,¹³³ have required a showing of “anticompetitive effects” in the tied product market.¹³⁴

Under the doctrine set forth by the Supreme Court in *Jefferson Parish*, two separate products exist “if there is ‘sufficient consumer demand so that it is efficient for a firm to provide them separately,’ even if the products are ‘functionally linked’ so that one is ‘useless without the other.’”¹³⁵ A separate line of cases involving challenges to the combination of previously separately marketed products through physical integration or product design, however, hold that such “technological tying” claims can succeed only if the plaintiff can show that “the challenged combination was carried out *solely* for the purpose of tying two separate products together ‘rather than to achieve some technologically beneficial result.’”¹³⁶

Tying arrangements that are not shown to be per se unlawful may still be found unreasonable, and therefore illegal, after a more extensive inquiry is conducted under the rule of reason.¹³⁷ The D.C. Circuit also held that the rule of reason, rather than the per se rule, was applicable to the Windows 98 tie in *Microsoft*, inasmuch as “the tying product is software whose major purpose is to serve as a platform for third-party applications and the tied product is complementary software functionality.”¹³⁸

To prevail under the rule of reason, the plaintiff must show that the challenged conduct is, on balance, unreasonable.¹³⁹ The plaintiff must first establish that the challenged conduct is *prima facie* unreasonable by showing a sufficient threat to

tying product to appreciably restrain free competition in the market for the tied product and a ‘not insubstantial’ amount of interstate commerce is affected.”); *Northern Pac. Ry. Co. v. U.S.*, 356 U.S. 1, 6 (1958) (same).

¹³² *Jefferson Parish*, 466 U.S. at 16.

¹³³ *See, e.g.*, *Microsoft*, 253 F.3d at 85 (listing only the four generally accepted elements of per se illegal tying).

¹³⁴ *See United Farmers Agents Ass’n, Inc. v. Farmers Ins. Exchange*, 89 F.3d 233, 236 (5th Cir. 1996); *Gonzalez v. St. Margaret’s House Housing Dev. Fund Corp.*, 880 F.2d 1514, 1516-17 (2nd Cir. 1989); *Wells Real Estate v. Greater Lowell Bd. of Realtors*, 850 F.2d 803, 815 (1st Cir.), *cert. denied*, 488 U.S. 955 (1988); *Power Test Petroleum Distributors v. Calcu Gas, Inc.*, 754 F.2d 91, 96 (2nd Cir. 1985); *Shop & Save Food Markets v. Pneumo Co.*, 683 F.2d 27, 30 (2nd Cir.), *cert. denied*, 459 U.S. 1038 (1982); *Yentsch v. Texaco, Inc.*, 630 F.2d 46, 56-57 (2nd Cir. 1980); *Driskill v. Dallas Cowboys Football Club*, 498 F.2d 321, 323 (5th Cir. 1974); *Coniglio v. Highwood Servs.*, 495 F.2d 1286, 1289 (2nd Cir.), *cert. denied*, 419 U.S. 1022 (1974); *but see Amey, Inc. v. Gulf Abstract & Title, Inc.*, 758 F.2d 1486, 1503 (11th Cir. 1985) (“A claim that a tying arrangement is illegal per se eliminates the requirement that the plaintiff show an actual anti-competitive effect.”).

¹³⁵ *Microsoft*, 1998 WL 614485, at *7 (quoting *Eastman Kodak Co. v. Image Technical Services, Inc.*, 504 U.S. 451, 462 (1992); *Jefferson Parish Hospital Dist. No. 2 v. Hyde*, 466 U.S. 2, 19 n.30 (1984)).

¹³⁶ *Id.* at *8 (emphasis in original) (citing *Foremost Pro Color, Inc. v. Eastman Kodak Co.*, 703 F.2d 534 (9th Cir. 1983); *Response of Carolina, Inc. v. Leasco Response, Inc.*, 537 F.2d 1307, 1330 (5th Cir. 1976); *Telex Corp. v. IBM*, 367 F.Supp. 258, 347 (N.D. Okla.1973), *rev’d on other grounds*, 510 F.2d 894 (10th Cir. 1975); *Innovation Data Processing, Inc. v. IBM*, 585 F.Supp. 1470, 1476 (D.N.J. 1984); *ILC Peripherals Leasing Corp. v. IBM*, 448 F.Supp. 228 (N.D. Cal.1978), *aff’d per curiam sub nom.*, *Memorex Corp. v. IBM*, 636 F.2d 1188 (9th Cir. 1980)).

¹³⁷ *See X AREEDA*, *supra* note 106, at ¶ 1728b, at 368-70.

¹³⁸ *United States v. Microsoft Corp.*, 253 F.3d 34, 95 (D.C. Cir. 2001).

¹³⁹ *See id.* at ¶ 1728c, at 370.

competition, which the defendant has an opportunity to rebut.¹⁴⁰ The defendant may also respond by presenting a prima facie case of justification, which the plaintiff has an opportunity to rebut.¹⁴¹

Tying arrangements may also be challenged as acts of monopolization or attempted monopolization under Section 2 of the Sherman Act.¹⁴² The standard for illegality of tying arrangements under Section 2 is the same as for other acts in furtherance of a monopoly or an attempt to monopolize: i.e., that the arrangement constitutes “exclusionary conduct ‘as distinguished from growth or development as a consequence of a superior product, business acumen, or historic accident.’”¹⁴³

D. Power in the Tying Product Market

While Microsoft has clear monopoly power in the market for PC operating systems,¹⁴⁴ it is a closer question as to whether Microsoft possesses sufficient market power in a relevant market for server operating systems to sustain a tying claim. Depending on the precise boundaries of the market, however, Microsoft’s share may already be sufficient to characterize as a dominant position.¹⁴⁵ Beyond this observation, a fuller inquiry into Microsoft’s power in the tying product market would fall outside the two analytical foci of this Article: first, the novel analytical issues presented by virtualization technology, and second, the analytical framework of the first principles approach to software product market definition.¹⁴⁶ For purposes of this Article, then, I will assume that Microsoft has durable market power in a relevant market for workgroup server operating systems.

E. The Separate Products Inquiry

The Supreme Court’s *Jefferson Parish*¹⁴⁷ “separate demand” test has generally been regarded as the controlling analysis for the separate products inquiry,¹⁴⁸ including

¹⁴⁰ See *id.*

¹⁴¹ See *id.*

¹⁴² 15 U.S.C. § 2.

