



SCHOOL OF LAW
TEXAS A&M UNIVERSITY

Texas Wesleyan Law Review

Volume 18 | Issue 4

Article 6

7-1-2012

Enterprise Software Licensing: New Options - New Obligations

Julie Machal-Fulks

Christopher Barnett

Follow this and additional works at: <https://scholarship.law.tamu.edu/txwes-lr>

Recommended Citation

Julie Machal-Fulks & Christopher Barnett, *Enterprise Software Licensing: New Options - New Obligations*, 18 Tex. Wesleyan L. Rev. 753 (2012).

Available at: <https://doi.org/10.37419/TWLR.V18.I4.5>

This Symposium is brought to you for free and open access by Texas A&M Law Scholarship. It has been accepted for inclusion in Texas Wesleyan Law Review by an authorized editor of Texas A&M Law Scholarship. For more information, please contact aretteen@law.tamu.edu.

ENTERPRISE SOFTWARE LICENSING: NEW OPTIONS — NEW OBLIGATIONS

By: Julie Machal-Fulks & Christopher Barnett†

TABLE OF CONTENTS

I.	INTRODUCTION	753
II.	HISTORICAL LICENSING MODELS	754
	A. <i>Per-Installation Licensing</i>	754
	B. <i>Server/Client Licensing</i>	755
	C. <i>Custom Licensing</i>	755
III.	NEW LICENSING MODELS	756
	A. <i>Workstations</i>	756
	1. Single-Seat/Stand-Alone	756
	2. Thin Clients	756
	3. The Cloud	757
	B. <i>Servers</i>	757
	1. Server Installations and Clients	758
	2. Processor Licensing	758
	3. Processor Capacity	758
	4. Function-Specific Capacity	759
	5. The Cloud	759
	C. <i>Self-Hosting</i>	760
IV.	COMMON LICENSING CONCERNS	761
	A. <i>Infrastructure Assessments</i>	761
	1. Difficulty Obtaining Deployment Information ..	761
	2. Various Licensing Requirements	761
	3. Raw-Data Complexity	761
	B. <i>Virtualization</i>	762
	C. <i>License Types</i>	762
	1. Retail EULA	762
	2. Volume Licensing	763
	3. Enterprise-Level Agreements	763
V.	SOFTWARE AUDITS	764
VI.	CONCLUSION	764

I. INTRODUCTION

Software licensing options for large enterprises are evolving almost as quickly as the mission-critical software solutions those companies deploy. In the past, most software licensing metrics were based on the software installation itself. Increasingly, software publishers are offer-

* Julie Machal-Fulks, partner at Scott & Scott, LLP, handles intellectual property and technology matters. She can be reached at jfulks@scottandscottllp.com. Christopher Barnett is an associate attorney at Scott & Scott. He can be reached at cbarnett@scottandscottllp.com.

ing more licensing options and flexibility to meet their customers' software needs; however, that increased flexibility often results in complex software asset management (“SAM”) risks and obligations. Licensing models that once would have required custom agreements with unique protocols, if technologically feasible in the first place, now are offered alongside the “traditional” licenses in increasingly dense menus of choices for IT teams to weigh. Businesses must equip themselves to recognize the unique challenges that accompany various options in order to avoid unnecessary licensing exposure.

The options available depend in large measure on the types of computers on which the software will reside. For workstations, many businesses that once relied on a one-license-per-installation model now are migrating to server-based installations accessed from terminals lacking hard drives (e.g., “thin-client” architectures) and to hosted software delivered through the cloud; each of these models presents unique infrastructure and licensing challenges. Server-based licensing options are complex, with many companies facing the prospect of having to determine license requirements using intricate calculations that depend on the processing capacity of the computer or on some other metric associated with a particular software product. Unsurprisingly, many companies are finding unique solutions to those challenges—including the formation of “captive IT services providers”—but many of those solutions present their own sets of challenges and risks.

II. HISTORICAL LICENSING MODELS

The first and most important challenge for software licensing from the software publisher's perspective is determining the correct licensing metric—the “yardstick” that measures a customer's usage of a software product in order to calculate an appropriate licensing fee.

