The Tech Contracts HANDBOOK

Second Edition

Cloud Computing Agreements, Software Licenses, and Other IT Contracts for Lawyers and Businesspeople

David W. Tollen

ABA AMERICAN BAR ASSOCIATION
Section of Intellectual Property Law

The Tech Contracts HANDBOOK

Second Edition

Cloud Computing Agreements, Software Licenses, and Other IT Contracts for Lawyers and Businesspeople

David W. Tollen

Section of
Intellectual Property Law

Cover design by Amy Mandel with the Section of Intellectual Property Law.

The materials contained herein represent the opinions of the authors and editors, and should not be construed to be the views or opinions of the law firms or companies with whom such persons are in partnership with, associated with, or employed by, nor of the American Bar Association or the Intellectual Property Section unless adopted pursuant to the bylaws of the Association.

Nothing contained in this book is to be considered as the rendering of legal advice for specific cases, and readers are responsible for obtaining such advice from their own legal counsel. This book is intended for educational and informational purposes only.

Library of Congress Cataloging-in-Publication Data

Tollen, David W., author.

The tech contracts handbook : cloud computing agreements, software licenses, and other IT contracts for lawyers and businesspeople / by David W. Tollen.—Second Edition.

pages cm

Includes index.

ISBN 978-1-63425-179-2 (epub)

1. Computer contracts—United States. 2. Technology transfer—Law and legislation—United States. 3. Computer software—Licenses—United States. 4. Copyright—Computer programs—United States. I. Title.

KF905.C6T65 2015

346.7302—dc23 2015020358

Discounts are available for books ordered in bulk. Special consideration is given to state bars, CLE programs, and other bar-related organizations. Inquire at ABA Publishing, American Bar Association, 321 North Clark Street, Chicago, Illinois 60654-7598.

www.ShopABA.org

Thank you to my wife, Wendy Pagett-Tollen. Thanks also to Mani Adeli, Paul Ambrosio, Marc Bernstein, Jay Botelho, Lee Bruno, Michael C. Carlson, Ralph Chandler, Guy Clarke, H. Ward Classen, Christopher B. Conner, Matteo Daste, Gary S. Davis, Mark K. Dickson, Karen Masterson Dienst, R. Oak Dowling, Roxanna Friedrich, Frederick Gault Jr., Kim David Greenwood, Jennifer Hanley, Michael A. Jacobs, Nels Johnson, John M. Keagy, Michael F. Kelleher, Cecilia Toman Mangoba, Alex Mann, Heather Meeker, Nathaniel D. McKitterick, Denise Olrich, Kathy O'Sullivan, Jay Parkhill, Martin Plack, Penelope A. Preovolos, Robert A. Preskill, Deborah Pulido, Mark F. Radcliffe, Ian A. Rambarran, Michelle M. Reichert, Aaron P. Silberman, Ken Stratton, Robert W. Tollen, Larry Townsend, Richard Vestuto, and Amy Ward. Finally, thank you to my excellent research assistant, Josh L. Young.

Please visit **http://TechContracts.com** *for contract clauses and full-length contract forms you can download, and for other resources.*

Contents

About the Author

Attorney David W. Tollen represents buyers and sellers in cloud computing and software licensing agreements and in other technology transactions. He also provides advice and assistance related to e-commerce, social media, industrial design, and intellectual property. Finally, he provides in-house and public training on drafting and negotiating technology contracts, for lawyers and for contract managers, salespeople, financial executives, and other businesspeople. He's the founder of Sycamore Legal, P.C., a technology and intellectual property law firm based in San Francisco (http://SycamoreLegal.com).

Mr. Tollen graduated with honors from Harvard Law School and has degrees from Cambridge University in England and U.C. Berkeley. He has served as general counsel of a publicly traded software company, as vice president of business development for a technology start-up, and as a lawyer in the Silicon Valley offices of Morrison & Foerster LLP.

Finally, Mr. Tollen writes fiction that teaches history and science. He's the author of the multiple-award-winner *The Jericho River, A Novel About the History of Western Civilization* (Winifred Press 2012, 2014) and also of *Secrets of Hominea,* to be published in 2016. (For more on these works of fiction, please visit http://DavidTollen.com.)

Introduction

This book will help you negotiate, draft, and understand information technology contracts. Specifically, it will help you with software licenses and other software transfers, cloud computing agreements, and technology professional services agreements. It addresses contracts between businesses, as well as business-to-consumer and business-to-government contracts. It also addresses both offline contracts and contracts related to the Internet and e-commerce.

This book is for both lawyers and nonlawyers. The text stays away from technical jargon—“legalese,” “engineerese,” and “programmerese”—and where it absolutely can’t avoid jargon, it provides a definition. In other words, this book is written in simple English, like a good contract.

You can use this book as a training manual or a reference guide or both. If you’re training, read this book cover to cover. It provides an overview of the key technology contracting concepts.

If you’re after a reference guide, you can pick and choose the chapters to read. When you’re negotiating a contract, or reading or writing one, look up the various clauses to learn what they mean and what’s at stake. You’ll find sample language in each chapter, which you can incorporate into your own contracts. Plus, if you visit this book’s website, http://TechContracts.com, you can copy the longer sample clauses and paste them into your document. You’ll also find several full-length contracts at the website, which you can download and revise to fit your deals.

Finally, you can also use this book’s table of contents as an issue spotter —as a checklist of clauses to consider.

This book can’t replace a lawyer—or a colleague with more information technology (IT) experience, if you are a lawyer. But it can help you understand your lawyer or colleague. And whether you have legal help or not, the better you understand your contracts, the more effective you’ll be.

I’m a technology lawyer, and this book grew out of seminars I teach, for both attorneys and nonattorneys. At the end of the program, students often asked where they could learn more—if I knew a good book on IT contracts. Most of the books I knew were massive tomes on intellectual property or contract law. They’re written for lawyers only, and their more practical lessons are spread across hundreds or thousands of pages. I’ve learned much of my trade on the job, rather than from a book. I’ve served as a technology lawyer with a global firm, as general counsel for a publicly traded software company, and as vice president of business development for an Internet start-up. I now practice through my own technology-focused law firm in San Francisco and the Silicon Valley. The material for my seminar came from the contracts I’ve negotiated and written in those positions. I’d never seen a really user-friendly outline of the issues. So I wrote this book.

• • • •

The rest of this introduction provides more detail about the types of contracts this book covers. It also explains the structure of a contract and of this book and offers a few explanations that will help you get the most out of your reading. Finally, it provides a short explanation of some IT industry language—just a little, particularly regarding cloud computing—and then offers three lessons about contracting in general.

Subject Matter: Types of IT Agreements

This book addresses four principal types of IT contracts:

1. ***Software License Agreements*** and ***Software Ownership Agreements*** transfer intellectual property rights from the vendor to the customer. They include end user licenses, enterprise licenses, distribution contracts, assignments, and work-for-hire agreements. In all these deals, the customer gains, at a minimum, the right to put one or more copies of the software on its computers: the right to make copies. These contracts need an intellectual property (IP) transfer because the right to make copies is protected by copyright and other IP laws.[1] The rights transferred might be limited, like the right to make a few copies or to distribute the software: a software license. Or the customer might receive all IP rights and become the software's new owner: a software ownership transfer.
2. ***IT Professional Services Agreements*** call on the vendor's staff to *help* the customer. The vendor's professionals are going to do something, rather than simply making software or other technology available (as in a software license or cloud services agreement). IT professional services include system integration, tech support, website development, software maintenance, and technology consulting.
3. ***Cloud Services Agreements*** call on the vendor to host a cloud computing system and to give the customer remote access, usually via the Internet. They're sometimes called *Cloud Hosting Agreements* or *Cloud Computing Services Agreements*. A cloud services agreement is neither a professional services contract nor a software license. It's not professional services because computers play the key role in providing the service, not human professionals. And it's not a license because the customer gets no copies of the cloud computing software—just remote access. (For more on cloud services, see "A Little Industry Language, Particularly re Cloud Computing," later in this Introduction.)
4. ***Combination Agreements*** call on the vendor to provide some combination of software, professional services, and cloud services. Computer programming contracts, for instance, usually call on the vendor to write software, a professional service, and to transfer IP rights in that software, a software license or ownership transfer. Cloud hosting and support contracts call on the vendor to host software (or other technology) and make it available to the

customer, a cloud services offering, and to help the customer figure out how to use that software, a professional service. A combination contract works like two or more contracts in one. It needs terms appropriate for two or three of our contract types: software licensing, professional services, and/or cloud services.

This book will also help you with purchase and lease agreements for computers and other IT hardware. That's because many of the clauses discussed here appear in hardware contracts too, and some of the contracts listed above can include hardware purchases. A software customer, for instance, might license software and buy computers to run it, all in one contract. That said, hardware agreements involve some terms you don't see in other types of IT contracts, like leases, security interests, and shipping terms. This book doesn't cover those clauses.

Finally, this book addresses both private contracts—business-to-business and business-to-consumer—and government contracts. A government contract is an agreement between a private vendor and a government agency customer. It can involve any of the contract types described above. Most government contracts use language required by government contracting laws and rules, like the federal acquisition regulations (FARs) and state procurement regulations. This book doesn't address those rules, but it should still help you understand the language they require.[2] And of course, where the government doesn't require its own language, you can use this book's sample clauses.

The Structure of a Contract and of This Book

IT contract terms can be organized into three groups: *prime clauses*, *general clauses*, and *boilerplate clauses*. This book is organized the same way.[3]

The prime clauses express the deal's central terms. There, the vendor grants a license or other rights to software, or promises to provide professional services or cloud services—or some combination of the three, in a combination contract. The customer, on the other hand, promises to pay. This book addresses prime clauses in Part I.

The general clauses account for most of the contract. They cover everything not addressed in the prime clauses or the boilerplate clauses, and they're usually the most heavily negotiated. This book addresses general clauses in Part II.

Boilerplate clauses cover the theoretically noncontroversial mechanics of a deal: terms on independent contractor status, contract interpretation, choice of law, etc. IT professionals tend to put most of these terms at the end of a contract. This book addresses boilerplate clauses in Part III.

Contracts usually start with two sets of boilerplate clauses: the introduction (including "recitals") and the definitions.[4] From there on, you should organize your clauses the way they're listed above: prime clauses, then general clauses, then the remaining boilerplate clauses. That makes agreements easy to understand. Unfortunately, though, you'll probably run across contracts with these clause types jumbled together in no particular order.

Using This Book

The following four brief notes and explanations will help you get the most out of this book.

First, almost any contract you write should be customized to fit your particular deal. So if you insert one of this book's sample clauses—which is what they're here for—don't do it thoughtlessly. You may need to edit it. Think through the unique issues raised by your deal. "We know the Windows version of our software has some serious bugs, so we don't want to give a broad warranty." "We've got a small IT department, so we need extra support." "Our CFO gets hives if we give personal injury indemnities." This book offers building blocks for a contract addressing issues like those, but the customizations are up to you.

Second, this book addresses U.S. law. That doesn't mean it's useless for contracts under other countries' laws. Most contract clauses mean what they say, regardless of the underlying legal system. But with some clauses, the underlying law really will affect the meaning. So if you're working outside the U.S., consider help from a local lawyer.

Within the U.S., the 50 states have similar contract laws, and federal law governs some of the issues discussed here, particularly related to copyrights and patents. So state law variations won't often lead your IT contracts astray. But sometimes state law variations really do matter. That's another reason to consider help from an experienced lawyer.

Third, like most contracts, the examples in this book use defined terms. When a contract creates a concept and uses it more than once, it usually defines it. For instance, a contract might list the vendor's services in Section 2, then mention them over and over in other sections. Rather than listing the services repeatedly, the contract defines the list as the "Services." Whenever the contract refers to the "Services" with a capital S, it means the whole list. This book's sample clauses work the same way: capitalized words that aren't proper names represent defined terms—e.g., "Software," "Effective Date," "Statement of Work," this "Agreement." (Some contracts mark defined terms with all caps instead—e.g., the "SERVICES.") The same goes for sets of initials in all caps, like "NDA" (for nondisclosure agreement). In this book, the sample clause often won't supply the definition. That's because, in a real contract, another section would define that term. Obviously, in your contracts, you should provide the definitions somewhere.

Fourth, most of this book's sample clauses use the defined terms "Vendor" and "Customer," and so does the text. Contracts you've worked with may use other names. "Vendor" stands in for "Licensor," "Provider," "Transferor," "Assignor," "Seller," and "Consultant," among others. And "Customer" stands in for "Licensee," "Transferee," "Assignee," "Recipient," "Buyer," and "Client." This book favors "Vendor" and "Customer" because they're generic.[5] But the text and sample clauses do occasionally use names like "Distributor" where necessary to avoid confusion.

A Little Industry Language, Particularly Re Cloud Computing

This book includes the world's shortest glossary, on page 271. It explains five terms you'll see in the text: *calendar (as in "calendar quarter"), including without limitation, object code, source code,* and *without limiting the generality of the foregoing.*

This book also uses some terms related to cloud computing, and they're important enough to explain up front. The key terms are "cloud computing" itself and "cloud services." It's also worth explaining "software-as-a-service," and while we're at it, we'll touch on two related terms: "platform-as-a-service" and "infrastructure-as-a-service." Experts actually disagree on these terms' definitions, and many would say this book oversimplifies some complex concepts. But the definitions given here work for our purposes.

"Cloud computing" is a model for delivering software and other IT resources through a particular type of computer network. The software sits on one or more central computers—servers—and the customer's users access it remotely, from their own computers (often desktops and other client computers). Usually, access is via the Internet. The customer might host the server computers and software itself, but often the vendor (or its reseller) hosts.

Where the vendor hosts the server computers and software, this book and many IT professionals call the arrangement "cloud services" (or sometimes "hosting services" or "cloud computing services"). As you'll see in Chapter I.E ("Subscription for Cloud Services"), cloud services don't involve a software license, since the vendor keeps the software to itself; it doesn't give the customer any copies. What the vendor really provides is a service: remote access to and use of its server computers and the software. On the other hand, if *the customer* hosts the software, it does need a license, since it has to make copies of the software. So an agreement letting the customer host is a software license agreement. This book doesn't talk much about "customer-hosted cloud computing," but that's not because it's unimportant. Rather, this book lumps it in with other software licenses.[6]

You've probably heard a lot about a particular type of cloud services: "software-as-a-service" (SaaS). SaaS refers to a cloud service where the remotely hosted IT resource is a software application, or several applications. An *application* is software for users, like a word processing or contacts management program—something a human being actually sits down and uses. It's contrasted with *platform* software (aka system software): a program that runs a computer, like an operating system. So in a SaaS relationship, the vendor puts software for users on its server computers and makes it available to the customer, usually via the Internet.

IT professionals talk about two other types of cloud services. In a "platform-as-a-service" (PaaS) offering, the vendor hosts a software platform. The customer installs or creates applications on that platform and uses them. And in "infrastructure-as-a-service" (IaaS), the vendor hosts server computers and other hardware infrastructure. The customer installs both platform and application software on that infrastructure, and uses them. In both cases, the customer usually accesses the systems via the Internet. This book rarely addresses PaaS or IaaS separately; it just discusses cloud services.

Three Lessons about Contracting

1. Good Fences Make Good Neighbors

Why do we sign contracts? It's not because we want to win a lawsuit later. It's not because we don't trust each other. It's not even because we're afraid lawyers will stir up trouble if they're not kept busy.

We sign contracts because good fences make good neighbors.

The best way to avoid arguments in a business relationship is to write down the parties' expectations ahead of time. That list becomes a boundary marker—like a fence between neighboring yards—explaining who's responsible for what. If the parties disagree, they can look at the list for guidance.

In other words, contracts *prevent* disputes—at least, good ones do. They prevent lawsuits.

Even if the parties never look back at the contract once it's signed, it has still probably played a vital role. When people put their business expectations on paper, they often find those expectations don't match. Just the act of negotiating a written[7] contract will uncover many mismatched expectations. The parties can address them before starting work.

Yes, it's true that we sometimes fight over contracts in lawsuits. And yes, in interpreting a contract, we often talk about what a judge would say it means. But that's only because courts have the ultimate say if the parties can't agree. Job number one for the contract is to keep the parties out of court.

2. There Is No Such Thing as "Legalese" or "Technicalese"

You may feel uncomfortable with contracts because of the unfamiliar language they use. Don't be intimidated. You can understand most contracts.

There really is no such thing as legalese. American contracts are written in English (or Spanish or Vietnamese or whatever language the parties speak). But contracts do sometimes use special shorthand: terms lawyers have developed to save time. And some IT contracts use "technology shorthand." Finally, contracts sometimes use formal, stilted language with long run-on sentences. Don't let shorthand or stilted language bother you.

If you run into an unfamiliar term in a contract—unfamiliar shorthand—don't worry. Look it up. If it's legal shorthand, you can probably find the definition in a standard dictionary, or online, or in *Black's Law Dictionary*,[8] found in many libraries. Treat technology terms the same way. Look them up in a dictionary or technical manual or online. Or ask someone with the right expertise.

Once you understand a term, feel free to use it in your own contracts. But you should also feel free *not* to use it. Shorthand is optional. If you do use shorthand, be sure the contract defines each technical term. Definitions can vary for IT terms like "sandboxing" and "bot,"[9] so the contract needs an agreed definition, unless there really can't be any doubt. Legal terms, on the other hand, often have widely accepted definitions, so you usually don't need to define them in the contract.

As for long sentences, just take a deep breath and read slowly. The same goes for formal language. There really is no reason to use terms like "heretofore" and "*lex loci*."[10] That sort of language often appears in form contracts from the olden days, when formal writing and Latin were more popular. It does crop up in modern contracts—often because someone wants to show off a big vocabulary. Be suitably impressed. Then take out your dictionary if necessary and figure out what each sentence says. And avoid terms like that in your own writing.

3. Ask Yourself "What's Our Best Option?" Not "What's Fair?"

Some businesspeople and lawyers ponder and argue a lot about whether proposed contract terms would be *fair*. I think that's an unhelpful view of contracts, for two reasons. First, it's hard or even impossible to define "fair" in contract negotiations. Second, a focus on what's fair may lead you to reject deals that make economic sense, or to accept deals that don't. The better question is: would doing the deal under these terms be more profitable than *not* doing it?

What does “fair” even mean in contract negotiations? Each party has a choice about whether to do a deal, so neither owes the other any particular terms. If some company insists on terms heavily slanted in its favor, and no one ever accepts them, that company’s dumb, not unfair. Or maybe it just doesn’t care whether it does any deals. On the other hand, if enough people do accept the bad terms, does it make sense to call them unfair? Does it even make sense to call them *bad terms*? The fact that “the market” accepts the terms legitimates them.[11]

So you might say “fair” can’t be defined in contract negotiations. Or you might say “fair” means “acceptable to enough people.” Either way, the guiding principle behind contract terms is leverage: whether the proposing party can get its terms often enough to do the deals it needs.

If you do focus on what’s fair, you might walk away from deals that make economic sense. If terms you don’t like seem unfair, and you can’t get the other party to budge, you’ll probably feel too screwed to sign the contract. But that’s a poor choice if you couldn’t get better terms from anyone else, and if accepting the terms would be more profitable than dropping the project. What if you focus on *best option available*, rather than fairness, and recognize that the other party wants its best option too? In that case, you’ll accept the deal if it’s the best option the market has to offer—and you’ll feel good about it. You’ll only walk away if you have better options.

A fairness focus could cut the other way too. You might make concessions because the other side’s requests sound fair—despite the fact that you’ve got options better than doing the deal under those terms. If you simply ask yourself whether you’ve got better options, and the answer’s *yes*, you’ll refuse the other side’s “fair” terms.

I’m not suggesting contract negotiators should be androids or Vulcans, who react only to logic. Most of us want the other side to leave the bargaining table happy. And we *do* respond to arguments about fairness. We just need to remember how slippery the concept is, and how ultimately the market for other options shapes the definition of “fair,” rather than some objective concept of right and wrong.

[1] For an explanation of IP and its role in tech contracts, see Appendix 1 (“Intellectual Property”).

[2] Many government contracting rules require terms unique to government procurement, like “buy American” provisions and clauses on conflicts of interest or use of recycled paper. This book doesn’t address those terms.

[3] You might not find these terms outside this book (the same goes for “combination contract,” above), but they’re meant to express common views of these clause types. The first edition of this book used “transactional clauses” instead of “prime clauses,” and “supporting clauses” instead of “boilerplate clauses.”

[4] See Chapters III.A (“Introduction and Recitals”) and III.B (“Definitions”).

[5] This book’s first edition used “Recipient” instead of “Customer” and “Provider” instead of “Vendor.”

[6] Of course, the vendor could host some cloud systems while the vendor hosts others. That would create a combination contract, with two prime clauses: a subscription for cloud services and a software license. See “Subject Matter: Types of IT Contracts,” earlier in this Introduction.

[7] Many contracts can be oral rather than written, but some can’t. And oral contracts lack the advantages of clarity and detail.

[8] From the West Publishing Company.

[9] *Sandboxing*: separating a computer program from other programs to limit the impact of errors and security issues. *Bot*: a software program that mimics human behavior in that it’s automated (short for *robot*).

[10] *Heretofore:* before now. *Lex loci:* Latin for the law of the place, usually referring to a contract’s choice-of-law clause.

[11] OK, that’s not entirely true, at least so far as the law is concerned. There are a few contract terms courts won’t enforce because they’re “opposed to public policy” or “unconscionable.”

PART I

Prime Clauses

The prime clauses are the key terms in most technology contracts. They provide for the fundamental transaction: the exchange of software or services for money or other consideration.

Most software and tech services contracts include two of the clauses described in this part: (1) a transfer or sale of software rights or of services (Chapters I.A through I.F) and (2) a promise of payment (Chapter I.G). But combination contracts—agreements with multiple transactions—include several prime clauses.[1]

A. Standard End User Software License

A license grants the customer rights to copy software or to exploit it in other ways. It leaves ownership with the vendor. A license works like a rental agreement. The vendor/landlord still owns the house, but the customer/tenant gets to use it.

This chapter looks at standard end user licenses: the central clause in an end user license agreement (EULA). In a standard license, the customer gets the right to run software for internal business purposes. It can't share the software with third parties or modify it.

A software license is a copyright license, but this chapter doesn't go far into the mechanics of copyright. That kind of knowledge isn't usually necessary for a standard license. If you want a deeper understanding of licensing, or if your license doesn't fit the "standard" model discussed here, see Chapter I.C ("Software Licenses in General").

Before drafting your license, ask yourself: *What* is being licensed? The contract should clearly define the "Software" or "Licensed Product"—usually in a separate definitions section. In a standard license, it's usually enough to give the software's name and version number and specify object code: "'Software' refers to Vendor's *GlitchMaster* software application, version 3.0, in object code format." But if the software has multiple modules or libraries or whatever, or if you see any chance of dispute about what's included, list the necessary elements: "'Licensed Product' refers to Vendor's *RoboSurgeon for the PC* software application, version 2.0, in object code format, including the following modules: RemoteScalpel, Anesthesia-Alarm, and MalpracticeManager." You might also specify the platform: Windows, Macintosh, Linux, etc. Finally, if the customer needs to reproduce user manuals and other documentation, the definition might include them: "The Licensed Product includes Vendor's standard user manuals and other documentation for such software."[2]

1. Reproduction and Use

End user licenses employ various terms for the rights granted. Most license clauses grant rights to “use,” “run,” “install,” “download,” “copy,” or “reproduce” software. These terms have commonsense meanings, but many of them overlap. This chapter sticks to “reproduce” and “use,” to avoid throwing around too many overlapping terms. I recommend you do the same for your end user licenses.

The customer should always get the right to *use* the software. Specifically listing *reproduction* rights, on the other hand, isn’t always necessary. If the vendor delivers ten copies—ten CDs, for instance—and the customer only needs ten, the license doesn’t need the right to reproduce. The same goes for downloaded software. If the customer can keep downloading copies until it has the correct number, it doesn’t need reproduction rights. Of course, technically speaking, you reproduce software every time you install it, but the right to *use* software implies the right to make an installed copy. The customer only needs reproduction rights if it can make more copies than the vendor delivers—for instance, if the vendor sends one CD or allows one download, and the customer needs ten copies. In that case, the license should grant the right to *reproduce* and *use* ten copies.[3]

Standard End User Reproduction and Use

Vendor hereby grants Customer a nonexclusive license to reproduce and use __ copies of the Licensed Product for Customer's internal business purposes, provided Customer complies with the restrictions set forth in this Section __.

• • • •

Vendor hereby grants Customer a nonexclusive license to reproduce and use the Software as necessary for Customer's internal business purposes, provided Customer complies with the restrictions set forth in Section __ (*Restrictions on Software Rights*). Such internal business purposes do not include use by any parent, subsidiary, or affiliate of Customer, or any other third party, and Customer shall not permit any such use.

• • • •

Vendor hereby grants Customer a nonexclusive license to use the Licensed Product for its internal business purposes, provided: (a) Customer may give no more than __ concurrent users access to the Licensed Product; and (b) Customer complies with the other restrictions set forth in this Section __.

• • • •

Vendor hereby grants Customer a nonexclusive license to use the Software for its internal business purposes, provided Customer: (a) deploys the Software to no more than __ seats; and (b) complies with the other restrictions set forth in this Section __.

If the customer can reproduce the software, the license should specify the number of copies, as in the first example in the clause box above. At least, that's the case for most deals. Some contracts call for an "enterprise license." In an enterprise license, like the second example in the clause box, the customer can make as many copies as it needs. A vendor should only grant an enterprise license if it knows the size of the customer's business and doesn't mind if it expands, along with the number of copies—or if the fees cover any likely expansion. Enterprise license vendors should also consider limiting software use to the customer itself and forbidding use by subsidiaries, parent companies, and other affiliates. Again, see the second example above.

Some software sits on a single server computer, and users access and use it from their own client computers (desktop, laptop, phone, etc.), without making new copies. That's the structure in the third example in the clause box above. If the license for one of these "client-server" systems allows 60 "concurrent users," the customer may allow 60 users at one time. The software could be physically available to hundreds of users and client computers, but it's restricted to 60 *at a time*.

A client-server license might instead allow a fixed number of “seats,” as in the fourth example in the clause box above. If the license authorizes 15 seats, it generally means 15 designated users can access the software, and no others. Jane Employee can’t access the software unless she’s one of those 15, even if fewer than 15 are accessing the software at any given time. (If Jane’s in the in-group, she probably has one of the 15 user IDs and passwords.) But in some cases, “seats” refers to a number of designated client computers, rather than individuals. Then, only those 15 client computers can access the software. Consider defining the term “seats” to avoid any doubt.

All four examples in the previous clause box grant license rights “provided” the customer complies with certain restrictions. Subchapter 2 addresses those restrictions.[4]

2. *End User Restrictions*

End user licenses generally limit the customer’s rights in various ways.

Standard End User Restrictions

Copies of the Software created or transferred pursuant to this Agreement are licensed, not sold, and Customer receives no title to or ownership of any copy or of the Software itself. Furthermore, Customer receives no rights to the Software other than those specifically granted in this Section __. Without limiting the generality of the foregoing, Customer shall not: (a) modify, create derivative works from, distribute, publicly display, publicly perform, or sublicense the Software; (b) use the Software for service bureau or time-sharing purposes or in any other way allow third parties to exploit the Software; or (c) reverse engineer, decompile, disassemble, or otherwise attempt to derive any of the Software’s source code.

Every license should confirm that the customer receives only the rights specifically granted and that the vendor retains ownership of the software. Vendors should also consider stating that individual copies of the software are “licensed,” not “sold.” In other words, the software isn’t like a book you’ve bought, which you can give away or sell. Rather, it’s like music you’ve downloaded, which you’re not allowed to pass around. See the first two sentences of the clause box above.[5]

An end user license should also list certain rights *not* granted. Copyright law grants several exclusive rights to copyright owners. Vendors should make sure the license doesn’t grant any of those except the right to reproduce (along with the right to use, which isn't actually mentioned in the copyright statute). That's why the example in the clause box provides that the customer can't exercise the other rights of copyright holders. It can't distribute, modify (create derivative works), or publicly display or perform the software. The customer also can't sublicense its rights to anyone else. Of course, if the clause is silent on restrictions, a court may consider the license limited to the rights specifically granted. But why take chances?

The vendor should clarify that the customer gets no time-sharing or service bureau rights, or any other rights to share the software with third parties. *Time-sharing* means sharing an application with customers or other third parties—letting them use the software too. *Service bureau* usage involves another type of sharing: the customer keeps the software, but it uses it to process third party data, instead of its own data, including in a cloud services offering. Either could cost the vendor sales.

The vendor should also be sure the license forbids reverse engineering and any other attempt to derive source code from the software.

Finally, the vendor should be sure the customer stops reproducing and using the software when the agreement terminates. The clause box above doesn't provide that requirement because this book addresses it in Subchapter II.V.4 ("Effects of Termination").

B. Standard Distributor Software License

This chapter addresses licenses to distribute software. In these clauses, the vendor authorizes a distributor to transfer copyrighted software to third parties—to end user customers. This chapter, therefore, talks about "distributors" rather than "customers." But to avoid throwing too many terms around, it sticks to "vendors" for the software's ultimate provider.

Software licenses are copyright licenses. But like Chapter I.A ("Standard End User Software License"), this chapter doesn't go far into the mechanics of copyright licensing. That kind of knowledge isn't usually necessary for a standard distributor license. But if you want a deeper understanding of licensing, or if your license doesn't fit the "standard" model discussed here, see Chapter I.C ("Software Licenses in General").

Before turning to the terms of a distributor license, ask: *What* is being licensed? The contract should clearly define the "Software" or "Licensed Product"—usually in a separate definitions section. It's usually enough to give the software's name and version number, and specify object code: "'Software' refers to Vendor's *Cookie-Cruncher* software application, version 6.02, in object code format." But if the software has multiple modules or libraries or whatever, and you see any chance of dispute about what's included, list the necessary elements. "'Licensed Product' refers to Vendor's *Pimp-My-Photo* software application, version 4.05, in object code format, including the following modules: Wardrobe Upgrade, Body Maximizer, and VirtualNoseJob." You might also want to specify the platform: Windows, Macintosh, Linux, etc. Finally, if the distributor needs to distribute user manuals or other documentation, the definition might include them: "The Licensed Product includes Vendor's standard user guides and other documentation for such software."[6]

1. Distribution

Not surprisingly, a distribution license grants the right to distribute the software—to pass it around.

Standard Distribution Rights

Provided Distributor complies with the restrictions set forth in Section __ (*Software Restrictions*) below, Vendor grants Distributor: (a) an exclusive license to distribute the Licensed Product within __________ (the "Territory"); and (b) a nonexclusive license to reproduce and use the Licensed Product within the Territory, solely as necessary to market it and to provide technical support to customers.

• • • •

Provided Distributor complies with the restrictions set forth in Section __ (*Software Restrictions*), Vendor hereby grants Distributor a nonexclusive, worldwide license to exploit the Software as follows, solely as an embedded component of Distributor's Product: (a) to distribute the Software; (b) to reproduce and use the Software for sales and marketing purposes and to the extent necessary to provide technical support to customers of Distributor's Product; and (c) to sublicense to its customers the right to reproduce and use the Software. Distributor may sublicense to its sub-distributors the rights granted in Subsections __(a) through __(c) above.

• • • •

Distributor may solicit sales of the Software within ________ (the "Territory"), and Vendor shall pay Distributor the commission set forth in Section __ (*Payment*) for all Software sales within Territory, whether or not initiated or closed by Distributor. Provided Distributor complies with the restrictions set forth in this Section __, Distributor may use and reproduce the Software to the extent reasonably necessary to market it as authorized in the preceding sentence.

Distribution rights are often restricted to a geographic territory (e.g., state, region, country, continent). The distributor has no right to distribute outside that area. The territory might also be defined by industry. For instance: "Vendor hereby grants Distributor the exclusive right to distribute the Software for use in Semiconductor Fabrication (as defined in Section __)." If you use an industrial territory, be sure to define the industry or segment clearly. For both types of territories, see the first and last examples in the clause box above.

The right to distribute may be exclusive or nonexclusive. If it's exclusive, no one, not even the vendor itself, has the right to distribute within the territory—or anywhere if the license is worldwide. See the first example in the clause box above. (Exclusive distribution creates some risks for the vendor. See Subchapter 3 below.)

Some distribution licenses include limited rights to reproduce and use the software, as in all three examples in the clause box above. These rights help with marketing and technical support. Technically speaking, the clause should grant those rights. But sometimes it's fair to assume a distributor has reasonable marketing and support rights, even if they're not spelled out.

Value-added reseller (VAR) licenses grant limited distribution rights. See the second example in the clause box above. The distributor can only give the software out as a component of some larger tool—often something the distributor itself produces. Imagine the vendor makes databases and the distributor makes factory-management applications, which use databases. A VAR license lets the distributor distribute the vendor's database *with* the distributor's application, giving end user customers a complete package. But the distributor can't distribute the database as a stand-alone product.

Original equipment manufacturer (OEM) licenses grant the same basic rights as VAR licenses. Technically speaking, in an OEM license, the distributor's product is always equipment—hardware—while a VAR license may involve hardware or software. But to many IT professionals, the terms are interchangeable.

Some distribution licenses let the distributor sublicense its rights to its sub-distributors, as in the second example in the clause box above. Vendors should make sure the contract's payment clause requires royalties or other payments, whether it's a distributor or sub-distributor that makes the sale. Some clauses also let the distributor sublicense certain rights to customers. In the second example in the clause box above, the distributor can grant its customers the right to reproduce and use the software. Few distribution licenses actually specify these rights, though. Most vendors and distributors assume that sublicensing rights are implied in the right to distribute.

Technically speaking, the third example in the clause box above is a sales representation clause, rather than a distribution clause. (And the example's first sentence is just a promise, not a copyright license.) The distributor markets the software in the territory but doesn't *distribute* it. The vendor signs contracts with the distributor's customers and distributes the software to them. The example also effectively gives the distributor exclusive rights to the territory. That doesn't necessarily mean the distributor is the only one marketing the software there. The vendor can market too, but the distributor gets a commission on every sale in the territory.

All three examples in the clause box grant license rights "provided" the distributor complies with certain restrictions. Subchapter 2 addresses those restrictions.[7]

2. Distributor Restrictions

License clauses usually restrict distributors in several ways.

Standard Distributor Restrictions

This Agreement grants Distributor no title to or ownership of the Software, and Distributor receives no rights to the Software other than those specifically granted in this Section __. Without limiting the generality of the foregoing, Distributor shall not reverse engineer, decompile, disassemble, or otherwise attempt to derive any of the Software's source code. Distributor may license copies of the Software to its customers but may not sell such copies, and neither Distributor nor its customers will receive title to or ownership of any copy or of the Software itself. Distributor will not distribute copies of the Software to (a) any sub-distributor that does not first execute a written contract with limits on Software rights no less restrictive than those set forth in this Section__; or (b) any customer or other third party that does not first execute a written end user license agreement in the form attached to this Agreement as Attachment __ (*EULA*).

The vendor should clarify that it still owns the software and that the distributor receives only the rights specifically granted. And usually the vendor should forbid reverse engineering and any other attempt to derive source code from the software. See the example in the clause box above.