¹⁴³ Microsoft, 253 F.3d 34, 58 (D.C. Cir. 2001) (quoting *United States v. Grinnell Corp.*, 384 U.S. 563, 571 (1966)).

¹⁴⁴ See *United States v. Microsoft Corp.*, 253 F.3d 34, 51-54 (D.C. Cir. 2001) (upholding district court’s finding of monopoly power in its entirety and noting finding that Microsoft possessed greater than 95% share of the x86 PC OS market).

¹⁴⁵ See Case COMP/C-3/37.792-Microsoft, Commission Decision of Mar. 24, 2004, para. 591, 2004 E.C. 299 (2004) <<http://ec.europa.eu/comm/competition/antitrust/cases/decisions/37792/en.pdf>> (noting IDC study finding Microsoft’s share of OS installations among under-\$25,000 workgroup servers had grown to 64.9% by 2002).

¹⁴⁶ Microsoft’s plans to include other features in the server version of Windows Vista are also being challenged by Symantec Corporation for misappropriation and infringement of a predecessor corporation’s intellectual property rights. Complaint, *Veritas Operating Corp. v. Microsoft Corp.*, Civ. No. 06-0703 (W.D. Wash. filed May 18, 2006). This litigation is also beyond the scope of this Article.

¹⁴⁷ *Jefferson Parish Hospital District No. 2 v. Hyde*, 466 U.S. 2 (1984).

¹⁴⁸ See, e.g., *Eastman Kodak Co. v. Image Technical Services*, 504 U.S. 451, 462 (1992) (citing *Jefferson Parish*, 466 U.S. 2, 21-22 (1984)) (“For service and parts to be considered two distinct products, there must

by Judge Jackson in his conclusions of law.¹⁴⁹ On appeal from Judge Jackson's conclusions of tying liability, however, the D.C. Circuit held that the per se approach to tying analysis was inapplicable to a tie between platform software and complementary software functionality, and that instead of using of the separate products inquiry as a "screen" to remove "false positives,"¹⁵⁰ the trial court should proceed directly to a balancing of anticompetitive effects against procompetitive justifications under the rule of reason.¹⁵¹ Deferring consideration of the D.C. Circuit's approach for now, I will first consider whether the alleged tying and tied products I have identified in Windows Server "Longhorn" constitute separate products under the *Jefferson Parish* test.

In *Jefferson Parish*,¹⁵² the plaintiff anesthesiologist challenged a requirement by the defendant hospital that its patients use the hospital's own anesthesiologists.¹⁵³ The hospital argued that the combination of general hospital services with anesthesiologists was not an illegal tying arrangement because it involved "a functionally integrated package of services," which constituted a single product.¹⁵⁴ Rejecting this argument, the Supreme Court held that "the answer to the question whether one or two products are involved turns not on the functional relation between them, but rather on the character of demand for the two items."¹⁵⁵ An alleged tying arrangement involves two separate products, the Court held, if it "link[s]" two distinct product markets that are "distinguishable in the eyes of buyers."¹⁵⁶ Specifically, there must be a "sufficient demand for the purchase of anesthesiological services separate from hospital services to identify a distinct product market in which it is efficient to offer anesthesiological services separately from hospital services."¹⁵⁷ Even though anesthesiological services would have no functional purpose without other hospital services, the Court noted that it had "often found arrangements involving functionally linked products at least one of which is useless without the other to be prohibited tying devices."¹⁵⁸

The Supreme Court subsequently applied the *Jefferson Parish* test in *Eastman Kodak Co. v. Image Technical Services*,¹⁵⁹ which concerned a challenge to defendant Kodak's policy of selling replacement parts for micrographic and copying machines only to buyers who use Kodak service or repair their own machines.¹⁶⁰ Defendant Kodak had moved for summary judgment, arguing, *inter alia*, that service and parts were a single

be sufficient consumer demand so that it is efficient for a firm to provide service separately from parts."); *Digidyne Corp. v. Data General Corp.*, 734 F.2d 1336, 1338-39 (9th Cir. 1984), *cert. denied*, 473 U.S. 908 (1985) (citing *Jefferson Parish*) (affirming separate products determination where separate demand was undisputed); II ABA SECTION OF ANTITRUST LAW, ANTITRUST LAW DEVELOPMENTS 178 (4th ed. 1997) (stating that *Eastman Kodak* and *Jefferson Parish* "supply the modern framework for antitrust analysis of tying").

¹⁴⁹ See *United States v. Microsoft Corp.*, 87 F. Supp. 2d 30, 50-51 (D.D.C. 2000)

¹⁵⁰ *United States v. Microsoft Corp.*, 253 F.3d 34, 87 (D.C. Cir. 2001).

¹⁵¹ See *id.* at 95.

¹⁵² *Jefferson Parish Hospital District No. 2 v. Hyde*, 466 U.S. 2 (1984).

¹⁵³ *Id.* at 4-7.

¹⁵⁴ *Id.* at 18-19.

¹⁵⁵ *Id.* at 19.

¹⁵⁶ *Id.* at 19-21.

¹⁵⁷ *Id.* at 21.

¹⁵⁸ *Id.* at 19 n. 30 (citations omitted).

¹⁵⁹ 504 U.S. 451 (1992).

¹⁶⁰ *Id.* at 458.

product “because there is no demand for parts separate from service.”¹⁶¹ The Court held that “[f]or service and parts to be considered two distinct products, there must be sufficient consumer demand so that it is efficient for a firm to provide service separately from parts.”¹⁶² Finding evidence in the summary judgment record that “service and parts have been sold separately in the past and still are sold separately to self-service equipment owners,”¹⁶³ the Court concluded that there was a triable issue of fact on the separate products issue. The Court specifically rejected Kodak’s argument that there was no demand for parts without service, reiterating that a tying arrangement between separate products may be found even when one product is useless without the other.¹⁶⁴

The description provided above¹⁶⁵ of the x86 virtualization industry, including its size,¹⁶⁶ market participants,¹⁶⁷ and competitive variables,¹⁶⁸ should suffice to show that there is *presently* sufficient demand for x86 virtualization software products separate from x86 OS software products to identify a distinct product market in which it is efficient to offer x86 virtualization software products separately from x86 OS software products. Moreover, industry analysts Klasell and Peck predict that even in the foreseeable future, when Microsoft and Linux will introduce more attractive x86 virtualization software products, there will remain a substantial market demand specifically for independently developed x86 virtualization software products *as long as such offerings are not precluded by OS/virtualization tie-ins*. They write:

“One thing that is striking about the x86 VM market versus the mainframe and UNIX market is that independent, non-OS vendors are driving the market, as Microsoft and the Linux community have yet to bring much to bear. Although the open-source community and Microsoft have started to focus more on this market, and they do have some advantages, the x86 world is more heterogeneous and we believe that many users will continue to want the multi-platform support that an independent vendor brings. This is especially true in management solutions, which should be a market that is less exposed to the O/S vendors’ abilities to bundle virtualization capabilities within the operating system.”¹⁶⁹

To demonstrate that the market for x86 virtualization software products is the relevant tied product market for purposes of a tying claim based on Windows Server “Longhorn,” it suffices to show that there is a well-defined segment of users interested in the end use of Windows Server “Longhorn” as a provisioning tool who are susceptible to (quality-adjusted) price discrimination without the possibility of arbitrage.¹⁷⁰ Given the

¹⁶¹ Id. at 462-63.

¹⁶² See id. (citing *Jefferson Parish*, 466 U.S. at 21-22).

¹⁶³ Id.

¹⁶⁴ Id. at 463. The apparent existence of “at least some” equipment owners who purchase parts without service, and service without parts, also factually contradicted Kodak’s argument. See *id.*

¹⁶⁵ See *supra* section III.A.

¹⁶⁶ See *supra* section III.A.1.

¹⁶⁷ See *supra* section III.A.2.

¹⁶⁸ See *supra* section III.A.3.

¹⁶⁹ Klasell & Peck, *supra* note 45, at 10.

¹⁷⁰ See Antitrust Analysis at 15-24.

value that users may place on the ability to run different operating systems simultaneously on the same computer, the ability to isolate the performance of one subsystem from another, and the ability to port the entire state of a system from one physical server to another,¹⁷¹ it is clear that the ability to construct a virtual hardware stack on which a guest operating system, and software stacks based on that platform, can be installed and run is a complete, meaningful, and well-defined consumer purpose.

To demonstrate the ability of a hypothetical x86 virtualization software monopolist to engage in quality-adjusted price discrimination against the segment of Windows Server “Longhorn” users who are interested in the end use of provisioning, it suffices to show that Microsoft has the legal and technical ability to remove the provisioning function from the Windows Server “Longhorn” product altogether.¹⁷² Recalling the “prototype removal program” developed by Princeton computer science professor Edward Felten to demonstrate Microsoft’s ability to offer a version of Windows 98 without Web browsing functionality,¹⁷³ it is easy to imagine a similar program demonstrating the ability to remove the provisioning function from the forthcoming version of Windows Server “Longhorn” with “integrated” virtualization.

Microsoft has already indicated that they plan to continue to allow users the option of not using virtualization. Assuming that Microsoft provides a method in Windows Server “Longhorn” by which users can exercise this option, it should be a simple matter to hard-code this option into the software, thereby removing all user access to the product’s provisioning function.

Even if Microsoft designs Windows Server “Longhorn” so that a user cannot completely turn off the provisioning function, a computer scientist could easily modify the “Longhorn” source code to do so. The partitioning function in Windows hypervisor would simply need to be hard-coded to allow only one configuration of the virtualization architecture: the parent partition and a single child partition running the Windows Server OS as a guest. Given that the Windows hypervisor will be designed as a “microkernelized” component with a minimal code base, the modification should be easy to perform.

Either of these approaches may be seen as a crude digital rights management mechanism that, when coupled with corresponding license use restrictions,¹⁷⁴ could effectively remove the tied provisioning tool product from the “integrated” release of Windows Server “Longhorn.” Microsoft’s continuing ability to offer such a provisioning-free version of Windows Server “Longhorn” demonstrates that there will remain a relevant product market for x86 virtualization software products in which Microsoft competes, even after Microsoft’s “integration” of its virtualization software into the OS has been consummated.

¹⁷¹ See *supra* section II.B,

¹⁷² Cf. Decoding *Microsoft* at 89 (performing a similar analysis with respect to the end use of Web browsing in Windows 98).

¹⁷³ See *United States v. Microsoft Corp.*, 84 F. Supp. 2d 9, 53-54, at ¶¶ 177, 178, 185.

¹⁷⁴ Microsoft’s ability to draft corresponding use restrictions is assured, *inter alia*, by the legal ability of a vendor to restrict a licensee’s right to run copyrighted software (and thereby create “copies” and “adaptations” of the software in RAM within the meaning of § 117) to the purpose “for which [the accompanying software product] was both sold and purchased.” See *Antitrust Analysis* at 71-72.

F. Existence of a Tying Condition

To prove per se tying liability, the plaintiff must show that the defendant did not merely enter into an agreement to sell two products together, but improperly “conditioned” the sale of the tying product on the purchase of the tied product.¹⁷⁵ Such conditioning may be found when there is an express contractual term,¹⁷⁶ an announcement,¹⁷⁷ or other conduct by the seller (also known as an “understood condition”)¹⁷⁸ that leads reasonable buyers to understand that they cannot get the tying product unless they also take the tied product.

When the defendant does not expressly state a tying condition but merely offers two products in a bundle, at most it is only possible to infer an understood tying condition. For an understood condition to be inferred, according to the Areeda treatise, the plaintiff must prove either that the defendant rebuffed requests for separate provision, or that circumstances indicate that customers reasonably believed that such requests would be futile or excessively burdensome.¹⁷⁹ For example, where a defendant packages products together in a box, “no reasonable retail consumer believes he can cause the manufacturer to take seriously a request for altering the contents of the box.”¹⁸⁰ Thus, the practice of packaging products in a box for consumer sale is generally regarded as a tie of the products.¹⁸¹ Similarly, according to Areeda, a form contract covering two products should be presumed to be a tie if the plaintiff also presents evidence that “buyers requested separate provision and were rebuffed or that many buyers actually understood that they had to buy [the tied product] in order to get the defendant’s [tying product].”¹⁸²

In a tie of two software products, the legal rights and technological capabilities that comprise the tying product are available only in combination with the legal rights and technological capabilities that comprise the tied product. If Microsoft carries out its plan to “integrate” its virtualization software into Windows Server “Longhorn,” Microsoft’s form license agreements for “Longhorn” will ensure that consumers can obtain the legal rights associated with Microsoft’s server operating system software product in Windows Server only in combination with the legal rights associated with Microsoft’s VMM software product. The software combination will then effectively complete the tie by preventing consumers from obtaining technological access to Microsoft’s operating system software product in Windows Server without also obtaining technological access to Microsoft’s VMM software product.