A. *Per-Installation Licensing*

With some exceptions, the metric for most software licensing in the recent past has been the software installation itself. Under this primarily copy-centric or computer-centric metric for determining licensing obligations, a customer simply would count the number of computers on which the software would be installed and would pay a fee based on a set, per-installation price, generally without regard to the performance qualities of the computers in question. Alternatively, many software products also commonly have been licensed on a per-user or per-“seat” basis, which might allow an unlimited number of workstation installations but only a set number of concurrently active user sessions limited by a common server or other network device.¹

1. See, e.g., Autodesk Software License Agreement § 2.2.3, http://images.autodesk.com/adsk/files/autocad-2011-eula-english-all_other_countries.pdf.

B. *Server/Client Licensing*

The most notable exception to this model has been the server/client model for licensing software used on servers, which represents an early and widely used attempt to apply usage variables to an installation-based metric. As with the above approach, this model also looks primarily at the number of devices in question. However, instead of looking only at the number of servers where the software is installed, customers also must count the number of software “clients”—other devices or users—that connect to those software installations remotely over a network. The customer then pays a per-installation licensing fee applied to each installation of the software, together with a per-client fee applied to each client connecting to the software. Alternatively, for certain products and depending on the number of clients in question, the customer might be able to purchase a more expensive per-processor license for each physical processor chip running on a server, which then allow an unlimited number of client connections. The most widely recognized example of this model is the server/client model used by Microsoft for many of its server products (e.g., Windows Server operating systems, Exchange Server messaging applications, and SQL Server database applications).²

C. *Custom Licensing*

While many software vendors may have used different approaches in the past, those exceptions often have been limited to custom, sometimes negotiation-intensive, licensing agreements based on unique characteristics of the software to be licensed and the network or networks where the software would be used. Under this approach, the prospective customer would provide information regarding its network, hardware, and user base to the software vendor, which then would prepare a custom quote. Licensing arrangements of this variety often have been limited to expensive, mission-critical software for which relatively high up-front costs—both in terms of license fees and in terms of legal or administrative tasking—are costs businesses have been willing to absorb.

2. See, e.g., Microsoft Software License Terms, Microsoft Windows Server 2008 § 3(a)(i)–(vii), http://download.microsoft.com/documents/useterms/Windows%20Server_2008%20Standard_English_c1719087-6590-430f-a09e-7d515551c1d0.pdf; see also Microsoft Software License Terms, Microsoft SQL Server 2008 R2 Standard § 2(e)(i)–(iii), http://download.microsoft.com/Documents/UseTerms/SQL%20Server%20Standard_2008%20R2_English_353c9407-2c4f-431e-9d8a-b5970e4773b5.pdf; Microsoft Software License Terms, Microsoft Exchange Server 2010 Standard § 4(a)(i)–(iv), http://download.microsoft.com/Documents/UseTerms/Exchange%20Server%20Standard_2010_English_f8bf0145-756d-49bc-bffa-1a179fab9389.pdf.

III. NEW LICENSING MODELS

Today, businesses have more options and more flexibility in acquiring licenses to accommodate their software needs. Businesses must equip themselves to recognize the unique challenges entailed with the various new options for workstation and server licensing in order to avoid unnecessary licensing exposure.

A. *Workstations*

Software licensing for workstations is typically a little easier for most IT teams to analyze because most of the available licensing follows traditional models. Some of the licensing models for workstations include:

1. Single-Seat/Stand-Alone

Single-seat/stand-alone is the basic licensing model for workstations and is ideal for smaller businesses. All that is required is a count of computers or users and a purchase of an equal number of installation or user licenses. As long as the counts are not too high, the accounting burden is fairly low, as is the level of technical expertise required from IT support.

However, there are significant limitations to this model. First, and most importantly, as the number of workstations or users grows, it becomes much more difficult to accurately determine what is installed on company computers, which makes SAM tasks more onerous. In addition, procurement teams must be able to accurately forecast software licensing needs into the future—most software licenses typically cannot be re-sold to third parties, so a downturn in business operations can result in a large capital expenditure for software licenses being rendered essentially valueless. Those risks can be mitigated somewhat through licensing adjustments available under certain kinds of license agreements for large businesses.