The vendor has another party to worry about, besides the distributor. What will the distributor's *customer* do with the software? That's why the license clause should require that the distributor have its customers sign agreements that restrict software use. Often, the vendor and distributor draft a full-length end user license agreement (EULA), as in the example above. But the parties might instead draft a set of minimum standards for the EULA. "Distributor shall not distribute copies of the Software to any third party that does not first execute a written end user license agreement ("EULA") that: (a) forbids distribution of the Software, service bureau or time-sharing use of the Software, or other exploitation of the Software, except internal use and reproduction to the extent specifically authorized by such EULA; (b) restricts use of the Software to the same extent as, or more than, Section __ (*Software Restrictions*) of this Agreement; (c) provides for Software audits, with terms no less restrictive than those of Section __ (*Software Audit*); (d) requires that such third party cease using and delete all copies of the Software after termination of such EULA; and (e) provides that Vendor may enforce the EULA as an intended third party beneficiary."[8]

For more on customer (end user) restrictions and contracts, see Subchapter I.A.2 (“End User Restrictions”).

The clause should also require that the distributor’s customers receive *license rights* to their copies of the software, not ownership of those copies. In other words, the software isn’t like a book the customer buys, which it can give away or resell. Rather, it’s like downloaded music, which the customer isn’t allowed to pass around.[9] Again, see the example in the clause box above.

In a sales representation deal, you don’t need all the restrictions in the clause box above. The last two sentences—on licensing of copies and customer contracts—aren’t necessary because the vendor, not the distributor, handles all the licensing and contracts directly.

Finally, the vendor should be sure the distributor stops distributing the software when the agreement terminates. This book addresses that issue in Subchapter II.V.4 (“Effects of Termination”).

3. Minimum Obligations to Distribute

What if the vendor grants an exclusive distribution license and the distributor doesn’t bother to market or sell? The software has been “shelved”—taken off the market. That’s particularly disastrous for the vendor if its compensation comes from royalties—from a portion of the distributor’s sales. Twenty percent of zero is zero.

> ***Minimum Obligations to Distribute***
>
> Distributor shall exercise commercially reasonable efforts to market and sell the Software. Without limiting the generality of the foregoing, if Distributor fails to achieve gross revenues of $_____ from Software distribution during any calendar year, Vendor may terminate this Agreement by written notice to Distributor.

The example in the clause box above requires that the distributor *try* to sell the software. But “commercially reasonable efforts”—the legal version of “try”—isn’t very clear. And the commercially reasonable requirement doesn’t help the vendor if the distributor has a lousy sales team. That’s why the clause box also provides that the distributor can lose the agreement if it fails to hit a certain sales figure. As an alternative, the clause could revoke the exclusivity provision, rather than the whole contract, if the distributor misses its numbers.

Another alternative: instead of “commercially reasonable efforts” to distribute, the clause could require “best efforts.” “Best efforts” calls for a standard higher than “commercially reasonable”—something like “try as hard as you can”—but it’s not clear either.

C. Software Licenses in General

Chapter I.A addresses standard end user software licenses, while Chapter I.B addresses standard distribution licenses. This chapter provides a more thorough review of software licensing—of copyright licensing. It provides concepts you can use to customize clauses that don't fit those two standard models. But the previous two chapters do explain some key concepts. So before reviewing this chapter, read I.A if you're working on an end user license, and I.B for a distributor license.

This chapter addresses two issues. Subchapter 1 asks: *What license rights does the vendor grant?* And Subchapter 2 asks: *What is the scope of the license?* Are the rights exclusive, temporary, restricted, etc.—and if so, how? Software licensing is a game of mix and match. You list the customer's rights and then match them with the appropriate scope terms.

Subchapter 3 uses all these lessons to show you an "unrestricted license." An unrestricted license grants the customer as many rights as possible without actually transferring ownership. Customers who pay for software development sometimes want this sort of license.

Always start by clearly defining the "Software" or "Licensed Product." For guidance on these definitions, see Chapter I.A for end user licenses and I.B for distributor licenses. Also, note that in technology development contracts, it's often impossible to identify all the software at the time the contract is drafted. There, the definition should read something like: "'Licensed Product' refers to all software to be created pursuant to this Agreement." (Unlike the examples in Chapters I.A and I.B, development contracts also sometimes include source code in the definition.)

1. Copyright License Rights

Software licenses are copyright licenses. Under U.S. federal law, the copyright owner has certain exclusive rights. In a license, the owner grants the customer some of those rights.[10]

The license should list the rights granted. And for the vendor’s sake, the license should also list the rights that are *not* granted, to make sure there’s no confusion. Finally, most licenses should confirm that the customer receives no *ownership* interest.

Often, the most important terms of a software license are the scope terms, addressed in Subchapter 2. But for now, let’s look at rights without scope terms, or with very limited scope terms.

Copyright Licenses (with Limited or No Scope Terms)

Vendor grants Customer a license: (a) to use the Software and to publicly perform it on the worldwide web, (b) to reproduce the Software to the extent reasonably necessary for such purposes, and (c) to sublicense the rights granted in Subsections __(a) and __(b) above to _________ (“Sublicensee”); provided Customer complies with the restrictions set forth in Subsection __ (*Restriction on Software Use*), and provided Customer does not modify the Software, distribute the Software to any third party other than Sublicensee, publicly display the Software, or attempt to exercise any copyright holder’s rights not specifically granted in this Section __. Copies of the Software created or transferred pursuant to this Agreement are licensed, not sold, and Customer receives no title to or ownership of any copy or of the Software itself.

• • • •

Provided Distributor complies with the restrictions set forth below in Section __ (*License Restrictions*), Vendor hereby grants Distributor a license: (a) to modify the Software as authorized in Attachment __ (*Specifications for Derivative Works*); (b) to reproduce the resulting derivative work (the “Derivative Work”); (c) to distribute the Derivative Work; (d) to reproduce and use the Derivative Work as reasonably necessary for marketing purposes; (e) to sublicense to its customers the right to reproduce the Derivative Work; and (f) to sublicense to its sub-distributors the rights granted in Subsections __(b) through __(e) above. Vendor retains full title to and ownership of the Software.

• • • •

Vendor grants Customer a license to use the Software, provided Customer complies with the restrictions set forth in this Section __. Customer may not distribute, modify, publicly perform, or publicly display the Software, and may not reproduce the software except as necessary to install it and to create one backup copy. Copies of the Software created or transferred pursuant to this Agreement are licensed, not sold, and Customer receives no title to or ownership of any copy or of the Software itself.

The following bullet points list the copyright license rights:

- ***Reproduce:*** The right to make copies. The license clause may authorize one copy, a thousand, or any number, including "such copies as are necessary for Customer's business operations" (an *enterprise license*). See the first two examples in the clause box above.
- ***Modify* or *Create Derivative Works:*** The right to change a copyrighted work, creating a new version. See the second example in the clause box above. (The customer—or whoever wrote the modifications—automatically owns the new code. But the vendor owns part of the derivative work too: the original software.)
- ***Distribute:*** The right to hand out copies, for payment or for free. This right is necessary for software distributors, including resellers. See the second example in the clause box above.
- ***Publicly Perform:*** The right to perform a copyrighted work. Public performance only applies to works that *can* be performed, like music, movies, and other audiovisual works. So for most software, this right doesn't apply, and there's no reason to include it in the agreement. But some software does dish up audio or visual content, so the public performance right makes some sense. See the first example in the clause box above.

  Arguably, public performance rights are implied by the right to reproduce or use software with audio or visual content, even if not specified. If the customer can't publicly perform software that plays movies on a website, what's the point? But there's no harm in clarity on this point, particularly if you're the customer.
- ***Publicly Display:*** The right to show copies to the public. This looks a lot like the public performance right discussed above, but it applies mostly to still images, like the individual photo frames of a movie. Similar to public performance, public display only applies to works that *can* be displayed, like photos, diagrams, and drawings. So for most software, this right doesn't apply, and there's no reason to include it. Of course, some software does provide these images. Even there, though, a grant of public display rights might cause more trouble than it's worth; it could be interpreted to grant rights to display the code itself, maybe even source code. And as with public performance, public display is usually implied by the right to *use* software that includes still images. So public display usually has no role in software contracts. The first

example in the clause box mentions public display, but only in the list of rights *not* granted.

- ***Use and Other Pseudo Rights:*** Some contracts grant the right to "use" software, or "run" it, or words to that effect. See all three examples in the clause box above. Those terms authorize the customer to *operate* software. This book calls these "pseudo rights" because they're not listed in the copyright statute, unlike the rights discussed above. Technically speaking, the pseudo rights aren't necessary. If the customer already has a copy of the software, or a license to reproduce, it doesn't need an additional grant of rights to use or run the software, because those aren't monopoly rights of copyright holders. (After all, you don't need a license to "use" a copyrighted book: to read it.) Still, there are some legal advantages to including pseudo rights, particularly for vendors.[11] And rights to "use" software have become so common that many of your contracting partners will insist on them.

  All the pseudo rights imply the right to make one copy of the software. If the customer can "use" or "install," it has to make a copy for its computer.[12] The problem with pseudo rights is that it's hard to tell what else they mean, since we don't have a statute to define them. If the customer can "install," can it also modify the software, if that's necessary to make the installation effective? Does "use" simply mean the customer can reproduce one copy and operate it, or does it also include rights to publicly perform? The answer will depend on the context—on whether the pseudo right implies other rights, as a result of the software's nature. Because of that uncertainty, both parties should handle pseudo rights with care. If you're the vendor, make sure to list the statutory license rights that are not granted, as in the third example in the clause box above. That way, you confirm that "use" or "install" or whatever does not include any rights you didn't intend. And if you're the customer, make sure the license specifically grants the other rights you need: the statutory rights to reproduce, distribute, etc. "Customer may use, reproduce, and modify ten copies of the Software." Don't assume ill-defined pseudo rights will cover all your needs.
- ***Sublicense:*** The right to pass license rights on to third parties. The license clause could authorize the customer to take one or more of the rights granted and pass them on to its own customers or distributors or whomever. See the first and second examples in the clause box above.

> The right to sublicense is sometimes confused with the right to distribute. Distribution rights allow the customer to hand out copies, not to transfer rights. However, sublicensing rights are sometimes implied by distribution rights, when the distributor gives its customers the right to reproduce the software. So the two terms can overlap.

Software vendors should consider addressing three other issues in their license rights clauses.

First, vendors should consider stating that the individual copies of the software are "licensed," not "sold." See the first and third examples in the clause box on pages 16–17. In other words, the software isn't like a book someone's bought, which he can give away or sell. Rather, it's like music he's downloaded, which he's not allowed to pass around.[13] Of course, if the vendor doesn't care what happens to the authorized copies—so long as the customer doesn't make new ones—or if the customer is *supposed* to pass the copy around, the language isn't necessary.

Second, vendors should be sure their customers or distributors stop reproducing, distributing, using, and otherwise exploiting the software when the agreement terminates. This book addresses that issue in Subchapter II.V.4 ("Effects of Termination").

Third, vendors should consider granting provisional rights, as in all three examples in the clause box on pages 16–17. The examples grant license rights "provided" the customer or distributor complies with certain restrictions. The restrictions in question are common license restrictions, like the prohibitions against reverse engineering listed in the clause box, as well as any restrictions included among the scope terms discussed in Subchapter 2. The point of the "provided" language is to help the vendor *enforce* those restrictions. The language helps establish that the restrictions are conditions on the license, as opposed to separate contract promises (aka "covenants"). If the restrictions qualify as license conditions, the customer (or distributor) loses its license rights if it violates the conditions. The customer also subjects itself to copyright infringement damages. On the other hand, if the restrictions are separate contract promises, the customer may not lose its license rights and is only liable for standard contract damages, which may be less effective than copyright infringement damages.

Unfortunately, the law hasn't clearly defined the type of restrictions that can be considered license conditions. Obviously, it's in the vendor's interest to tie any and all contract provisions to a software license, so it can cancel the license and get copyright damages if the customer breaches. But I doubt the courts will consider any old restriction a license condition, just because it follows a license clause and the word "provided." Until the law gets clearer, I recommend that you use the "provided" proviso for restrictions closely related to the right to exploit the software. For instance, restrictions on the number of concurrent users who can access software probably qualify as license conditions. The same goes for restrictions on reverse engineering and on distribution outside a given territory. But a clause saying the customer will make payments by the first of each month probably doesn't qualify. Nor does a promise to keep the vendor's business plans confidential. Those restrictions have little to do with exploitation of the software, so I doubt courts will treat them as license restrictions, even if attached to a "provided" proviso.

2. Scope Terms

Scope terms add extra detail to a copyright license, once the copyright holder's rights have been granted. They're limited only by your imagination. Once you've granted rights to the software, you can restrict or define those rights in almost any way, or not at all.

Copyright License with Scope Terms

Vendor grants Distributor an exclusive license to distribute the Licensed Product in __________ (the "Territory").

• • • •

Vendor authorizes Reseller to provide Software access to Reseller's end customers as a software-as-a-service offering, and Vendor hereby grants Reseller a license to reproduce the software as necessary to support such use, provided: (a) Reseller shall deploy the Software to no more than _____ seats; (b) Reseller shall not use the Software to process more than ____ Transactions per day; and (c) Reseller shall comply with the other restrictions on use of the Software set forth in this Section __.

• • • •

Vendor grants Customer a nonexclusive license, for __ years from the Effective Date: (a) to install the Software on computers owned and operated by Customer, fitting the description on Attachment __ (*Platform Specifications*); and (b) to reproduce the Software as necessary to exercise its rights in Subsection __ (a) above and to create a reasonable number of backup copies.

• • • •

Provided Customer complies with the license restrictions set forth in this Section __, Vendor hereby grants Customer a perpetual, irrevocable, worldwide, nontransferable, nonexclusive, fully paid, royalty-free license to reproduce, use, and modify the Licensed Software.

The following bullet points list the most typical scope terms:

•***Exclusivity:*** License rights may be exclusive to the customer. The vendor is promising not to grant the same rights to anyone else, or to exercise those rights itself. So in the first example in the clause box above, the right to distribute in the territory is exclusive to the distributor. Or the license clause might do the opposite. The customer might receive a "nonexclusive" right to distribute, reproduce, etc., as in the third and last examples in the clause box above.[14]

As noted in Subchapter I.B.3 ("Minimum Obligations to Distribute"), an exclusive right to distribute software can blow up in the vendor's face if the customer/distributor has no clear obligation to market the software.

•***Territory:*** This scope term restricts license rights to certain areas—usually geographic, but sometimes defined by industry. In the first example in the clause box above, you'll have a geographic territory if you fill in the blank with "the States of Oregon and Washington." The following defines territory by industry: "Distributor may distribute the Licensed Product within the Dental Office Equipment Market."

On the other hand, if you want to clarify that there are no territorial restrictions, the license should be "worldwide," as in the last example in the clause box above. (The term "worldwide" isn't strictly necessary because the law will usually assume a license has no territorial limits if it doesn't list any, but it helps to be clear.)

•***Duration:*** Unless the contract provides otherwise, license rights last as long as the term of the agreement. But a license can last longer than the underlying contract, or it can end earlier. The end of a license can be pegged to a calendar date, as in the third example in the clause box above, or to a date to be determined. Here's a date to be determined: "The license rights granted in this Section __ will continue so long as Customer is a party to a services contract with Vendor's subsidiary, Neeto-Service, Inc." Another option is a "perpetual" license, as in the last example in the clause box. With a perpetual license, the underlying contract may terminate—ending the customer's payment obligations and most other promises—but the license rights last forever, unless they're revoked.

The termination clause should confirm that "perpetual" really means "survives termination," to remove any doubt. See Subchapter II.V.4 ("Effects of Termination").

- ***Revocability:*** If a license is “irrevocable,” the vendor can’t take it away, even if the customer breaches the contract. See the last example in the clause box above. If the customer doesn’t pay, the vendor’s only remedy is to sue for the money. The vendor has given up any right to a court order forcing the customer to stop using the software. At least, that’s the generally understood meaning of “irrevocable.” But courts’ interpretations vary. Customers can increase their chances of a broad interpretation by clarifying: “Vendor’s remedies for breach, including without limitation breach of Customer’s payment obligations, may include monetary damages, but Vendor hereby waives any right to termination of the license granted in this Section __.”[15] Vendors, on the other hand, can protect themselves with terms that delay irrevocability until payment: “The license granted in this Section __ will become irrevocable upon Customer’s payment of the License Fee.” (An irrevocable license is not necessarily perpetual. The vendor can’t revoke the rights, but the contract might still specify a natural expiration date.)
- ***Payment Scope:*** A “royalty-free” license, like the last example in the clause box above, requires no royalty payments. That doesn’t necessarily mean the customer gets the license for free. The contract might call for a fixed payment or for payment under some other scheme (usually appearing in the payment clause, not the license clause). The point is that the customer doesn’t have to pay more every time it makes a copy or otherwise exercises its license rights. If the license is “fully paid”—also like the last example in the clause box—the parties agree that whatever payments are required, if any, have already been made as of the moment the license is granted. If the license is effective “upon Customer’s payment of the License Fee,” for instance, and there are no royalties, the license is “fully paid” when granted. (When used together, as in the last example, the two terms provide overlapping protection for customers.)
- ***Transferability:*** A license clause can provide that the rights granted are “nontransferable,” as in the last example in the clause box above. In other words, the customer can’t give the license to anyone else. Or the clause can say that the license *is* “transferable.” Note that sometimes these terms are unnecessary. Most contracts have an assignment clause governing the transfer of rights. If the assignment clause says the contract is or is not transferable (“assignable”), there’s no need to repeat

it in the license clause.[16] (A transfer is not the same as a sublicense. In a transfer or assignment, the customer/licensee gives away all its license rights—the whole contract—and is left with none. In a sublicense, the customer/licensee keeps its rights, but authorizes a third party to exercise them.)

•***Restrictions on Use: Internal Use, Seats, Cloud Access, Transactions, Installations, etc.:*** The vendor can limit the customer to certain uses of the software. (As noted in Subchapter 1 under *Use and Other Pseudo Rights*, some licenses grant permission to "use" software or "install" it or whatever, in the grant of rights. Here, we're talking about a different application of the word "use": as a scope term *restricting* the customer's rights to the software, rather than a license term granting rights. It's definitely confusing.)

Vendors often restrict customers to "internal use." But the license clause can do just the opposite, authorizing "service bureau use": "Vendor grants Customer a license to use the Software for 'service bureau' processing of third party data and to reproduce the software as necessary to support such use." Or the clause might go even further, authorizing the customer—truly a reseller in this case—to give its own customers remote access to the software, providing software-as-a-service or other cloud services, as in the second example in the clause box on pages 21–22.[17]

Many vendors also limit customers to a fixed number of seats, concurrent users, or transactions.[18] The second example in the clause box on pages 21–22 limits both seats and transactions. The following limits concurrent users: "Customer may reproduce two copies of the Software, provided no more than 25 concurrent users may access either copy."

Many vendors require that software be used (and installed) only on certain computers or on a certain number or type of computers. The third example in the clause box on pages 21–22 specifies the acceptable platforms, or computers. In that description, you might include any sort of restriction on platforms, including computer make, model, operating system, and number of processors or "cores," as well as limits on enhanced computing power resulting from virtualization.

If your deal requires restrictions on use, don't hesitate to get creative about drafting them. For instance, the following would be perfectly

> legitimate: “Customer agrees not to use the Software on the first Monday of any calendar month, not to install the Software on any computer used to process pet food inventories, and not to permit access to the Software by anyone other than a podiatrist certified to practice in the State of Maryland.”

Scope terms may apply to all the license rights granted or only to some. For instance, the right to reproduce software may be perpetual, while the right to distribute lasts only one year. Or the right to distribute may be exclusive within a particular territory and nonexclusive outside the territory.

Scope terms should appear in the license clause itself. That makes the contract easier to understand. But you may run into contracts with scope terms spread around.

3. Unrestricted License

An unrestricted license throws in the kitchen sink. The vendor grants all the rights of copyright holders, with a broad scope. The license authorizes the customer to do just about anything with the software, but the vendor still owns it and can grant licenses to third parties.

Customers sometimes want these super-broad licenses when they pay for software development. The customer is paying the vendor to create the software and the project’s other “deliverables,” so it wants a more or less unlimited right to exploit those deliverables. Of course, the contract could simply give the customer ownership of the deliverables—of the copyright. But vendors often charge more for ownership because it keeps them from selling the deliverables to third parties, on future projects.

Unrestricted Copyright License

Vendor hereby grants Customer a nonexclusive, perpetual, irrevocable, worldwide, transferable, fully paid, royalty-free, license: (a) to reproduce, modify, distribute, publicly perform, publicly display, and use the Deliverables, in each case without any restrictions; and (b) to sublicense any or all such rights to third parties.

It’s actually possible to draft a broader license than the example in the clause box above. The rights granted could be exclusive. But then the vendor would have no copyright left. The transaction would work like an assignment—a transfer of copyright ownership to the customer—despite the word “license.”[19]

The example in the clause box doesn’t include the “provided” language discussed in Subchapter 1—the language saying the license is conditional on the customer’s compliance with the contract’s various restrictions. Many vendors granting super-broad licenses don’t care about ensuring copyright remedies (which is what’s at stake in these “provided” provisos). But if the vendor does care, it should add the language.

D. Technology Ownership: Assignment and Work-for-Hire

An ownership clause provides that the customer will *own* software from the vendor, particularly intellectual property rights in software. It may give the customer other assets too, like user manuals, schematics, designs, and logos.

Before executing an ownership contract, the customer should ask itself why it wants to *own* the software or whatever else the vendor is creating. In software development agreements, the customer often argues: "We're paying to have it made, so we own it." But if the customer only plans to *use* the system—if it doesn't sell software—it may get little value from ownership. A license could provide all the necessary rights, particularly if it's an "unrestricted license," like the one in Subchapter I.C.3 ("Unrestricted Copyright License"). And the vendor might charge less for a license because if it keeps ownership, it can provide the software to other customers.[20]

Ownership clauses are complicated, and a lot can go wrong, particularly for the customer. Among other issues, the customer should satisfy itself that the vendor actually has or will have the rights it's transferring. An IP warranty makes that promise, so customers should consider one. Both parties, of course, should always consider experienced legal help when drafting IT contracts, but all the more so for the clauses in this chapter.[21]

This chapter explains the key types of ownership transfers—work product clauses and transfers of existing assets—in Subchapters 1 and 2. It also addresses, in Subchapter 3, an issue related to both: getting IP rights from the vendor's employees. But first, this chapter explains the legal vehicles used in all ownership clauses: assignments and work-for-hire provisions.

The text here talks about software ownership, but assignment and work-for-hire work the same for books, website content, music, paintings, and other works of authorship.

• • • •

An assignment clause transfers ownership from the vendor to the customer. “Vendor hereby assigns to Customer all its right, title, and interest in and to the Software.”[22]

A work-for-hire clause, on the other hand, doesn’t transfer ownership because the vendor isn’t the owner and never was. Under a work-for-hire clause, the customer owns the software from the moment it’s written, even though the vendor writes it. In other words, work-for-hire reverses the law’s usual assumption: that the author of a writing owns it.

Work-for-hire applies to copyright only, not to patents or other forms of IP. Some contracts with work-for-hire clauses, however, also include assignment clauses that transfer patent and other rights.

You can’t create a work-for-hire relationship just by writing it into a contract. Work-for-hire status applies to two fact patterns. In the first, the “vendor” is actually the customer’s employee and writes the software within the scope of his or her duties. In the second, the vendor is an independent contractor and the software or other assets fit into one of copyright law’s nine “eligible” categories, listed below, and the parties agree in writing on work-for-hire status.

The first fact pattern is simple, so long as there can be no doubt about the software author’s employment status. If an employee wrote the software, and his or her job involves the software programming in question, the deal passes the test. In fact, a written contract isn’t necessary, since generally the law will consider the employment a work-for-hire relationship. But for the employer, it’s better to avoid doubt by including a work-for-hire clause in the employment contract, or if that agreement isn’t in writing, by signing an IP ownership contract. Also, the employee might create assets that aren’t subject to work-for-hire treatment—patentable inventions, for instance—and a written contract would address them too. See Sub-chapter 1 below.

It’s often hard to tell whether a relationship qualifies as “employment.” What if the software author serves part-time or works from home with no health benefits? Unfortunately, the law doesn’t provide a clear test. Courts consider a long list of factors in determining employment status. So if you want to use work-for-hire and you’re not absolutely sure the author is an employee, consider experienced legal help. You should also back up your work-for-hire clause with an assignment, as explained in Subchapter 1 below.

The second fact pattern allows work-for-hire treatment through a written contract, even if the person doing the programming isn't an employee working in the scope of his or her employment.[23] "The Software will be considered works made for hire pursuant to the U.S. Copyright Act, 17 U.S.C. Sections 101 *et seq.*, and will be Customer's sole property." But that clause only works if the deal fits one of copyright law's nine eligible categories. The customer has to order the software "as a contribution to a collective work, as a part of a motion picture or other audiovisual work, as a translation, as a supplementary work, as a compilation, as an instructional text, as a test, as answer material for a test, or as an atlas."[24] Weirdly random, huh? Legal scholars debate the extent to which software can fit these nine categories. One thing is clear: the categories are complicated, so again, if in doubt, seek experienced legal help. And again, consider backing up your work-for-hire clause with an assignment, as explained in Subchapter 1.

For customers, work-for-hire is better than assignment because it can't be revoked. Revocation of assignments is rare, but it's possible. And actually *all* copyright assignments and licenses can be revoked after 35 years, no matter what the contract says.[25] But of course, work-for-hire treatment isn't always available.

1. Ownership of Work Product

This subchapter addresses ownership of "work product": software or other assets created through professional services relationships, research and development agreements, and similar collaborations—and also through employment relationships. In most cases, the work product doesn't yet exist when the parties execute the contract.[26]

If you're the customer, you should consider a work product ownership clause when you engage a software programmer (either an individual or a company), assuming you want to own the software created through the relationship. But technology creation isn't restricted to programmers. Almost any professional services vendor could create IT-related assets. For instance, a tech support contractor might write a set of instructions for handling tech support calls—or even come up with an improvement to the customer's product. So customers should consider work product clauses in all their IT-related professional services contracts. They should also add them to their staff's employment agreements.

Some work product clauses give the customer copyright ownership but not patents or other forms of IP. They also tend to leave "preexisting assets"—software created before the deal—with the vendor. These more limited clauses often appear in professional services agreements where the vendor is a company, rather than an individual—and they rarely appear in employment agreements. Some clauses, on the other hand, give the customer all imaginable rights to the work product, including patents. They often direct the vendor *not* to include any preexisting assets in the work product. Those broader work product ownership clauses most often show up in contracts between the customer and its employees or individual contractors—human contractors, rather than companies, who join the staff and act like employees.

Let's start with the more limited type of ownership clause.

Ownership of Work Product: Limited Transfer

Effective upon full payment pursuant to Section __ (*Fees*), and subject to Subsection __(a) below (*Preexisting Assets*), Vendor hereby assigns to Customer all Vendor's ownership, right, title, and interest in and to any and all copyrights in the software and other assets listed on Attachment __ (*Deliverables*) or otherwise created pursuant to this Agreement (collectively, the "Work Product").

(a)Preexisting Assets. The assignment in the preceding sentence does not include any component of the Work Product created before the Effective Date (any "Preexisting Asset").

(b)License. To the extent that this Section __ does not provide Customer with full ownership, right, title, and interest in and to the Work Product, including without limitation Preexisting Assets, Vendor hereby grants Customer a perpetual, irrevocable, fully paid, royalty-free, worldwide license to reproduce, create derivative works from, distribute, publicly display, publicly perform, and use the Work Product, with the right to sublicense each and every such right; provided Customer may not: (i) reproduce or use Preexisting Assets other than as components of the Work Product, (ii) distribute Preexisting Assets, or (iii) sublicense any rights in Preexisting Assets to third parties other than in support of Customer's internal business operations.

(c)Further Assistance & Survival. Vendor shall reasonably assist Customer in obtaining and enforcing copyrights in the Work Product, at Customer's expense. The rights granted in this Section __ will survive any termination or expiration of this Agreement or of Vendor's engagement with Customer.

The clause box above transfers copyright ownership upon payment, which is obviously best for the vendor. But some clauses transfer ownership immediately ("as of the Effective Date"), even for software to be created in the future. Either way, customers should be sure the clause says the vendor "hereby" assigns ownership, as in the clause box, not that the vendor "shall assign." That latter isn't an assignment but rather a promise to execute an assignment document in the future. If the vendor breaches that promise, or goes bankrupt before signing the future document, the customer won't own the work product; it'll just own the (possibly worthless) right to sue the vendor for breach of contract.

Notice that the example in the clause box above doesn't include a work-for-hire provision. It could, and you could add the language in this subchapter's next clause box, on page 35. But work-for-hire doesn't fit professional services provided by a company, as opposed to work by an employee or by an individual human vendor who operates like an employee.

The example in the clause box above transfers copyrights but not patents. Most software isn't patented, so this rarely matters. But in the rare case where the vendor invents something patentable while working on the project, the right to file that patent stays with the vendor. If the vendor gets a patent, it'll have the right to enforce it—to keep anyone from writing new software based on the invention. But that won't restrict the customer's rights to the original software, thanks to its ownership of the copyright and thanks to the license in Subsection (b), which is effectively a patent license (among other things). The customer can use the work product, modify it, resell it, and keep third parties from using it. But the customer *can't* keep third parties from writing *new* software based on the invention covered by the vendor's patent. That right belongs to the vendor.[27]

Subsection (a) in the clause box says the vendor keeps "preexisting assets." In other words, the vendor doesn't have to give up software or other assets from its own toolkit added to the work product. But the vendor does grant the customer a broad, permanent right to *use* preexisting assets with the rest of the work product, in Subsection (b). To make sure the customer doesn't use the preexisting assets to compete with the vendor, the license in Subsection (b) is limited: the customer can use the preexisting assets only as components of the work product and only for the customer's own internal business purposes. If the customer wants to go into business selling the work product, it'll have to replace the preexisting assets.

Some jurisdictions, particularly certain foreign ones, bend over backwards to protect IP creators from their own contracts and don't always enforce assignments. So a good assignment clause includes a royalty-free backup license, as in Subsection (b) of the clause box above. (Subsection (b) plays other roles too, as we've seen.) The license grants the customer full rights to exploit the software, even if it doesn't get full ownership. Note that the backup license in the clause box isn't *exclusive* to the customer. Yes, the point of the whole clause is to give the customer exclusive rights: ownership. But if a court refuses to enforce the assignment, it almost certainly won't enforce an exclusive license either. The backup license salvages what it can.

Sometimes courts and government agencies won't honor an assignment unless the vendor signs special forms or cooperates in other ways. It's hard to know in advance what kind of cooperation you'll need. So Subsection (c) in the clause box above requires whatever reasonable cooperation the customer may eventually request. Subsection (c) also clarifies that the ownership transfer and license don't end when the contract terminates.

• • • •

Now let's look at a broader (scorched earth) work product ownership clause —the type typically used for the customer's employees, as well as for individual contractors who operate like members of the customer's staff.

The example in the next clause box assumes the vendor is one of those contractors—a freelance software programmer, for instance. But it works just as well for employees—though in that case you might change "Vendor" to "Employee," "Customer" to "Employer" or "Company," and "engagement" to "employment."[28]

Before learning about the next clause, you should read the material earlier in this Subchapter 1 about the more limited work product ownership clause. Some of the necessary concepts explained there aren't repeated here.

Ownership of Work Product: Broad Transfer

(a)*Reporting of Inventions.* Vendor shall promptly disclose to Customer all computer software programs, other works of authorship, formulas, processes, compositions of matter, databases, mask works,[29] improvements, logos, symbols, designs, and other inventions that Vendor makes, conceives, reduces to practice, or creates, either alone or jointly with others, during the period of the Vendor's engagement with Customer (collectively, "Inventions"), whether or not in the course of such engagement and whether or not such Inventions are patentable, copyrightable, protectable as trade secrets, or otherwise subject to intellectual property protection.

(b)*Customer Ownership of Work Product.* An Invention will be considered "Work Product" and will be Customer's sole property if it fits any of the following three criteria: (1) it is developed using equipment, supplies, facilities, or trade secrets of Customer; (2) it results from Vendor's work for Customer; or (3) it relates to Customer's business or its current or anticipated research and development.

(i)Work-for-Hire. To the extent permissible under applicable law, Work Product will be considered work made for hire pursuant to the U.S. Copyright Act, 17 U.S.C. §§ 101 et seq., and any foreign equivalent thereof.

(ii)Assignment. To the extent, if any, that Customer does not own full right, title and interest in and to the Work Product pursuant to Subsection __(b)(i) above, Vendor hereby assigns to Customer all of its ownership, right, title, and interest in and to all Work Product, including, without limitation: (A) all copyrights, patents, rights in mask works, trademarks, trade secrets, and other intellectual property rights and all other rights that may hereafter be vested relating to the Work Product, arising under U.S. or any other law, together with all national, foreign, state, provincial, and common law registrations, applications for registration, and renewals and extensions thereof; (B) all goodwill associated with Work Product; and (C) all benefits, privileges, causes of action, and remedies relating to any of the foregoing, whether before or hereafter accrued (including without limitation the exclusive rights to apply for such registrations, renewals, and/or extensions, to sue for all past infringements or violations of any of the foregoing, and to settle and retain proceeds from any such actions).

(c)*Backup License.* To the extent, if any, that this Section __ does not provide Customer with full ownership, right, title, and interest in and to the Work Product, Vendor hereby grants Customer a perpetual, irrevocable, fully paid, royalty-free, worldwide license to reproduce, create derivative works from, distribute, publicly display, publicly perform, use, make, have made, offer for sale, sell or otherwise dispose of, and import the Work Product, with the right to sublicense each and every such right. Exercise of Customer's rights pursuant to this Subsection __(c) does not excuse any breach of Vendor's obligations pursuant to Subsection __(b) above or its breach of the warranty in Section __ of this Agreement (Ownership/Infringement Warranty).

(d)*Prior Inventions.* Vendor represents that Attachment __ (Prior Inventions) attached to this Agreement is a list of all Vendor's Inventions prior to the Effective Date which Vendor has not separately assigned to Customer (collectively "Prior Inventions"), and that if Attachment __ is blank or not included, there are no Prior Inventions. Vendor shall not incorporate any Prior Invention into the Work Product or otherwise use any Prior Invention in its work pursuant to this Agreement without Customer's prior written consent.

(e)*Moral Rights.* In addition to the foregoing transfers and allocations of rights, Vendor hereby irrevocably transfers and assigns to Customer any and all "moral rights" Vendor may have in or with respect to the Work Product. Vendor also hereby forever waives and agrees that it shall never, even after termination of its engagement with Customer, assert any moral rights with respect to the Work Product. "Moral rights" include any rights to claim authorship of or credit on a work of authorship, to object to or prevent the modification or destruction of a work of authorship, or to withdraw from circulation or control the publication or distribution of a work of authorship, and any similar right, existing under judicial or statutory law of any country or subdivision of a country, or under any treaty, regardless of whether or not such right is described as a "moral right."