Given that Windows Server “Longhorn” will be licensed under a form contract, it is likely that no reasonable consumer will believe he can cause Microsoft to take

¹⁷⁵ See X AREEDA, *supra* note 106, ¶ 1752b, at 280-81.

¹⁷⁶ See *id.* at ¶ 1753, at 292-300.

¹⁷⁷ See *id.* at ¶ 1754b, at 301-03.

¹⁷⁸ See *id.* at ¶ 1754c, at 303-05.

¹⁷⁹ See *id.* at ¶ 1756e, at 325.

¹⁸⁰ See *id.* at ¶ 1756e, at 326.

¹⁸¹ See *id.* at ¶ 1756e, at 325-26 (citing *Data General Corp. Antitrust Litigation*, 490 F. Supp. 1089, 1010-11 (N.D. Cal. 1980)).

¹⁸² See *id.* at ¶ 1756d1, at 322-23. Areeda suggests that the proposed presumption should be rebuttable by proof that buyers frequently requested and received the tying product separately, or generally understood that contract revisions were readily available.

seriously a request to redesign “Longhorn” to remove its provisioning tool product. Accordingly, Microsoft’s plan to market the forthcoming “integrated” release of Windows Server “Longhorn” appears to be an understood condition tying its provisioning tool product to its server OS platform product.

It should be noted here that Microsoft’s plan to combine its server OS platform and provisioning tool products would not be problematic if the company also continued to offer the option of obtaining the products separately. Microsoft’s marketing practice, however, is to limit the duration for offering and supporting each version of its software products. For example, Microsoft discontinued all support for Windows 95, the last Windows PC OS without “integrated” Web browsing functionality, in December 2001.¹⁸³ Thus, Microsoft’s plan to “integrate” its virtualization software into its Windows Server OS beginning in 2009-10 will have the effective of conditioning the sale of Microsoft’s server OS product on the purchase of Microsoft’s provisioning tool product.

In characterizing the “integration” of virtualization software into “Longhorn” as an understood tying condition, I explicitly reject two other possible characterizations of Microsoft’s conduct: first, the view that Microsoft is only exercising rights within the scope of its copyright in the software it has developed, and second, the idea that Microsoft is engaging only in the kind of “technological tie” that warrants judicial deference to product design.

1. The Limited Scope of Microsoft’s Copyright

The ability of a consumer to use a software product depends not only on the development of usable software, but on the legal rights that constitute the software product itself, as defined by the terms of the accompanying software license. Most of these license terms track the Copyright Act’s default allocation of rights, either by granting to the user certain nonexclusive rights within the scope of the copyright owner’s exclusive rights, or by leaving the default allocation unchanged.¹⁸⁴ Violations of such license terms may constitute both a contractual breach and a copyright infringement.¹⁸⁵ Other license terms, however, purport to enlarge the scope of the copyright owner’s rights under the Copyright Act. These terms are enforceable, if at all, under state contract law, and may be subject to preemption by federal intellectual property law¹⁸⁶ and/or susceptible to challenge under the antitrust laws to the extent that they exceed the

¹⁸³ See Microsoft Support Lifecycle (visited May 15, 2006) <<http://support.microsoft.com/lifecycle/?C2=9980>>.

¹⁸⁴ See Maureen A. O’Rourke, *Drawing the Boundary Between Copyright and Contract: Copyright Preemption of Software License Terms*, 45 DUKE L.J. 479, 490 (1995) (noting that “with few exceptions, . . . [software license] terms track those of the Copyright Act”).

¹⁸⁵ See, e.g., *S.O.S., Inc. v. Payday, Inc.*, 886 F.2d 1081 (9th Cir. 1989) (“A licensee infringes the owner’s copyright if its use exceeds the scope of its license.”).

¹⁸⁶ See 17 U.S.C. § 301(a) (preempting grants of exclusive rights “within the general scope of copyright” under “the common law or statutes of any State”); see also *Vault Corp. v. Quaid Software Ltd.*, 847 F.2d 255, 269-70 (5th Cir. 1988), *citing* *Sears, Roebuck & Co. v. Stiffel Co.*, 376 U.S. 225, 229 (1964) (holding prohibition against decompilation in a standard form software licensing agreement unenforceable despite contrary state statute because the state statute “touches upon an area” of federal copyright law).

legitimate exercise of a copyright owner's exclusionary rights.¹⁸⁷ In addition, software license agreements usually contain terms other than those relating to rights in the software, such as warranties, limitations of liability, and choice of law.¹⁸⁸ Such terms are regarded as independent contractual covenants; they do not bear on the scope of the license and are enforceable only under contract law and not copyright law.¹⁸⁹

It is settled law that the Copyright Act extends protection at least to the software code¹⁹⁰ that accompanies a software product. Section 102(a) expressly provides copyright protection to "original works of authorship fixed in any tangible medium" including literary, musical, dramatic, choreographic, pictorial, graphic, sculptural, audiovisual, recorded audio, and architectural works.¹⁹¹ A work is fixed in a tangible medium when its embodiment in a copy "is sufficiently permanent or stable to permit it to be perceived, reproduced, or otherwise communicated for a period of more than transitory duration."¹⁹² Literary works are defined as "works, other than audiovisual works, expressed in words, numbers, or other verbal or numerical symbols or indicia."¹⁹³

Software code is expressed in "verbal or numerical symbols or indicia," and is therefore considered a "literary work" within the meaning of the Copyright Act.¹⁹⁴ The software code that accompanies a software product is fixed in a tangible medium insofar as it is distributed in a form that at least permits its reproduction in the random access memory ("RAM") of the user's computer.¹⁹⁵ Copyright law thus prohibits at least the