2. Thin Clients

With the thin-client model, the computer hardware at the workstation has very limited, or no, local data-storage capacity, and locally installed software is limited to a client operating system that allows the hardware to communicate with a central server. For many businesses, thin-client licensing represents a more efficient approach, since it allows central control over software deployments and, at least in terms of hardware resources, tends to be more easily scalable.

However, thin clients have their limitations. Most thin-client models represent derivations of the single-seat approach since the licensee typically remains obligated to purchase a number of application licenses commensurate with the number of workstation clients. The principal advantage to this approach—central control of deploy-

ments—is an important one, but it also means that IT teams need to have the technical skills to manage the client network. In addition, initial capital expenditures on network infrastructure and server systems can be substantial.

3. The Cloud

The third workstation-licensing model commonly seen in modern IT environments consists essentially of the transfer of licensing responsibility to a third-party provider of hosted software solutions delivered over the Internet. The hosting provider pays the software vendor for a sufficient number of licenses (typically on a month-to-month basis) to cover the level of access requested by the end-user customer. While the end-user usually must agree to license terms that prohibit copying or reverse-engineering the software, its software access is typically technologically limited to the number of users or devices in its order.³ This allows the end-user to avoid most SAM-related responsibilities. The other principal benefit of this approach is that it is extremely easy to scale—in many cases, it may be possible to adjust the number of licenses included in a monthly order on an incremental basis simply by making adjustments to an online account profile made available by the hosting provider.

Unfortunately, all that flexibility comes with what can be a significant disadvantage for many businesses—the loss of control. By shifting the SAM burden to the hosting provider, a cloud customer relies on that provider to maintain both the right to license software through its hosting solution and the security of any remotely stored data. This usually makes cloud solutions less attractive in direct proportion both to the extent to which the licensed applications are critical to the company's business operations and to the sensitivity of any data stored online. As a result, while the licensing burdens may be minimized, they often are replaced by a different front-end burden related to ensuring that the contractual relationship with the hosting provider includes appropriate insurance, indemnity, and data-security terms (to the extent that the provider will agree to negotiate those terms to any extent).

B. Servers

Server-software licensing is often much more complicated than workstation licensing, even though there are a limited number of licensing models. Most server software is licensable only under one or two of the following models:

3. See, e.g., Autodesk Cloud Terms of Service § 1.3, <http://usa.autodesk.com/adsk/servlet/item?siteID=123112&id=17784802>.

1. Server Installations and Clients

With server licensing, server/CAL licensing, or processor licensing, the advantages and disadvantages essentially mirror those associated with single-seat/stand-alone licensing for workstations. Although it is fairly easy to count the number of servers, it is not always obvious how many users or devices are connecting to the servers. In addition, as with workstations, the flexibility and scalability of the approach can be somewhat limited.

2. Processor Licensing

Under this model, the number of licenses required typically equals the number of physical processors located on the server where the software is installed.⁴ In many instances, products licensed using a processor license do not require separate client access licenses. However, it is important to review the licensing terms specific to the product to determine whether a processor license will allow an unlimited number of users or devices to access the server. This model is most appropriate when the number of users is large or difficult to determine.

3. Processor Capacity

Processor-capacity licensing is a model most commonly associated with a number of server applications published by IBM.⁵ Under a common variant of this approach, the software publisher assigns a processor value to a make and model of physical processor chip or processor core (per IBM, a “Processor Value Unit” or “PVU” per core value).⁶ The intent is for the value to reflect the power or processing capacity of a server chip relative to other server chips. In order to arrive at the cost to run the software on a particular server, the per-processor value then is multiplied first by the number of processors or cores running on that server and then by the per-value-unit price set by the publisher.

In a sense, processor-capacity licensing is little more than a more granular variant of processor licensing, and it shares some of the same limitations of that approach. Unlike processor-based licensing, processor capacity allows for a greater degree of flexibility for businesses that deploy virtual servers or logical partitions on physical machines. For example, with PVU-based products published by IBM,

4. See, e.g., Microsoft Software License Terms, Microsoft SQL Server 2008 R2 Standard § 3(a)–(b), http://download.microsoft.com/Documents/UseTerms/SQL%20Server%20Standard_2008%20R2_English_353c9407-2c4f-431e-9d8a-b5970e4773b5.pdf.