(f)*Further Assistance.* Vendor shall help Customer obtain and enforce patents, copyrights, rights in mask works, trade secret rights, and other legal protections for the Work Product in any and all jurisdictions throughout the world. Vendor shall execute any documents Customer reasonably requests for use in obtaining or enforcing such rights and protections. To the extent that such assistance occurs after Vendor's engagement with Customer, Customer shall compensate Vendor at a reasonable rate for time and expenses spent at Customer's request pursuant to this Subsection __(f). Vendor hereby appoints Customer or its designated representative as Vendor's attorney-in-fact to execute documents on Vendor's behalf for the purposes set forth in this Subsection __(f).

(g)*Survival.* The rights and obligations of this Section __ will survive any termination or expiration of this Agreement or of Vendor's engagement with Customer.

Subsection (a) in the clause box above addresses a particular concern for customers (and employers): How will the customer know what software and other assets the vendor has created, which the customer might be able to claim as work product? The solution is for the vendor (or employee) to report all software and other inventions created during the engagement, even if they're not related to the work for the customer.

Next, the clause determines which "Inventions" are work product—software and other assets owned by the customer. Subsection (b)'s definition includes any software or other asset related to the customer's business, even if it's created after hours, and any asset created with the customer's computers or other facilities, even if it has nothing to do with the customer's business. That's a common definition, but it's also pretty favorable to the customer. The vendor might try to narrow it. For instance: "'Work Product' refers to any Invention conceived, developed, or reduced to practice during Vendor's work for Customer." Or, to narrow the definition even further: "'Work Product' refers to any software related to the Pest Management Industry created during Vendor's work for Customer." (That assumes the contract defines "Pest Management Industry.")

The example in the clause box above includes both a work-for-hire clause and an assignment, in Subsections (b)(i) and (b)(ii). Work-for-hire only relates to copyright, so the assignment helps the customer claim patentable inventions and other IP rights.[30] The assignments also serve as backup for the work-for-hire terms. If the latter can't be enforced, everything is assigned. Whatever happens, the customer should own all work product. And in case both the assignment and work-for-hire terms fail, Subsection (c) provides a backup license.

Subsection (f) in the clause box requires that the vendor help obtain patents, copyright registrations, and other IP law tools. It also provides that the customer will be the vendor's "attorney-in-fact." In other words, if the vendor ever can't sign IP ownership or enforcement documents, the customer can sign them as the vendor's representative.

Subsection (e) of the clause box addresses a similar concern. Some foreign countries give authors "moral rights" over their work. (The United States does too, but to a very limited extent.) Moral rights vary, but they often include rights to be identified as the author and rights to prevent mutilation or revision. They apply more to artistic works than software, but if you're the customer, why risk leaving any such rights with the vendor? Subsection (e) of the clause box waives moral rights, to the extent possible.

Finally, Subsection (d) asks the vendor to list all his or her prior inventions. Work product assignments cover new work only, and often the vendor couldn't assign prior inventions even if he or she wanted to, because third parties own them. So customers want to identify prior inventions in advance, in order to head off disputes, and they want those inventions kept out of the vendor's work. Some work product clauses go a step further. If the vendor includes a prior invention in the work product despite all these precautions, the vendor grants the customer a broad license to exploit it (like the backup license in Subsection (c)). Vendors, of course, should hesitate before granting a license like that, particularly if third parties own some or all of their prior inventions.

2. Transfer of Existing Assets

This subchapter addresses software and other assets the assignor created on its own and wants to give to the customer, rather than "work product" created specifically for the customer. We'll keep calling the parties the “vendor” and “customer,” for consistency, but transfers of existing assets often take place between strategic partners, without a vendor/customer relationship. (In the next clause box, you might replace “Vendor” with “Assignor” or “Provider” and “Customer” with “Recipient” or “Assignee.”)

As with a license, the contract should clearly define the “Software,” “Assigned Materials,” or whatever you’re calling the assets in question. If there’s any chance of dispute, make sure to list all modules, libraries, bug fixes, documentation, etc. Software subject to an IP ownership transfer usually includes all forms of code: object code, source code, etc. It might also include any documentation necessary to understand the software. For instance: “The ‘Assigned Materials’ refers to Vendor’s *TaxMadness* software application, including without limitation: (1) all versions thereof; (2) object code, source code, machine code, and all other forms of software code; (3) all user manuals and related technical documentation Vendor has at any time published or distributed to customers, or included in internal development and quality assurance manuals, for use with the *TaxMadness* software; and (4) the trademarks and logos reproduced on Attachment __ (*TaxMadness Trademarks*).”

The example in the following clause box includes an assignment, a license, a moral rights clause, and a further assistance clause—just like the work product ownership language discussed in Subchapter 1. So for explanations of those terms, see Subchapter 1.

Ownership of Existing Assets

(a)*Assignment.* Vendor hereby assigns to Customer all of its ownership, right, title, and interest in and to the Software, excluding patents but including, without limitation: (i) all copyrights, trademarks, and trade secrets, including all such rights that may hereafter be vested relating to the Software, arising under U.S. or any other law, together with all national, foreign, state, provincial, and common law registrations, applications for registration, and renewals and extensions thereof; (ii) all goodwill associated with the Software; and (iii) all benefits, privileges, causes of action, and remedies relating to any of the foregoing, whether before or hereafter accrued (including, without limitation, the exclusive rights to apply for such registrations, renewals, and/or extensions, to sue for all past infringements or violations of any of the foregoing, and to settle and retain proceeds from any such actions).

(b)*License.* To the extent, if any, that this Section __ does not provide Customer with full ownership, right, title, and interest in and to the Software, Vendor hereby grants Customer a perpetual, irrevocable, fully paid, royalty-free, worldwide license to reproduce, create derivative works from, distribute, publicly display, publicly perform, use, make, have made, offer for sale, sell or otherwise dispose of, and import the Software, with the right to sublicense each and every such right.

(c)*Moral Rights.* In addition to the foregoing transfers and allocations of rights, Vendor hereby irrevocably transfers and assigns to Customer any and all "moral rights" Vendor may have in or with respect to the Software. Vendor also hereby forever waives and agrees never to assert any moral rights with respect to the Software. "Moral rights" include any rights to claim authorship of or credit on a work of authorship, to object to or prevent the modification or destruction of a work of authorship, or to withdraw from circulation or control the publication or distribution of a work of authorship, and any similar right, existing under judicial or statutory law of any country or subdivision of a country, or under any treaty, regardless of whether or not such right is described as a "moral right."

(d)*Further Assistance.* Vendor shall help Customer obtain and enforce copyrights and other legal protections for the Software in any and all jurisdictions throughout the world. Vendor shall execute any documents Customer reasonably requests for use in obtaining or enforcing such rights and protections. Customer shall compensate Vendor at a reasonable rate for time and expenses spent at Customer's request pursuant to this Subsection __(d). Vendor hereby appoints Customer or its designated representative as Vendor's attorney-in-fact to execute documents on Vendor's behalf for the purposes set forth in this Subsection __(d).

(e)*Survival.* The rights and obligations of this Section __ will survive any termination or expiration of this Agreement.

The clause above gives the customer all IP rights in the software, except patents. The customer's IP will include copyrights, of course, and it may include any brand name or logo used to sell the software, depending on how the contract defines "Software." In other words, it's a lock, stock, and barrel transfer of a product from vendor to customer.

Like the first clause box in Subchapter 1, this clause box doesn't include patents because that would give the customer more than the software. It would give the customer the right to keep anyone else, including the vendor, from creating new code similar to the transferred software. But the clause box does give the customer a broad patent license, in Subsection (b), in case the software includes technology covered by the vendor's patents (current or future).[31]

This clause box uses an assignment, in Subsection (a), but not a work-for-hire provision. That's because work-for-hire fits software and other assets to be created specifically for the customer, rather than the *existing* assets at issue here: software the vendor presumably created before executing the contract. (That distinction lies behind all the differences between this clause and the work product ownership clauses in Subchapter 1.)

3. Transfers from the Vendor's Employees and Contractors

What if the vendor has employees or contractors who wrote the work product or existing assets? Is each contributor subject to a valid employment relationship or contract giving the vendor full rights it can transfer to the customer? If not, those contributors will own part of the software. As this chapter's introduction mentions, customers should protect themselves through IP warranty terms.[32]

Customers should also consider terms affirmatively requiring that the vendor's employees and contractors sign a transfer contract.

Staff Ownership Transfers

Vendor shall require that all its employees and contractors in any way involved in creating the Software execute agreements with Customer in the form attached hereto as Attachment __ (*Staff Transfer*). Vendor shall reasonably cooperate with Customer in assuring such employees' and contractors' compliance with the terms of Attachment __.

• • • •

Vendor shall ensure that all its employees and contractors in any way involved in creating the Software are subject to written agreements with Vendor that, on or before the effective date of the assignment to Customer in Section __ above (*Ownership Transfer*), grant Vendor all such employees' or contractors' ownership and other rights in and to the Software.

The first example in the clause box above requires that vendor employees and contractors sign a contract with the customer, assigning their rights directly to the customer. The form for these separate assignment contracts would be an attachment to the main agreement, and its central clause would match the clause box in Subchapter 2 (but replace "Vendor" with something like "Assignor," "Employee," or "Staff Member").[33]

Instead of a contract with the customer, the second example in the clause box requires a contract between the vendor itself and its employee or contractor, transferring rights to the vendor. If the vendor has all the necessary rights and transfers them to the customer, the customer should have the rights it needs. Agreements with that structure don't usually include an attached contract for the vendor to sign with its staff. Rather, they require that the vendor make sure it has received the necessary rights, again as in the second example in the clause box. (That supplements the Vendor's IP warranty, if any.)

E. Subscription for Cloud Services

This chapter addresses prime clauses for cloud services. It's also useful for similar services provided through computers, like Internet connectivity and telecommunications. But the discussion here focuses on the cloud.

As this book's Introduction explains, cloud services are separate from professional services, like support and consulting, which the vendor provides primarily through human professionals, rather than through computers. Cloud services are also separate from software licenses. Both involve software, but in a licensing deal, the vendor gives the customer a copy of the software. In cloud services, the customer doesn't get a copy; rather, it gets remote access, usually through the Internet.[34]

1. Cloud Services Subscriptions

In a cloud services agreement, the customer gets the right to access and use the software or other technology: a subscription. Cloud subscriptions are simpler than copyright licenses, so they call for a shorter, simpler clause.

Cloud Services Subscriptions

During the term of this Agreement, Customer may access and use Vendor's _______ service (the "Service") pursuant to Vendor's policies posted on Vendor's website at www.____________, as such policies may be updated from time to time. Vendor retains all right, title, and interest in and to the Service, including without limitation all software used to provide the Service and all logos and trademarks reproduced through the Service, and this Agreement does not grant Customer any intellectual property rights in the Service or any of its components.

• • • •

Customer may access and use the computer system described on Attachment __ (the "System") from ________ until _________ (the "Subscription Period"). Vendor retains all right, title, and interest in and to the System, including without limitation all computers, other hardware, and software incorporated into or used by the System, and this Agreement does not grant Customer any intellectual property rights in the System or any of its components.

The first example in the clause box above envisions a typical software-as-a-service (SaaS) offering. The second envisions a bigger, more complicated system—platform-as-a-service (PaaS) or infrastructure-as-a-service (IaaS)—where the vendor gives the customer remote access to server computers in a data center. (Usually, those servers run some software from the customer and some from the vendor.) But the two clauses aren't radically different, and either could work for SaaS, PaaS, or IaaS (maybe with slight modification).[35]

Where possible, customers should include a detailed description of the system or service, as suggested in the second example in the clause box above ("the computer system described on Attachment __"). Vendors should consider services descriptions too. A description can list the tasks the system *won't* perform, and that protects the vendor.

This book doesn't provide an example of a cloud services description. But Chapter II.A ("Technical Specifications") covers specifications for both software and cloud services, and Chapter II.B ("Service Level Agreements") covers SLAs. Both will help you describe cloud services.

If your deal involves a standardized contract, like online terms of service, you should review Appendix 3 ("Clickwraps, Browsewraps, and Other Contracts Executed without Ink"). And if your customers 46 **The Tech Contracts Handbook** execute cloud services contracts online, review Appendix 4 ("Online Policy Documents").

2. Cloud Resale

Cloud vendors can delegate the right to sell their services, just as software vendors can license the right to distribute their software.

Unlike software vendors, cloud services vendors don't usually give the right to *distribute* copies of their software. Remember, cloud services end customers don't get software copies.[36] So there's nothing for the reseller to distribute. End customers just get a subscription to access the software and the rest of the system—to use it. So what rights does the reseller need? It depends who hosts the software for the end customers: the reseller or the vendor.

If the reseller hosts the software and provides it to its end customers as a cloud service, it needs a software license. Just like a software customer, it's going to put a copy of the software on its own computers and use it. The contract's prime clause, then, is a license to reproduce and use the software. But the terms differ from typical software licenses in one way. The reseller needs the right to grant access to its end customers, so any terms forbidding "service bureau" operation of the software should be reversed. The clause should specifically grant the right to let third parties access and use the software. For reseller-hosted resale terms, see the second example in the clause box in Subchapter I.C.2 ("Software Licenses in General: Scope Terms").

If the vendor hosts the software, on the other hand, the reseller doesn't need a license. It's not making copies. Instead, the prime clause simply authorizes resale.

SaaS Resale: Vendor Hosted

Subject to Section __ (*Qualification of End Customers*), Reseller may sell subscriptions to the System authorizing its end customers to access and use the System.

The example in the clause box above lets the reseller sell cloud services subscriptions, with one caveat. Usually, the vendor doesn't authorize sales to just anyone. The contract may require that the vendor preapprove the end customers or that they fit some set of criteria, like industry, size, or credit rating. Those restrictions would go in the section mentioned at the start of the example ("Section __ (*Qualification of End Customers*)").

Finally, does the vendor care what terms the reseller gives end customers? (Probably.) If so, the contract might include a whole Terms of Service ("ToS")—a contract the reseller is required to use with end customers. Or it might spell out certain terms for the ToS, like an exclusion of any responsibility, liability, or support obligations from the vendor. It might even make the vendor an intended third party beneficiary[37] of the reseller's ToS rights, or just of the clauses relating to the vendor, like the exclusion of liability.

F. Promise of Professional Services

Professional services contracts call on the vendor's staff to give the customer some kind of *help*. So they differ from software contracts, where the vendor gives the customer software or IP rights. They also differ from cloud services agreements, though both of course involve services. Cloud services come primarily through computers, while professional services come primarily from those human staffers.[38]

Obviously, professional services clauses appear in contracts addressing programming, IT consulting, and other human-based services. But they also crop up in many software and cloud services contracts. In either case, the vendor could (1) agree to develop some or all of the technology provided through the deal, a professional service, and (2) grant a license or cloud services subscription. Or the vendor could promise phone support or maintenance to help customers use technology, adding a promise of professional services to what's otherwise a software license or cloud services agreement. Those deals call for combination contacts with multiple prime clauses: one for professional services and one for software licensing or cloud services.[39] This chapter addresses just the professional services prime clause.

Promise of Professional Services

Vendor will provide the following services to Customer: ____________________.

Subchapter 1 below addresses the blank in the clause box above: the terms defining the professional services. Subchapter 2 explains how to craft a statement of work procedure and form, while Subchapter 3 does the same for a change order procedure.

1. Defining the Professional Service

Your description of professional services might appear in the main body of the contract or in an attached statement of work. Either way, you'll need to figure out how to describe the services.

Professional services descriptions can be task driven or outcome driven. Task-driven descriptions favor the vendor. In provisions like the first two examples in the clause box below, all the vendor has to do is perform the tasks listed. “Vendor will provide two employees to do the following . . .” The customer might be unhappy with the outcome, but that’s not the vendor’s problem (at least, legally).

Outcome-driven descriptions, like the last example in the clause box below, favor the customer. “Vendor will achieve the following . . .” The vendor can’t point to a list of tasks and say: “I tried.” If the vendor doesn’t achieve the outcome, it hasn’t met its obligations.

Description of Professional Services

During Business Hours, Vendor shall staff the Help Desk with no fewer than __ [number] technicians. Through such technicians, Vendor shall exercise reasonable efforts to resolve computer and software malfunctions and user errors promptly, in response to Customer technical support requests.

• • • •

Throughout the term of this Agreement, Vendor shall: (a) make __ [number] Consultants available to Customer during Business Hours to assist with system integration; and (b) produce a system development report __ or more days after the end of each calendar _____ [month or quarter], providing detailed descriptions of system development progress.

• • • •

Vendor shall write an employee benefits software application that conforms to the technical specifications set forth in Attachment __ (the “Software”).

Clarity is particularly important in professional services descriptions, and particularly difficult. On fixed price deals, vendors often suffer from “scope creep”: the job gets bigger but the price doesn’t. A clear description of the professional services helps prevent scope creep, because the vendor can say: “Look, that’s not included in the contract; it’ll cost extra.”

Customers should protect themselves from unclear descriptions too. They should make sure the contract promises all the expected help. Is technical support required? How about bug fixes? After-hours support? Is special equipment required, and if so, who supplies it?

Both parties should be sure to keep the customer’s tasks out of the professional services description. Don’t draft a description like the following: “Vendor shall provide the following services: (1) Customer’s project manager shall meet with Vendor by June 17 and outline the specifications, and Vendor shall then prepare the first specifications draft . . .” That makes no sense because the vendor can’t make the customer show up at that meeting. To avoid confusion, list the customer’s tasks separately, in their own attachment or clause. With the customer’s tasks elsewhere, the previous provision might read: “Within ten business days of the first specifications meeting with Customer, Vendor shall deliver the first specifications draft.”[40]

2. Statements of Work

You should consider putting long or detailed service descriptions in a separate document: a “statement of work” (SoW) attached to the main contract. The separation helps avoid confusion between what IT professionals often call “business terms”—services descriptions, payment requirements, etc.—and the rest of the contract.[41] Also, an SoW system lets you add future projects—future statements of work—under the same master services agreement, without amending it and without renegotiating terms already agreed.

The body of the contract should set up the statement of work procedure, particularly if you expect multiple SoWs, some to be executed after the contract. It’s usually best to include a statement of work *form* to fill out, again particularly if you plan on multiple SoWs. That’s better than letting the parties use any old Word document, e-mail, notepad, or napkin. Use of a form helps keep careless employees from agreeing to new projects without realizing it, and without proper review. Some clauses go further and provide: “No proposed statement of work will become part of this Agreement or bind either party unless signed by each party’s Project Manager” (or some other officer).

Statement of Work Procedures

Vendor shall provide such services (“Services”) as are required by any statement of work in the form attached hereto as Attachment A (*Statement of Work Form*), executed by each party (each a “Statement of Work” or “SoW”). Upon execution, a Statement of Work will become part of this Agreement. In the event of any conflict with a Statement of Work, the terms of this main body of this Agreement will govern.

The procedure should require that both parties sign the statement of work, as in the example in the clause box above. And it shouldn't be binding until they do.

A contract with statements of work should also address conflicts among attachments. The last sentence in the clause box above says that if an SoW's terms contradict the main contract's, the main contract governs. For more on that issue, see Chapter III.N ("Conflicts among Attachments").

• • • •

The clause box below is a statement of work *form*. It has a blank at the top for an SoW number, so it's intended for a multiple SoW master agreement where the parties give each statement of work a number, to keep track. But the form below works for single SoW agreements too (without the number blank).

Statement of Work Form

STATEMENT OF WORK NUMBER ____
To Technology Services Agreement

Project Title: ________________________________

This Statement of Work Number __ (this "SoW") is entered into pursuant to the ______________ [date] Master Services Agreement (the "Agreement") by and between ___________ ("Vendor") and _____________ ("Customer").

This SoW is incorporated into the Agreement. In the event of any conflict with this SoW, the main body of the Agreement will govern. The provisions of this SoW govern only the subject matter hereof and not any other subject matter covered by the Agreement. Capitalized terms not otherwise defined in this SoW will have the meanings given in the main body of the Agreement.

I.*Professional Services & Deliverables.* Vendor shall provide the following services: [Insert description of professional services. Include technical specifications for any technology to be created, or include reference to specifications attached to this SoW.] _______________ ____

II.*Customer Cooperation.* Customer shall reasonably cooperate with Vendor in the provision of services and shall provide the following assistance to Vendor: *[Insert description of Customer responsibilities, or insert "N/A" if not applicable.]*________________________________ ____

III.*Payment.* Customer shall pay Vendor as follows: [Insert payment schedule. Insert any payment/invoicing terms not already covered in main body of Agreement.] ____ ____

IV.*Additional Provisions.* In addition, the parties agree as follows: [Insert additional terms or "N/A" if not applicable.] ____ _____

This SoW is effective as of the latest date of execution set forth below.

signature block for both parties

In the clause box above, the italicized text in brackets is part of the form. It provides instructions for future statement of work drafters. They can remove that text when they do their drafting, but you shouldn't when you add the blank form to the contract.

3. Change Orders

Often, the customer wants to change the professional services partway through the project.

Change Order Procedures

The parties may agree to modify the Services through a written change order specifically referencing both this Agreement and the applicable Statement of Work. Such change order will become part of the applicable Statement of Work when executed by both parties, and the services described therein will become part of the Services.

• • • •

Customer may request that Vendor add features to the Software not in the Specifications by submitting a written proposed change order to Vendor, in the form attached hereto as Attachment __ (*Change Order Form*). Vendor shall negotiate in good faith regarding change order prices and shall not require rates higher than those set forth in Section __ (*Service Rates*). Such change order will become part of the applicable Statement of Work when executed by both parties, and the services described therein will become part of the Services.

Change orders let the parties modify the professional services without a new contract or even a new statement of work. See both examples in the clause box above. You don't really need a change order; you could just amend the contract. But a change order procedure is easier. Plus, for the customer, clauses like the second example above require that the vendor offer relatively reasonable prices and terms for the revised work. ("Good faith," in the clause, isn't the clearest of contractual obligations, but it's better than nothing for the customer.)

In both examples, the change order, once signed, becomes part of the statement of work in question: the SoW describing the services to be changed. If your contract doesn't use statements of work, substitute "this Agreement" for "the applicable Statement of Work" in the clause.

If you do create a change order procedure, consider a change order *form*, as in the second example in the clause box above. That way, a casual exchange of letters or e-mails won't count as a change order. The form could read: "Pursuant to the April 13, 2015, Technology Services Agreement between Obsequio, Inc. ('Vendor') and ScopeCreep LLC ('Customer'), Vendor shall provide the additional professional services listed below, and Customer shall pay the additional fees listed below . . ." Or you could create a more elaborate form. If you'd like a more elaborate model, consider modifying the statement of work form in Subchapter 2 above. (Obviously, you’ll need to replace “Statement of Work” and “SoW” with “Change Order” throughout, among other edits.)

G. Payment

In the payment clause, the customer or distributor promises to pay the vendor. In some cases, a cloud vendor promises to pay its reseller, but we'll focus on payments to vendors, except in Subchapter 2 below, where we address cloud resellers.

The payment clause should specify the technology or services triggering the obligation, particularly if the contract has more than one deliverable. The parties might eventually need to know what's been paid for and what hasn't. Also, a specific payment obligation can help the customer if the vendor breaches or goes bankrupt. If the contract has separate payment obligations for software licenses and support services, for instance, and the vendor stops providing services, the customer could stop paying service fees but keep paying license fees, and keep the software license. If the customer paid in a lump sum for both license rights and services, on the other hand, it might have to keep paying for both to keep the software, even if the vendor stopped providing the services.

1. Customers' Fees

Cloud services agreements often call for periodic payments, usually monthly or annually, as in the first example in the clause box below. The cloud subscription and services stop if the customer fails to pay for the next subscription period. Software license agreements may call for similar periodic "royalty" payments, as in the second example below. Many other license agreements call for a simple one-time payment, as in the third and fifth examples.

Customers’ Fees

Customer shall pay Vendor $_______ (the “Subscription Fee”) for each Subscription Period. No new Subscription Period will go into effect unless such payment is made 30 or more days before the scheduled start thereof.

• • • •

Customer shall pay Vendor a license royalty of $___________ per ________ (the “License Term”). No new License Term will go into effect unless such payment is made 30 or more days before the scheduled start thereof.

• • • •

Customer shall pay Vendor $__________ for its rights granted in Section __ (*Software License*), subject to invoice upon delivery of the Software.

• • • •

Customer shall pay Vendor the following amounts, each due 30 days after the milestones listed in Attachment B (*Development Schedule*):

- ***Milestone 1:*** $_____________.
- ***Milestone 2:*** $____________.
- ***Final Milestone:*** $_____________.

• • • •

Customer shall pay Vendor: (a) $____________ in license fees, due within 30 days of Acceptance (as defined in Section __, *Acceptance Procedures*); and (b) $____________ per calendar quarter for the Support Services, due on the last day of the preceding calendar quarter.

• • • •

Customer shall pay for Professional Services on a time and materials basis, according to the rate schedule in Attachment __ (*Rates*). Vendor shall invoice all amounts due on the last day of each calendar quarter. Payment against all invoices will be due within ____ days of receipt thereof.

A thorny issue crops up in development agreements where payment is required when the work is done. What does “done” mean? If the vendor says the job’s finished but the software doesn’t do what the customer hoped, is it done? The best way to address this is through an acceptance procedure, as suggested by the fifth example in the clause box. (The fourth example may call for acceptance too, because each of the milestones listed could be acceptance of a deliverable.) See Chapter II.F (“Delivery, Acceptance, and Rejection”).

Professional services fees can be paid all at once or on a continuing basis, just like license fees. For continuing fees, see Subsection (b) of the fifth example in the clause box above. The parties might also agree to milestone payments; the vendor gets paid each time it achieves a goal listed in the contract, like delivery or acceptance of a deliverable, as in the fourth example above.

Vendors often bill professional services on a time and materials basis, as in the last example in the clause box. The customer pays for the vendors' employees' billable hours and for equipment and other materials. Sometimes the materials part of the charge is a simple reimbursement, but sometimes it includes a markup, so the vendor can make a profit on the materials. Time and materials payments can lead to disputes. What if the vendor spends 200 hours solving a problem the customer thinks should take 50? One solution is a clause providing: "Fees will not exceed $_______ for any given calendar month unless Customer agrees in writing in advance." That doesn't mean the vendor has to work for free once it hits the dollar cap. More likely, it will then stop work and request instructions.

2. Distributors' and Resellers' Fees

Software distributors and cloud resellers often pay their vendors (suppliers) a percent of revenues collected from end customers. The first example in the clause box below calls for such a revenue share—a "royalty"—in a software license agreement. But the example could work for the cloud too, with fees calculated "per Seat" or "per Subscription," instead of "per copy of the Software distributed." In some cases, however, the distributor or reseller pays a fixed fee per copy, seat, or subscription, as in the second example below.

Distributors' Fees

Distributor shall pay Vendor a royalty of __ % of gross revenues collected from End Customers; provided such royalty will not fall below $___ per copy of the Software distributed, regardless of the revenues collected by Distributor, including without limitation if Distributor collects no revenues. On the ______ business day of each calendar quarter, Distributor shall report all Software transactions for the preceding calendar quarter, including customer names, units provided, and amounts receivable and received, as well as such additional detail as Vendor reasonably requests. Distributor shall pay all amounts due within 30 days of such report.

• • • •

Reseller shall pay Vendor $___ per calendar month for each Seat provided to End Customers, and Vendor shall invoice Distributor monthly for all Seats activated during the prior calendar month.

• • • •

Vendor shall pay Reseller $___ per Referred Subscription provided to End Customers. On the ______ business day of each calendar quarter, Vendor shall report all Referred Subscriptions for the preceding calendar quarter, including customer names and amounts receivable and received. Vendor shall pay all amounts due within 30 days of such report.

Royalty and other revenue share obligations can lead to disputes. The solution is detail. The party receiving payments should make sure the paying party has to report all sales in detail and on a regular basis, as in the first and third examples in the clause box above. Usually, that's the only way to learn how much is due.[42] And if the recipient gets a percentage of revenues rather than a fixed payment, both parties should think through the meaning of "revenues." The recipient gets 20 percent of *what*? The two examples in the clause box give the recipient a percentage of "gross revenues." That's usually relatively simple, but think about whether the parties might dispute the definition. And if the recipient gets a percentage of *net* revenues, you definitely need a definition. Decide whether your figure includes sales taxes, commissions, cost of delivery, etc., and wrap those decisions into a definition. For instance: "'Net Revenues' refers to all revenues received from third parties for rights to the Software, including revenues from maintenance, minus: (a) sales tax, use tax, and value-added taxes; (b) return credits; and (c) salespeople's commissions."

Recipients paid through a percentage of revenues should also consider a minimum royalty. See the first example in the clause box. Without a minimum, the payee could sell for nothing or next to nothing—as a promotion, for instance—and the recipient would get nothing.[43]

The last example in the clause box reverses the flow of payments: the vendor pays the reseller. It's meant for a resale agreement where the vendor hosts a cloud service and charges the end customer, but the reseller actually makes the sale. The last example calls for a fixed payment per subscription. The payment could be per seat instead, which wouldn't change the clause much. Or the clause could call for a share of the vendor's income from referred customers: "__ % of gross revenues collected from Referred Subscriptions."

3. *Due Dates and Invoices*

An invoice can start the clock ticking on a payment obligation, as in the first example in the clause box below. Or it can simply remind the customer of payment deadlines already set by the contract, as in the second example.

Invoices

Vendor shall submit itemized invoices to Customer for the payments required by this Section __, and all invoices will be due and payable within 30 days of Vendor's transmission of the invoice.

• • • •

Invoices serve as confirmation of amounts owed, and Customer's payments are due on the dates required pursuant to this Section __ (*Fees*), regardless of whether the Customer receives an invoice, or when.

[1]. For combination agreements, see "Subject Matter: Types of IT Contracts" in the Introduction.

[2]. See Chapter II.D ("Documentation").

[3]. Actually, the explanation above glosses over some legal complications related to "use" and "reproduction" rights. But we don't need to address those complications to understand typical end user licenses. If you'd like to know more, see Subchapter I.C.1 ("Copyright License Rights")—particularly the bullet point on *Use and Other Pseudo Rights*.

[4]. The "provided" language helps the vendor enforce the restrictions. Subchapter I.C.1 ("Copyright License Rights").

[5]. Vendors use this "license vs. sale" language to avoid copyright's "first sale doctrine." See footnote 13.

[6]. See Chapter II.D ("Documentation").

[7]. The "provided" language helps the vendor enforce the restrictions. Subchapter I.C.1 ("Copyright License Rights").

[8]. For software audits, see Chapter II.P ("Software Audits"). For deletion of software after contract termination, see Subchapter II.V.4 ("Effects of Termination"). Finally, an "intended third party beneficiary" is someone who benefits from promises made in a contract and has the right to enforce them, but who isn't a party to the contract.

[9]. Vendors use this "license vs. sale" language to avoid copyright's "first sale doctrine." See footnote 13.

[10]. The copyright statute is found at Title 17 of the United States Code (U.S.C.). Just to muddy the waters, some software is patented, as well as copyrighted. But software customer contracts don't need patent licenses. See Appendix 1 ("Intellectual Property").

[11]. By granting the right to "use," "install," or "run," vendors bolster their argument that the customer doesn't *own* its copy of the software. That helps vendors avoid copyright's "first sale doctrine," explained in footnote 13.

[12]. In addition to the obvious implication of terms like "use," the copyright statute clarifies reproduce-to-install rights for some customers. "[I]t is not an infringement for the owner of a copy of a computer program to make . . . another copy . . . provided: (1) that such a new copy . . . is created as an essential step in the utilization of the computer program in conjunction with a machine . . . or (2) that such new copy . . . is for archival purposes only. . . ." 17 U.S.C. § 117(a).

[13]. Vendors use this "license vs. sale" language to avoid copyright's "first sale doctrine" (17 U.S.C. § 109). The first sale doctrine provides that if you own a copy of a book or painting or photo or other copyrighted work, you can sell that particular copy without infringing copyright. You just can't make new copies. (It's not entirely clear how the first sale doctrine applies to software, so the strategy recommended above may not always protect the vendor. But it's a good precaution.)

[14]. Exclusivity has some consequences. For one, the customer essentially *owns* the transferred right; it owns a piece of the copyright. So the customer can sue third parties for infringement of that particular right. In other words, if the customer has an exclusive right to distribute in Virginia, and someone else distributes in Virginia, the customer has legal standing to sue that person. (To help enforce their rights, customers with exclusive licenses should consider registering their licenses with the U.S. Copyright Office.)

[15]. Actually, "irrevocable" licenses can be revoked in one of two ways. First, the copyright statute lets the author (the programmer) revoke a license after 35 years, no matter what any contract says. 17 U.S.C. § 203. Usually, no one cares after that long. Second, it's always possible that a court won't honor irrevocability, particularly if the customer refuses to pay and has no good reason. Courts don't like clauses that seem unfair.

[16]. See Chapter III.I (”Assignment"). Note that there are two kinds of "assignments." Here, we're talking about an assignment of an entire contract, with all its rights and obligations. "Assignment" also refers to a transfer of ownership rights in a copyright or other intellectual property. (For that sort of assignment, see Chapter I.D, "Technology Ownership: Assignment and Work-for-Hire.") If the license rights are nontransferable, but the entire contract is assignable, what you have is a mess. The two clauses contradict each other, at least arguably, and it's hard to say which governs.

[17]. See Subchapter I.E.2 ("Cloud Resale").

[18]. For an explanation of "seats" and "concurrent users," see Subchapter I.A.1 ("Reproduction and Use").

[19]. See Chapter I.D ("Technology Ownership: Assignment and Work-for-Hire").

[20]. Licenses do have a disadvantage. A license is a contract right, and contracts can be terminated. An assignment gives an ownership right, which is harder to terminate, and the same goes for a work-for-hire clause, which is essentially non-terminable. (See the following explanation of work-for-hire.) In general, however, if your license agreement is well crafted, termination shouldn't be a major concern.

[21]. See Subchapter II.I.2 ("Intellectual Property/Ownership Warranty"). The customer should also consider registering any copyrights acquired, and filing patents and trademarks where

appropriate. The U.S. Library of Congress handles copyright registration, while the U.S. Patent and Trademark Office handles patents and trademarks.

[22]. Here we're talking about an assignment of ownership in a copyright or other intellectual property. "Assignment" also refers to the transfer of an entire contract, with all its rights and obligations. For that sort of assignment, see Chapter III.I ("Assignment"). For the differences between copyrights and patents, see Appendix 1 ("Intellectual Property").

[23]. A warning for California customers: a couple of bizarre state laws provide that, if you sign a contract with a work-for-hire clause, the contractor becomes your employee, like it or not—at least for purposes of workers' comp and unemployment insurance. (Cal. Labor Code § 3351.5(c) and Cal. Unemploy. Ins. Code §§ 621(d), 686.)

[24]. 17 U.S.C. § 101.

[25]. See 17 U.S.C. § 203.

[26]. Some contracts use "deliverables" instead of "work product," but that can lead to confusion. A deliverable is something the vendor is required to produce and deliver. "Work product" usually includes deliverables but also means assets produced on the job but not delivered, like software that didn't work and incidental inventions.