¹⁸⁷ See, e.g., *United States v. Loew's, Inc.*, 371 U.S. 38, 47-48 (1962) (condemning the block booking of separately copyrighted motion pictures for television exhibition as a tying arrangement under § 1 of the Sherman Act); *In re Independent Service Organizations Antitrust Litigation*, 203 F.3d 1322, 1329 (Fed. Cir. 2000) (rejecting unilateral refusal to license claim for lack of evidence that copyrights were obtained by unlawful means or were used to gain monopoly power beyond the statutory copyright granted by Congress); *Data General Corp. v. Grumman Sys. Support Corp.*, 36 F.3d 1147, 1187 (1st Cir. 1994) ("[W]hile exclusionary conduct can include a monopolist's unilateral refusal to license a copyright, an author's desire to exclude others from use of its copyrighted work is a presumptively valid business justification for any immediate harm to consumers"); *Montgomery County Ass'n of Realtors, Inc. v. Realty Photo Master Corp.*, 878 F. Supp. 804, 817 & n. 23 (D. Md. 1995), *aff'd*, 91 F.3d 132 (4th Cir. 1996) (holding that a copyright owner's refusal to license the copying of its database was authorized by § 106 of the Copyright Act, and was therefore a "legitimate business purpose" negating concerted refusal to deal claim); see generally HOVENKAMP ET AL., *IP AND ANTITRUST: AN ANALYSIS OF ANTITRUST PRINCIPLES APPLIED TO INTELLECTUAL PROPERTY LAW* § 13.3, at 13-10 (2003) (stating that antitrust law generally imposes a duty to license intellectual property only in cases where "an intellectual property owner has sought to expand the scope of its right beyond what the intellectual property laws grant it").

¹⁸⁸ See O'Rourke, *supra* note 184, at 490 n. 39.

¹⁸⁹ See generally *Sun Microsystems, Inc. v. Microsoft Corp.*, 188 F.3d 1115, 1121-23 (9th Cir. 1999).

¹⁹⁰ See *supra* note 104.

¹⁹¹ 17 U.S.C. § 102(a).

¹⁹² 17 U.S.C. § 101.

¹⁹³ 17 U.S.C. § 101.

¹⁹⁴ See *Atari Games Corp. v. Nintendo of America, Inc.*, 975 F.2d 832, 838-39 (Fed. Cir. 1992) (finding that computer programs fall within the terms of the Copyright Act, but noting that § 102(b) limits protection to "the expression adopted by the programmer" and excludes "the actual processes or methods embodied in the program").

¹⁹⁵ See *MAI Systems Corp. v. Peak Computer, Inc.*, 991 F.2d 511, 517-18 (9th Cir. 1993) (finding that copies of software in random access memory can be "perceived, reproduced, or otherwise communicated for a period of more than transitory duration" within the meaning of § 101).

unauthorized literal copying of software code associated with simple forms of software piracy, such as the duplication of software CD-ROMs or diskettes.

The Copyright Act is less clear, however, regarding the eligibility of nonliteral elements of software for copyright protection. Section 102(b) of the Copyright Act expressly denies copyright protection to ideas, procedures, processes, systems, methods of operation, concepts, principles and discoveries.¹⁹⁶ In general, this provision serves to codify the longstanding rule that “copyright does not protect ideas, but only expression of ideas,”¹⁹⁷ also known as the “idea/expression dichotomy.”¹⁹⁸ The purpose of the idea/expression dichotomy is to balance competing constitutional values by placing the original expressions of authors within the scope of copyrightable subject matter while preserving the public’s First Amendment interest in the free communication of ideas.¹⁹⁹

Although the idea/expression dichotomy is well-settled as a matter of principle,²⁰⁰ it has been difficult for courts to apply in practice, particularly in cases involving functional works.²⁰¹ In such cases, the aim is to preserve “the balance between competition and protection reflected in the patent and copyright laws.”²⁰² With respect to software, the idea/expression dichotomy stands for the principle that copyright in a computer program should not prevent another developer from writing different code that performs the same functions as the copyrighted program when executed. Thus, section 102(b) serves “to make clear that the expression adopted by the programmer is the copyrightable element in a computer program, and that the actual processes or methods embodied in the program are not within the scope of copyright law.”²⁰³

¹⁹⁶ 17 U.S.C. § 102(b).

¹⁹⁷ *Whelan Associates v. Jaslow Dental Laboratory, Inc.*, 797 F.2d 1222 (3rd Cir. 1986) (quoting *Baker v. Selden*, 101 U.S. (11 Otto) 99 (1879)).

¹⁹⁸ See H.R. Rep. No. 1476, 94th Cong., 2d Sess. 57 (1976), *reprinted in* 1976 U.S.C.C.A.N. 5659, 5670 (“Section 102(b) in no way enlarges or contracts the scope of copyright protection under the present law. Its purpose is to restate . . . that the basic dichotomy between expression and idea remains unchanged.”)

¹⁹⁹ See *Harper & Row Publishers, Inc. v. Nation Enterprises*, 471 U.S. 539, 556-57 (1985) (citation omitted) (noting that the dichotomy “strike[s] a definitional balance between the First Amendment and the Copyright Act by permitting free communication of facts while still protecting an author’s expression.”); *Religious Tech. Center v. Netcom On-Line Communications Servs., Inc.*, 907 F. Supp. 1361, 1377 (N.D. Cal. 1995) (noting that the dichotomy “balance[s] the important First Amendment rights with the constitutional authority for ‘promoting the progress of the science and useful arts.’”).

²⁰⁰ See *Feist Publications, Inc. v. Rural Telephone Service Co., Inc.*, 499 U.S. 340, 350 (stating that the dichotomy “applies to all works of authorship”); *Baker v. Selden*, 101 U.S. 99 (1879).

²⁰¹ See, e.g., *John Shepard Wiley, Jr., Copyright at the School of Patent*, 58 U. CHI. L. REV. 119, 121-29 (1991) (arguing that the dichotomy has become an “incoherent” doctrine that “announces results but does not determine or justify them”).

²⁰² *Apple Computer*, 714 F.2d 1240, 1253 (3rd Cir. 1983) (quoting *Herbert Rosenthal Jewelry Corp. v. Kalpakian*, 446 F.2d 738, 742 (9th Cir. 1971)).