5. See e.g., *Processor Value Unit [PVU] licensing for Distributed Software*, IBM, http://www-01.ibm.com/software/lotus/passportadvantage/pvu_licensing_for_customers.html (last updated Nov. 15, 2011).

6. *Id.*

businesses have a choice between licensing the full capacity of the physical processors running on a server or only the capacity of the virtual processors accessed by the software. This can result in substantial cost savings. It is important to keep in mind, though, that for products where it is available, sub-capacity licensing on virtual servers typically involves very complex licensing arithmetic and, in some cases (including IBM), additional licensing requirements related to monitoring the server resources actually accessed by the software. These requirements can put a substantial strain on IT team resources.

4. Function-Specific Capacity

The wide variety of functions addressed by server applications is mirrored by an increasingly wide variety of function-specific licensing metrics for those applications. For example, data-backup software might be licensed based on the volume of data subject to backup operations, or database software might be licensed based on the maximum number of concurrent connections to the database during a reporting period.⁷ These models can offer a high degree of flexibility and scalability over other models since licenses often can be purchased on a more incremental basis as processing needs grow over time. However, function-specific models still require accurate forecasting and planning for perpetual license needs. In addition, with increasingly creative license metrics often come increasingly complex measurement requirements, and it can be very difficult—especially in the event of an audit—to show to everyone’s satisfaction that the licenses previously acquired for a product adequately support the usage levels detected during an inventory or network-scanning process.

5. The Cloud

As they do with workstations, many businesses also look to hosting services providers to deliver server-based application solutions over the Internet. This carries the same kinds of advantages in flexibility and scalability as cloud-based workstation solutions as well as the same kinds of concerns related to control. However, server applications usually magnify those risks—the nature of the data stored typically is of a wider scope, and the mission-criticality of the applications accessed often is significantly more pronounced. Businesses are wise to think long and hard before ceding primary control of their most important IT assets to a vendor bound only by the terms contained in a service level agreement.

With regard solely to acquisition costs, the ideal server-licensing solution typically involves some combination of the various licensing

7. See e.g., *Learn about Software Licensing: Resource Value Unit*, IBM, http://www-01.ibm.com/software/lotus/passportadvantage/about_software_licensing.html#rvu (last visited Feb. 12, 2012).

models identified above depending on the particular solution deployed and the nature of the affected IT infrastructure. Nevertheless, correct license-counting in an environment containing multiple licensing platforms can become very difficult and fraught with challenges. License agreements or license-agreement attachments documenting different license acquisitions under different licensing models can become very difficult to track over time, leaving businesses with an inaccurate understanding of what they actually own in terms of license rights. In addition, even if SAM teams accurately understand the extent of their companies' license-rights, IT staff in a mixed environment may be unfamiliar with all applicable licensing obligations and, for example, may make incorrect assumptions about what is permissible or required for one product based on past experience with different products. Finally, while many IT professionals rely on software-based solutions to help gather inventories of software deployed on company computers, those tools have their limits. Networks running products deployed under several different licensing metrics may require an equal number of data-gathering tools in order to accurately determine what is running and what must be licensed. Challenges such as these may be essentially unavoidable for many businesses, but it is important to keep in mind that it may be possible to mitigate the resulting SAM burdens by considering other licensing models at an early stage when alternatives exist.

C. *Self-Hosting*

One remaining licensing model increasingly used as a solution both for workstation- and server-licensing challenges is self-hosting through a "captive" IT-services entity. Under this model, a separate legal entity delivers the software through the Internet to the enterprise. Many businesses have discovered that such a model allows them to realize both the pricing predictability and scalability of cloud-based solutions without ceding fundamental control over the IT environment.

The principal downside of this approach is that, unlike with true cloud offerings, the enterprise also retains responsibility for ensuring that all software deployments are properly licensed under the business-partner license from the software vendor. Those licensing obligations typically include potentially burdensome, monthly reporting obligations. In addition, most of the agreements under which third-party hosting providers are able to deliver software to their end-users (such as Microsoft's Services Provider License Agreement or "SPLA") are not intended as mechanisms for companies to use for self-hosting. In fact, those agreements often contain internal-use restrictions or other terms that represent obstacles to implement such an approach for many businesses. Any companies planning for self-hosting need to spend a substantial amount of effort at the outset of the

process carefully structuring the hosting entity and its relationship to the enterprise to ensure that internal-use restrictions are not triggered, leading to potential compliance gaps down the road.