[27]. For more on patents and how they differ from copyrights, see Appendix 1 ("Intellectual Property"). This is a complex area, so if patents matter, you should consider help from an experienced IP lawyer.

[28]. Some state laws restrict employers' rights to require assignments of inventions. (For instance, see California Labor Code §§ 2870–2872.) If in doubt about your state, get experienced legal help.

[29]. For mask works, see footnote 2 in Appendix 1 ("Intellectual Property").

[30]. For more on patents and the difference between them and copyrights, see Appendix 1 ("Intellectual Property"). This is a complex area, so if patents matter, you should consider help from an experienced IP lawyer.

[31]. See Appendix 1 ("Intellectual Property").

[32]. See Subchapter II.I.2 ("Intellectual Property/Ownership Warranty"). And consider an experienced IP lawyer's help if you're treading the complicated border between patent and copyright.

[33]. The forms library at this book's website includes a sample assignment you can use for employees. Please visit http://TechContracts.com.

Note that the first example in the clause box above gives the customer a contract right against the vendor, not against the employees or contractors. If the employees never actually sign their separate assignment contracts, the customer can sue the vendor for breach of contract, but it may not be able to get any ownership rights from the employees. (Also, see footnote 28 for restrictions on employers' rights to require assignments from employees.)

[34]. See "Subject Matter: Types of IT Contracts," in the Introduction.

[35]. See the explanations of SaaS, IaaS, and PaaS under "A Little Industry Language, Particularly re Cloud Computing" in the Introduction.

[36]. Here we're talking about a pure cloud services deal. With some cloud offerings, the vendor also provides software to run on the customer's computers. This book calls contracts like that "combination agreements," rather than pure cloud services agreements. See "Subject Matter: Types of IT Agreements" in the Introduction.

[37]. An "intended third party beneficiary" is someone who benefits from promises made in a contract and has the right to enforce them, but who isn't a party to the contract.

[38]. See "Subject Matter: Types of IT Contracts" in the Introduction.

[39]. See the description of Combination Agreements under "Subject Matter: Types of IT Contracts" in the Introduction.

[40]. For more on the interplay between customer and vendor responsibilities, see Chapter II.E ("Schedule and Milestones").

[41]. The distinction is actually artificial. They're all contract terms, and they all matter. But often the businessperson who "owns" the project will draft the "business terms," while the lawyer or contract manager will draft the rest: the "legal terms."

[42]. Software audits give the vendor additional security. See Chapter II.P ("Software Audits").

[43]. The solution proposed above fixes a minimum *royalty* (or share) paid to the vendor (or reseller), not a minimum *price* for end customers. The latter would be "vertical price fixing," and it violates some states' antitrust laws.

PART II

General Clauses

“General Clauses” is a catch-all category, referring to everything that’s not a prime clause or a boilerplate clause. The following terms account for most of the ink spread across most software and IT services contracts. The one characteristic shared by all these clauses is that they generate a lot of disagreement, debate, and compromise.

A. Technical Specifications

Technical specifications describe software and computer systems. They say what the technology will do: how it's supposed to perform. They're appropriate for most software-related contracts, including licenses, assignments, distribution agreements, and cloud services agreements. They also appear in professional services agreements, describing the intended functionality of technology deliverables. In many contracts, technical specifications, or "specs," provide the most important terms. Yet businesspeople and lawyers usually pay them little attention.

Some IT professionals distinguish *functional specifications* from technical specifications. A hazy line separates the two, but in general, functional specs describe software from the user's point of view: what it's supposed to achieve—everything from broad business outcomes, like "tracking inventory shipping times and destinations," to fine details, like how screen shots should appear. Technical specs are more . . . technical, looking at issues like system architecture and programming languages. This book makes no such distinction. It addresses both under the "technical specifications" heading.

The specs provide information used in *other* contract clauses. For instance, here's a warranty clause: "Vendor warrants that, during the first 1 year after the Effective Date, the Software will perform according to its technical specifications listed in Attachment A." And here's a maintenance clause: "Vendor shall maintain the System so that it performs materially in accordance with its Specifications." Or a development professional services clause: "Vendor shall design a software application that conforms to the technical specifications attached hereto as Attachment B." Or an acceptance clause: "In the event that the software fails the acceptance tests, Customer shall provide a written description of each deviation from the specifications listed in Attachment A, and Vendor shall repair the software so that it performs materially in accordance with all such specifications." Finally, in cloud services agreements, specs often play a role in the service level agreement (not really a separate agreement): "As used in this SLA, an

‘Error’ is any failure of the System to perform materially as required in the Technical Specifications.”[1]

1. The Importance of Specifications

Because specs are usually technical, many customers and vendors pay them little attention or leave them out. That’s a bad choice.

Imagine you’re the customer and you license a widget-tracking computer system for your factory floor. You soon realize the system takes too long to generate reports. Worse, it won’t sync with your floor managers’ handheld computers. You complain to the vendor, but you signed a contract with no technical specs. All the contract says is that you bought “WidgetTracker Server Edition 5.02, a computer system for tracking and managing widgets on a factory floor.” That doesn’t address speed or synchronization. Maybe the vendor’s salespeople told you the system was fast and could sync, but those promises didn’t find their way into the contract, so they don’t do you much good.

Now imagine you’re the vendor. You have a customer who thinks you were dishonest, and you have a dispute on your hands. Ideally, you would point to the contract and say: “Look, the contract says what the system does; it’s not required to do anything else.” But you can’t because the contract has no specs. As the vendor, you need specs to clarify what the system will *not* do.

In other words, as this book’s introduction explains, good fences make good neighbors. Detailed specifications are excellent fences.

There *are* contracts that arguably don’t need detailed specifications. If there’s little chance reasonable minds could differ about what the system is supposed to do, specs aren’t necessary. For instance, if the deal involves standard off-the-shelf software, with widely known functions, you might leave out the details. Or, better, you might just provide: “The System will perform according to its technical specifications published by Vendor.” That assumes, of course, that the vendor has published something.

But if in doubt, assume you need specs.

2. Responsibility for Specifications

Who should draft specifications? Customer or vendor? Lawyer, businessperson, or IT staffer?

As between customer and vendor, there is no standard answer. Whoever understands the system best will usually write the first draft. Often that's the vendor. But sometimes a customer drafts specs for an RFP (request for proposal) regarding customized technology, and those specs become part of the contract.

Both customers and vendors often leave specs drafting to programmers and engineers. That's usually appropriate, but if the businesspeople and lawyers responsible for the deal don't get involved, the specs may not reflect the business's goals. Let's take a contract for a customized human resources computer system. The customer's technical folks might draft the specs because they're familiar with the technology. But it's the HR staff members who know best what the system should do. If they're not involved, the specs won't fully address HR's needs, and neither will the system.

In other words, whoever is responsible for the deal should play a role in drafting the specs, even if that's a technically challenged businessperson or lawyer.

That doesn't mean a businessperson or lawyer has to write specs from scratch. But if a non-tech businessperson or lawyer is responsible for the deal, he or she should go over the specs with the technical staff, to make sure they describe the desired system. And he or she should edit the specs for clarity. If the technically challenged business manager or lawyer can understand the specs, they must be clear.

3. Organizing and Editing Specifications

The job of drafting technical specifications lies outside the scope of this book, but this subchapter will help you organize and edit your specs.

There is no standard length. Specs should run as long as necessary to express the business goal for the technology. Nor is there a standard organization. Specs might appear as a narrative: an essay describing cloud computing offerings or software. But unless the specs are very short, numbered clauses usually work better. The specs could appear as a list of numbered bullet points, for instance, or as an outline.

Technical Specifications Form

ATTACHMENT A: TECHNICAL SPECIFICATIONS

The System will provide the functionality listed below.

Module A: _____________. *[Insert name of module. Insert each function below.]*

•Function #1: _____________.
•Function #2: _____________.
•Function #3: _____________.

Module B: _______________. *[Insert name of module. Insert each function below.]*

•Function #1: _____________.
•Function #2: _____________.
•Function #3: _____________.

Module C: ______________. *[Insert name of module. Insert each function below.]*

•Function #1: _____________.
•Function #2: _____________.
•Function #3: _____________.

The clause box above is a blank form—one of many options for organizing technical specifications.

The language should be clear and simple. For instance, Function #1 of Module A might read: "Module A will process employment applications and create an Excel spreadsheet for each, in the format described below under 'Reports.'" That's a very functional description. But even if the specs are more technical, the language should remain clear.

Edit specifications the same way you'd edit any contract clause. Make sure each concept gets a simple and clear description. Limit technical jargon. And when the specs must use a technical term, provide a clear definition.

4. Specifications as a Moving Target

In some deals, the specs won't exist until after the contract's signed, or they'll change after the contract's signed. How can the contract deal with requirements that don't yet exist?

In many contracts calling for creation of technology, the parties won't know exactly what the technology's supposed to do until after some kind of "discovery phase." During that phase, the vendor studies the customer's existing systems and needs. Detailed specs won't be available until that's done. But the parties must have *some* vision of the technology when they sign the contract—some idea what the vendor's supposed to create. That vision should be written into the contract, as high-level specs. More detailed specs follow later, as a deliverable, and they become part of the contract once they're accepted.

Specifications as a Deliverable

Vendor shall draft technical specifications for the Software on or before ______ [date], and such technical specifications shall be materially consistent with the requirements of Attachment A of this Agreement (*High-Level Specifications*). Upon acceptance of the proposed technical specifications pursuant to Section __ (*Acceptance of Deliverables*), they shall become the Software's "Technical Specifications," and Vendor shall design the Software so that it materially complies with such Technical Specifications.

If your deliverables include specifications, the contract should include a system for accepting or rejecting them, just like any other deliverable. See Chapter II.F ("Delivery, Acceptance, and Rejection").

Specifications for cloud services might change even without a deliverable. The vendor runs the system, so it can easily modify it. Customers should be wary of an open-ended right to change specs, since the vendor could remove necessary functionality. But cloud vendors usually change their systems to improve them, or to keep up with changing IT environments, and that benefits customers. A simple clause will usually address both parties' needs: "Vendor may revise the Technical Specifications at any time by posting a new version at the Specs Website and giving Customer written notice, provided no such revision may materially degrade the functionality required by this Agreement."[2]

B. Service Level Agreements

"Service level agreement" refers to a clause or set of clauses addressing the performance of a service. An SLA might govern cloud services or a professional service, like software support or maintenance.

Despite the name, a service level agreement generally isn't a separate contract. It could be a separate document incorporated into the contract—maybe something attached or posted at a website. Or it could simply appear in the contract's main body: "Section 10. Service Level Agreement."

SLAs provide minimum standards for services. In a cloud services agreement, the SLA will lay out minimum standards for the cloud-based system: uptime, speed of communications, speed of performance, etc. An SLA might also lay out timelines for the vendor's staff to fix tech failures—in a cloud services agreement, or in a professional services agreement covering tech maintenance, or even in the maintenance section of a software license agreement. Finally, an SLA may lay out credits or other remedies for service failures.[3]

SLAs often refer to or include technical specifications, discussed in the last chapter (Chapter II.A, "Technical Specifications"). If an SLA addresses standards for technology performance, it'll either reference a separate specifications document or include specs within the SLA itself (though it won't always use the term "specifications"). See the first example in the first clause box in Subchapter 1 below.

Subchapter 1 addresses vendor responses to technology failures, including both repairs and credits. Subchapter 2 addresses the customer's right to terminate in response to service failures. Finally, Sub-chapter 3 covers the vendor's right to modify SLAs (or lack thereof).

1. Response, Repair, and Remedy

Most SLAs address the vendor's obligations to acknowledge service requests, to fix broken technology, or to provide credits or other remedies—or some combination of the three.

SLA Response & Remedy Terms

Vendor shall address System faults as follows:

•*Level 1 Error:*Response within __ minutes; Remedy within __ hours.

•*Level 2 Error:*Response within __ minutes; Remedy within __ hours.

•*Level 3 Error:*Response within __ hours; Remedy within __ business days.

As used above:

(a)"Error" refers to any failure of the System to perform as required in the technical specifications set forth _______ (the "Specifications").

(i)"Level 1 Error" refers to ________________________.

(ii)"Level 2 Error" refers to any Error that is not a Level 1 or Level 3 Error.

(iii)"Level 3 Error" refers to ________________________.

(b)"Remedy" refers to a solution that returns the System to material compliance with the Specifications at issue.

(c)"Response" refers to an e-mail, telephone, or in-person acknowledgment of a technical support request.

• • • •

Vendor shall exercise reasonable efforts to achieve ______________ [insert uptime, latency, or other IT performance term] ("Performance") better than _________ [insert minimum Performance] (the "Target"). In the event that average Performance falls below the Target during any calendar month, Vendor shall credit Customer __% of such month's applicable service fees for each ______ [insert Performance metric] below the Target; provided such credit will not exceed __% of any month's otherwise applicable service fees. The credits set forth in the preceding sentence are Customer's sole remedy for Performance below the Target. Credits issued pursuant to this SLA apply to outstanding or future invoices only and are forfeit upon termination of this Agreement. Vendor is not required to issue refunds or to make payments against such credits under any circumstances, including without limitation termination of this Agreement.

SLAs often provide deadlines both for acknowledging support requests and for fixing the underlying problem. See the first example in the clause box above: the deadlines for "Response" and "Remedy."

Many SLAs also provide credits. If the service doesn't work, or if the vendor doesn't fix it fast enough, the customer gets fees deducted from its next bill. See the second example in the clause box above.

Usually, the customer has no remedy other than credits, as in the second example above. But if the contract includes a warranty of function—a warranty that technology will "work"—it should leave the door open for warranty remedies. "The credits set forth in this SLA are Customer's sole remedy for Performance below the Target, except as may be set forth in Section __ (*Warranty of Function*).” Of course, many vendors won’t give both an SLA and a warranty of function. For more in this issue, see Subchapters II.I.1 (“Warranty of Function”) and II.I.6 (“Remedies for Breach of Warranty and Similar Failures”).

Most vendors refuse refunds: the credit can’t be turned into cash back. So if the contract terminates before the customer uses the credit, tough luck. Again, see the second example in the clause box above. For the vendor, that policy preserves revenues and gives customers an incentive to stick around. But the clause could allow refunds: “Vendor shall issue refunds against any outstanding credits issued pursuant to this SLA within __ days of termination of this Agreement for any reason other than Customer’s breach.”

In rare cases, the vendor gets an incentive: extra fees for performance better than required. “If average Performance exceeds the Target during any calendar month, Vendor may invoice Customer for additional fees equal to __% of that month’s applicable services fee for each ______ [insert unit of Performance] above the Target, up to __% of the applicable service fees.” Incentive fees make sense in some cases, but the customer should always ask itself whether better-than-expected performance is worth extra money.

2. The Material Breach Issue

Many SLAs say nothing on a key issue: at what point is service so bad that the customer can terminate the contract? In other words, at what point do service level errors count as material breach of contract?

Vendors generally don’t want customers terminating. But failure to address material breach in the SLA doesn’t necessarily protect the vendor. At some point, bad service will probably authorize termination, even if the contract says nothing on the issue. By addressing material breach in the SLA, vendors can make termination more predictable.

SLA Material Breach & Termination

In the event of __ or more Errors higher than Level 3 during any calendar month, Customer may terminate this Agreement for material breach pursuant to the provisions of Section __ (*Termination for Cause*), provided Customer notifies Vendor in writing of termination within __ days of the end of such calendar month.

The example in the clause box above authorizes termination for breach if service falls below a certain level. It also protects the vendor by setting a time limit on termination. The customer can’t terminate the contract in July because of bad service in January.

3. SLA Modification

SLAs generally regulate services that might change. New technology or evolution in the business environment may render SLA obligations impractical or meaningless. So vendors often insist on the right to change the SLA—unilaterally, without a contract amendment.

If you’re the customer, you might react to this suggestion with some suspicion. But often the change benefits the customer. And some vendors provide the same SLA to all their customers, so they can’t afford to “freeze the SLA in time” on any particular contract.

SLA Revisions

Vendor may revise this SLA on 30 days’ written notice to Customer, but only to the extent that Vendor revises service level agreements for its customers generally.

• • • •

Vendor may revise this SLA by posting a new version at the SLA Website and providing written notice to Customer, provided no such revisions may materially degrade this Agreement’s required performance of the System.

The first example in the clause box above limits SLA revisions to those the vendor applies to all its customers, or at least to the vast majority. So the customer might lose service quality, but it can’t be singled out for a reduction. In theory the vendor could still ruin the service through SLA revisions, but it won’t unless it’s willing to upset all its customers.

The unilateral revision right in the first example could create a problem for the vendor. As Chapter III.S (“Amendment”) explains, courts might not enforce a right to change contract terms without the customer’s consent—and even adding such a clause could, in theory, invalidate the whole contract. Unfortunately, the law isn’t clear on this issue, and the risk depends on your state. In general, I think courts won’t invalidate a contract because of a clause like the first example in the clause box above because it essentially says: “You get the same SLA as all our customers, as it may change from time to time.” That doesn’t sound like a vendor dictating ever-changing terms to helpless customers. Vendors can further limit the risk by avoiding any right to amend other clauses without the customer’s consent and by giving the customer the right to terminate if the SLA really loses value: “Customer may terminate this Agreement for convenience within 30 days of any SLA revision that materially degrades this Agreement’s required performance of the System.”[4]

The second example in the clause box works better for customers. The vendor can change the SLA so long as it doesn’t materially reduce promised performance. I don’t think that clause raises issues about enforceability.

Of course, the vendor could give itself the right to revise the SLA with no restrictions: “Vendor may revise this SLA at any time and in any way, on 30 days’ written notice to customer.” That increases the vendor’s risk of an invalid clause or contract. Again, see Chapter III.S (“Amendment”). If you’re the vendor and you want an unlimited right to revise, get some legal advice about your state’s contract laws—and again consider giving your customers the right to terminate in response to SLA reductions.

C. Maintenance, Including Updates and Upgrades

A software license or cloud services agreement may give the customer the right to maintenance: to a repair service for technology. Many maintenance clauses also include a right to updates and upgrades. So long as it's paying for maintenance, the customer gets new versions of the software, ranging from bug fixes and other minor tweaks to whole new releases. The maintenance and update/upgrade obligations can stand alone—each in its own section—but this book and most contracts put them together.

Maintenance clauses often appear in software license agreements. They also appear in some cloud services agreements, where they play a similar role to service level agreements (SLAs), addressed in the last chapter.[5] The two concepts overlap, and you might see a maintenance clause *within* an SLA or *vice versa*, depending on what names the drafter gives the contract's various sections and attachments. But at heart, maintenance clauses apply to narrower sets of services than SLAs. An SLA may address any type of tech-related service, but a maintenance clause only addresses what you might call "dedicated services." The vendor will fix and update *the customer's copy* of the software. So in cloud services, maintenance clauses make the most sense where the vendor hosts a server or a copy of the software *dedicated to the customer*. You're more likely, then, to see maintenance terms in infrastructure-as-a-service (IaaS) and platform-as-a-service (PaaS) deals than in software-as-a-service (SaaS).

Technically speaking, maintenance is a professional service. So you might find it helpful also to review Chapter I.F ("Promise of Professional Services").

Maintenance

Starting upon delivery of the Software, Vendor shall maintain the Software according to its ____ [Platinum, Gold, Tinfoil, etc.] Maintenance Plan, as further described on Vendor's Website, and Customer shall pay such fees for the ____ Maintenance Plan as are set forth on Vendor's Website.

• • • •

During the Term, (a) Vendor shall exercise commercially reasonable efforts promptly to correct any failure of the System to perform according to its Specifications and (b) Customer shall pay the maintenance fees set forth in Section ___ (*Fees*).

• • • •

During each Maintenance Term (as defined below), Vendor shall maintain the Software so that it performs in material compliance with the Specifications during no less than _____% of each calendar month ("Maintenance"). In exchange for Maintenance, Customer shall pay Vendor $_____ per Maintenance Term, with each payment due __ days before the start of such Maintenance Term. The Maintenance Term shall renew automatically at the end thereof unless Customer gives written notice of its intent not to renew ____ days before the renewal date. After the _______ [2nd, 3rd, 4th . . .] renewal of the Maintenance Term, Vendor may refuse further renewal by written notice _____ days before the next renewal date. ("Maintenance Term" refers to the _______ period following Go-Live. "Go-Live" refers to the earlier of (a) Acceptance of installation and customization of the Software pursuant to Section ___ (*Acceptance*) or (b) Customer's first use of the Software in production.)

Some vendors have standard maintenance plans, so their contracts don't describe maintenance obligations but rather refer to a plan written down somewhere else. See the first example in the clause box above.

Other contracts do describe maintenance obligations. Descriptions favorable to the vendor simply require commercially reasonable efforts to keep the technology working, as in the second example in the clause box above. Other maintenance descriptions set a clear performance standard, as in the third example in the clause box. That's better for the customer, unless the standard is low.

The parties sometimes argue over *when* maintenance starts. Does it start the day the vendor delivers the software, or on the contract's effective date? If the vendor will be customizing the software or needs time for delivery or installation, an early date like either of those could mean the customer pays for maintenance it won't ever use. If the software's not running, or if the vendor's working on installing it, there's nothing to maintain—and the customer probably doesn't need updates or upgrades. So why pay for maintenance? That's why the third example in the clause box above starts maintenance on *go-live.* If you use a clause like that, make sure to define "Go-Live" in a way that fits your deal. You might, for instance, define it as "acceptance of the final Deliverable pursuant to the Implementation Statement of Work," or "the date on which customization and installation are complete and the Software materially performs according to its Technical Specifications."[6] However, if acceptance serves as the trigger, the customer could start using the software before maintenance begins. That's why the third example in the clause box says maintenance starts if the customer begins using the software ("in production," as opposed to testing), even if it hasn't formally accepted it yet.

Some customers push for an even later start date. Maintenance starts at the end of the warranty period. In many cases, a warranty of function is essentially a temporary free maintenance clause, so customers might not need maintenance during the warranty period. See Subchapter II.I.1 ("Warranty of Function").

The vendor usually has a financial incentive to start maintenance, and maintenance *fees*, as early as possible—ideally on delivery or on the contract's effective date, as in the first two examples in the clause box above. And the vendor may have good arguments for starting early—even during the implementation process. It might make certain staff and other resources available through the maintenance process, and customization and installation might be difficult without those resources. Plus, many maintenance clauses include updates and upgrades (discussed below). If maintenance doesn't start until after implementation, or after the warranty period, how does the customer pay for updates added during implementation? Also, for some vendors, maintenance fees "upon delivery" are built into the price of the technology. Maybe the customer really *doesn't* need maintenance until after customization and installation, but license fees or cloud subscription fees would be higher if maintenance fees started later.

There is no right answer. As with most contract terms, the party with the most leverage will probably get the maintenance start date it wants.

• • • •

The right to updates and upgrades might be baked into the vendor's standard written maintenance plan. If not, the contract should spell it out.

Updates and Upgrades

During each Maintenance Term, Vendor shall provide Customer with copies of all new versions, updates, and upgrades of the Software (collectively, "Upgrades"), without additional charge, promptly after commercial release. Upon delivery to Customer, Upgrades will become part of the Software and will be subject to the provisions of Section __ (*License*) and the other provisions of this Agreement.

• • • •

Vendor shall deliver all Minor Upgrades (as defined below) to Customer promptly after release, and such Minor Upgrades will then become part of the Licensed Product. Customer may acquire copies of Major Releases (as defined below) at a __ % discount off Vendor's standard retail price, and such Major Release will become part of the Licensed Product upon Vendor's receipt of payment. "Major Release" refers to any new version of the Licensed Product Vendor releases commercially (at its sole discretion) with a new version number to the left of the version decimal point. "Minor Upgrade" refers to any other new version of the Licensed Product.

The contract will generally state that updates and upgrades become part of the software or other technology, as in both examples in the clause box above.

Often the vendor wants to distinguish between minor upgrades and major new releases. If the customer has rights to *CrashProne* v. 3.00, it should get a free copy of v. 3.04, which corrects bugs and makes minor improvements. But *CrashProne* v. 4.00 has all new features and may cost more. So provisions like the second example in the clause box above require that the customer pay for these "major releases," though sometimes at a discount.

D. Documentation

Proper use of technology often requires documentation. These documents range from user manuals to design descriptions for IT staff. If documentation is necessary, a software agreement should require that the vendor deliver it.

Some software contracts define the “Licensed Product” or “Software” to include “such software’s standard user manual.” With a definition like that, the license and delivery clauses will usually give the customer all the necessary rights. But some software contracts address documentation separately, and cloud services contracts rarely include documentation in the definition of the “Service.”

Documentation Requirements

Upon delivery of the Software, Vendor shall also deliver __ copies of the Software’s standard user manual.

• • • •

Vendor shall draft and provide such documentation as is reasonably necessary to operate the Customized Software (the “Documentation”). Vendor shall deliver the Documentation to Customer upon delivery of the Customized Software and shall revise the Documentation as reasonably necessary in the event of changes to the Customized Software, without further charge. Customer may reproduce the Documentation as reasonably necessary to support internal use of the Customized Software.

The first example in the clause box above addresses standard software, with little or no customization. The second addresses software customized for the customer.

Documentation is written text, just like software, so if the customer needs to make copies, it should get a license to reproduce or otherwise exploit the documentation. See the second example in the clause box.

E. Schedule and Milestones

Some professional services need a schedule or end date. For example, a technology development project will usually have deadlines for designing a computer system, building it, and getting it ready for use. Those deadlines appear in a scheduling clause.

Schedule of Professional Services

Vendor will complete the Services on or before ______________.

• • • •

1.*Milestones*. Vendor will complete the Service by the following deadlines (“Milestones”):

A.Alpha Version functioning according to Specifications (“Operational”): _________ [days] after Effective Date;

B.Beta Version Operational: ________ [days] after Alpha Version Operational;

C.Full System Operational and submitted for Acceptance Testing: _______ [days] after Beta Version Operational.

2.*Payment*. Customer will pay Vendor in the following installments:

A.Milestone A: __ % of the Development Fee

B.Milestone B: __ % of the Development Fee

C.Milestone C: __ % of the Development Fee

D.Acceptance: __ % of the Development Fee

The simplest way to handle scheduling is to provide a deadline for completion of the project, as in the first example in the clause box above. But for a long or complex project, you often need several deadlines, or milestones. See the second example. Often, one of the milestones is “Acceptance” or submission to “Acceptance Testing.” Those terms are usually defined in an acceptance clause. See the next chapter (II.F, “Delivery, Acceptance, and Rejection”).

Milestones can serve as powerful incentives if they’re linked to the customer’s payment obligations, as in the second example above. The vendor may perform much more quickly if it gets paid at each important step—an obvious benefit for the customer. And for the vendor, a milestones payment structure may be the only way to get some of the fee before finishing the job.

A source of dispute hangs over all scheduling clauses. What if the vendor needs the customer's cooperation to finish on time, and the customer doesn't cooperate? What if the vendor needs instructions from the customer, or equipment, or access to the building—and the customer takes forever? The vendor shouldn't be held responsible for the delay. Sometimes you can handle that through a lockstep scheduling clause: "Customer shall provide an instructions memo within 60 days of the Effective Date. Vendor shall complete Phase 1 within 30 days of receiving the instructions memo." In a clause like that, the clock doesn't start ticking on the vendor's performance until after the customer cooperates. In the second example in the clause box above, 84 one of the milestones uses a similar system. Per Subsections 1.C and 2.C, "submission for Acceptance Testing" triggers a payment, even if the customer delays the actual testing.

Often, however, customer cooperation isn't so easily defined. The vendor may need general and miscellaneous cooperation: a million small favors that will make the project run smoothly. This is one of the hardest problems to address in a contract. One solution is extra time for each milestone, so that if the customer fails to cooperate, the vendor can still get done on time. But the vendor will be safer with something like: "All deadlines are subject to such extension as is reasonably necessary if Customer does not cooperate in good faith with Vendor, including without limitation by providing the following forms of assistance: . . ." The problem, of course, is that terms like "cooperate," "good faith," and "reasonably necessary" are vague. Try to draft the scheduling clause as clearly as possible, but recognize that you may have to accept some of these wiggle words.

At some point, the customer's failure to cooperate crosses over into breach of contract, giving the vendor the right to terminate for cause, and maybe also damages (though the latter might take a lawsuit). Vendors can strengthen their termination for breach remedy by specifying the type of delay that justifies it. Vendors should also consider a right to terminate for convenience in response to customer delays, since that's usually less contentious, and it might offer an easier way to get some compensation for losses related to a canceled project.

Termination for Customer's Failure to Cooperate

Customer's failure to make the Facility available to Vendor's personnel for more than __ business days out of any calendar month will constitute a material breach of this Agreement. The preceding sentence will not be interpreted to limit Customer's other failures that may constitute material breach of this Agreement.

• • • •

Customer's failure to provide the staff, facilities, or equipment required by the Statement of Work for any Customer task or joint task, within __ business days of Vendor's written request, will authorize Vendor to terminate this Agreement pursuant to Section __ (*Termination for Convenience*). The termination right set forth in the preceding sentence: (a) shall expire if not exercised within __ business days of Vendor's written request referenced above; and (b) does not limit any right of Vendor to terminate this Agreement for breach or restrict the definition of "material breach."

The first example in the clause box above provides that a particular type of delay qualifies as material breach, justifying termination. The breach in question might qualify anyway, but this way, there's no doubt and, the vendor hopes, no dispute.[7]

The second example gives the vendor the right to terminate for convenience ("without cause"). That may be better than terminating for breach because it's less likely to trigger disputes. See Sub-chapter II.V.3 ("Termination for Convenience"). The second example in the clause box also requires that the vendor exercise its termination right within some set period after the customer's failure to cooperate. That way, the vendor can't keep working and then use the termination right long after, for some unrelated reason.

F. Delivery, Acceptance, and Rejection

Delivery, acceptance, and rejection clauses are appropriate for most software agreements, as well as for cloud services and professional services agreements with deliverables. At their simplest, these clauses provide instructions for the vendor about the time and place of delivery. But some clauses go further and call for “acceptance tests.” The customer can test the software to make sure it works. If the software fails, the customer can reject it, and the vendor usually has to fix it or refund the money.

Delivery, Acceptance, and Rejection

> Vendor shall deliver each Deliverable to Customer’s facility located at ____________ on the following timeframe: ____________ (“Delivery”). Each Deliverable will be considered accepted (“Acceptance”) (a) when Customer provides Vendor written notice of acceptance or (b) _____ days after Delivery, if Customer has not first provided Vendor with written notice of rejection. Customer may reject a Deliverable only in the event that it materially deviates from its Technical Specifications and only via written notice setting forth the nature of such deviation. In the event of such rejection, Vendor shall correct the deviation and redeliver the Deliverable within _____ days. Redelivery pursuant to the previous sentence will constitute another Delivery, and the parties shall again follow the acceptance procedures set forth in this Section ___. Vendor’s failure to provide Deliverables that materially conform to the Technical Specifications may constitute breach of this Agreement, and this Section ___ does not limit any remedy Customer may have for such breach.

The first example in the clause box above is a simple delivery provision. The second is an acceptance clause. Acceptance clauses and acceptance testing are most common for customized software: systems the vendor creates or modifies to fit the customer’s needs. They often appear in contracts that include professional services, where the services involve creation of software or other deliverables.

The vendor shouldn’t give the customer freedom to reject goods for any old reason, or because they don’t meet expectations the customer never mentioned. The clearest test provides that the goods fail if they don’t conform to their technical specifications, as in the second example in the clause box above.[8] But some contracts lay out narrower tests, defining steps the customer can take to test the goods. “The Software will be considered accepted if it passes all three tests listed on Attachment __ (*Acceptance Testing*).”

One risk for the vendor is that the customer will never get around to testing the goods, or will take a long time. That’s particularly troubling if the vendor doesn’t get *paid* until acceptance. That’s why many acceptance clauses have a “deemed acceptance” provision, as in the second example in the clause box. If the customer doesn’t either accept or reject within X days, the goods are *deemed* accepted.

What if the goods fail the test? The contract might require that the vendor fix them, as in the second example in the clause box, or that it refund the customer’s money.

In some cases, the customer wants to terminate the agreement if deliverables fail their acceptance tests. In the clause box above, the last sentence of the second example says that the deliverables’ failure may mean the vendor has breached the contract. The example doesn’t address how many failures or what type would allow the customer 88 **The Tech Contracts Handbook** to terminate—would qualify as “material breach.” That’s because it’s almost impossible for the contract drafter to describe in advance what qualifies. The courts address that on a case-by-case basis, so the clause just clarifies that the contract’s repair and redelivery terms don’t necessarily rule out breach-of-contract claims.

Some contracts give the customer a refund for nonconforming deliverables. If you’re the vendor and you agree to such a refund clause, consider also providing that the customer won’t have any other remedy.

Refund for Failed Deliverables

If a Deliverable does not conform to its Technical Specifications on the third or any subsequent Delivery, Customer may require a refund of fees paid for such Deliverable. Vendor shall provide such refund within 30 days of Customer’s written request, and Customer shall promptly return the Deliverable. The preceding two sentences set forth Customer’s sole remedy for the failure of any Deliverable to conform to its Technical Specifications.

The clause box above does *not* answer an important question: If the project has multiple deliverables, what happens to the rest after one gets rejected? What if the other deliverables won’t work without the rejected deliverable? You could extend the refund right to those other deliverables: “Vendor shall likewise refund the fees paid for any Deliverable Customer received primarily to interface with the rejected Deliverable, if such other Deliverable's functionality is materially impaired by removal of the rejected Deliverable." Obviously, that could easily unravel all the vendor's revenues from the project. The customer might instead need the right to cancel the whole project. And it might also need the right to sue for breach, which would mean the last sentence in the clause box above doesn't work.

Finally, some contracts add another remedy for goods that don't pass acceptance tests, or pass them on time: late fees. The vendor usually "pays" late fees through a credit or partial refund: "In the event of Rejection, Vendor shall credit Customer 1% of the License Fee for every business day until the delivery date of Software that passes the Acceptance Tests, up to a maximum of 25 business days, as liquidated damages and as Customer's sole and exclusive remedy for such 25-day delay." If you add late fees, review Chapter II.Q ("Liquidated Damages") and add the language suggested there.

G. Nondisclosure/Confidentiality

In a nondisclosure clause, one party commits to keep the other's sensitive information confidential, particularly trade secrets. These clauses can appear in almost any IT contract. Nondisclosure terms can also serve as the prime clause in a separate nondisclosure agreement (NDA).[9]

A nondisclosure clause may operate in both directions or only one. In a one-way clause, one party discloses confidential information and the other receives it and keeps it secret. In a two-way or "mutual" clause, either party may disclose or receive confidential information. Usually, two-way clauses call the parties "Discloser" and "Recipient," or something like that. Those names aren't attached to one party but rather rotate. This chapter uses them the same way.

The examples in this chapter are two-way clauses. If you'd like to use them for a one-way provision, delete the definitions of "Discloser" and "Recipient" in the first clause box (on pages 91–92). Then, throughout the clause, replace those terms with whatever names you're using for your parties ("Vendor," "Customer," "Reseller," etc.).