²⁰³ H.R. Rep. No. 1476, 94th Cong., 2d Sess. 57 (1976), *reprinted in* 1976 U.S.C.C.A.N. 5659, 5670; S. Rep. No. 473, 94th Cong., 2d Sess. 54 (1976). Several of the federal circuit courts have referred to this legislative history in discerning the purpose of § 102(b). See, e.g., *Gates Rubber Co. v. Bando Chemical Industries, Ltd.*, 9 F.3d 823, 836-37 (10th Cir. 1993); *Atari Games Corp. v. Nintendo of America Inc.*, 975 F.2d 832, 838-39 (Fed. Cir. 1992); *Computer Associates Intern., Inc. v. Altai, Inc.*, 982 F.2d 693, 703 (2nd Cir. 1992); *M. Kramer Mfg. Co., Inc. v. Andrews*, 783 F.2d 421, 434-35 (4th Cir. 1986); *Apple Computer, Inc. v. Franklin Computer Corp.*, 714 F.2d 1240, 1252-53 (3rd Cir. 1983).

In effect, the section 102(b) inquiry requires courts to attach legal significance to the subtle distinction between a programmer’s expression in the “verbal or numerical symbols or indicia” of a computer program and the “processes or methods embodied in the program.” As a result, courts have historically taken widely divergent approaches toward copyright infringement cases in which the defendant has not literally copied the plaintiff’s code, but has duplicated other elements of the plaintiff’s software, such as its structure, sequence, organization, hardware interfaces, programming interfaces, user interfaces, and “look and feel.”

A 1992 decision by the Second Circuit in *Computer Associates v. Altai* is now recognized as providing the leading formulation of the idea-expression dichotomy in copyrighted software.²⁰⁴ The Second Circuit set forth a three-step analytical procedure that should be followed in determining whether protected non-literal elements of a computer program have been copied:

1. Abstraction: “[D]issect the allegedly copied program’s structure and isolate each level of abstraction contained within it. This process begins with the code and ends with an articulation of the program’s ultimate function.”²⁰⁵
2. Filtration: “[E]xamin[e] the structural components at each level of abstraction to determine whether their particular inclusion at that level was ‘idea’ or was dictated by considerations of efficiency, so as to be necessarily incidental to that idea; required by factors external to the program itself; or taken from the public domain and hence is nonprotectable expression.”²⁰⁶ The filtering of “structural components . . . required by factors external to the program” specifically excludes from copyright protection those elements where the “freedom of design choice”²⁰⁷ available to the developer of the allegedly infringed program²⁰⁸ was “circumscribed by extrinsic considerations such as (1) the mechanical specifications of the computer on which a particular program is intended to run; (2) compatibility requirements of other programs with which a program is designed to operate in conjunction; (3) computer manufacturers’ design standards; (4) demands of the

²⁰⁴ See, e.g., Peter S. Menell, *Envisioning Copyright Law’s Digital Future*, 46 N.Y.L. SCH. L. REV. 63 (2003) (stating that the Second Circuit’s approach “has been universally adopted by the courts since 1992”).

²⁰⁵ Id. at 707.

²⁰⁶ Id. (citing 3 MELVIN B. NIMMER & DAVID NIMMER, NIMMER ON COPYRIGHT § 13.03[F]; Mark T. Kretschmer, Note, *Copyright Protection for Software Architecture: Just Say No!*, 1988 COLUM. BUS. L. REV. 823, 844-45 (1988)).

²⁰⁷ Id. at 709.

²⁰⁸ See *Mitel, Inc. v. Iqtel, Inc.*, 124 F.3d 1366, 1375 (10th Cir. 1997) (stating that analytical focus should be on “the external factors that dictated [the plaintiff’s] selection” of non-literal program elements); see also *Computer Associates*, 982 F.2d at 714 (stating that preferred approach is “filtering out the unprotected aspects of an allegedly infringed program and then comparing the end product to the structure of the suspect program”).

industry being serviced; and (5) widely accepted programming practices within the computer industry.”²⁰⁹

3. Comparison: Determine “whether the defendant copied any aspect of [the remaining] protected expression,” and assess “the copied portion’s relative importance with respect to the plaintiff’s overall program.”²¹⁰

Even though the present inquiry is not in the posture of a copyright infringement claim, the *Computer Associates* framework can inform the inquiry as to whether the exclusionary effect of Microsoft’s conduct on third-party software development falls within the scope of the exclusionary rights accorded to Microsoft by virtue of its copyright in the Windows Server “Longhorn” software.

As I will argue more fully in Section IV.G, Microsoft’s “integration” of its virtualization software into Windows Server will largely foreclose third-party vendors from offering a rival virtualization software product that supports Windows Server VMs and utilizes Intel’s and AMD’s virtualization support (i.e., by running the VMM at privilege level –1, thereby allowing guest software to run at its native privilege level). The *Computer Associates* decision, however, makes clear that copyright law alone could not confer upon Microsoft a right to exclude third-party VMM software vendors from incorporating design elements that are “circumscribed by . . . compatibility requirements of other programs with which a program is designed to operate in conjunction . . . [or] computer manufacturers’ design standards.” Because the exclusionary effect of Microsoft’s “integration” of virtualization software into Windows Server will extend to the idea of interoperation with Windows Server and the idea of incorporating Intel’s and AMD’s design standards for virtualization support, Microsoft’s conduct there is no warrant in copyright law for such a restraint. Accordingly, Microsoft’s plan to market a version of Windows Server “Longhorn” with “integrated” VMM software exceeds the legitimate exercise of a copyright owner’s exclusionary rights, and remains susceptible to challenge under the antitrust laws notwithstanding Microsoft’s ownership of copyright in the resulting work.

2. The Applicability of “Technological Tying” Doctrines

A tying condition that has been implemented at least in part through technological means may be referred to as a “technological tie.” Bundling the tying and tied products together physically (e.g., by bolting them together or placing them into the same box) would constitute a technological tie in this sense.²¹¹ The term “technological tie” is sometimes used more narrowly,²¹² however, to describe a situation where “a defendant has market power in a primary product that works better with his complementary product

²⁰⁹ 982 F.2d at 709-10 (quoting 3 NIMMER, *supra* note 206, at § 13.03[F][3], at 13-66 to 13-71).

²¹⁰ *Id.* at 710 (citing 3 NIMMER, *supra* note 206, at § 13.03[F][5]; *Data East USA, Inc. v. Epyx, Inc.*, 862 F.2d 204, 208 (9th Cir. 1988)).