IV. COMMON LICENSING CONCERNS

Regardless of the nature of the environment to be licensed or the licensing model selected by a company, there are a number of challenges common to most or all enterprise-level, software-licensing initiatives. These include the following:

A. *Infrastructure Assessments*

It is not possible to correctly license any kind of software unless you know what hardware you own and what software is installed on each hardware asset. Therefore, some level of internal auditing is necessary in almost all cases in order to gather information about the enterprise's IT infrastructure. The inventory process can itself be quite challenging depending on the network characteristics and on the software to be licensed. Common challenges include:

1. Difficulty Obtaining Deployment Information

If a company does not already have a solution in place to track software deployments in its organization, it can be a difficult process to initiate. Often tools designed to provide insight into what is installed on the network take months or years to implement. Additionally, the raw output does not contain the information necessary to determine what licenses are required. For instance, if a product is licensed on a processor basis but the software inventory does not include the number of processors on the physical machines, the inventory report will be insufficient to determine the number of licenses required.

2. Various Licensing Requirements

As noted above, many businesses run or need to run a variety of software products on a diverse collection of hardware systems. IT staffs need to be familiar with the different licensing requirements applicable to different software products and hardware types. In many cases, the resulting complexity means that IT staff may be unequipped to handle all aspects of the licensing assessment without the input of a third party or the company's legal team.

3. Raw-Data Complexity

After a business gathers the software deployment information assessment, it still must be prepared to interpret the data. Many tools include hotfixes, updates, and trial versions and a manual review is

required to eliminate irrelevant information from the software inventory.

B. *Virtualization*

Many enterprises are moving to IT environments that utilize virtual technologies, and many of those virtualized environments also may utilize server clustering—using two or more physical servers with shared and managed resources to host the virtual servers. Seeking to capitalize on this trend, most software publishers limit their customers' ability to license software in virtualized environments. For example, Microsoft often limits the number of virtual operating system environments in which a software product may be installed depending on the edition of the software to be deployed (e.g., SQL Server Datacenter versus SQL Server Enterprise).⁸ In addition, IBM often requires that a server or cluster be licensed to its full processor capacity for a software product—even if only a single virtual machine hosted on the server or cluster is running that product—unless the company agrees to the technical and procedural requirements for sub-capacity licensing, allowing for license acquisition at the virtual-server level.⁹

C. *License Types*

Another significant licensing concern common to all businesses is the need to identify the appropriate kind of license agreement to pursue with a software publisher, in those cases where different agreement forms are available. The three most common varieties are the following:

1. Retail EULA

The retail shrink-wrap or click-wrap end-user license agreement (“EULA”) remains the most familiar kind of license agreement for the software industry and most individual end users. This kind of agreement typically consists of take-it-or-leave-it terms that accompany installation files for software products. By opening a software package or installing the software product, the user impliedly consents to the terms of the agreement, in the vast majority of cases without any opportunity for negotiation or modification.

Large enterprises generally do not purchase significant amounts of software governed by retail EULAs. These products do not offer any

8. See MICROSOFT, SQL SERVER 2008 R2 LICENSING QUICK REFERENCE GUIDE, 9 (version 1.1 2011), available at http://download.microsoft.com/download/4/8/E/48E9A4EB-125A-49CB-9644-169B82C45611/SQL2008R2_QRG_2011.pdf.

9. See IBM, SUB-CAPACITY (VIRTUALIZATION) LICENSE COUNTING RULES: LICENSE COUNTING IN A SERVER “CLUSTER” 7 (2009), available at ftp://ftp.software.ibm.com/software/passportadvantage/SubCapacity/Scenarios_VMware.pdf.

opportunity for custom terms, and pricing typically is not competitive for large license purchases. In addition, retail EULAs make license tracking for large organizations very difficult, especially when employees who have no authority to bind the company purportedly accept the terms on the organization's behalf.