The law actually provides protection for certain sensitive information even if it's not covered by a nondisclosure clause. As Appendix 1 ("Intellectual Property") explains, a "trade secret" is information that's (a) valuable because it's not widely known or easily discovered by people who could use it and (b) subject to reasonable efforts to maintain secrecy.[10] Trade secret infringement breaks the law even without a contract. But trade secrets law only protects against *unauthorized* taking or use of information. If the discloser *gives* the information without any contractual or other restriction, trade secrets law won't help. So nondisclosure clauses play a role in trade secrets protection, by defining authorized and unauthorized use.

Finally, don't confuse nondisclosure clauses with data management and security clauses. Data clauses mostly address electronic data used by *computers*; humans usually get little or no access—and what they do get may be subject to a separate NDA. In a nondisclosure clause, human access is the whole point: the recipient's staff gets the information and promises to keep it confidential. And the two clauses differ in other ways, explained in the next chapter (specifically Subchapter II.H.4, "Data Clauses vs. Nondisclosure Clauses and NDAs").

1. What's Confidential?

The clearest clauses define "Confidential Information" as anything the discloser marks "confidential," as in Subsection (a) in the clause box below. In many contracts, that's enough. In some deals, however, confidential information will be disclosed orally: "OK, what I'm gonna tell you next is confidential, per the contract." Because memories aren't reliable, the clause should require that the discloser confirm any oral designation in writing, as in Subsection (b).

Confidential Information Defined

"Confidential Information" refers to the following items one party to this Agreement ("Discloser") discloses to the other ("Recipient"): (a) any document Discloser marks "Confidential"; (b) any information Discloser orally designates as "Confidential" at the time of disclosure, provided Discloser confirms such designation in writing within __ business days; (c) any source code disclosed by Vendor and any names of actual or potential customers disclosed by Customer, whether or not marked as confidential; and (d) any other nonpublic, sensitive information disclosed by Discloser. Notwithstanding the foregoing, Confidential Information does not include information that: (i) is in Recipient's possession at the time of disclosure; (ii) is independently developed by Recipient without use of or reference to Confidential Information; (iii) becomes known publicly, before or after disclosure, other than as a result of Recipient's improper action or inaction; or (iv) is approved for release in writing by Discloser.

Subsections (a) and (b) in the clause box above require that the discloser mark or designate confidential information, so the recipient knows what it's getting. But Subsection (c) expands the definition by "pre-marking" certain information. The contract itself hereby slaps a "confidential" mark on the information in Subsection (c), so the discloser doesn't have to remember. That protects against careless failure to mark particularly sensitive assets: source code and customer lists, in the clause box example.

Subsection (d) also protects the discloser against its own carelessness. It expands the confidential information definition even further, to any trade secret. It's valuable because the marking requirement actually creates a risk for the discloser. If it discloses a trade secret without marking it, and that information’s not pre-marked per Subsection (c), it could lose its trade secret status. Subsection (d) protects against that outcome, but it comes with a price. The *anything “nonpublic” or “sensitive”* definition isn’t very clear, so it creates a risk of disputes. So if you’re confident you can avoid disclosing trade secrets without marking them, or pre-marking them through the clause itself, delete Subsection (d).

Any clause that lets the discloser decide what’s confidential creates a risk for the recipient. The discloser could designate too much. It could disclose confidential information the recipient doesn’t want—information the recipient can’t easily protect or that restricts its business. If this risk is high (e.g., because of low trust), the clause could include a safety valve: “Before disclosure, Discloser shall provide Recipient with a nonconfidential written summary of any data intended to be Confidential Information. Within 5 business days of receipt of such a summary, Recipient may reject the information in writing. Information disclosed without such a summary, or after such rejection, will not be considered Confidential Information.”

Whatever the definition of confidential information, some data should be excluded. The example above excludes the key types: data the recipient already has or develops independently, and data that’s already public.

2. Restrictions on Use

The core terms of a nondisclosure clause tell the recipient what *not* to do with confidential information. In short, the recipient can’t pass the information on to third parties without permission.

Nondisclosure

Recipient shall not use Confidential Information for any purpose other than to facilitate the transactions contemplated by this Agreement (the “Purpose”). Recipient: (a) shall not disclose Confidential Information to any employee or contractor of Recipient unless such person needs access in order to facilitate the Purpose and executes a nondisclosure agreement with Recipient with terms no less restrictive than those of this Section __; and (b) shall not disclose Confidential Information to any other third party without Discloser’s prior written consent. Without limiting the generality of the foregoing, Recipient shall protect Confidential Information with the same degree of care it uses to protect its own confidential information of similar nature and importance, but with no less than reasonable care. Recipient shall promptly notify Discloser of any misuse or misappropriation of Confidential Information that comes to Recipient’s attention. Notwithstanding the foregoing, Recipient may disclose Confidential Information as required by applicable law or by proper legal or governmental authority. Recipient shall give Discloser prompt notice of any such legal or governmental demand and reasonably cooperate with Discloser in any effort to seek a protective order or otherwise to contest such required disclosure, at Discloser’s expense.

The example in the clause box above has three key directives. First, the recipient won’t use the confidential information for anything but the transaction outlined in the agreement. Second, the recipient won’t give confidential information to anyone not authorized. And third, the recipient will take reasonable steps to protect the information. That three-part structure is common but not required. Some clauses simply forbid disclosure to anyone unauthorized.

The example in the clause box requires that the recipient have its employees sign NDAs if they get access to confidential information. That’s not always necessary. Often the discloser can rely on the recipient’s supervision of employees. If you do need separate NDAs, you can take the clause box examples in this chapter and put them into a separate contract between the employee and the recipient. (In some cases, the discloser wants to sign the employee NDA too, or wants to sign instead of the recipient.) The NDA usually won’t need any other terms except boilerplate clauses, addressed in Part III of this book.[11]

A nondisclosure clause doesn’t have to address procedures for protecting confidential information. Those that do usually require “reasonable care” or something similar, like the example in the clause box above. But there’s nothing to keep you from drafting more detailed standards or procedures. For example: “Recipient shall keep all copies of Confidential Information in a locked safe at its corporate headquarters, with keys or combinations available only to its General Counsel. Recipient shall shred all copies of documents containing Confidential Information within __ days of disclosure."

Finally, most nondisclosure clauses—including the example above—allow the recipient to share confidential information with the government and with courts (including litigants with discovery rights) to the extent required by law. That's a necessary exception.

3. Injunction, Termination, and Retention of Rights

Most nondisclosure clauses give the discloser the right to a court-issued injunction against leaks. And most also address termination of the agreement and "ownership" of information.

Injunction, Termination, and Retention of Rights

(a)*Injunction*. Recipient agrees that breach of this Section __ would cause Discloser irreparable injury, for which monetary damages would not provide adequate compensation, and that in addition to any other remedy, Discloser will be entitled to injunctive relief against such breach or threatened breach, without proving actual damage or posting a bond or other security.

(b)*Termination and Return.* The obligations of Section __ above (*Nondisclosure*) will terminate ___________ after the Effective Date. Upon termination of this Agreement, Recipient shall return all copies of Confidential Information to Discloser or certify, in writing, the destruction thereof.

(c)*Retention of Rights.* This Agreement does not transfer ownership of Confidential Information or grant a license thereto. Except to the extent that another section of this Agreement specifically provides to the contrary, Discloser will retain all right, title, and interest in and to all Confidential Information.

If the recipient leaks the confidential information or threatens to leak it, the discloser will probably want to plug the leak before it's too late. It needs a court order or "injunction" directing the recipient to protect the information. If the contract doesn't address this injunction issue and the parties wind up in court, the recipient could argue that the discloser doesn't really need an injunction—that money damages, granted after the fact, would be enough. To defeat that argument, Subsection (a) in the clause box above has the recipient admit in advance that money damages wouldn't do the trick. The leak would injure the discloser's business in a way that no amount of money could compensate.

Subsection (b) of the clause box puts a time limit on nondisclosure obligations. Keeping secrets can be burdensome, and most secrets grow less sensitive over time anyway. So nondisclosure clauses usually fix an end date for the recipient's obligations. In six months, five years, or whenever, the recipient is off the hook. Subsection (b) sets a consistent end date for all confidential information, but you might instead provide that nondisclosure obligations terminate "___years after disclosure of the item of Confidential Information in question." And of course, you don't need any expiration date, if you think your confidential information will remain sensitive indefinitely. If so, delete the first sentence of Subsection (b).

Finally, the discloser should consider terms requiring return or destruction of confidential documents after the relationship ends, and confirming that the contract doesn't grant a license to confidential information, or transfer ownership. See Subsection (c) and the last sentence of Subsection (b). Of course, if the contract's deliverables include Confidential Information, the clause should create some exceptions. "This Agreement does not transfer ownership of Confidential Information or grant a license thereto, subject to the rights specifically granted in Sections __ (*Software License*) and __ (*Ownership of Deliverables*)."[12]

Some confidentiality clauses also require return of confidential information on the discloser's request. "Recipient shall return any Confidential Information promptly after Discloser's request, retaining no copies thereof."

H. Data Management and Security

Data management and security clauses address information stored or processed by computers. They appear most often in cloud services contracts. They may also appear in professional services agreements and in software licenses with maintenance clauses, where the vendor gets remote access to data. The vendor isn't getting a copy of the data, but its computers can access customer computers that do have a copy, so data management and security concerns still arise.

A data clause should begin by defining its key term: the "Customer Data"—the information to be protected and managed. Subchapter 1 below addresses that definition and its consequences. Then, Subchapter 2 addresses data *management*, while Subchapter 3 addresses data *protection*. The line between those two may be a bit thin, but it breaks this chapter and its concepts into smaller bites. Finally, Subchapter 4 compares data clauses to nondisclosure clauses or agreements (NDAs), to eliminate a common source of confusion.

This chapter mostly offers customer-friendly clause boxes. That's because customers generally benefit from more detailed data terms, and this chapter aims to address all the issues—all the details. So if you're the vendor, view the clause boxes below as guides to the terms your sophisticated customers might want. View them also as *starting points* for your own clause, but then eliminate any terms you don't need to get the deal done, which in some cases will be almost everything. (The sample software-as-a-service form at this book's website, http://TechContracts.com, has a bare-bones data management and security clause more appropriate for vendors, at least as a pre-negotiation form.)

If you're the customer, recognize that the clause boxes below throw in the kitchen sink, or almost, and many vendors won't give anywhere near that much. In fact, in an online form contract, like the ones many cloud services vendors use, you'll get a small fraction of these terms, if any. But even where you can't get all the terms you'd like, the clause boxes below give you a list of topics to consider.

1. Customer Data Defined

Data clauses usually begin by defining “Customer Data” or “Project Data”: the protected information.

Customer Data Defined

”Customer Data” means all information processed or stored on computers or other electronic media by Customer or on Customer’s behalf, or provided to Vendor for such processing or storage, as well as any information derived from such information. Customer Data includes, without limitation: (a) information on paper or other nonelectronic media provided to Vendor for computer processing or storage, or information formerly on electronic media; (b) information provided to Vendor by customer’s customers or other users or by other third parties; and (c) personally identifiable information from such customers, users, or other third parties.

The definition in the clause box above focuses on electronic data, accessible to computers. But protection shouldn’t disappear if someone prints the data, or if the vendor receives it on paper and then inputs it into a computer. That’s the purpose of Subsection (a) in the clause box, and that’s also why the definition includes “any information derived from [electronic] information.”

Subsection (b) clarifies the data’s source. It’s not necessarily directly from the customer, and it doesn’t necessarily relate to the customer. It may be data provided by the customer’s own customers or by its employees or other third parties.

Subsection (c) in the clause box assumes the customer’s data includes consumer *private* information—aka “personally identifiable information,” or “PII”—not just information about the customer’s business. If not, the customer data becomes less sensitive, and the vendor should delete (c). In that case, the vendor should also make sure the contract *excuses* liability for PII: “Customer represents and warrants that Customer Data does not and will not include, and Customer has not and shall not transmit to Vendor’s computers or other media, any personally identifiable information (‘PII’). Customer recognizes and agrees that: (a) Vendor has no liability for any failure to manage or protect PII, including without limitation as required by applicable law; and (b) Vendor’s systems are not intended for management or protection of PII and may not provide adequate or legally required security for PII.” A vendor that doesn’t want PII should also consider an indemnity from the customer, protecting against third party suits about PII leaks, just in case the customer does include unwelcome data. (See Subchapter II.J.4, “Indemnities from the Customer, Distributor, or Reseller”: the second example in the clause box on page 138.)

Usually, however, PII protection—privacy protection—is half the point of a data management and security clause. The rest of this chapter assumes the customer data does include PII.

2. Data Management and E-Discovery

Data management clauses address the customer’s right to control its data—to access and copy it and to require that the vendor retain or erase it. They’re most appropriate for vendors that store or host customer data, even temporarily, like SaaS and other cloud services. Some data management terms, though, also fit vendors that get access to data but never host it, like software vendors providing remote maintenance.

Most large companies operate under data management policies, addressing questions like how long to keep electronic data and when to erase it. These policies may also address “e-discovery.” If the company finds itself in court, other litigants may issue discovery demands for access to its data. Once the company anticipates a lawsuit, it’ll need to place a “litigation hold” on relevant data, to make sure nothing gets erased. (Erasing relevant data can lead to serious court-imposed sanctions.) Obviously, these policies get complicated if a third party holds the data. Data management clauses address that problem.

Data Management

(a)*Access, Use, & Legal Compulsion.* Unless it receives Customer's prior written consent, Vendor: (i) shall not access, process, or otherwise use Customer Data other than as necessary to facilitate the Services; (ii) shall not give any of its employees access to Customer Data except to the extent that such individual needs access to facilitate performance under this Agreement and is subject to a reasonable written nondisclosure agreement with Vendor protecting such data, with terms reasonably consistent with those of this Section __ (*Data Management*) and of Section __ (*Data Security*); and (iii) shall not give any third party access to Customer Data, including without limitation Vendor's other customers, except subcontractors subject to Subsection __(d) below. Notwithstanding the foregoing, Vendor may disclose Customer Data as required by applicable law or by proper legal or governmental authority. Vendor shall give Customer prompt notice of any such legal or governmental demand and reasonably cooperate with Customer in any effort to seek a protective order or otherwise to contest such required disclosure, at Customer's expense.

(b)*Customer's Rights.* Customer possesses and retains all right, title, and interest in and to Customer Data, and Vendor's use and possession thereof is solely on Customer's behalf. Customer may access and copy any Customer Data in Vendor's possession at any time, and Vendor shall reasonably facilitate such access and copying promptly after Customer's request.

(c)*Handling, Retention, & Deletion.* Vendor shall observe the policies attached to this Agreement as Attachments __ (*Privacy Policy*) and __ (*e-Discovery Policy*), including without limitation policies regarding retention and deletion of Customer Data. Customer may revise either such policy by providing new written versions to Vendor; provided Vendor is not required to accept any such revision without reasonable additional compensation if it materially increases Vendor's obligations. Except as permitted in such policy, Vendor shall not erase Customer Data, or any copy thereof, without Customer's prior written consent and shall follow any written instructions from Customer regarding retention and erasure of Customer Data. Unless prohibited by applicable law, Vendor shall purge all systems under its control of all Customer Data at such time as Customer may request. Promptly after erasure of Customer Data or any copy thereof, Vendor shall certify such erasure to Customer in writing. In purging or erasing Customer Data as required by this Agreement, Vendor shall leave no data recoverable on its computers or other media, to the maximum extent commercially feasible. Finally, Vendor shall not transfer Customer Data outside _________ (the "Approved Region") without Customer's prior written consent.

(d)*Subcontractors.* Vendor shall not permit any subcontractor to access Customer Data unless such subcontractor is subject to a written contract with Vendor protecting the data, with terms reasonably consistent with those of this Section __ (*Data Management*) and of Section __ (*Data Security*), specifically including without limitation terms consistent with those of Subsection __(a)(ii) above as applied to subcontractor employees. Vendor shall exercise reasonable efforts to ensure that each subcontractor complies with all of the terms of this Agreement related to Customer Data. As between Vendor and Customer, Vendor shall pay any fees or costs related to each subcontractor's compliance with such terms, including without limitation terms in Section __ below (*Data Security*) governing audits and inspections.

(e)*Applicable Law.* Vendor shall comply with all applicable laws and regulations governing the handling of Customer Data and shall not engage in any activity related to Customer Data that would place Customer in violation of any applicable law, regulation, government request, or judicial process; provided the foregoing does not require that Vendor comply with or be aware of any of the following laws or regulations: ________________________.

(f)*Injunction.* Vendor agrees that violation of the provisions of this Section __ (*Data Management*) or of Section __ (*Data Security*) below would cause Customer irreparable injury, for which monetary damages would not provide adequate compensation, and that in addition to any other remedy, Customer shall be entitled to injunctive relief against such breach or threatened breach, without proving actual damage and without posting a bond or other security.

Subsection (a) in the clause box above provides the contract's key data management terms. The vendor agrees not to use or disclose the data except as necessary. Subsection (a) looks more like a nondisclosure clause or NDA than any of the other data terms in this chapter. In fact, Subsection (a)(ii) requires that the vendor's employees sign NDAs protecting any data they may review.[13]

Subsection (b) confirms the customer's ownership of the data—or at least confirms that the vendor has no independent rights to it. It also calls on the vendor to give the customer access to the data and to allow copying. Copying may be important if litigants send the vendor e-discovery demands for the customer's data. Usually, neither party wants that. The vendor doesn't want the expense and hassle of complying with e-discovery demands, and the customer wants to control the discovery process by providing the data itself. If the customer has a copy of all data in the vendor's possession, it can tell the court that there's nothing to be gained from e-discovery directed at the vendor.

Subsection (c) of the clause box refers to the customer's privacy and e-discovery policies and requires that the vendor obey them. Many contracts, however, incorporate the vendor's policies, rather than the customer's. E-discovery policies generally include instructions about erasing data. But just in case the policy isn't clear, or the customer doesn't have one, the example lets the customer stop any data deletion, in case of a litigation hold (or for other reasons).

The last sentence of Subsection (c) restricts the movement of data around the world. Some customers don't want their data moved out of their own country or out of a limited set of countries and jurisdictions. That's usually because the laws in those jurisdictions do a good job protecting the customer's data, and it's afraid laws elsewhere won't—or because applicable law actually forbids moving the data. If you use a restriction like Subsection (c), make sure to draft a clear definition of the “Approved Region.” “The United States,” “the member nations of the European Union,” and “Australia and New Zealand” are all clear enough. “North America” and “Europe,” on the other hand, leave room for dispute. (Does North America include Central America, Cuba, or Hawaii? Does Europe include Iceland, Turkey, or Russia?)

Subsections (a)(iii) and (d) of the clause box address the vendor’s subcontractors’ access to data. Data management clauses typically require that subcontractors agree to data management and security terms consistent with the clause itself, as in Subsection (d). Some contracts go further and limit data disclosure to subcontractors listed in the contract or approved by the customer. And some go even further: “Vendor shall take all necessary steps to make Customer a third party beneficiary of all subcontractor contracts related to Customer Data and shall provide copies to Customer upon request, provided Vendor may redact contract language not related to Customer or Customer Data.” That sort of “third party beneficiary” provision lets the customer enforce subcontractor contracts against the subcontractor itself, in court. But provisions like that may not be possible where the vendor formed relationships with the subcontractors before the deal with the customer.

If the vendor fails to manage or protect the data, the customer will probably want to plug the leak as soon as possible. It needs a court order or “injunction” telling the vendor to do its data protection job. If the contract doesn’t address this injunction issue and the parties wind up in court, the vendor could argue that the customer doesn’t need an injunction—that money damages granted after the fact would be enough. To defeat that argument, Subsection (f) of the clause box above has the vendor admit in advance that money damages wouldn’t do the trick.

Various laws and regulations impose data security standards on companies that handle private data.[14] In Subsection (e) of the clause box above, the vendor promises to obey those laws. So if the vendor breaks some legal rule on data security, it’s liable to the customer for breach of contract, on top of potential liability to consumers and the government. This double or triple liability makes the most sense where the vendor offers services specifically tailored to the customer’s industry or to the type of data in question. But even there, the vendor might refuse Subsection (e), particularly if it’s not equipped to keep up with changes in the law. Or it might promise only “reasonable efforts” to comply. Of course, the vendor’s not off the hook if it doesn’t promise compliance with laws. It still has potential liability to the government or to consumers for breaking the law; it’s just not necessarily liable to the customer for breach of contract.

Sometimes the compliance clause goes further and lists the laws in question, as well as the other standards. It might list whatever’s then the toughest set of U.S. state laws, and it might even include foreign laws. “Vendor shall comply with the Massachusetts Standards for the Protection of Personal Information of Residents of the Commonwealth (201 CMR 17.00), the EU Member State laws or regulations implementing the European Union Data Protection Directive, 95/46/EC, and the most recent Payment Card Industry Data Security Standard from the Payment Card Industry Data Security Standards Council.”

Even when they do promise to comply with laws, vendors often insist on limiting their obligations. The provision after the semicolon in Subsection (e) of the clause box above says the vendor doesn’t have to comply with certain laws, and in the blank at the end of the sentence, you’d list the excluded laws. Vendors often refuse to promise compliance with data management laws they’re not used to: laws specific to the customer’s business that don’t govern the vendor’s other customers’ data. For instance, if the customer holds data related to stock trading, and the vendor can’t or won’t take responsibility for laws governing that data, the clause could excuse compliance with “the Public Stock Trade Information Handling Act of 1992[15] and any other statute or regulation governing management of data related to publicly traded stock exchanges.”

Usually the vendor should do more than just disclaim any obligation to know laws governing data it doesn’t want. If you’re not set up to comply with a law, you don’t want the data it governs. The data may bring legal burdens even if you weren’t supposed to get it.

Excluded Data

Customer represents and warrants that Customer Data does not and will not include, and Customer has not and shall not upload or transmit to Vendor’s computers or other media, any data (“Excluded Data”) regulated pursuant to ____________________ (the “Excluded Data Laws”). Customer recognizes and agrees that: (a) Vendor has no liability for any failure to provide protections set forth in the Excluded Data Laws or otherwise to protect Excluded Data; and (b) Vendor’s systems are not intended for management or protection of Excluded Data and may not provide adequate or legally required security for excluded data.

The terms in the clause box above parallel the no-PII terms suggested in Subchapter 1 above. As suggested there, the vendor should also consider an indemnity protecting it from third party suits over leaks of excluded data—just in case the customer does include it in the customer data. (See Subchapter II.J.4, “Indemnities from the Customer, Distributor, or Reseller”: the second example in the clause box on page 138.)

3. Data Security

Data security clauses require affirmative steps to protect data and to address data breaches. They apply to vendors that store or host customer data, particularly cloud services vendors. They may also apply to vendors that just access data, or could access it, like software vendors with remote access to customer computers for maintenance, as well as some professional services vendors. But if the vendor doesn’t host the data and has limited, supervised access, data security terms become less important.

If your deal involves data that’s not private or highly sensitive, you might not need a data security clause at all. The data management terms in Subchapter 2 might cover all your needs.

Data Security

Vendor shall exercise commercially reasonably efforts to prevent unauthorized exposure or disclosure of Customer Data. In addition, and without limiting the generality of the preceding sentence:

(a)*DataSec Program.* Vendor shall maintain, implement, and comply with a written data security program (the "DataSec Program") that requires commercially reasonable policies and procedures to ensure compliance with this Section __ (*Data Security*) and with Section __ (*Data Management*). The DataSec Program's policies and procedures shall contain administrative, technical, and physical safeguards, including without limitation: (i) guidelines on the proper disposal of Customer Data after it is no longer needed to carry out the purposes of the Agreement; (ii) access controls on electronic systems used to maintain, access, or transmit Customer Data; (iii) access restrictions at physical locations containing Customer Data; (iv) encryption of electronic Customer Data; (v) dual control procedures; (vi) testing and monitoring of electronic systems; and (vii) procedures to detect actual and attempted attacks on or intrusions into the systems containing or accessing Customer Data. Vendor shall review the DataSec Program and all other Customer Data security precautions regularly, but no less than annually, and update and maintain them to comply with applicable laws, regulations, technology changes, and best practices.

(b)*Employee Background Checks.* Vendor shall not allow any of its employees or subcontractor personnel to access Customer Data except to the extent that such individual has received a clean report with regard to each of the following: (i) verifications of education and work history; (ii) a 7-year all residence criminal offender record information check; and (iii) a 7-year federal criminal offender record information check. (A clean report refers to a report with no discrepancies in education or work history and no criminal investigations or convictions related to felonies or to crimes involving identity theft or other misuse of sensitive information.) However, the requirements of the preceding sentence shall not apply to the extent forbidden by applicable law.

(c)*Audits & Testing*

(i)Vendor shall retain a certified public accounting firm to perform an annual audit of the Services' data protection features and to provide a SOC 2 Type II report, pursuant to the standards of the American Institute of Certified Public Accountants (the "AICPA"). The most current report shall be due to Customer within ___ business days of the Effective Date and thereafter annually within ___ business days of Vendor's receipt from the audit firm. If the AICPA revises its relevant reporting standards, Vendor shall provide the report that then most closely resembles a SOC 2 Type II report. In addition, Vendor shall annually conduct its own internal security audit and address security gaps in compliance with its security policies and procedures, including without limitation the DataSec Program.

(ii)If requested by Customer, Vendor shall, on a quarterly basis: (A) permit security reviews (e.g., intrusion detection, firewalls, routers) by Customer on systems storing or processing Customer Data and on Vendor policies and procedures relating to the foregoing; and (B) permit unannounced inspection of any or all security processes and procedures during the Term, including without limitation penetration tests; provided vendor is not required to permit any review or inspection that may compromise the security of Vendor's other customers or of their data.

(iii)Any report or other result generated through the tests or audits required by this Subsection __(c) will be Vendor's Confidential Information pursuant to Section __ (*Nondisclosure*). If any audit or test referenced above uncovers deficiencies or identifies suggested changes in Vendor's performance of the Services, Vendor shall exercise reasonable efforts promptly to address such identified deficiencies and suggested changes, including without limitation by revising the DataSec Program.

> (d)*Data Breaches*. Vendor shall implement and maintain a program for managing unauthorized disclosure or exposure of Customer Data stored by or accessible through the Services ("Data Breaches"). In the event of a Data Breach, or in the event that Vendor suspects a Data Breach, Vendor shall (i) promptly notify Customer by telephone and (ii) cooperate with Customer and law enforcement agencies, where applicable, to investigate and resolve the Data Breach, including without limitation by providing reasonable assistance to Customer in notifying injured third parties. In addition, Vendor shall provide 1 year of credit monitoring service to any affected individual, unless the Data Breach resulted from Customer's act or omission. Vendor shall give Customer prompt access to such records related to a Data Breach as Customer may reasonably request; provided such records shall be Vendor's Confidential Information pursuant to Section __ (*Nondisclosure*), and Vendor shall not be required to provide Customer with records belonging to, or compromising the security of, its other customers. The provisions of this Subsection __ (d) do not limit Customer's other rights or remedies, if any, resulting from a Data Breach.

Long as it is, the example in the clause box above provides only the skeleton of a data security clause. The flesh comes from the technical and physical security requirements, which will go in the data security program (DataSec Program) mentioned in Subsection (a). That program should address the technical and physical details of data protection: encryption, firewalls, locked computer cages, etc. Like software technical specifications, the technical and physical requirements might be drafted by either party's IT staff, but whoever's ultimately responsible for the deal—lawyer, contract manager, executive, etc.—should make sure they're clear and complete. (See Subchapters II.A.2, "Responsibility for Specifications," and II.A.3, "Organizing and Editing Specifications.")

Of course, the vendor may already have a data security program, and it might not be willing to customize it to the customer's needs. In that case, Subsection (a) could say something like: "Vendor shall comply with its Data Security Program in effect as of the Effective Date, as such program may be modified from time to time, provided no such modification may materially reduce protection of Customer Data."

A few data security clauses require employee background checks, as in Subsection (b) of the clause box above. And some go even further, requiring drug testing. But background checks and drug tests are intrusive, and some vendors won't do them. Each party should consider how much risk comes from rogue employees.

The customer's best assurance of data security usually comes from an outside audit of the vendor's systems. Subsection (c)(i) in the clause box above requires an accounting firm audit based on standards from the

American Institute of Certified Public Accountants (the AICPA). "Accounting firm" and "technology" may sound like an odd match, but the AICPA puts out the U.S.'s most relied-upon standards for reporting on data security. The "SOC 2" report, required in the clause box, refers to the AICPA's "Service Organization Control" standard, number 2. It's designed to test IT security, particularly for cloud computing systems.[16] The clause box also calls for a "Type II" report: another AICPA term, referring to a report that tests the effectiveness of the system's security-related controls over some period, like twelve months. The clause box could instead have called for a "Type I" report, which tests the system's controls at a single point in time and only reports on whether the vendor has the controls it claims, rather than how well they work. Customers don't get as much assurance from a Type I report, but it does have real value, and it costs less.

SOC 2 isn't the only audit option, and it's possible to use more than one. To choose, you need advice from a data security expert, but here's a bare-bones summary. "SOC 1," also known as "SSAE 16," provides an alternative to SOC 2.[17] It resembles SOC 2 but focuses on systems that manage financial data, like pension plans, payroll systems, and accounts receivable. "SOC 3," another alternate audit, also resembles SOC 2, but the report provides less information, and it includes a certification with a high level summary of the report, fit for publication (e.g., on the vendor's website). Whatever sort of SOC audit you use, you can require a Type I or Type II report. Finally, the "ISO 27001" audit standard comes from the International Organization for Standardization (ISO), rather than the AICPA. It's popular in Europe and Asia and provides another set of tests, more focused on technical data security than SOC 1, 2, or 3, and less on business processes.

Some customers also get the right to audit or test their vendors' systems themselves, as in Subsection (c)(ii) of the clause box. Vendors agree to accounting firm audits more readily than to customer testing. And they particularly dislike customer penetration tests: authorized hacking to uncover vulnerabilities. Vendors that do allow customer testing should consider terms protecting their *other* customers' data, as in Subsection (c)(ii) above.

Finally, data security clauses often address data breaches or leaks, as in Subsection (d) in the clause box above. In a data breach, someone hacks into a system holding sensitive data, or steals a laptop holding the data, or otherwise gets unauthorized access. A data breach often creates an emergency, with the parties scrambling to plug the leak, limit harm to themselves and third parties, cooperate with law enforcement, and manage bad publicity. Arguments between customer and vendor about rights and responsibilities can derail that process. The parties should get at least some of the arguing done in advance, during contract negotiations, and make sure the contract provides clear guidance.

In addition to the protections in the clause box above, the customer might want warranty or indemnity terms addressing data security (and the vendor might say "no way"). (See Subchapters II.I.4, "Other Warranties," and II.J.2, "Indemnities from the Vendor.")

4. Data Clauses vs. Nondisclosure Clauses and NDAs

A lot of contract drafters use nondisclosure clauses or agreements (NDAs) for data management and security. After all, nondisclosure clauses (discussed in Chapter II.G, "Nondisclosure/Confidentiality") address protection of sensitive information. But I don't recommend them for data—for information processed or stored by computers. To understand why, let's look at three key differences between nondisclosure clauses and data clauses.

First, as we've seen, data clauses mostly facilitate *computer* use of sensitive information, and they often forbid or limit human use. Nondisclosure clauses facilitate *human* use. The NDA's whole point is to allow human access while imposing obligations to keep the information confidential. So if you used a nondisclosure clause for data, you'd probably have to rewrite it to change its central focus, shifting attention from human use to computer use of data.

Second, as we've seen, data clauses protect all the data in a particular computer system. The parties can't easily figure out what's sensitive, so the clause doesn't try. Nondisclosure clauses, on the other hand, protect *specific* information: material marked "confidential" or designated "confidential" in the contract itself, like secret business plans and source code libraries. So again, you'd have to revise a nondisclosure clause to change its sphere of protection.

Third, data clauses focus on procedures for managing and protecting sensitive information. Nondisclosure clauses, on the other hand, forbid disclosure but generally say little or nothing about procedures. Few nondisclosure clauses require anything more than “reasonable” steps to protect the information, “no less extensive than those Recipient uses to protect its own information of similar sensitivity.” So if you used a nondisclosure clause for data, you’d need to add terms about protection procedures. By the time you finish that revision and the others suggested above, you’ve transformed your nondisclosure clause into a data clause and risked mistakes along the way. Better just to start with a data clause.

In some contracts, you’ll have both nondisclosure and data protection terms, sometimes covering the same information. For instance, you might have a nondisclosure clause forbidding disclosure of secret business plans and a data clause requiring firewalls and other systems to protect data in a computer system. If the secret business plans end up stored in the computer system, both clauses would cover them. And in some data clauses, the vendor agrees to have its employees sign NDAs covering the data, if they’re given access. (See Subsection (a)(ii) in the clause box “Data Management” in Subchapter 2 above, page 100.) Again, both types of clauses would then protect the data.

I. Warranty

A warranty guarantees that something is true or will happen. Or to be more precise, it provides that if the guaranteed fact *doesn't* turn out true, the promisor will be on the hook for some remedy. A warranty can cover any topic, and it can come from either the customer or the vendor, though you'll see vendor warranties more often.

IT contracts offer a variety of warranties, particularly promises that software or other goods will work, at least for a certain period, and guarantees of the vendor's right to transfer intellectual property. Most IT contracts also *disclaim* certain warranties. Finally, many contracts specify remedies for breach of warranty.

In most contracts, one party "represents and warrants" some set of facts. Technically speaking, those two terms have different meanings. A "representation" is a statement of present fact, while a "warranty" is a promise that something will be true in the future. But that distinction plays little role in many modern contracts (and many lawyers don't know about it). As most often used today, both representations and warranties state facts offered to convince a party to enter into a contract. You don't gain or lose much by using one term instead of the other, and there's generally no harm in using "represent and warrant." (Plus, if you're the vendor and you use "represent" without "warrant," your customer will inevitably complain: "Where are my warranties?")[18]

Warranty clauses appear in all types of IT contracts.

1. Warranty of Function

The warranty of function promises that software will "work."

Warranties of function appear in many or even most software license agreements and in lots of cloud services agreements. They appear in some professional services contracts too, but they don't serve much purpose if the vendor isn't providing software or other deliverables that might malfunction. (See Subchapter 3, “Professional Services Warranty and Related Promises,” for warranties that do fit professional services.)

Warranty of Function

Vendor represents and warrants that, during the _________ period following delivery, the Software will perform materially as described in the technical specifications set forth in Attachment __.

• • • •

Vendor represents and warrants that, during the first _________ after installation, each New Module will perform materially according to its documentation issued by Vendor under the heading “Official Product Documentation.”

What does it mean to warrant that software or services will *work*? A clear warranty clause refers to the contract’s technical specifications, as in the first example in the clause box above. In other words, the warranty says that the software will perform as required in the technical specs attached to the contract.[19]

Unfortunately, warranties of function are often much less clear. For instance: “Vendor warrants that the Software will be in good working order.” Ugh. What does that mean? A slight improvement might read that the system will “perform according to its documentation.” But what is documentation? Brochures, e-mails from salespeople, ads posted online . . . ? If you’re going to reference documentation that isn’t attached to the contract, state which documents you mean. For instance, the vendor might put a label or stamp on official documentation, as in the second example in the clause box above. That clause is adequate, assuming the vendor takes care to stamp “Official Product Documentation” in the appropriate places—and assuming there can be no doubt about which documents qualify, those documents are clear, and they explain what the product is supposed to do (like technical specs).