²¹¹ *See* HOVENKAMP ET AL., *supra* note 127, at 21.5b2, at 21-104; *see also* X AREEDA, *supra* note 129, ¶ 1757a, at 335 n.2 (noting that the term is sometimes used “to include the combination of two items in one device”); *United States v. Microsoft Corp.*, 147 F.3d at 9

²¹² *See* X AREEDA, *supra* note 129, ¶ 1757a, at 335 & n.2 (distinguishing between the two usages of the term); *but see* HOVENKAMP ET AL., *supra* note 127, at 21.5b2, at 21-104 (not distinguishing between the two usages).

than with rival versions” and the “practical effect” of this “product interdependence” is that rivals are foreclosed in the complementary market *even if the defendant sells the two products separately*.²¹³ Because this product interdependence may be the result of procompetitive innovation, some courts have been reluctant to find a tying condition in such cases.²¹⁴

In light of first principles, the “integration” of Microsoft’s provisioning tool product into Windows Server will be a technological tie in the first sense but not the second. Microsoft plans to implement the understood tying condition in part through the technological method of combining its server OS software and virtualization software in the same code base.

Microsoft’s Windows Server OS software product, however, cannot be said to “work better” with Microsoft’s VMM software product than with third-party versions. Microsoft’s inclusion of Windows “enlightenments” and paravirtualization in Windows Server may reduce the virtualization overhead of Microsoft’s VMM software relative to what it would have been in their absence, but it cannot be said that these modifications will necessarily produce a result that is more efficient, let alone “better” (in terms of the many other competitive variables that influence the market for VMM software products²¹⁵) than that which would be achieved by running Windows Server VM on a third party VMM.

Moreover, as I will argue in Section IV.G, the foreclosure from the challenged tie does not arise from the fact that the Windows Server OS will be optimized to run on the accompanying VMM software, but from the fact that Microsoft will refuse to sell its OS and VMM products separately. By requiring Windows Server users to install Microsoft software that runs on Intel and AMD microprocessors at privilege level –1, Microsoft will largely foreclose third-party vendors from offering a rival virtualization software product that supports Windows Server VMs without paravirtualization and utilizes Intel’s and AMD’s virtualization support. Thus, a tying claim based on the “integration” of virtualization software into “Longhorn” does not represent a challenge to the kind of innovation that results in product interdependence, but to the practice of tying itself.

G. Foreclosure and Anticompetitive Effect in the Tied Product Market

For a tying arrangement to be per se illegal, it must affect “a ‘not insubstantial’ amount of interstate commerce” in the tied product.²¹⁶ While this requirement calls for a showing as to the dollar-volume of affected business, the dollar-volume need only be

²¹³ See X AREEDA, *supra* note 129, ¶ 1757a, at 335-36.

²¹⁴ See, e.g., *Foremost Pro Color, Inc. v. Eastman Kodak*, 703 F.2d 534, 542 (9th Cir. 1983), *cert. denied*, 465 U.S. 1038 (1984) (“[W]e decline to place . . . technological ties in the category of economic restrictions deemed per se unlawful by *Northern Pacific* and its progeny.”); see also *Microsoft*, 87 F. Supp. 2d at 51 (citations omitted) (noting that other courts “resist a strict application of the ‘separate products’ test to similar questions of ‘technological tying’”); X AREEDA, *supra* note 129, ¶ 1757, at 335-41 (arguing that most product interdependence is procompetitive and should raise a presumption of no tying condition).

²¹⁵ See *supra* section III.A.3.

²¹⁶ *Jefferson Parish*, 466 U.S. at 16.

“substantial enough . . . so as not to be merely *de minimis*.”²¹⁷ In this context, the Supreme Court has held that as little as \$60,800 was not insubstantial.²¹⁸

Some doctrinal approaches to tying call for a closer examination of the actual and/or potential anticompetitive effects of the challenged conduct. For example, even though the per se analysis of a tying claim does not permit a detailed measurement of market foreclosure,²¹⁹ the Supreme Court has stated that “as a threshold matter there must be a substantial potential for impact on competition in order to justify per se condemnation.”²²⁰ For per se liability to be appropriate, then, the challenged conduct must be shown to be “of the type that could cause the kind of foreclosure that anti-tying rules seek to prevent.”²²¹ Accordingly, lower courts that include “anticompetitive effects” as an element of per se tying liability have not undertaken detailed measurements of market foreclosure, but instead have used the requirement to excuse certain types of tying arrangements that do not present the potential for a substantial anticompetitive effect in the tied product market.²²²

Microsoft’s inclusion of a VMM software product in Windows Server “Longhorn” will not completely preclude an OEM or consumer from installing and using a rival VMM software product, but it will present some significant obstacles.

First, to the extent that computer users tend to become most familiar with the software products that are available to them by default, Windows Server users will be unlikely to switch from Microsoft’s preinstalled virtualization software product to a rival product. In *Microsoft*, Judge Jackson found that “once [consumers] have acquired, found, and used one browser product, most are reluctant — and indeed have little reason — to expend the effort to switch to another.”²²³ Switching costs are likely to be even higher for virtualization software products than for Web browser software products, because of the lack of common standards for hardware partitions for running VMs and file formats for stored VMs. Some virtualization software products may even require wiping the hard drive in preparation for a “clean install,” so that all hardware resources can be made available for allocation to the virtual address space.

Second, Windows Server users who wish to use a third-party VMM software product may find it confusing to be presented with two or more different virtualization software products. Given the powers that are delegated to virtualization software products, any errors resulting from such confusion may have serious consequences.

²¹⁷ *Fortner Enterprises, Inc. v. U.S. Steel Corp.*, 394 U.S. at 495, 501 (1969).

²¹⁸ *United States v. Loew’s, Inc.*, 371 U.S. 38, 49 (1962).

²¹⁹ *See, e.g., Ohio-Sealy Mattress Mfg. Co. v. Sealy*, 585 F.2d 821 (7th Cir. 1978), *cert. denied*, 440 U.S. 930 (1979) (“Because the Supreme Court has repeatedly held that tying, if it fits within the *Northern Pacific* standard, is a *per se* violation, we are not free to inquire whether such tying in any given case injures market competition. . . . [H]owever . . . if a given tying arrangement has no potential to foreclose access to the tied product market, it does not exemplify the vice that led the [Supreme] Court to declare tying a per se offense.”).

²²⁰ *Jefferson Parish*, 466 U.S. at 16 (noting that “as a threshold matter there must be a substantial potential for impact on competition in order to justify per se condemnation”).