2. Volume Licensing

For those companies that are ready to transition from retail EULAs to a more flexible license agreement and are able to commit to purchase requirements over a set term, most large software publishers offer standardized license agreements with more competitive pricing and more scalable license-management options. Many of these agreements also include ancillary benefits (for example, downgrade rights, which is the ability to use an older version of a licensed product instead of the most current version included in a purchase order) and online, license-management portals, both of which facilitate license tracking and thereby can help to reduce audit exposure.

However, like retail EULAs, most volume licensing agreements remain pre-printed forms that offer little, if any, room for customization and that also typically include onerous audit rights in favor of the publisher. In addition, most form volume-licensing agreements still require a substantial front-end capital expenditure and limited options for spreading license costs over the useful life of the software in question. In order to achieve a licensing deal that meets the unique needs of a larger enterprise, something more usually is needed.

3. Enterprise-Level Agreements

Large enterprises often seek the availability of license agreements covering multiple (or all) divisions within the enterprise. Typically, only such enterprise-level agreements offer even the possibility of more favorable terms for scalability and auditing. For example, unlike other volume-licensing models offered by Microsoft, only the Enterprise Agreement allows for annual, retrospective license true-ups in lieu of predictive license purchases based on current or future need. In addition, depending on the licensing commitments that a company is in a position to make, enterprise-level agreements also may offer the possibility of terms customization, allowing some businesses to tailor reporting, audit, or other obligations to the demands or flow of their business operations.

That level of flexibility has a price, which most often takes the form of substantially larger purchase commitments over the term of the agreement. Where retail EULAs typically require no purchase commitments and lower-level volume agreements may require orders of several thousand dollars in order to participate, most enterprise-level agreements contemplate commitments in the tens or hundreds of thousands of dollars, if not more. In addition, given the dollars in

question and the scope of potential exposure, enterprise-level deals ideally should include the input of legal counsel, which might slow down the contracting process (though often with the benefit of more favorable terms when the deal is signed). Finally, enterprise-level agreements also typically require IT teams to implement new solutions or procedures to correctly track deployments and licenses under the applicable agreement or agreements.

V. SOFTWARE AUDITS

One remaining concern that deserves special mention is the subject of publisher-initiated software audits. Most software publishers, by default, include relatively onerous audit-rights provisions in their form agreements, such as:

- Very few restraints on audit timing and frequency, as opposed to defined limits as to when audits may be demanded;
- No express limitations on the scope of potential audits (either legal, geographic or product-specific);
- Few or no meaningful protections for information disclosed by the enterprise during the course of the audit (either as to confidentiality or to admissibility in court, in the event litigation arises); and
- Burdensome resolution terms, including:
 - License purchases for unlicensed deployments, regardless of use;
 - Back-maintenance purchases for unlicensed deployments (or, in some cases, a percentage over the MSRP licensing costs);
 - Obligation to pay the publisher's third-party auditor, in the event that any compliance gap exceeds a stated threshold.

For these reasons, before a company reports any deployment or entitlement information to a publisher in response to a software audit, it is crucial for that company to consult with counsel to determine the appropriate response to the audit. The most common mistake that many businesses make at this stage is to assume that those demands are consistent with the negotiated terms, which often results in over-reporting and unnecessary exposure.

VI. CONCLUSION

Enterprise software licensing is a complex undertaking that demands careful and dynamic attention to an often-shifting set of licensing details. There are few points along the process where corporate counsel cannot make positive contributions to implement appropriate licensing options, and there are certain points where legal involvement is crucial to ensuring that the company remains compliant with applicable license agreements. Unfortunately, many businesses discover this only in the context of an audit, when what may have been an uncomfortable cost associated with preventative legal review is re-

2012]

ENTERPRISE SOFTWARE LICENSING

765

placed with painful or even crippling compliance costs resulting from under-licensed, mis-licensed, or unlicensed software deployments. Licensing agreements at the enterprise level often are multi-million-dollar transactions with large corporations that have sizeable legal teams tasked with securing the most favorable terms for the publisher. Enterprise-level customers enter negotiations on such agreements at their peril, unless they are equally well prepared.