The vendor can warrant just about anything in terms of functionality. Some warranties are customized and address the particular needs of the deal. For instance: “Vendor warrants that no Deliverable, when installed, will impair the System’s ability to process purchase and sales transactions at the speeds set forth in Attachment __ (*Processing Speeds*).”

Vendors usually qualify warranties of function by requiring only "material" conformity with specs or with other requirements, as in the both examples in the clause box above. If every glitch counted as a breach of warranty, the vendor would be in trouble. The use of "material" excludes unimportant errors.

Finally, warranties of function usually have time limits, as in both examples above. If the system stops working the day after the warranty expires (as required by Murphy's Law), the vendor is off the hook. But a time limit isn't required. The vendor could warrant the system indefinitely or "during the term of this Agreement."

If the warranty does include a time limit, the customer should think about when the warranty *starts*. If the vendor is installing software or customizing it, the warranty probably shouldn't start until that job is done. While the vendor's hard at work getting the software installed and working, the customer doesn't need a separate promise that the system *will work*. If the system fails during installation, the vendor already has to deal with it. That's why the last clause provides that the warranty continues for X period "after installation." Even better for the customer, the warranty might continue for X period "after Acceptance of the final Deliverable."[20]

Warranties of function and service level agreements (SLAs) cover similar subject matter. SLAs often make promises about how technology will work too, particularly cloud services, and they often cite specifications.[21] So for technology covered by an SLA, the customer might not need a warranty, and vice versa. However, SLAs and warranties of function often offer different remedies for technology failure. So before giving up on the warranty, or the SLA, the customer should think through its remedies. See Subchapter 6 below ("Remedies for Breach of Warranty and Similar Failures").

2. Intellectual Property/Ownership Warranty

The intellectual property or ownership warranty guarantees rights to technology, particularly rights in intellectual property. It promises that no third party will come along and keep the customer from using the technology through a claim that it infringes a copyright, patent, or other IP right. In other words, the vendor is saying, "We guarantee we have the authority to provide this technology."

IP warranties appear in most types of IT contracts. But like warranties of function, they don't serve much purpose in professional services agreements with no software deliverables or other deliverables that might infringe someone's IP, like copyrightable written content. If the vendor simply provides advice or tech support, or configuration of someone else's technology without adding anything new, the customer doesn't face much risk of an infringement claim related to the vendor's work.

IP Warranty

Vendor represents and warrants that it is the owner of the System and of each and every component thereof, or the recipient of a valid license thereto, and that it has and will maintain the full power and authority to grant the intellectual property and other rights granted in this Agreement without the further consent of any third party.

Some vendors balk at the IP side of the ownership warranty: "How can we guarantee that? There are millions of patents covering all kinds of technologies. How can we possibly be sure our product doesn't infringe one of them?" The answer is that the vendor *can't* be sure. Nor can most vendors be sure their engineers didn't illegally copy a few lines of code, infringing someone's copyright. But vendors don't need to be sure because the warranty isn't about certainty. It's about *allocation of risk*. The typical IP warranty says that the vendor, not the customer, bears the legal risk that the goods infringe some third party's IP. That's often appropriate because it's the vendor's product. The vendor is in a better position than the customer to create safeguards: to hire honest and careful engineers, to run patent searches, to license IP the technology might infringe, etc.

That's not to say the vendor has to accept the risk. It can refuse to guarantee IP rights or other ownership rights. Or it can simply warrant that it doesn't *know* of any IP infringement, limiting its risk. "Vendor represents and warrants that it is not aware of any copyright, patent, or other intellectual property right infringed by the Software, and that it is not aware of any claim of intellectual property infringement related to the Software."

Technology vendors sometimes argue that IP warranties cover the same subject matter as IP indemnities. *Ergo*, since the indemnity seems to be the stronger tool, the customer doesn't need an IP warranty. There's some sense to the argument, but it's based on a misunderstanding, and both customers and vendors will negotiate better armed with the real facts. As we'll see in

Chapter II.J ("Indemnity"), IP indemnities address what happens in litigation: what happens if a third party sues the customer claiming its use of the vendor's technology infringes intellectual property rights. IP warranties, on the other hand, mostly address what happens *after* the litigation, or after an injunction—if the vendor and customer lose to the third party plaintiff, and the customer has to stop using the technology. Without an IP warranty, the customer may then have no claim against the vendor. The vendor has fulfilled its only obligation: the indemnity—defense of the case and payment of judgments. The vendor can walk away, leaving the customer to pick up the pieces after losing what may be mission-critical technology. So customers really do give something up if they get an IP indemnity but no IP warranty.

The vendor's strongest argument is *not* that the IP warranty would add nothing on top of the indemnity. Rather, the indemnity provides *most* of the protection the customer needs—and more would cost extra. Even if the customer had an IP warranty, damages would capped by the limitation of liability clause, and the same clause would provide that the customer can't get consequential damages, which form a big part of the customer's likely loss.[22] Plus, IP warranties often have specified remedies, and vendors can promise those same remedies *without* a warranty. In fact, many vendors do so as a matter of course. For remedies, see Subchapter 6 below.

The customer is still better off with an IP warranty because if it has a breach of warranty claim, it has more leverage over the vendor. But by offering remedies either way, the vendor narrows the gap. In the end, the resolution will come down to who cares more and who has more leverage.

• • • •

A close cousin of the IP warranty appears in some professional services contracts where the vendor creates software or other work product for the customer. Let's call it a warranty of originality.

Warranty of Originality

Vendor represents and warrants that the Work Product will be its own original work, without incorporation of software, text, images, or other assets created by third parties, except to the extent that Customer consents to such incorporation in writing.

In the clause box above, the vendor warrants that it won’t copy third party content—that work product will be original (unless the customer agrees to third party content). That’s just about the same as a copyright warranty, since you can’t infringe copyright without copying—without creating something unoriginal. But the warranty of originality does *not* promise that work product won’t infringe third party patents or other IP rights. It *is* possible for original work to infringe a patent or trademark, through no fault of the creator.[23]

In other words, the warranty of originality doesn’t protect the customer as much as the IP warranties discussed above in this sub-chapter. Customers sometimes accept this more limited protection from vendors who create written work—website content, user manuals, etc.—since written work isn’t likely to infringe a patent (or trademark). But there’s no reason the parties can’t substitute an originality warranty for a more typical IP warranty in any agreement, even one calling for creation of technology.

3. Professional Services Warranty and Related Promises

In a professional services agreement with deliverables subject to IP rights, like software, vendors often give an intellectual property warranty like the ones described in Subsection 2 above. Professional services vendors also warrant the quality of their staff members and services.

Professional Services Quality Warranties

Vendor represents and warrants that all Professional Services will be performed in a professional and workmanlike manner.

• • • •

Vendor shall ensure that all its staff members staffing the Help Desk (a) have received a certification from ________ for operation and maintenance of _________ (“Included Systems”) and (b) have no fewer than __ years’ full-time work experience operating or maintaining Included Systems.

The first example the clause box above is a typical professional services warranty. It promises that the services will be “professional and workmanlike"—a somewhat vague but common term meaning *businesslike and skilled*.

The second example in the clause box above isn't phrased as a warranty because vendors tend to deliver it as a garden-variety promise, though how a court would treat it depends on the state and the situation. It promises professional staff with a minimum level of experience. An experience clause makes the most sense alongside a *task-driven* professional services clause, as Subchapter I.F.1 ("Defining the Professional Service") defines that term. In a task-driven clause, the vendor promises to provide the service, but it doesn't promise any particular outcome (malfunctions fixed, software programmed, etc.). For the customer, then, a promise of qualified staff increases confidence in the service. Qualifications clauses appear in all sorts of professional services agreements.

In an experience clause, avoid vague qualifications like "adequate experience." No one knows what that means. Terms like "industry standard experience" work a bit better, if the industry really has a recognized standard. But the best option is to specify concrete qualifications. In the clause box above, fill in the blanks with the name of the specific system to be maintained or the names of respected training institutions and certification programs.

4. Other Warranties

Warranties can cover almost any topic. The examples in the clause box below are common, but you should craft whatever language fits your deal.

Special Warranties

Each party represents and warrants that it has the full right and authority to enter into, execute, and perform its obligations under this Agreement and that no pending or threatened claim or litigation known to it would have a material adverse impact on its ability to perform as required by this Agreement.

• • • •

Vendor represents and warrants that the Software and any media used to distribute it contain no viruses or other computer instructions or technological means intended to disrupt, damage, or interfere with the use of computers or related systems.

• • • •

Vendor represents and warrants that the Licensed Program does not include software subject to any legal requirement that would restrict Distributor's right to distribute the Licensed Program, or any modification thereof: (a) for a fee, (b) with or without source code or source code rights, or (c) with such restrictions as Distributor sees fit to place on its customers' modification or distribution rights.

• • • •

Vendor represents and warrants that the Services will comply with all applicable laws, including without limitation federal, state, and local.

• • • •

Vendor represents and warrants that it will employ industry standard or better protections to prevent unauthorized disclosure or exposure of personally identifiable information Customer provides to the System.

The examples in the clause box above are mostly self-explanatory. But the third example may confuse you. It protects software distributors against open source software provided with a "copyleft" license (sometimes called a "viral" license). See Appendix 2 ("Open Source Software Licenses").

Remember that warranties don't truly promise a state of affairs. Rather, *they shift legal risk.* So the vendor might not actually be able to guarantee that it won't deliver a computer virus or that its services comply with every conceivable law—as in the second and fourth examples above. But the vendor *can* promise to take the blame for any of those events. It can accept the legal risk.

That said, vendors often refuse that legal risk; they limit their warranties to events they can control. So instead of the virus warranty in the second example in the clause box, for instance, the contract might read: "Vendor represents and warrants that it will analyze each Deliverable with the industry standard antivirus software and will not deliver any Deliverable with a virus discovered by such software or with any other computer instructions or technological means, discovered by such software, intended to disrupt, damage, or interfere with the use of computers or related systems." (That's roughly the standard of the last example in the clause box above, the data security warranty.)

5. Disclaimers of Warranties

Most IT warranty clauses include disclaimers. In fact, for many vendors, the clause's key job is disclaiming warranties, not granting them.

Warranty Disclaimers

EXCEPT FOR THE EXPRESS WARRANTIES SPECIFIED ABOVE IN THIS SECTION __, VENDOR MAKES NO WARRANTIES, EITHER EXPRESS OR IMPLIED, INCLUDING WITHOUT LIMITATION ANY IMPLIED WARRANTY OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

• • • •

CUSTOMER ACCEPTS THE SERVICE "AS IS," WITH NO REPRESENTATION OR WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING WITHOUT LIMITATION IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NONINFRINGEMENT OF INTELLECTUAL PROPERTY RIGHTS OR ANY IMPLIED WARRANTY ARISING FROM STATUTE, COURSE OF DEALING, COURSE OF PERFORMANCE, OR USAGE OF TRADE. WITHOUT LIMITING THE GENERALITY OF THE FOREGOING, VENDOR HAS NO OBLIGATION TO INDEMNIFY OR DEFEND CUSTOMER AGAINST CLAIMS RELATED TO INFRINGEMENT OF INTELLECTUAL PROPERTY RIGHTS.

• • • •

Vendor does not warrant that the Software will perform without error or that it will run without immaterial interruption. Vendor provides no warranty regarding, and will have no responsibility for, any claim arising out of: (a) a modification of the Software made by anyone other than Vendor, unless Vendor approves such modification in writing; or (b) use of the Software in combination with any operating system not authorized in the Documentation or with hardware or software specifically forbidden by the Documentation.

• • • •

Vendor shall: (a) pass through to Customer any warranty right it receives from a third party provider of System components not authored or manufactured by Vendor ("Third Party Components"); and (b) reasonably cooperate with Customer in enforcing such rights, at Customer's expense. VENDOR PROVIDES NO WARRANTIES, EXPRESS OR IMPLIED, WITH REGARD TO THIRD PARTY COMPONENTS, AND VENDOR WILL NOT BE LIABLE FOR ANY FAILURE OF ANY THIRD PARTY COMPONENT TO FUNCTION AS EXPECTED OR INTENDED.

Some software contracts disclaim all warranties. The software is sold "as is" or "with all faults." See the second example in the clause box above. Vendors should be aware that sometimes an "as is" disclaimer won't work, particularly in an agreement with consumer customers. If the deal seems unfair to the consumer in the extreme, some courts will brush aside the "as is" provision.

Whether the contract disclaims all warranties or only some, it should address implied warranties. State laws impose certain warranties on sales contracts, even if the parties don't actually write those warranties into the agreement. The three implied warranties of greatest concern are the *implied warranty of merchantability*, the *implied warranty of fitness for a particular purpose*, and the *implied warranty of noninfringement*. See the first and second examples in the clause box above. Some vendors also disclaim implied warranties related to *course of dealing*, *course of performance,* and *usage of trade*. See the second example in the clause box.

"Merchantability" warrants, among other things, that the goods will do what they're supposed to: they're fit for their ordinary purposes. That makes sense for lamps and toasters, because everyone knows what they're supposed to do. But software and IT services have many complex functions, and reasonable minds can differ about their ordinary purposes. So vendors almost always disclaim the implied warranty of merchantability.

"Fitness for a particular purpose" warrants that a product will be appropriate for the customer's unique needs. In the IT business, the vendor often doesn't fully understand the customer's needs, or know them at all. So vendors specifically disclaim the implied warranty of fitness for a particular purpose.

"Noninfringement" warrants that the product won't infringe third party intellectual property rights. It's not clear that the law actually includes an implied warranty of noninfringement, but vendors should disclaim it, just in case, if they want to avoid the warranty. They should also disclaim any obligation to indemnify the customer for infringement, if that's not part of the deal. See the second example in the clause box above. Obviously, many vendors do warrant IP and do indemnify the customer for IP issues. That's why the first example in the clause box above lacks a noninfringement disclaimer.[24]

Finally, "course of dealing," "course of performance," and "usage of trade" warrant that the vendor will perform the way it usually does or the way members of the industry usually do. See the second example in the clause box above. Again, these implied warranties might not apply to IT, but if you're the vendor, why risk it? Customers, on the other hand, should hesitate before accepting these last three disclaimers. An implied warranty related to usage in trade could be particularly valuable to the customer, since it may require that the vendor live up to IT industry standards.

Disclaimers of implied warranties should be conspicuous, appearing in all caps, as in the first two examples in the clause box above. And for the disclaiming party, it never hurts to put other disclaimers in caps, particularly if they cover an entire product or service. See the last example in the clause box.

Not all disclaimers address implied warranties. One set of alternate disclaimers relates to misuse of the Software. Vendors often disclaim responsibility for customers' unauthorized modification of software. Vendors also disclaim warranties related to customers' use of software on an unauthorized platform or with forbidden hardware or software. See the third example in the clause box above. And for intellectual property warranties, vendors should consider a broad array of disclaimers, similar to those found in many IP indemnity clauses. See Subchapter II.J.3 ("Exclusions from IP Indemnity"), and consider importing the provisions listed there into your disclaimer of warranties. Or you might simply reference your indemnity disclaimers: "The intellectual property infringement warranty in this Section __ does not apply to the extent that the infringement arises out of any of the conditions listed in Subsection __ (*Exclusions from IP Indemnity*)."

Still another common disclaimer relates to third party components. A vendor might design and sell a computer system that includes hardware and software from third parties, as well as its own products. Since the vendor didn't produce the third party components, it might not be able to trust them. Worse, the third parties might have given the vendor a weak warranty, or none. So the vendor could find itself on the hook for someone else's product, alone. The solution is to pass through any third party warranties to the customer and disclaim any other warranty on third party components, as in the last example in the clause box above.

Of course, this sort of pass-through creates problems for the customer. If the system doesn't work, the vendor and third party manufacturer will likely blame each other and refuse responsibility. That's why many customers argue that if the vendor *resells* the third party component, it should take responsibility for it. Otherwise, why doesn't the customer purchase directly from the third party—for less, without the vendor's markup? Customers also argue that the whole reason for hiring a single technology integrator (the vendor) is to get a single point of contact: one party responsible for the system.

Finally, vendors often disclaim specific issues in the warranty clause. For instance: "Vendor does not warrant the Software's interoperability with any transaction reporting system other than *BigBrother* version 7.01." Craft whatever disclaimers fit your deal.

6. Remedies for Breach of Warranty and Similar Failures

A contract doesn't have to specify a remedy for breach of warranty. If it doesn't, a court will impose money damages or other solutions (or more likely the parties will negotiate something). But by agreeing on the remedies in advance, the parties remove much of the element of chance.

You can craft almost any remedy terms—anything that compensates the customer for its loss, limits the vendor's obligations, or both.

Remedies for Breach of Warranty

In the event of breach of the warranty in Section __ (*Warranty of Function*), Vendor shall: (a) repair the Software in question; (b) replace the Software in question with software of substantially similar functionality; or (c) if such attempts do not succeed after ___ days, refund all amounts paid by Customer for such Software. The preceding sentence, in conjunction with Customer's right to terminate this Agreement for breach where applicable, states Customer's sole remedy and Vendor's entire liability for breach of the warranty in Section __.

• • • •

In the event of a breach of the warranty in Section __ (*IP Warranty*), Vendor, at its own expense, will promptly take the following actions: (a) secure for Customer the right to continue using the Software; (b) replace or modify the Software to make it noninfringing, provided such modification or replacement will not materially degrade any functionality listed in the Specifications; or (c) refund __% of the licensee fee paid for the Software for every month remaining in its license term following the date after which Customer is required to cease operation of the Software. The remedies set forth in the preceding sentence are not exclusive of any others Customer may have.

• • • •

In the event of a breach of the warranty in Section __ (*Services Warranty*), Vendor, at its own expense, shall promptly re-perform the Services in question. The preceding sentence, in conjunction with Customer's right to terminate this Agreement for breach where applicable, states Customer's sole remedy and Vendor's entire liability for breach of the warranty in Section __.

Usually, the vendor promises to repair or replace defective goods, as in the first example in the clause box above. And for infringement warranties, the vendor promises it will get the customer a license to keep using the goods, or replace them with something noninfringing, as in the second example. Finally, in both cases, the vendor generally promises to refund the customer's money if those remedies fail—if it can't fix or replace or license the technology.

Distributors should consider terms extending the vendor's warranty remedies to their customers. For instance: "Vendor's obligations set forth in this Subsection __ include, without limitation, repair or refund of Software provided to Distributor's customers."

In the first example in the clause box above, the customer gets a full refund if the vendor can't fix the software or replace it. That's common for a warranty of function. In most cases, the warranty lasts for a short time, so if the technology doesn't work, the customer probably hasn't gotten much use out of it. But for longer warranties of function, and for IP warranties, vendors often give a pro rata refund, as in the second example in the clause box above. The customer may have gotten months or years out of the technology by the time a warranty problem crops up. A full refund would mean the customer pays nothing for that time. So the vendor might promises to return an amount proportionate to the time left in the license term. If the License term is 36 months and the customer has to stop using the technology after 12, the refund would be equivalent to 24 months' fees: 24/36—aka 2/3, or 66.6%—times the license fee.

For a services warranty—a warranty of professional and workmanlike services, like the first example in Subchapter 3's clause box—vendors often limit remedies to re-performance. See the third example above. Without that limit, the vendor could find itself in a rat's nest of demands to fix problems springing indirectly from bad services.

In most tech contracts, warranty remedies are exclusive, as in the first and third examples in the clause box. If the vendor provides one of the remedies in the clause, the customer can't sue for more (or at least, can't win), though it can terminate the contract. Vendors should always specify exclusivity of warranty remedies if possible. But exclusivity creates a problem for the customer. What if the vendor can't repair or license around the functionality or IP problem, and the customer has to stop using the technology? The remedies may give the customer a refund, or a partial refund, but that might not make the customer whole. Imagine the product is a bookkeeping system, and the customer has already thrown away the old system. If the customer has to stop using the new one, it has *no* bookkeeping system, and it's in trouble—trouble a refund won't solve. That's why some customers ask for terms like the second example in the clause box above, which lets the customer seek damages. That should motivate the vendor to go the extra mile looking for a solution, and it lets the customer go after damages if the problem really can't be solved.[25]

If the contract includes a service level agreement (SLA), terms about exclusive remedies should recognize that the SLA might provide remedies too: "The remedies listed in Section __ (*Remedies*) state Customer's sole remedy and Vendor's entire liability for breach of the warranty in Section __ (*Warranty of Function*), in conjunction with Customer's right to terminate this Agreement for breach, where applicable, *and any remedy set forth in the SLA*" (emphasis added). Of course, that assumes the customer gets both SLA and breach of warranty remedies for the same malfunction. The two clauses' remedies overlap, but they usually focus on separate loses, so overlapping remedies might make sense. Most tech contracts provide SLA remedies to compensate customers for temporary malfunctions, while warranty remedies compensate for complete failure. For an SLA failure, the customer gets a small credit, usually, but keeps the technology. For an unfixable breach of the warranty of function, the customer gets its money back, usually, and stops using the product. Still, either set of remedies could intrude on the other's territory, so they may overlap enough that it doesn't make sense to include both in the contract (as noted in Subchapter 1 above).

You don't actually need a warranty to provide the types of remedies usually offered for breach of an IP warranty, as Subchapter 2 notes. Vendors often refuse a literal intellectual property warranty but offer the customer *remedies* for IP infringement. In other words, the vendor doesn't warrant that the technology won't infringe third party IP, but it does promise to replace the software if an IP problem crops up, or to get a license or provide a refund—and it specifies that those are exclusive remedies. For the vendor, this strategy avoids breach of warranty claims and other troubling fallout from a warranty, while still offering the customer a solution.

IP Remedies without a Warranty

In the event the Software infringes the intellectual property rights of a third party, Vendor, at its own expense, shall promptly take the following actions: (a) secure for Customer the right to continue using the Software; (b) replace or modify the Software to make it noninfringing, provided such modification or replacement does not materially degrade any functionality set forth in the Specifications; or (c) refund __% of the licensee fee paid for the Software for every month remaining in its license term following the date after which Customer is required to cease operation of the Software. In conjunction with Section __ (*IP Indemnity*) and Customer's right to terminate for breach where applicable, the preceding sentence states Vendor's sole obligation and liability, and Customer's sole remedy, for potential or actual intellectual property infringement by the Software.

Warranty-free IP remedy provisions often appear in the IP indemnity clause. See Subchapter II.J.2 ("Indemnities from the Vendor").

J. Indemnity

In an indemnity clause, one party promises to protect the other from lawsuits and other claims: to hire and pay lawyers and to pay judgments or settlements. Indemnity clauses are appropriate for software, cloud services, and IT professional services contracts.

Which party should give an indemnity, and for what? There is no standard answer, but in general, indemnity clauses make sense when one party faces a significant risk of third party claims, just by virtue of doing business with the other. IT contracts often include an intellectual property indemnity from the vendor, for instance. That's because the customer buys a risk of IP litigation just by using the vendor's technology. The vendor hasn't necessarily done anything wrong to create the risk. If it provides an indemnity, it's because the risk comes from its business and technology, not from the customer's. Also, the vendor can limit the risk more easily than the customer, by hiring careful engineers, running patent searches, etc.

So the vendor often serves as the "indemnitor," the party making the promise, while the customer is the "indemnified party" (aka the "indemnitee"). But there's no reason the vendor shouldn't *get* an indemnity, particularly where *the customer's* business creates a risk of claims and lawsuits. Customers sometimes indemnify vendors against personal injury claims, for instance. If one of the vendor's contractors gets hurt while working at the customer's plant and makes a claim against the vendor, the customer takes responsibility for the case.

I think IT customers and vendors negotiate (and fight over) indemnities more than any other clause. They raise complex issues that most lawyers misunderstand. That's why this is the longest chapter in this book. Subchapter 1 addresses the basics of indemnity. It provides the concepts used in the other subchapters. Subchapter 2 addresses the most common indemnities: those from the vendor. Subchapter 3 continues the discussion of vendor indemnities and addresses typical exclusions from the vendor's IP indemnity. Sub-chapter 4 addresses indemnities from the customer. Finally, Subchapter 5 addresses mutual indemnities.

This chapter does *not* address a set of terms frequently added to the IP part of the indemnity clause: remedies for infringement of intellectual property. This chapter limits itself to indemnities, which in the IP field address claims of infringement and any settlements or judgments that resolve those claims. Remedies clauses, on the other hand, address what happens *after* the claim is resolved, or at least partly resolved—if the vendor/indemnitor loses the legal battle and the customer has to stop using the technology. This book addresses those issues and clauses in the warranties chapter, in Subchapter II.I.6 ("Remedies for Breach of Warranty and Similar Failures").

1. Indemnity in General

This subchapter explains how indemnities work, without getting into specifics about any particular *type* of indemnity. So it refers to the "Indemnified Claims" but doesn't address whether the claim in question relates to IP or data breach or whatever. We'll address specific indemnified claims in Subchapters 2 and 4.

In most indemnity clauses, the indemnitor promises to "defend, indemnify, and hold harmless" the indemnified party. So even though IT professionals and this book call the clause an *indemnity*, "indemnify" isn't the only key promise.

In addition to listing the indemnitor's obligations, most indemnity clauses provide procedures for handling claims.

Generic Indemnity Clause

(a)*Indemnity.* Indemnitor shall defend and indemnify Indemnified Party and its Indemnified Associates (as defined below) against any Indemnified Claim (as defined in Subsection __, *Indemnified Claims*). Indemnitor's obligations set forth in the preceding sentence include retention and payment of attorneys and payment of court costs, as well as settlement at Indemnitor's expense and payment of judgments. (The "Indemnified Associates" are the Indemnified Party's officers, directors, shareholders, parents, subsidiaries, agents, successors, and assigns.)

(b)*Litigation.* Indemnitor's obligations set forth in Subsection __(a) above will be excused to the extent that Indemnified Party's or any Indemnified Associate's failure to provide prompt notice of the Indemnified Claim or reasonably to cooperate materially prejudices the defense. Indemnitor will control the defense of any Indemnified Claim, including appeals, negotiations, and any settlement or compromise thereof; provided Indemnified Party will have the right, not to be exercised unreasonably, to reject any settlement or compromise that requires that it admit wrongdoing or liability or subjects it to any ongoing affirmative obligations.

In Subsection (a) of the clause box above, the indemnitor promises to *defend* and *indemnify* the indemnified party against certain third party claims. ("Third party claims" refers to claims brought by someone outside the contract: neither the indemnitor nor the indemnified party.) "Defend" means hire and pay attorneys to defend the claim. "Indemnify" means reimburse for judgments—for liability to the third party who filed suit. "Indemnify" generally implies an obligation to pay settlements too, but the indemnified party should spell it out, again as in Subsection (a) above.

Most indemnity clauses also have a *hold harmless* requirement: "Indemnitor shall defend, indemnify, and hold harmless Indemnified Party . . . ." Authorities differ on whether "hold harmless" is redundant, adding nothing to "defend and indemnify." A few courts have ruled that "hold harmless" does mean something: the indemnitor can't sue the indemnified party for losses related to the third party claim. Imagine the customer is the indemnified party, and it uses the software in a way forbidden by the contract—a way that infringes a third party's IP rights. If the contract has an "indemnify and defend" promise covering any IP claim, the vendor/indemnitor would have to defend the claim. But it could still sue the customer for breach of contract, for using the software in a way the contract forbids. However, if the contract also says the vendor/indemnitor will "hold harmless" the customer, it can't; it has waived that claim. At least, some courts have taken that position.[26]

The examples in this book don’t use “hold harmless.” If it adds nothing to “defend and indemnify,” it’s useless. And if it does have the meaning suggested above, it doesn’t fit most IT vendors’ and customers’ expectations. But if you’re the indemnified party, consider adding “hold harmless.” And if you’re the indemnitor, check your state’s law on whether “hold harmless” actually adds anything. If not, it does no harm.[27]

• • • •

Most indemnity clauses require that the indemnified party give the indemnitor prompt notice of a claim and let the indemnitor run the defense, including settlement negotiations. But many give the indemnified party authority to reject any settlement that requires that it take some affirmative action in the future, like reporting to the third party plaintiff on its use of software. See Subsection (b) in the clause box above. Some indemnified parties also demand the right to veto any settlement that “restricts any of Indemnified Party’s rights under this Agreement.” (In an IP indemnity, covered in Subchapter 2 below, the vendor/indemnitor should then clarify that “rights under this Agreement” do not include the right to use the technology subject to the claim. To settle an IP claim, the vendor/indemnitor may *have* to agree that its customers won’t use the technology.)

Most indemnity clauses require that the indemnitor indemnify and defend both the indemnified party and its employees, officers, and insurers. See the “Indemnified Associates” definition in Subsection (a) of the clause box above. Customers planning to give their affiliates technology access should add them to the definition too: “The ‘Indemnified Associates’ are . . . any subsidiary, parent, or other affiliate of Customer authorized to use the System pursuant to this Agreement.” Distributors and resellers should think through the definition too, when they’re indemnified parties. Does the indemni-tor—the tech vendor—have to protect the distributor’s *end customers* against IP suits, or just the distributor itself? Those end customers will look to the distributor for protection, so without a broad enough definition of indemnified associates, the distributor could find itself dealing with a case against its end customers with no help from the ultimate vendor, who created the technology. Wherever possible, distributors should add “Distributor’s licensees” or “end user customers” to an “Indemnified Associates” definition.

In many states, indemnity obligations don't apply when the indemnified party's negligence triggers the claim, unless the contract specifically says so. You might think the indemnified party should *never* get protection against cases related to its own negligence, but in some deals that protection actually makes sense, as we'll see in Sub-chapters 4 and 5 below. For those deals, add the following: "Indemnified Claims include, without limitation, claims arising out of or related to Indemnified Party's own negligence." (We'll see these negligence terms in action below, in Subchapters 4 and 5.)

Finally, the indemnity may do the indemnified party little good if it's restricted by the limitation of liability clause. The cost of a patent infringement suit, for example, usually far exceeds the demands permitted by a limit of liability. So IT contracts often—but not always—exempt the indemnity clause from the limit of liability. Those terms usually appear in the limit of liability section, rather than the indemnity clause.[28]

2. Indemnities from the Vendor

This subchapter fills in the definition of "Indemnified Claims" in clauses where the vendor is the indemnitor. Instead of "indemnitor" and "indemnified party," this subchapter simply uses "vendor" and "customer." And to avoid confusion, this subchapter uses "Customer Associates" instead of the customer's "Indemnified Associates." (Some contracts use "Customer Indemnitees" instead.)

Indemnities from the Vendor[29]

Vendor shall defend and indemnify Customer and the Customer Associates (as defined _____) against any "Indemnified Claim," meaning any third party claim, suit, or proceeding arising out of, related to, or alleging: (i) infringement of any patent, copyright, trade secret, or other intellectual property right by the System; (ii) injury to or death of any individual, or any loss of or damage to real or tangible personal property, caused by the act or omission of Vendor or of any of its agents, subcontractors, or employees; or (iii) disclosure or exposure of personally identifiable information or other private information caused by the act or omission of Vendor or any of its agents, subcontractors, or employees.

The most common indemnified claim is an IP infringement claim against the customer, addressed in Subsection (i) of the clause box above.[30] The customer risks a patent, copyright, or trade secret claim or lawsuit by using the vendor's technology, and the vendor/ indemnitor promises to protect the customer against such a claim.

Some indemnity clauses address claims about personal injuries or property damage. In Subsection (ii) of the clause box above, the concern is that, while providing professional services, the vendor's staff will drop a computer on someone's foot, or sexually harass someone, or burn down a building. An injured third party makes a claim against the customer, and the vendor/indemnitor takes responsibility for the case.

Contracts for cloud services sometimes provide an indemnity for leaks of private data, as in Subsection (iii) in the clause box above. If the customer receives private data from consumers or other third parties (social security numbers, credit card numbers, etc.), and puts it on a cloud vendor's computers, the customer will likely get sued for any leak from those computers. Again, the vendor/indemnitor promises to step in and protect its customer.

Most cloud vendors refuse to give a data security indemnity. On the other hand, a few cloud *customers* not only insist on the indemnity but add broader terms than the example above. Subsection (iii) of the clause box only covers claims related to the vendor's (and its contractors') *acts and omissions*. That might not include cases involving leaks from the vendor's computers that can't easily be blamed on the vendor's acts or omissions—including leaks resulting from *the customer's* actions. So some indemnified claims definitions include cases involving "disclosure or exposure of personally identifiable information or other private information from Vendor's or its contractor's computers."[31] Customers argue that even if the customer somehow triggers the leak while using the vendor's systems, the vendor should indemnify and defend because it's responsible for creating a system that protects data *against customer mistakes*. To put the argument another way, the vendor is responsible for making the system idiot-proof.

3. Exclusions from IP Indemnity

Vendors should try to exclude certain intellectual property claims from their indemnity obligations. The point is to avoid covering claims the customer brings on itself.

Occasionally, the customer grants an IP indemnity. The terms and suggestions in this subchapter work for those indemnities too, though the text below treats the vendor as the indemnitor.

Four Exclusions from IP Indemnity

Vendor's obligations set forth in Section __ (*IP Indemnity*) do not apply to the extent that an Indemnified Claim regarding intellectual property infringement arises out of:

(a)Customer's breach of this Agreement;

(b)revisions to the Software made without Vendor's written consent;

(c)Customer's failure to incorporate Software updates or upgrades that would have avoided the alleged infringement, provided Vendor offered such updates or upgrades without charges not otherwise required pursuant to this Agreement;

(d)Vendor's modification of the Software in compliance with specifications provided by Customer; or

(e)[See next clause box below.]

Obviously, the vendor should try to avoid indemnities for IP suits triggered by the customer's breach of contract or by unauthorized software modification. See Subsections (a) and (b) in the clause box above. Also, if the customer has refused a software update or upgrade that would have avoided the IP claim, it's arguably the customer's own fault—and again the vendor should avoid indemnity obligations. See Subsection (c) in the clause box above.

Even if the vendor created the technology that triggered an IP suit, it should avoid an indemnity obligation if the customer *thought up* the technology—if the customer provided the specifications. See Subsection (d) in the clause box above. And remember that the example in the clause box only excludes the indemnity "to the extent that" the customer's specifications trigger the infringement claim. So if the claim really results from the way the vendor chose to build based on the specifications—using unlicensed software, for instance—the indemnity should still apply. But if you're the customer, you might want to clarify by adding the following to the end of Subsection (d): "provided the exception in this Subsection __(d) does not apply if the alleged infringement results from the Vendor's method of implementing Customer's specifications, rather than as an inevitable result of implementing such specifications."[32]

Vendors should also limit or avoid responsibility for IP suits triggered by the software's use in combination with someone else's technology.

The Fifth Exclusion from IP Indemnity: Combination

Vendor’s obligations set forth in Section __ (*IP Indemnity*) do not apply to the extent that an Indemnified Claim regarding intellectual property infringement arises out of: . . . [Choose one version of Subsection (e) below.]