²²¹ IX AREEDA, *supra* note 129, ¶ 1722, at 286.

²²² *See* IX AREEDA, *supra* note 129, ¶ 1722c, at 291.

²²³ *United States v. Microsoft Corp.*, 84 F. Supp. 2d 9, 46-47, ¶ 144 (D.D.C. 1999).

Third, even though Microsoft has expressly authorized users to maintain inactive copies of Windows Server VMs for the purpose of supporting automated backup-and-recovery procedures,²²⁴ Windows Server's Product Activation mechanism will restrict third-party VMM users to only one installed copy of the Windows Server software, thus preventing them from creating inactive backup copies.

Fourth, the inclusion of virtualization support with Windows Server is likely to eliminate demand for type-2 Windows Server-hosted VMM software products altogether. Such software would simply impose an extra level of overhead while providing little if any additional functionality to the user.

Fifth, Windows Server users who wish to use a third-party type-1 VMM software product will have to use a system on which there are two or more different virtualization software products installed, each of which enjoys the ability to run at privilege level -1. Unless each of these products also allows the user to disable it completely, the resource control requirement for virtualization will be violated.

Finally, if a Windows Server user wishes to use a third-party VMM software product that runs at privilege level -1, and wishes to run Windows Server as the guest OS on one or more VMs, she will have to cope with the fact that Windows Server uses privilege levels -1, 0 and 3. To accommodate Windows Server in privilege levels 0, 1, 2 and 3, her VMM will have to perform ring deprivileging or some other paravirtualization technique, resulting in increased virtualization overhead for any Windows Server VMs and defeating the purpose of Intel's and AMD's virtualization support.

Taken together, these obstacles to the adoption of third-party VMM software products by Windows Server users are likely to result in significant foreclosure in the market for VMM software products. The last of these obstacles is particularly serious, as it adversely impacts the performance of all rival VMM software products relative to Microsoft's VMM software product, thereby preventing competition on the merits.

Microsoft's integration of a hardware provisioning tool with an OS platform software product also violates a principle of consumer choice that held true in the days before virtualization: consumers should be free to make an independent hardware provisioning decision for each OS platform they wish to use.²²⁵

²²⁴ See *supra* text accompanying note 71.

While Microsoft's express authorization makes resort to statutory rights unnecessary, users of third-party VMM software might find support in § 117(c) of the Copyright Act for the right to make copies of the Windows Server OS on archival VMs "for purposes only of maintenance or repair," provided that such copies are "made solely by virtue of the activation of a machine that lawfully contains an authorized copy of the computer program." 17 U.S.C. § 117(c). It is easy to envision a design for VMM software that automatically creates backups of VMs upon activation. Whether such a design would satisfy the statute is an interesting open question.

²²⁵ VMware's President Diane Greene has made a similar observation:

"[B]undling virtualization in the operating system puts the ability to exploit freedom of choice around the software stack in jeopardy. Operating system vendors have a financial incentive to move customers to their latest operating system, not to provide software choice to the consumer."

LinuxWorld: Virtualization Bake-In Off and Running, EWEEK.COM (visited May 20, 2006) <<http://www.eweek.com/article2/0,1895,1945205,00.asp>>.

Beyond these qualitative observations, it is difficult to forecast the anticompetitive effect that Microsoft’s virtualization plans will have on the market for VMM software products. As technology journalist Simon Bisson notes, however, the stakes are high:

“[T]he VMM (the hypervisor) installed has an incredible amount of power — it controls what runs and how it runs. Install yours first, and the machine is yours So what would the hypervisor wars mean? Firstly an end to the open systems model that’s been at the heart of enterprise IT for the last 25 years. If Microsoft and VMware fell out, VMware could reduce the priority of Windows partitions. Other hypervisors might have licensing conditions that make it impossible to run non-free OSes as clients. You could end up with a situation where each OS installation would attempt to insinuate its own hypervisor onto the system partition.”²²⁶

Microsoft has now spoken: the hypervisor wars are here.

H. Possible Justifications

[I have some preliminary thoughts toward a rule of reason analysis of the tying claim, which I will raise during my discussions with Greg Vetter and our discussions with you.]

V. CONCLUSIONS

Over the past two decades, successive generations of Microsoft’s dominant operating system software product (“Windows”) have incorporated an ever-expanding array of features and functionalities. Antitrust claims against Microsoft for tying other software products to Windows or leveraging Microsoft’s OS monopoly into other software product markets have challenged this practice as an “embrace, extend and extinguish” strategy,²²⁷ wherein the nearly ubiquitous distribution of a new functionality with Windows effectively forecloses competitors from selling that functionality to Windows users in a standalone software product. The “browser wars” between Microsoft and Netscape Corporation that led to the government’s Sherman Act prosecution of Microsoft between 1998 and 2003 are probably the best-known example of this pattern.

Microsoft’s “embrace, extend and extinguish” strategy may have thus far eluded the antitrust agencies, but it has remained fertile ground for scholarship. It is not difficult to imagine undertaking a full-blown tying analysis in response to the addition of every

²²⁶ Simon Bisson, *The Hypervisor Wars*, A NEW IT WORLD, Jan. 11, 2006, available at <http://itphasechange.blogspot.com/2006/01/hypervisor-wars.html>.

²²⁷ See, e.g., *Comes v. Microsoft Corp.*, 403 F.Supp.2d 897, 901 (S.D. Iowa 2005) (quoting complaint); *United States v. Microsoft Corp.*, (D.D.C. 1998), 11/12/98 A.M. Tr., 1998 WL 788010, at *31 (reporting cross-examination of McGeedy on allegation that “Mr. Maritz, or someone from Microsoft, used the phrase ‘embrace, extend and extinguish’” in describing Microsoft’s Web strategy).

major new functionality to Windows. Some such ties, however, are of greater antitrust concern than others.²²⁸ In this Article, I have explained why Microsoft’s plan to “integrate” virtualization software into Windows Server “Longhorn” warrants particular concern not only as the tying of an important software product to Windows, but as an effort to blunt a significant threat to Microsoft’s OS monopoly.

²²⁸ See, e.g., Chin, *Decoding Microsoft*, at 115-28 (describing foreclosure and anticompetitive effects from the OS/browser tie in Windows 98); Schneider, *supra* note 4, at 20 (discussing foreclosure from the OS/security software tie).