(e)use of the Software in combination with hardware or software not provided by Vendor.

• • • •

(e)use of the Software in combination with hardware or software not provided by Vendor, unless the Documentation or Specifications refers to a combination with such hardware or software (without directing the user not to perform such a combination).

• • • •

(e)use of the Software in combination with hardware or software not provided by Vendor, unless: (i) the Documentation or Specifications refers to a combination with such hardware or software (without directing the user not to perform such a combination); or (ii) such combination achieves functionality described in the Documentation or Specifications (and neither the Documentation nor Specifications directs the user not to perform such combination).

Most IT vendors refuse all IP indemnities related to combinations of their technology with a third party’s. See the first example in the clause box above. That’s the best solution for the vendor, but customers should think it through. What if the IP claim results from combination with a system the customer *had* to use to operate the vendor’s technology? What if the customer had to use the third party browser, operating system, spreadsheet system, or whatever? Most customers would expect a defense and face a rude awakening if the vendor says, “not covered.”

In most IT deals, the customer just lives with that risk (usually without realizing it). But where possible, customers should try to limit the combination exception. In the second example in the clause box above, the customer gets protection for claims resulting from combination of the vendor’s technology with third party technology recommended in the documentation or specifications. So if the vendor’s documentation says, “Run the Software on an ACME operating system,” the IP indemnity would cover a combination with an ACME operating system.

A very few clauses go further, requiring an indemnity for any combination that *achieves functionality* described in the specifications. The third example in the clause box above is the best for the customer. There, the documentation doesn’t have to name the third party technology. If the documentation says the vendor’s technology can achieve some function, and the customer combines the technology with a third party system to achieve that function, the indemnity covers the combination. So if the documentation says, “Here’s how to download data into a spreadsheet report,” the indemnity would cover the combination of the vendor’s system and third party spreadsheet software, whether or not the documentation names the spreadsheet software in question. Obviously, vendors significantly increase their risk through terms like those.

None of the examples in the clause box above provides an indemnity for combinations resulting from the customer’s creative, off-the-reservation use of the technology. If you’re the vendor and you *have* to give some type of combination indemnity, that’s a line you might want to draw in the sand. If the customer combines your technology with a third party system to achieve functionality never suggested in the documentation or specifications, you’re not responsible for any resulting IP suits.

4. Indemnities from the Customer, Distributor, or Reseller

In this subchapter, the customer serves as the indemnitor, or in some cases the distributor or reseller does. So instead of “indemnitor,” this subchapter refers to the “customer,” “distributor,” or “reseller.” And instead of “indemnified party,” it uses “vendor,” while instead of “Indemnified Associates,” it uses “Vendor Associates.”

Indemnities from the Customer [33]

Customer shall defend and indemnify Vendor and the Vendor Associates (as defined ____) against any "Indemnified Claim," meaning any third party claim, suit, or proceeding arising out of, related to, or alleging: (i) infringement or violation of a copyright, trademark, trade secret, privacy, or confidentiality right by written material, images, logos, or other content uploaded to the System through Customer's account; (ii) that use of the System through Customer's account harasses, defames, or defrauds a third party or violates the CAN-Spam Act of 2003 or any other law or restriction on electronic advertising; or (iii) injury to or death of any individual, or any loss of or damage to real or tangible personal property, caused by the act or omission of Customer or of any of its agents, subcontractors, or employees.

• • • •

Customer shall defend and indemnify Vendor and the Vendor Associates (as defined ____) against any third party claim, suit, or proceeding arising out of, related to, or alleging exposure or disclosure of personally identifiable information or other private information input into the System through Customer's account (whether such data belongs to Customer, to one of Customer's customers or users, or to other third parties) (any "Indemnified Claim"). Indemnified Claims include, without limitation, claims arising out of or related to Vendor's negligence, provided that to the extent that a court holds that injuries result from Vendor's negligence, Customer's obligation to pay judgments or settlements shall be excused.

• • • •

Customer shall defend and indemnify Vendor and the Vendor Associates (as defined ____) against any third party claim, suit, or proceeding arising out of or related to Customer's alleged or actual use of, misuse of, or failure to use the Services, including without limitation (i) claims by Customer's users, subscribers, and employees, as well as by Customer's own customers, and (ii) claims related to unauthorized disclosure or exposure of personally identifiable information or other private information (collectively, any "Indemnified Claim"). Indemnified Claims include, without limitation, claims arising out of or related to Vendor's negligence.

Imagine the customer hauls toxic chemicals, and the vendor provides software that tracks the customer's trucks. People injured in a chemical spill might sue both the customer and any supplier remotely involved in managing the trucks, like the vendor. So the vendor wants a personal injury and property damage indemnity, as in Subsection (iii) of the first example in the clause box above.

SaaS and other cloud services vendors sometimes ask for indemnities related to customers' online conduct. In many cloud systems, customers upload written content and logos: online postings, artwork, e-mail, etc. A third party might claim that material infringes its copyright, trademark, or trade secret. Customer content might also harass a third party or violate anti-spam laws. So many cloud vendors ask for a content indemnity, as in Subsections (i) and (ii) of the first example in the clause box above.[34]

Some cloud services vendors also ask for a data security indemnity. This takes most customers by surprise or even offends them. After all, isn't the vendor supposed to be protecting the customer's data, not vice versa? (Subchapter 2's clause box has a data security indemnity from the vendor.) Cloud services vendors who ask for these indemnities generally accept whatever data the customer provides and can't review it to decide whether it's worth the risk. The customer could be putting explosively sensitive information on the vendor's computers, like thousands of social security numbers or children's addresses (or plans for building nuclear weapons). From the vendor's point of view, the indemnity serves as the customer's price for this open-lidded black box, and the service would cost more without it. So even though the vendor might leak the data and injure the customer's end users, *the customer* defends and indemnifies. See the second example in the clause box above. Of course, many customers push back.

Some cloud services vendors go further and ask for an indemnity covering *any* claim by the customer's end users. If the customer serves its end users through the vendor's cloud system, the vendor risks suits by those end users for anything that goes wrong. The indemnity puts all those suits on the customer's shoulders. Of course, the customer/indemnitor could find itself defending and indemnifying the vendor even though the vendor caused the problem, as with the data security indemnity discussed above. The same justification applies: the cloud services vendor makes the system available to the customer without figuring out the risk of lawsuits, and the indemnity serves as the price of that open door. See the third example in the clause box. Again, many customers push back.

As explained in Subchapter 1, some states won't enforce indemnity obligations triggered by the indemnified party's negligence, unless the contract specifically includes those claims. So vendors that want full protection from their customers should specifically include claims based on their own negligence. The third example in the clause box gives the vendor that sort of total protection. The second example offers more limited protection, clarifying the customer/indem-nitor's obligation to *defend* claims about the vendor's negligence, but excusing any obligation to pay damages for resulting injuries once a court has found the vendor at fault.

This vendor negligence indemnity *really* sounds wrong to most customers. But the logic supporting a data security or end user indemnity in the first place supports this too. In each case explained above, the vendor accepts data or user relationships without figuring out its risk. The price is protection from the customer—even against the vendor's own negligence. But of course, the customer might refuse to pay that price.

• • • •

Software distributors and SaaS resellers sometimes indemnify their vendors too. Simply by doing business, the distributor/reseller creates all kinds of lawsuit risks. And the risks grow if the distributor makes wild claims in marketing materials or modifies the software or documentation. An injured customer could easily drag the vendor into a lawsuit, even if the vendor has no direct relationship with the customer.

Indemnities from the Distributor [35]

Distributor shall defend and indemnify Vendor and the Vendor Associates (as defined _____) against any third party claim, suit, or proceeding by any customer of Distributor, as well as any such customer's employee, contractor, or other end user, (any "Indemnified Claim"), except to the extent that such claim, suit, or proceeding arises out of, relates to, or alleges: (i) intellectual property infringement by the Software; or (ii) an injury caused by the Software's failure to conform to its Documentation or Specifications provided by Vendor. For the avoidance of doubt, Indemnified Claims include claims related to injuries caused by the Software's failure to perform as represented by Distributor but not by the Documentation or Specifications provided by Vendor.

5. Mutual Indemnities

In a mutual indemnity, each party promises to protect the other against the same type of claim. In a data breach mutual indemnity, for instance, each party promises to provide protection against claims related to data breaches that party caused, or breaches it's accused of causing. In an IP indemnity, each promises to provide protection against infringement claims related to technology or content that party provided.

Mutual Indemnities [36]

> Each party ("Indemnitor") shall defend and indemnify the other party ("Indemnified Party") and its Indemnified Associates (as defined ____) against any third party claim, suit, or proceeding arising out of, related to, or alleging any of the following (any "Indemnified Claim"): (i) unauthorized disclosure or exposure of personally identifiable information resulting from Indemnitor's acts or omissions or from those of its contractors; (ii) injury to or death of any individual, or any loss of or damage to real or tangible personal property, resulting from Indemnitor's acts or omissions or from those of its contractors; or (iii) intellectual property infringement by software or content Indemnitor contributed to the System.

The example in the clause box above covers all three common indemnities: data breach, personal/property injury, and IP infringement. It provides typical terms for mutual indemnities.

However, a complicated problem lurks behind Subsections (i) and (ii) in the clause box—and behind almost all mutual indemnities, other than those dealing with IP.

Subsections (i) and (ii) say that the party who caused a data breach or personal injury (accident) becomes the indemnitor. What if that's not clear when the injured plaintiff makes its claim? What if each party denies doing anything to cause the leak or accident? Worse, what if each blames the other? Who indemnifies and defends whom? Who's the indemnitor?

You might think the courts will answer this question. Maybe, but that doesn't solve the problem. In a lawsuit, the court could eventually decide who's responsible for a data breach or accident, and then the parties will know who's the indemnitor. But the obligation to hire lawyers and defend arises *at the start of the claim*, right when the plaintiff files a lawsuit (or maybe even earlier, when the indemnified party needs legal help dealing with a demand for compensation). The court won't rule until *the end of the litigation*, when it's too late for the indemnitor to defend the case. And the court might never rule, if the case settles for instance, as most do.

So in a mutual indemnity triggered by the indemnitor's acts or omissions, the parties might not be able to identify the indemnitor until it's too late for that party to defend the case. And that defeats half the clause's point.

Few contracting parties worry about this issue. Those that even notice it hope they'll work something out if and when it crops up. They hope they'll be able to cooperate about mutual defense of the claim until they know who's really responsible. In other words, they take the risk that the mutual indemnity clause won't provide clear guidance. That's not a bad choice since there’s no consistently better alternative. But there *is* an alternative. It’s very complicated, and it’s not always an improvement, so stop reading here if you’re satisfied with the typical mutual indemnity terms. But if you’d like to know more, let’s discuss the status-based mutual indemnity.

• • • •

I break indemnified claims into two types: conduct-based and status-based. In a *conduct-based indemnity*, the indemnitor takes responsibility for cases related to its own acts or omissions (and usually those of its contractors). Subsection (i) and (ii) in the clause box on pages 141–142 are conduct-based, and it’s conduct-based *mutual* indemnities that create the problem discussed above. If each party denies the conduct in question, or blames the other’s conduct, who’s the indemnitor?

In a *status-based indemnity*, on the other hand, the indemnitor is responsible for cases related to its status, regardless of conduct—regardless of who’s actually responsible. A status-based mutual indemnity about data security, for instance, might say that whichever party’s computers the data leaked from serves as the indemnitor. A status-based mutual indemnity about personal injury might say that whichever party operates the building where the victim got hurt serves as the indemnitor. The trigger for responsibility lies in the indemnitor’s status as the owner or controller of the computers or of the building.

Status-based indemnities solve the problem discussed above. The parties don’t have to know who’s at fault. When a claim arises, they can almost always identify the indemnitor, right from the start. But status-based indemnities come with their own problem.

Status-Based Mutual Indemnity, Data Breach[37]

Each party (“Indemnitor”) shall defend and indemnify the other party (“Indemnified Party”) and its Indemnified Associates (as defined _____) against any third party claim, suit, or proceeding arising out of, related to, or alleging unauthorized disclosure or exposure of personally identifiable information from Indemnitor’s or its contractor’s computers.

The status-based indemnity in the clause box above addresses data breach. But it could also address personal injury and property damage. It could add an indemnified claim for “injury to or death of any individual, or any loss of or damage to real or tangible personal property, occurring in

Indemnitor's plant or other facility." I didn't include those terms because you don't see a lot of status-based personal/property damage indemnities. But it might make sense where one party runs a particularly dangerous plant.

The disadvantage of the status-based indemnity is that the indemnitor might be defending a case about the indemnified party's mistake. A cloud services vendor might own the computers that hold the data, but the customer has access to those computers through the cloud service, and its error might trigger the leak. And the fact that one party owns a building doesn't keep the other from causing an accident there.

You can go round and round about whether a status-based mutual indemnity works. You might argue that whoever owns a computer system or building should make it as idiot-proof as possible and so should take responsibility for related claims. You might also reassure yourself that most states won't make the indemnitor pay judgments if the court holds that the injury sprang from the indemnified party's negligence. So the status-based indemnitor's only obligation would be to defend. (Of course, that doesn't work if the clause specifically extends the indemnitor's obligations to claims about the indemnified party's negligence, as discussed in Subsection 1.)

Or you might turn back to the conduct-based indemnity. There, defense obligations might not be clear when the case gets filed, but at least no one's stuck defending the other party's mistake.

You should consider a third option too: *no* indemnity. If either party could trigger a particular type of claim, indemnity may offer a poor solution for handling it. Remember, you don't need an indemnity to hold the responsible party accountable. The data leak or personal injury or whatever is probably a breach of contract. So the innocent party might have a right to damages, even if it doesn't have a right to a defense against third parties. Plus, in many cases, the law would require that the guilty party reimburse the other for damages paid to the plaintiff, even without an explicit contractual indemnity.

• • • •

As noted above, mutual *IP* indemnities don't suffer from the mutuality problem. That's because intellectual property indemnities are almost always status-based. And unlike other status-based indemnities, IP indemnities don’t trigger a lot of claims where the indemnitor has to defend the indemnified party’s bad conduct, though you shouldn’t ignore the risk.

The mutual IP indemnity in the clause box on pages 141–142 is status-based. Subsection (iii) in the clause box says that each party indemnifies and defends claims “alleging intellectual property infringement by software [the] Indemnitor contributed to the System.” In other words, a party’s *status* as the contributor of the accused technology makes it the indemnitor, not its conduct. So when a third party makes an IP claim, the contracting parties don’t fight about whose conduct triggered the alleged infringement.

The indemnity in Subsection (iii) of the clause box on page 142 creates little risk that the indemnitor will have to defend a case about the indemnified party’s technology or conduct. By definition, the case will address *the indemnitor’s* technology. The indemnitor’s risk level falls even further if it does a good job limiting its IP indemnity obligations, per Subchapter 3 above. That risk level won’t hit zero, though, if the two parties’ technologies blend together so thoroughly that it’s hard to tell whose contribution allegedly infringes the plaintiff’s patent, copyright, or trade secret—or if the combination itself creates the alleged infringement. So even with IP claims, mutual indemnities create some risk of confusion and dispute.

K. Limitation of Liability

Limitation of liability clauses appear in almost all software, cloud services, and IT professional services contracts. They almost always protect vendors and distributors, but sometimes they protect customers too.

To newcomers, limitation of liability often seems bizarre. The clause says that if one party injures the other, it's not liable for the full damages. Imagine the vendor supplies defective software. The software malfunctions, and as a result, the customer loses a million dollars. Imagine also that the limitation of liability clause caps the vendor's liability at $50,000. The result: the vendor is liable for *one-twentieth* of the customer's loss. Even if everyone agrees the vendor's to blame for the whole loss, it owes $50K, and that's all.

What customer would agree to such a thing, and why? The answer is, *almost every customer*, and there's a good reason why.

The feature of the IT industry that makes it so profitable makes limitation of liability standard. That feature is *scalability*. Information technology is an unusually scalable tool: it can be used to achieve goals geometrically more valuable than the tool itself. You can use a $5,000 software program to design a half-billion-dollar bridge. You can use a $10,000 computer to manage a billion-dollar asset portfolio. And that same low-cost software application or computer can *ruin* a half-billion-dollar bridge or a billion-dollar asset portfolio.

If the vendor faced potential liability of a billion dollars, or even a half-million dollars, with every $5,000 sale, it couldn't do business. One malfunction could wipe out ten years of profits, or the whole company. That's why IT vendors insist on limitations of liability.

Still, limitations of liability can be a bit hard to swallow. In fact, courts often won't enforce the clause if they think the customer didn't grasp its importance, particularly if the customer is a consumer. That's why the limitation of liability usually stands out, printed in capital letters (like the examples in the clause boxes below). The vendor wants to establish that if the customer didn't notice the clause, or recognize its importance, it's the customer's own fault. Vendors should take other precautions to increase their chances of enforcement, explained in Subchapter 3 below.

Limitation of liability clauses come in two flavors: dollar caps and exclusions of consequential damages. Most contracts feature both, as overlapping protections. Many clauses also have exceptions: types of liability that are *not* limited. We'll address each in turn.

Finally, as mentioned above, some IT contracts limit the customer's liability too. The usual customer rationale is: "If your liability's limited, so is ours." The examples in the following clause boxes protect only the vendor, but you can easily make them two-way clauses. For instance, instead of "Vendor's liability will not exceed . . . ," your contract would say: "Neither party's liability will exceed . . . ." Vendors should be aware, however, that two-way clauses create some special risks for them. See Subchapter 4 below.

1. Dollar Cap

The simplest part of the limitation of liability clause caps the parties' liability at a dollar figure.

Dollar Cap Limitation of Liability

> VENDOR'S LIABILITY ARISING OUT OF OR RELATED TO THIS AGREEMENT WILL NOT EXCEED $______.
>
> • • • •
>
> VENDOR'S LIABILITY ARISING OUT OF OR RELATED TO THIS AGREEMENT WILL NOT EXCEED THE ANNUAL LICENSE FEE.

The dollar cap could be a million dollars or $5,000 or anything else. Sometimes the contract doesn't set the figure. Instead, it's calculated through some formula, as in the second example in the clause box above.

Many contracts cap liability at the total fees due under the contract or under a particular order or statement of work, or the fees due for a single year. But the figure could be higher or lower. You can pick almost any figure and almost any way of calculating it—except that vendors should avoid an excessively low dollar cap, as explained in Subchapter 3 below.

Contracting parties often debate whether a proposed dollar cap is “fair.” Fairness plays even less role in the dollar cap than in other contract clauses. It’s arbitrary.

2. Exclusion of Consequential Damages

The exclusion of consequential damages focuses on the *type* of liability, not the dollar amount. The clause may exclude “indirect,” “special,” “punitive,” and various other types of damages. All of those but *punitive*—discussed below—are flavors of consequential damages.

> ***Exclusion of Consequential Damages***
>
> IN NO EVENT WILL VENDOR BE LIABLE TO CUSTOMER FOR ANY CONSEQUENTIAL, INDIRECT, SPECIAL, INCIDENTAL, OR PUNITIVE DAMAGES ARISING OUT OF OR RELATED TO THIS AGREEMENT.

To understand consequential damages, let’s return to our example of a software application used to design a half-billion-dollar bridge. The software malfunctions, and as a result, the bridge is defective and falls apart. The customer designed the bridge and owns it, and it wants two kinds of compensation from the vendor: direct and consequential damages. The *direct* damage is compensation for the type of losses usually caused by software failures. Half-billion-dollar bridge design isn’t the norm for that application (or for any, arguably), so direct damages probably wouldn’t cover the cost of the bridge. They might cover the cost of a much smaller defective design, and the cost of replacing the software. The *consequential* damage is the price tag on all the unique consequences of *this* particular failure. That would cover the bridge, as well as the cost of compensating injured or killed bystanders, loss of the customer’s time, loss of other business opportunities, loss of reputation, etc.[38]

In other words, consequential damages are unpredictable and theoretically unlimited (though some state laws impose limits). That’s why IT vendors generally insist on limiting them.

Obviously, a dollar cap would also prevent liability for much of the money at stake in a consequential damages claim. The two clauses overlap.

Finally, limitation of liability clauses sometimes exclude punitive damages. Punitive damages go beyond compensation; they punish. They’re usually not available in contract cases. But the customer could also make a tort claim, and anyway the law isn't perfectly predictable. There's no harm in throwing "punitive" into the list with "consequential," "special," etc., as in the example in the clause box above.

3. Unconscionability and Required Clarifications

In theory, a limitation of liability is perfectly enforceable. Still, courts often look for a way out if the clause seems unusually unfair—particularly if the customer is a consumer. Courts will set aside the clause if it's so unfair as to be "unconscionable" or "opposed to public policy." It's hard to define "unconscionability," so it's hard to give clear instructions on avoiding it. Usually, though, courts don't mind exclusions of consequential damages, so the unconscionability review focuses on the dollar cap. Is the cap so low that the vendor avoids any meaningful liability for its own wrongdoing? If so, you may have a problem. Would a sane customer sign the contract after reading the clause? If not—if you're hoping customers just won't notice the limitation of liability—again, you may have a problem.

Certain disclaimers and clarifications also help avoid unconscionability. These disclaimers relate to both the dollar cap and the exclusion of consequential damages.

Clarifications and Disclaimers

THE LIABILITIES LIMITED BY SECTIONS __ (*Dollar Cap*) AND __ (*Exclusion of Consequential Damages*) APPLY: (a) TO LIABILITY FOR NEGLIGENCE; (b) REGARDLESS OF THE FORM OF ACTION, WHETHER IN CONTRACT, TORT, STRICT PRODUCT LIABILITY, OR OTHERWISE; (c) EVEN IF VENDOR IS ADVISED IN ADVANCE OF THE POSSIBILITY OF THE DAMAGES IN QUESTION AND EVEN IF SUCH DAMAGES WERE FORESEEABLE; AND (d) EVEN IF CUSTOMER'S REMEDIES FAIL OF THEIR ESSENTIAL PURPOSE. If applicable law limits the application of the provisions of this Section __, Vendor's liability will be limited to the maximum extent permissible.

Most courts won't enforce a limitation of negligence liability unless it's explicit, so negligence should appear in the disclaimers and clarifications, as in Subsection (a) in the example above. The vendor should also list just about every form of action the parties are likely to address, so no one can reasonably doubt the intended reach of the clause, as in Subsection (b).

Courts sometimes refuse to enforce the limitation of liability because the vendor had warning of the loss in question, or the loss was foreseeable. Subsection (c) in the clause box addresses that risk. Courts also sometimes refuse enforcement where the clause leaves the customer no meaningful remedy. The best solution is to allow a meaningful remedy, as suggested above, but the vendor should also have the customer agree that no remedy is required, just in case. That's why Subsection (d) in the clause box says the clause survives even if the customer's remedies "fail of their essential purpose."

Finally, the vendor should think about salvaging something from the limitation of liability clause even if some of these precautions fail—even if a court refuses to enforce part of the clause. The last sentence in the clause box gives courts instructions on paring back an unconscionable or otherwise enforceable clause, without eliminating it.

4. Exceptions: Liability That's Not Limited

The parties often agree that the clause won't limit certain forms of liability.

Exceptions to Limitation of Liability

Section __ (*Limit of Liability*) does not apply to: (a) claims pursuant to any provision of this Agreement calling for liquidated damages; (b) claims pursuant to Section __ (*Indemnity*); or (c) claims for attorneys' fees and other litigation costs Customer becomes entitled to recover as a prevailing party in any action.

• • • •

The limitations of liability in Section __ (*Limit of Liability*) do not apply to: (a) Customer's obligation to pay fees pursuant to Section __ (*License and Service Fees*); or (b) any claims against Customer for infringement of Vendor's intellectual property, including without limitation copyrights in the Software.

Generally, the exceptions carve out liabilities created by the contract itself. The most common exceptions are for indemnity and liquidated damages—both addressed in the first example in the clause box above.[39] The indemnity exception is particularly important and particularly contentious in some deals. Customers argue that if the limitation of liability restricts the vendor's IP indemnity obligations, for instance, those obligations could be almost meaningless, since the cost of a patent suit might be millions more than the limitation of liability allows. The same concern applies to other indemnities. Vendors, on the other hand, argue that if the vendor bore the whole risk of indemnified claims, the customer would pay much higher prices. In other words, the vendor's prices assume limited indemnity liability.

This is a debate without a right answer, but one possible compromise would give the vendor some limit to indemnity liability, but make it higher than the limit for other liabilities. "The Provisions of Section __ (*Limit of Liability*) do not apply to liability pursuant to Section __ (*Indemnity*). VENDOR'S LIABILITY ARISING OUT OF OR RELATED TO SECTION __ (*Indemnity*) WILL NOT EXCEED __ TIMES THE ANNUAL LICENSE FEE." That removes the consequential damages cap for indemnity liability, and it provides, presumably, a higher dollar cap than for other losses.

Some clauses also exclude liability for breach of a nondisclosure agreement or clause. And a few exclude liability for data breaches. In those cases, the first example in the clause box would provide that the limits of liability do not apply to: "(d) liability pursuant to Section __ (*Nondisclosure*); or (e) liability pursuant to Section __ (*Data Management & Security*)."

In a two-way limitation of liability clause—protecting the customer as well as the vendor—vendors should consider two additional exceptions. The first has to do with intellectual property infringement. What if the customer infringes the vendor's copyright by creating too many copies of the software? What if, instead of creating two copies, as authorized in the license section, the customer creates a thousand? If the limitation of liability clause caps damages at the contract price, the customer arguably gets a thousand copies for the price of two. Of course, a court might not enforce such contract-assisted theft, but the vendor shouldn't take the risk. That's why the second example in the clause box above carves out liability for IP infringement by the customer.

Second, the vendor should be certain the clause can't be interpreted as a guarantee that the customer won't have to pay anything other than license fees, even if it breaches the contract. What if the customer breaches the contract by failing to pay a year's worth of fees, and also in some other way? If the dollar cap limits each party's liability to one year's worth of fees, the customer could argue that its only liability is for the unpaid fees. It could breach the contract's other terms with impunity, thanks to the dollar cap. At least, the contract could be interpreted that way. So the vendor should protect itself by clarifying that the liability for fees stands outside the clause (or at least outside the dollar cap). See the second example in the clause box above.

Vendors granting mutual clauses should peruse the contract for other liabilities that should *not* be limited. Breach of a government restricted rights clause might cost the vendor a fortune, for instance. So the vendor might want terms saying: "The limitations of liability of this Section __ do not apply to breach of the provisions of Section __ (*Government Restricted Rights*)."[40] In general, vendors should consider the limitation of liability clause *their* shield. If the protection shields the customer too, the vendor should think through the risks and create exceptions for any it can't accept.

L. Use of Trademarks

A trademark license is like a software copyright license. It authorizes use of trademarks without transferring ownership. Businesses use these licenses to facilitate marketing.[41]

You don't need a trademark license if one party just wants to issue a press release talking about its relationship with the other. But if a party wants to make extensive marketing use of the other party's name, it needs a trademark license. Such "extensive" use would include putting the other party's name on product packages, for instance, in a distribution relationship (e.g., "Intel Inside®").[42] And use of logos almost always requires a license.

The trademark owner should supervise use of its marks. It should review press releases, customer lists, and other marketing documents to make sure they're consistent with its public image. And in a distribution agreement, the trademark owner should check on the quality of any product sold using its trademarks. Finally, the trademark owner should be sure the other party includes a trademark symbol: "™" for trademark and "®" for registered trademark. This type of supervision doesn't just make good business sense; it's necessary to preserve trademark rights. Trademarks are like friends: if you don't treat them well, you lose them. A *naked license*—a grant of trademark rights without supervision—can invalidate the trademark.

Use of Names and Trademarks

Customer hereby grants Vendor a license to include Customer’s primary logo, illustrated on Attachment __ (the “Logo”), in any customer list or press release announcing this Agreement, provided Vendor first submits each such press release or customer list to Customer and receives written approval, which approval shall not unreasonably be withheld. Goodwill associated with the Logo inures solely to Customer, and Vendor shall take no action to damage the goodwill associated with the Logo or with Customer.

• • • •

Vendor hereby grants Distributor a license to reproduce its trademarks listed on Attachment __ (*Trademarks*) on marketing and advertising materials and packaging related to any Product (collectively, “Advertisements”); provided (a) the Product conforms to the quality requirements listed in Attachment __ (*Quality Standards*) and (b) Distributor observes Vendor’s standard guidelines on trademark usage, attached hereto as Attachment __ (*Vendor Trademark Policy*), including any written amendment to such policy provided by Vendor in its sole discretion. All goodwill associated with such trademarks inures solely to Vendor, and Distributor shall take no action to damage the goodwill associated with the Trademarks or with Vendor. In the event that Vendor notifies Distributor in writing that any Product or Advertisement (pending or published) does not conform to the requirements of this Section __, Distributor shall promptly withdraw it or remove all Vendor trademarks; provided Vendor shall not unreasonably issue such notice.

Both examples in the previous clause box help avoid naked licensing by granting supervision rights. In the first example, the trademark owner has to approve trademark usage in advance. That’s the best system for the owner, but it’s not always practical, particularly when the other party issues hundreds of ads. So the second example does away with advance approval. Instead, it requires that the other party observe the owner’s trademark usage guidelines. (Those guidelines should include, among other things, use of “™” or “®” with the marks.) And if the owner does notice misuse of its mark, the other party has to withdraw the offending ad. The second example also addresses the quality of the underlying product—promising the owner that its trademark won’t be associated with shoddy merchandise.

But contract language might not do the trick alone. To preserve trademark status, the owner should *actually supervise* use of its marks.

The trademark licensee should also agree not to damage the owner’s goodwill, as in both examples in the clause box above. “Goodwill” is the value behind trademarks: the reputation for quality associated with products and services sold under the mark. Of course, the licensee more or less agrees to preserve goodwill by agreeing to obey the owner’s trademark policies and rules. But the more general promise adds to the owner’s protection.

Finally, trademark owners should usually ensure that goodwill “inures to” the owner, as in both examples in the clause box above. In other words, the owner gets the legal benefit of goodwill, even though the other party may be creating some of that goodwill through its use of the trademark.

M. Training

Training clauses are most common in software licenses and cloud services agreements. In these clauses, the vendor promises to help the customer learn how to operate technology.

Training

Vendor shall provide training courses on operation of the System, at Customer's ________ facility, at such times during business hours as Customer may reasonably request. Each training course will last __ hours. Customer may enroll up to ___ of its staff members in any training course, and Vendor shall provide a hard copy of the Licensed Product's standard training manual for each enrollee. Vendor shall ensure that each training course is taught by a technician with no fewer than ___ years' full-time experience operating ____________ systems. Vendor shall provide the first ___ trainings without additional charge and shall provide additional trainings at its standard rates.

• • • •

Without additional charge, Vendor shall provide such training on use of the Software as Customer may reasonably request, and the parties shall negotiate in good faith regarding the time and place of such training.

Some clauses provide significant detail about training parameters—duration and size of courses, expertise of instructors, cost (if any), etc.—as in the first example in the clause box above. But some training clauses leave the details out, on the assumption that the parties can work them out later, as in the second example.

Training is a professional service. So if training forms a major part of the transaction, review Chapter I.F ("Promise of Professional Services").

N. Feedback Rights

Many tech vendors worry that their customer will provide feedback about their products or services and that, as a result, the customer will own those ideas. In other words, because the customer came up with the feedback, the vendor won't be able to use it, thanks to the customer's intellectual property rights, or some other rights. To avoid that, the vendor drafts a feedback clause, ensuring that it *can* use the idea.

IT vendors also use feedback terms with their subcontractors. And sometimes customers ask for feedback clauses. They worry about rights in vendor feedback related to *customer* products or services. Feedback clauses can work in any of these situations, and you can modify the sample clauses below to fit them. But this chapter sticks to the most common type: terms addressing customer feedback, protecting the vendor.

Much of the concern about feedback is actually pretty far-fetched, and it depends on a misunderstanding of IP law. That misunderstanding often leads to a "feedback license," like the clause box below. That's not necessarily a bad solution for the vendor, if it drafts a good license and if the customer's dumb enough to accept it. But it's overbroad, and if the customer won't agree, the vendor can still protect itself through terms like the second clause box below, which this chapter calls a "feedback disclaimer."

Feedback License

Customer hereby grants Vendor a perpetual, irrevocable, worldwide license to use any Feedback (as defined below) Customer communicates to Vendor during the Term, without compensation, without any obligation to report on such use, and without any other restriction. Vendor's rights granted in the previous sentence include, without limitation, the right to exploit Feedback in any and every way, as well as the right to grant sublicenses. ("Feedback" refers to any suggestion or idea for modifying any of Vendor's products or services, including without limitation all intellectual property rights in any such suggestion or idea.)

Feedback licenses generally give the vendor rights to "use" feedback. The parties don't know in advance what sort of IP or other rights the customer might have in feedback, so the license has to be pretty broad and general. The clause box above isn't a model of clarity—feedback licenses never are—but it probably gives the vendor an IP license to do anything it wants with inventions or other assets included in feedback. For instance, if the customer gets a patent on an invention described in the feedback, the clause box would probably serve as a patent license (without the usual 20 pages of rights and restrictions).

If you're the vendor, the clause box above should protect you against the feedback concern. If you're the customer, you shouldn't sign a license like the one above, or any feedback license, if you can avoid it. It's anyone's guess what message to the vendor might qualify as feedback. In offering a suggestion about the vendor's technology, the customer's staff might say something about the customer's own products or services. If the customer gets a patent or other IP rights covering inventions hinted at in that message, or already has rights, the vendor could have a broad license under that IP. The vendor could even transfer that license to the customer's competitors. Worse, you'll probably never know the extent of the vendor's feedback license unless the parties go to court. The license could cast a shadow over the customer's rights to future innovations. None of this is likely, but the consequences could be serious.

If the customer does reject the typical feedback license, the vendor has an alternative. It doesn't need a broad IP license to feedback, because the customer probably won't ever own assets the vendor cares about: ideas related to the vendor's products or services. That's because, contrary to common belief, no one can own an abstract idea. As Appendix 1 ("Intellectual Property") explains, IP law doesn't protect ideas but rather copyrighted writings, patented inventions, and trade secrets (as well as trademarks in names and brands). None of those is likely to play a role in feedback related to the vendor's products or services.

Let's look at copyrights. If the customer writes software that improves the vendor's systems, the customer *will* own the copyright in that software. So the vendor won't be able to use it without the customer's consent. But software ownership isn't the concern driving feedback clauses. The vendor just wants the right to use the *ideas* in the feedback. Copyright won't protect the ideas behind the software. The vendor could still write its own software based on those ideas.

Nor would copyright keep the vendor from using an idea in a memo or e-mail from the customer. The customer might have a copyright in the memo or e-mail, meaning the vendor couldn't copy the words or circulate them (though even that's not likely, thanks to legal doctrines like "fair use" and "implied license"). But the vendor could still take the idea from the memo or e-mail—the feedback—and use it.

What about patents? Abstract ideas aren't patentable, so the customer can't get a patent on ideas included in feedback, and the vendor's free to use them. There *is* a chance a member of the customer's staff could invent some new technology that improves the vendor's systems and describe it in feedback to the vendor. If the customer later gets a patent on that invention, the vendor couldn't use it without the customer's consent. But again, the concern driving feedback clauses isn't inventions of new technology; it's ideas. Plus, how likely is it that the customer (1) invents new technology related to the vendor's systems, as opposed to just offering a suggestion, and
(2) describes that invention in feedback? And how likely is it that the customer (3) realizes it's invented something, (4) pays patent lawyers to analyze an invention *related to someone else's product or service*, (5) concludes it's patentable, (6) pays to prepare and file a patent, and (7) gets a patent awarded?[43]

Finally, what about trade secrets? Ideas *can* count as trade secrets, so the customer could, in theory, use trade secrets law to stop the vendor from using ideas described in feedback. But the very fact that the customer communicates the feedback to the vendor means it's *not* a trade secret (and probably means the customer doesn't care about protecting it anyway). Again, see Appendix 1 ("Intellectual Property"). The feedback would only qualify for trade secret protection if it's subject to a nondisclosure clause or NDA.

That's where our alternative clause comes in. To protect its rights to use feedback, the vendor needs to make sure the feedback isn't a trade secret—or confidential information protected by a contract or some other type of promise.

Feedback Disclaimer

Vendor has not agreed to and does not agree to treat as confidential any Feedback (as defined below) Customer provides to Vendor, and nothing in this Agreement or in the parties' dealings arising out of or related to this Agreement will restrict Vendor's right to use, profit from, disclose, publish, keep secret, or otherwise exploit Feedback, without compensating or crediting Customer. ("Feedback" refers to any suggestion or idea for improving or otherwise modifying any of Vendor's products or services.)

The clause box above doesn't give the vendor an IP license. Rather, it confirms that feedback isn't protected by trade secrets law and isn't confidential. It also confirms that nothing in the agreement restricts the vendor's right to use feedback. If the agreement includes a nondisclosure clause, you might clarify even further: "Feedback will not constitute Confidential Information, even if it would otherwise qualify as such pursuant to Section __ (*Nondisclosure*)."

Arguably, the feedback disclaimer above protects the vendor less than a feedback license, because of the small risk that feedback will include an invention related to the vendor's systems and the customer will patent it. The feedback disclaimer wouldn't give the vendor a license under that patent. On the other hand, the feedback disclaimer provides clear terms, while it's difficult to know how a court will interpret a feedback license. That helps the vendor. And for the customer, of course, the feedback disclaimer eliminates the risk that feedback includes some idea related to *its* intellectual property.

O. Non-Compete and Employee Non-Solicit

A non-compete clause provides that one party won't poach the other's customers or, in some cases, provide services to the other party's competitors. An employee non-solicit clause forbids attempts to hire away the other party's staff. Either may appear in a professional services contract or in a software license or cloud services agreement requiring integration, maintenance, or other support. (Either could also appear in an employment contract.)

Non-compete and employee non-solicit clauses butt heads with antitrust, unfair competition, and employment laws—rules favoring competition and protecting everyone's right to make a living. Terms that don't comply aren't enforceable. Worse, they can trigger government fines and other liabilities. You might even find yourself on the wrong end of a class action. Federal antitrust law rules out most non-compete and employee non-solicit clauses between competitors—between IT companies in related fields, for instance. So before using one of those clauses, you'll need to figure out whether the law would consider your parties competitors—and that's not easy. State laws, for their part, rule out both types of clauses in a variety of settings, and those laws vary—a lot. So even if you pass the "competitor" hurdle, you'll need to craft your clause to fit the laws in each state touched by your deal. In some cases, those laws will leave *no* room, or very little room, for non-compete or employee non-solicit terms.

Because the "competitor" definition depends so much on your contracting parties, and because state laws vary so much, this chapter can't guide you as much as the rest of this book. Most of this book's suggested terms work most of the time in most states. And though you're always better off consulting an expert, you're taking a reasonable risk if you use most of this book's sample clauses without that help. That's not the case with the samples in this chapter. *I don't recommend using any of this chapter's sample clauses without consulting a lawyer who knows the* *antitrust, unfair competition, and employment laws applicable to every state touched by your deal.*[44]

1. Non-Compete

A non-compete clause can restrict either party, but usually it's the customer who's worried about competition, so the clause restricts the vendor. It says the vendor won't poach the customer's end customers or serve its competitors. That's the case with the examples and discussion below, but keep in mind that you can reverse the parties or create a mutual non-compete, restricting both.

Before using a non-compete, ask yourself whether an NDA or nondisclosure clause would offer the protection the customer wants. If the customer discloses lists of its end customers, or strategies for pursuing end customers, and those resources count as "Confidential Information" under an NDA, the vendor can't use them to compete or to serve competitors. True, you might get more protection from a non-compete clause, particularly since you can't enforce an NDA unless you *know* your information's been misused, but vendors accept nondisclosure terms more readily than non-competes. And NDAs aren't as likely to run afoul of pro-competition laws.[45]

If an NDA doesn't work, you'll want to think through two types of non-compete clauses. The first restricts *direct competition*. It provides that the vendor won't poach the customer's own customers. The second restricts *serving competitors*. It provides that the vendor won't offer the customer's competitors the same services it provides to the customer. Sometimes it says the vendor won't serve competitors at all.

The restriction on direct competition springs from the concern that, by providing services to the customer, the vendor will gain information or contacts it could use to compete with the customer. Obviously, that rarely worries anyone in a typical vendor-customer relationship, where the vendor works in IT and the customer operates a non-tech business (selling shoes or airplanes, running a government, etc.). The restriction makes more sense in deals between IT companies or between parties whose markets overlap in some other way—in other words, *between competitors*. So if you're interested in a restriction on direct competition, there's a good chance antitrust law would consider your contracting parties competitors. That means you could get into serious trouble for trying a non-compete clause. *Some* version of the restriction on direct competition *might* be kosher, but it's got to address legal issues specific to your parties, your deal, and your state. Unfortunately, that means I can't give you a generally useful sample clause. So this subchapter doesn't include a clause box for the restriction on direct competition.[46]

We can do more with the restriction on *serving* competitors because it doesn't necessarily imply that the customer and vendor *are* competitors. A customer selling insurance might hire an IT vendor to design software, for instance. The customer worries that the vendor will learn about its operations through the project and then use that knowledge to serve other insurance companies: the customer's competitors. A clause forbidding that *might* pass legal muster.

Non-Compete: Restriction on Serving Competitors

During the term of this Agreement and for ___ days following termination, Vendor shall not provide Comparable Services to a Competitor for use or consumption in ______ [geographic territory]. For breach of this Section __, Vendor shall pay Customer $____ per Competitor, as liquidated damages. ("Comparable Services" refers to _________________. "Competitor" refers to _______________.)

The clause box above operates only during a set period of time—180 days, 365, etc.—and only in some particular region, like "the states of New York, New Jersey, and Pennsylvania." Shorter durations and smaller regions work better for the vendor, of course. They also reduce both parties' risk of trouble with pro-competition laws.

The clause box above says that the vendor won't provide comparable services to the customer's competitors. The trick, then, is defining "comparable services" and "competitor." Don't simply call comparable services "the services provided pursuant to this Agreement." No two projects are identical, so that definition gives no guidance on how similar the services can be. The clause needs to *describe* comparable services. Imagine the vendor develops a factory management system for the customer. You might then define comparable services as "development or provisioning of software that manages fabrication in factories, or advice regarding such software, including both installed software and remote-hosted or cloud computing software." That leaves the vendor free to offer services related to other sorts of technology. You could create a narrower definition instead, covering only a particular type of factory, for instance, or focusing on the type of technology used in the software. Again, narrower definitions serve the vendor better and reduce both parties' risk of trouble with pro-competition laws.

The clause's definition of "competitor" won't necessarily match the definition from antitrust or unfair competition laws. We're just defining what *the contract* considers a competitor (and hoping pro-competition laws don't get involved). The clearest definition lists the companies in question: "'Competitor' refers to SiblingRival LLP, Leading Brand Solutions, Inc., and Nemesis Corporation." But that's not always possible. If you have to draft a looser definition, consider two questions. First, what products or services must a company provide to be a competitor—or, if you absolutely can't pin that down, what industry must it work in? Second, do competing *affiliates* make a party a competitor? For instance, if a company wouldn't count as a competitor but its parent company has another subsidiary that would, is it a competitor? Here's a definition that addresses both: "'Competitor' refers to a person or entity that provides products or services that track or otherwise manage livestock waste. A Competitor includes, without limitation, any entity directly or indirectly controlling, controlled by, or under common control with a Competitor, with 'control' referring to the beneficial ownership or control of 50% or more of the equity interest in an entity or the ability to direct or cause the direction of the management or affairs of an entity, whether through the direct or indirect ownership of voting interests, by contract, or otherwise."

Even that lengthy definition has areas of uncertainty. Defining "competitor" isn't easy.

Finally, the clause box above calls for liquidated damages. It's usually hard to tell how much breach of a non-compete hurts the injured party. So instead of risking a court battle over a hard question, the parties set a compensation figure in advance. That figure could be a fixed fee for each competitor served, as in the clause box, or a percentage of revenues from a project the vendor shouldn't have sold. The key requirement is that the amount add up to a reasonable guess at the likely losses. If you include liquidated damages, review Chapter II.Q ("Liquidated Damages") and add the language suggested there.

2. Employee Non-Solicit

Employee non-solicit clauses can restrict either the vendor or the customer. They make sure professional services relationships don't turn into employee-poaching expeditions. As a result of close contact, one party might identify valuable members of the other's staff.

If federal law considers the parties competitors, an employee non-solicit clause won't be enforceable, and it might trigger fines and all sorts of other trouble. State laws might conflict with the clause, too. The parties could even find themselves liable to employees, including in a class action. So, as always with this chapter, get legal help before drafting an employee non-solicit clause.

Employee Non-Solicit

During the term of this Agreement and for ___ days after termination, neither party shall solicit any of the other's employees involved in the Services to consider alternate employment. For the avoidance of doubt, the preceding sentence does not forbid a solicitation to the general public. For each employee who quits as a result of breach of this Section __, the soliciting party shall pay the other party $____, as liquidated damages.

The clause box above forbids solicitation of employees involved in the "Services": the parties' work together. You might be tempted instead to forbid solicitation of any employee, but why? That would be overbroad if you're trying to prevent poaching of employees one party meets through its work with the other. It would also greatly increase your risk of trouble with pro-competition and employment laws (and virtually guarantee that trouble in some states).

Of course, solicitation can be difficult to prove. What if one party slyly hints to the other's employees that better opportunities might be available elsewhere? What if a party offers a job but then claims the employee started the conversation? Some contracts address this by going beyond soliciting employees to forbid *hiring*: "During the term of this Agreement and for ___ days thereafter, neither party shall hire, as an employee or independent contractor, any of the other's employees involved in the Services." Again, the broader clause increases your risk of trouble with pro-competition and employment laws.

Non-solicit clauses can address independent contractors instead of employees. “During the term of this Agreement and for ___ days after termination, neither party shall offer employment, including work as an independent contractor, to any contractor involved in the Services who provides technology development services to the other during the term of this Agreement.” In drafting a contractor clause, remember that every non-employee who provides services counts as an independent contractor. Do you really want to keep the other party from retaining your accountant, or your plumber, janitor, or phone company? Specify the type of contractor you mean.

Many employee non-solicit clauses include liquidated damages provisions, like the last sentence in the clause box above. Like breach of a non-compete, damages from employee poaching would be hard to calculate, so the parties agree on a figure in advance. If you use liquidated damages, review Chapter II.Q (“Liquidated Damages”) and add the language suggested there.

P. Software Audits

An audit clause helps the vendor protect against unauthorized copying and use of software. Audit terms are appropriate for end user licenses and distribution agreements. They usually don't show up in cloud services agreements because the vendor knows who's using the system; it doesn't need to audit.

An audit clause authorizes the vendor to review the customer's books and computers. The vendor searches for evidence of copies in excess of the license, use beyond the scope authorized, or distribution without royalty payments.

Software Audit

During the term and for ___ thereafter, Vendor may audit Customer's use of Licensed Software on ___ days' advance written notice. Customer shall cooperate with the audit, including by providing access to any books, computers, records, or other information that relate or may relate to use of Licensed Software. Such audit shall not unreasonably interfere with Customer's business activities. If Vendor discovers unauthorized use, reproduction, distribution, or other exploitation of Licensed Software, in excess of ___% of the copies or fees that would have applied to authorized exploitation, Customer shall reimburse Vendor for the reasonable cost of the audit, or of the next audit in case of discovery without an audit, in addition to such other rights and remedies as Vendor may have. Vendor may not conduct an audit more than once per ________.

In the clause box above, the customer reimburses the cost of the audit if it turns up misuse of the software in excess of some minimum amount, like five percent. (If *any* misuse triggered the reimbursement, the customer could find itself on the hook because of an immaterial mistake, like making 1,001 copies when it only had a license for 1,000. Few vendors demand terms that strict.) The clause box also clarifies that reimbursement isn't the vendor's only remedy. The contract doesn't have to spell out those other remedies, and the clause box above doesn't. The law will automatically impose rights to compensation, including fees for unlicensed software—and possibly damages for copyright infringement.

Many vendors prefer that the audit clause say nothing about their remedies for unlicensed software because they don't want to *limit* those remedies. If the contract spelled out the vendor's right to fees for past use, a court might hold that the vendor has no right to additional remedies, like damages. Copyright infringement can lead to hefty damages, so many vendors prefer to preserve possible claims. But some want a quick and clear route to payment for unlicensed software, rather than a possible legal battle, so they spell out their rights to "back fees," as in the next clause box below. If you consider spelling out those rights, ask yourself if your clause would create a perverse incentive. Would it give the customer an incentive to cheat, since cheating won't cost it anything if it doesn't get caught and won't cost more than playing by the rules if it does?

If you're the vendor and you do spell out your back fees rights, make sure to include whatever you would have earned if the customer had licensed the unauthorized software in the first place. Would the customer have paid maintenance fees for the extra software? Would it have paid for a whole term or several, even though unauthorized reproduction started late in a term? Would the vendor have benefited from the extra software rights in any other way?

Vendor's Fees for Unauthorized Use (Back Fees)

If Vendor discovers unauthorized reproduction of the Licensed Software, Vendor may, by written notice, direct customer to delete all unauthorized copies or add them to the license granted in Section __ (*License*). In either case, Customer shall pay: (a) the per-copy license fee and per-copy maintenance fee for each unauthorized copy, for the period from the creation-date of such copy to the date of Vendor's notice; and (b) interest at the rate of __% per month or the maximum rate permitted by law, whichever is less, compounded daily from the date any payment would have been due until the date paid. For continuation of any copy after the date of Vendor's notice, Vendor may charge Customer a special license fee of $______; thereafter, such copy will be subject to the fees set forth in Attachment __ (*License and Maintenance Fees*).

In the clause box above, the vendor can choose whether to let the customer keep using the unauthorized software. And it gets paid for past use—back fees—either way. Plus, the vendor can charge an extra fee for continued use of the software—to extend the license outside the normal licensing procedures. Keep in mind that penalties aren't enforceable in contracts. The vendor can't force the customer to pay some extra amount just for having been caught red-handed. But a special license fee probably won’t be considered a penalty, since the customer can refuse to pay it and just stop using the extra software.

Q. Liquidated Damages

When one party breaches a contract, the other often has a right to damages: to money that compensates for any injuries. But in some relationships, the parties know in advance that damages won't be easy to calculate. So they specify the amount the breaching party will have to pay, in a liquidated damages clause.

Imagine the customer wants customized software to improve its efficiency, and the vendor agrees to write it within four months. If the vendor delivers late or not at all, the customer will have wasted a lot of time. It will also have missed out on savings and business opportunities. But it's hard to put a dollar figure on lost time or improvements brought by an untried system. So the parties agree in advance on *liquidated* damages. The vendor will pay $500 for every day of delay and $25,000 if it never delivers the software at all. This liquidated damages pact should help prevent disputes about the consequences of delay. And if the parties do go to court, the clause should prevent an expensive dispute about the amount of damages.

Don't confuse liquidated damages with early termination fees. As explained in Subchapter II.V.3 ("Termination for Convenience"), some contracts let a party terminate early in exchange for a fee. This termination for convenience is not a breach of contract; it's authorized. So the fee isn't a damages calculation. It's the contract *price* for early termination. Liquidated damages, on the other hand, are *damages for breach* of contract.

A court won't enforce a liquidated damages clause unless it meets two conditions. First, at the time the parties sign the contract, likely damages have to be uncertain or difficult to prove. Second, the damages have to serve as compensation for injuries, not as a penalty. Penalty clauses are unenforceable. So the liquidated damages amount should roughly approximate the injured party's projected losses. Of course, you don't know how much the injured party would lose if the other party breached. But you can at least guess at a range and fix the liquidated damages somewhere in that range. It doesn't matter if you guess wrong. What matters is that, at the time of executing the contract, the guess was reasonable. If the parties just pick some high number, with little or no relationship to likely losses, a court will probably consider it a penalty.

A liquidated damages clause should have two parts: a specification of damages and a justification.

Specification of Liquidated Damages

Vendor shall credit Customer __% of the License Fee, as liquidated damages, for each business day between the Due Date and any later date Vendor delivers the Software, up to a maximum of 30 business days. Such liquidated damages are Customer's exclusive remedy for any delay of 30 business days or fewer, but they do not preclude other remedies for other injuries, including without limitation for delay in excess of 30 business days.

• • • •

In the event that Customer materially breaches this Agreement and Vendor terminates on that basis, Customer shall pay Vendor, as liquidated damages, __% of the Service Fees not yet invoiced. The provisions of the preceding sentence state Vendor's exclusive remedy for such breach but do not limit Customer's obligation to pay Service Fees already invoiced.

• • • •

In the event that the System does not provide the functionality listed in Item __ of the Technical Specifications, Vendor shall pay Customer $_______, as liquidated damages. Such liquidated damages are Customer's exclusive remedy for such breach, but they do not preclude other remedies for other injuries.

The parties can set a specific dollar amount as liquidated damages, as in the last example in the clause box above. Or they can provide a formula for calculation of damages, as in the first two examples.

The first example in the clause box imposes fees for late delivery (of software), and these late fees go up for every additional day the delivery's late. That's fine, but remember that the total has to be reasonable, or it's not enforceable. That means the daily increase can't be so large that, after a short time, the receiving party—the customer in this case—gets its deliverable for little or nothing. By the same token, the total shouldn't keep rising indefinitely. That's why the first example in the clause box has a 30-day limit: delay after day 30 doesn't trigger additional liquidated damages. That, however, doesn't mean the injured party has *no* remedy for additional delay. The example specifies that the customer could get regular contract damages for losses triggered by delay *after* day 30.

Liquidated damages should be exclusive: the injured party can’t get any other compensation for the injury in question. But as the prior paragraph suggests, it’s often a good idea to clarify that the clause doesn’t rule out additional damages for *other* injuries. See the first and last examples in the clause box above.

• • • •

The justification part of the clause addresses the requirements for liquidated damages, rather than the amount. The parties agree that liquidated damages are necessary because figuring out actual losses would be so difficult. They also agree that they didn’t intend the clause as a penalty and that they used an estimate of actual losses to set the amount.

Justification for Liquidated Damages

The parties agree that the damages set forth in this Section __ are liquidated damages and not penalties and that they are reasonable in light of the harm that would be caused by breach, the difficulties of proof of loss, and the inconvenience and infeasibility of otherwise obtaining an adequate remedy.

The fact that the parties state these justification claims doesn’t make them so. A court will make its own assessment. But neither party should be able to argue that it didn’t understand the clause’s purpose.

R. Disaster Recovery

A disaster recovery clause lays out procedures the vendor should follow to keep technology up and running in the face of natural disasters, wars, and other calamities. It may also lay out steps to restart technology quickly if it does go down. Disaster recovery (DR) terms appear most often in cloud services agreements.

Disaster Recovery Plan

Vendor shall comply with the Disaster Recovery Plan set forth in Attachment __ (*DR Plan*).

• • • •

Vendor shall maintain and comply with a reasonable written plan (the "DR Plan") setting forth procedures for: (a) keeping the Service functioning during and after an earthquake, hurricane, other natural disaster, war, act of terrorism, act of cyberterrorism, and other natural or man-made disaster, including without limitation force majeure (as that term is used in Section __, *Force Majeure*) (collectively, a "Disaster"); and (b) restoring Service functionality promptly after a Disaster. The DR Plan will include procedures no less protective than industry standard, and Vendor shall update the DR Plan as the industry standard changes.

Neither example in the clause box above lays out the vendor's actual DR obligations. Those would appear in a disaster recovery plan, usually a separate document, possibly attached to the contract. The DR plan might call for anything from a set of steps for getting a crashed system running to a whole backup data center ready to take over on a minute's notice. The second example in the clause box above suggests two topics for a DR plan: procedures for *preventing* system failures and procedures for *recovering* after a disaster does disable the system. A DR plan could address either or both.

If the DR plan does require a whole backup system, look at Sub-chapter II.S.4 ("Cloud Services Escrow and Pseudo-Escrow"), particularly the clause box "Cloud Services Pseudo-Escrow" on page 185. That subchapter lays out terms for a backup (mirror) system meant to address vendor breach or bankruptcy. But it works for disaster recovery, too, possibly with slight modification.

The second example in the clause box above refers to the contract's force majeure clause: a common set of terms providing that a party isn't in breach of contract if forces beyond its control keep it from performing. Force majeure clauses often include a list of disasters and other forces that might interfere, so they can help define "disaster." See Chapter III.J ("Force Majeure").

The examples in the clause box above, and the DR plans they suggest, address *procedures* for disasters. In other words, they require that the vendor use disaster recovery systems, not that it actually prevent technology failures or recover from disasters. If the vendor follows its procedures but a disaster still disables the system, or keeps it down longer than expected, the vendor hasn't breached the DR clause. And the force majeure clause reinforces that conclusion.

Of course, with best-in-class disaster management systems, a disaster won't take the technology down anyway, unless it's a zombie apocalypse or the election of a Tea Party president. So contracts often don't address what happens if the DR plan fails to do its job. But it's worth some thought.

Failure of Disaster Recovery

No failure of the System and no Vendor obligation shall be excused pursuant to Section __ (*Force Majeure*) if the failure could have been avoided but for Vendor's breach of this Section __ (*Disaster Recovery*) or of any of Vendor's other obligations set forth in this Agreement.

• • • •

The Service's performance requirements and Vendor's other obligations pursuant to this SLA shall apply at all times, including without limitation during natural disasters, wars, or other event of force majeure (as that term is used in Section __, *Force Majeure*) (collectively, "Disasters"), except in case of a Disaster that _____________ (a "Mega-Disaster").

The first example in the clause box above confirms the intent of most DR clauses. It says that disasters don't excuse system failures if the vendor breached the DR procedures, and those procedures could have prevented the loss. Arguably, you don't need the first example if that's your intent, since most DR and force majeure clauses read that way anyway. But why not make it clear?

The second example in the clause box imposes tougher obligations on the vendor. Its obligations go beyond following DR procedures; the vendor has to keep the system running in the face of a disaster, unless it’s catastrophic. That only makes sense for truly critical systems and well-heeled vendors.

Vendors shouldn’t give up all force majeure protection. So the second example above still lets the vendor off the hook for force majeure, provided it’s an unusually big disaster. The blank at the end of the clause box lets you define “Mega-Disaster.” The definition depends on the nature of the technology and of the DR systems required. You might describe an event that disrupts Internet access to all the vendor’s data centers or to the majority of customers in the vendor’s state, for instance.

S. Technology Escrow

Software licensing customers often receive object code but not source code. That usually means the customer can’t maintain the software, since technicians need source code to see how the system is supposed to work. That’s not a problem if the vendor offers any necessary maintenance service. But what if the vendor goes out of business? Or what if the vendor breaches its maintenance obligations? If the system’s vital, the customer will be in trouble. That’s why some software licenses include technology escrow provisions.

Cloud services agreements sometimes include escrow provisions too. But escrow doesn’t fit cloud services as well as licensed software. This chapter addresses cloud services escrows in Subchapter 4 below. The rest of this chapter focuses on source code escrows for licensed software. But you can actually put almost any asset in escrow.

In an escrow clause, the vendor gives a reliable third party a copy of its source code, and of any supporting documentation. This third party, the “escrow agent,” holds the materials for the customer’s benefit. The customer gets the source code if an agreed “release condition” happens. Release conditions usually include the vendor’s bankruptcy or breach of its service obligations.

The deal needs the escrow agent because the customer can’t assume it’ll get the source code from the vendor, if and when the time comes. If the vendor someday breaches its service obligations, it might also breach any promise to turn over source code. And if the vendor goes bankrupt, the law will relieve it of most contract obligations, like promises to turn over source code.

The escrow agent could be almost anyone the parties trust, but the most reliable services come from companies that specialize in technology escrow.[47] These professional escrow agents generally have their own form contract setting up the relationship. The “escrow agreement” often becomes an attachment to the parties’ license agreement. Customer, vendor, and escrow agent all sign it—usually at the same time as the customer and vendor sign the license agreement.

Separate Escrow Agreement

Concurrent with execution of this Agreement, the parties shall execute a third party escrow agreement in the form attached hereto as Attachment __ ("the Escrow Agreement"), in conjunction with ______________ (the "Escrow Agent").

Most terms of the separate escrow agreement relate to the mechanics of the parties' relationship with the escrow agent: storage of materials, payment of the agent, etc. This book doesn't address those terms. It does address terms governing the escrow relationship between the customer and the vendor. Those terms can appear either in the escrow agreement or in the main license agreement. If they appear in the escrow agreement, they're usually subject to negotiation and revision by the vendor and the customer, unlike the other terms of the standard escrow agreement.

1. Deposit and Verification

The customer should make sure the vendor gives the escrow agent the right source code and supporting materials.

Escrow Deposit and Verification

(a)*Deposit.* Within ___ business days of the Effective Date, Vendor shall deposit with the Escrow Agent, pursuant to the procedures of the Escrow Agreement, the source code for the Software, as well as the Documentation and names and contact information for each author or other creator of the Software. Promptly after release of any update, upgrade, patch, bug fix, enhancement, new version, or other revision to the Software, Vendor shall deposit updated source code, documentation, names, and contact information with the Escrow Agent. ("Deposit Material" refers to material required to be deposited pursuant to this Subsection __(b).)

(b)*Verification.* At Customer's request and expense, the Escrow Agent may at any time verify the Deposit Material, including without limitation by compiling source code, comparing it to the Software, and reviewing the completeness and accuracy of any and all material. In the event that the Deposit Material does not conform to the requirements of Subsection __(a) above: (i) Vendor shall promptly deposit conforming Deposit Material; and (ii) Vendor shall pay the Escrow Agent for subsequent verification of the new Deposit Material. Any breach of the provisions of Subsection __(b)(i) above will constitute material breach of this Agreement, and no further payments will be due from Customer until such breach is cured, in addition to such other remedies as Customer may have.

First and foremost, the escrow clause should list the deposit material, particularly source code and documentation, as in Subsection (a) of the clause box above. If the contract doesn't call for documentation, or the documentation isn't very complete, customers should try to require deposit of "all documentation necessary to enable a person of reasonable skill with software to compile and build machine-readable code for the Software, to maintain the Software, and fully to operate the Software."

Customers should also consider adding employee contact information to the escrow material. No one understands software better than its authors, and the customer might want to hire them if the vendor stops providing maintenance.[48] Subsection (a) in the clause box above gives the customer contact information for the programmers. (For a larger IT system, the language might limit itself to certain key programmers.) Of course, programmer contact information doesn't necessarily have to pass through the escrow. If the information isn't sensitive, the vendor might just hand it over.

What if the vendor deposits inadequate material? The customer won't know what's been deposited unless and until it gets its hands on the material—after a release condition. By then the vendor will probably be out of business, or at least uncooperative, and it will be too late to insist on compliance. That's why many escrow clauses include verification procedures, as in Subsection (b) of the clause box above. The escrow agent checks to see if the vendor has deposited the right material. This verification system is another reason to hire a professional escrow company rather than use a trusty friend. Verification can be a big job, requiring significant technical expertise and resources.

2. License and Confidentiality

If the customer does someday receive source code and other deposit material, it will need the right to exploit them. In other words, it will need a license.

In some cases, the contract's license clause already provides the necessary rights. If the licensed software includes all elements of the code and documentation, including source code and other deposit material, the customer doesn't need an additional license. If not, the customer will need a separate license like the one in the following clause box. The license should include the same terms as the license to the underlying software, except that if the customer would handle its own maintenance and support after a release, it probably also needs the right to create derivative works. The source code license should also include some or all of the restrictions in the underlying software license.[49]

Escrow License

Vendor hereby grants Customer a license to use, reproduce, and create derivative works from the Deposit Material, provided Customer may not distribute or sublicense the Deposit Material or make any use of it whatsoever except for such internal use as is necessary to maintain and support the Software. Copies of the Deposit Material created or transferred pursuant to this Agreement are licensed, not sold, and Customer receives no title to or ownership of any copy or of the Deposit Material itself. The Deposit Material constitutes Confidential Information of Vendor pursuant to Section __ (*Nondisclosure*) of this Agreement (provided no provision of Section __ calling for return of Confidential Information before termination of this Agreement will apply to the Deposit Material).

The escrow license should be effective immediately, even though the customer doesn’t yet have access to the source code and may never get it. In other words, don’t write: “Upon the occurrence of a Release Condition, Vendor grants Customer a license . . .” If the vendor goes bankrupt, it can rescind a license like that. The only license that can reliably survive bankruptcy is one that’s effective *before* bankruptcy proceedings start.[50] That’s why the example above provides: “Vendor *hereby* grants Customer a license . . . .” That license is effective immediately. But keep in mind: the fact that the license is effective immediately doesn’t mean the customer can use the source code immediately. It has the legal right to do so, but it doesn’t have a *copy*. The escrow agent has the source code and won’t give it to the customer unless and until a release condition occurs.

Vendors should make sure the license limits the customer’s rights to deposit material. After all, if the vendor comes out of bankruptcy or recovers from whatever kept it from supporting the software, it will still want to protect its source code. So the license should limit the customer to internal use: it can only use the source code to support and maintain the software, as in the previous example.

Vendors should make sure the customer protects the secrecy of the source code. If the contract has a nondisclosure clause, the deposit material should be considered confidential information, as in the example in the clause box above.[51] And if there is no nondisclosure clause, the escrow clause should include some kind of confidentiality provision. For instance: “Customer recognizes and agrees that the Deposit Material includes trade secrets of Vendor, and Customer shall not disclose it to any third party except to the extent required by law.”

3. Release and Other Issues

What triggers release of material to the customer? Any event can serve, but release conditions usually fall into two categories: (1) the vendor breaches its obligation to maintain or support the software; and (2) the vendor goes out of business. (In many deals, these release conditions appear in the escrow agreement, rather than the main contract.)

Escrow Release Conditions

The term “Release Conditions,” as used in the Escrow Agreement, refers to any of the following: (a) material breach by Vendor of Section __ (*Maintenance*) of this Agreement, if such breach remains uncured ___ or more business days after Customer’s written notice; (b) any failure of Vendor to function as a going concern; (c) appointment, application for, or consent to a receiver, trustee, or other custodian for Vendor or its assets; (d) Vendor becomes insolvent or unable to pay its debts as they mature in the ordinary course or makes an assignment for the benefit of creditors; (e) Vendor is liquidated or dissolved; or (f) any proceedings are commenced with regard to Vendor under any bankruptcy, insolvency, or debtor’s relief law, and such proceedings are not dismissed within 60 days.

The vendor should make sure the contract talks about “material” breaches of support obligations, not just any technical breach. It should also make sure a breach doesn’t count as a release condition unless it goes uncorrected for some period of time, like 30 days. See Section (a) in the clause box above.

Vendors might argue that the only necessary release condition is breach of the maintenance obligation. After all, the customer doesn’t need the source code or other materials if the vendor continues maintaining the system, even if it does file bankruptcy or otherwise faces business trouble. So release conditions (b) through (f) above aren’t necessary, the argument goes. But lots of agreements include these “business trouble” release conditions, because vendors usually *do* stop providing maintenance soon after any of these going-out-of-business warning signs. Customers don't want to risk delays in accessing crucial source code.

What if, at some point, the customer tells the escrow agent a release condition has occurred, and the vendor disagrees? The escrow agent finds itself caught in the middle. Usually, the escrow agreement solves this problem with an arbitration clause. In case of dispute, the escrow agent doesn't have to release the material until it gets an order from an arbitrator. That's a good solution, but it can cause problems for the customer if the software is critical and needs quick maintenance. So customers should consider a contract clause calling for *expedited* arbitration, a very fast procedure available through many arbitration associations. See Chapter II.U ("Alternate Dispute Resolution").

Finally, vendors should be sure the escrow agreement includes a termination clause. It should provide that if the license agreement is terminated for any reason, other than breach by the vendor, the escrow agent will return the escrow material to the vendor.

4. Cloud Services Escrow and Pseudo-Escrow

Cloud services customers worry about their vendors going bankrupt too, or breaching their service obligations. In fact, a cloud services vendor's failure would likely be worse than a comparable software vendor's, because the cloud vendor hosts the system. If the vendor disappears, the customer doesn't just lose the ability to maintain the software; it loses the whole system.

As a result, many cloud services customers ask for escrows. But what do you put in escrow? The source code alone won't do much good because the customer never had the software anyway: the object code. You could put the object code in escrow, or both object code and source code, but even that might do the customer little good. To run the cloud system, the customer would probably also need appropriate computers and other hardware—possibly a whole data center—along with supporting software it's never even heard of, not to mention trained staff. Unless the customer and its contractors can manage a technology system almost as well as the vendor, the escrow won't help. And even if the customer *is* tech-savvy and has tremendous resources, the cloud services software might be so clunky that no one but the vendor's staff can operate it. After all, this isn't commercial licensed software: its authors may not have imagined that any customer would ever host it.

For many cloud services deals, escrow doesn't work. In a very large deal, however, with critical systems and rich parties (or at least one rich party), you should consider what this book calls "a pseudo-escrow." The vendor agrees to create a mirror system: a set of computers and other infrastructure more or less identical to the cloud system's, along with identical software, documentation, and data. Someone other than the vendor should host the mirror system, or at least have permanent access—or what's the point? An escrow agent could serve as the host, and one of the big escrow companies might muster the necessary resources and expertise. Or if a third party provides a data center for the system, that company could host the mirror system. Either way, the mirror system might have to be live online frequently, possibly permanently (though it's not used in production), so it can download the most current customer data and system updates.

Finally, someone has to agree to pay the escrow agent/host and to pay for all the extra hardware and software and the staff to set them up and run them.

Cloud Services Pseudo-Escrow

Throughout the Term, Vendor shall maintain a copy of the System, including without limitation all software, hardware, and other infrastructure listed on Attachment __ (*System Components*) or otherwise necessary to operate the System in material conformity with the Specifications (the "Mirror System"). The Mirror System shall be maintained in a data center run by __________ ("Escrow Host"), and Vendor shall grant Escrow Host such rights, and provide Escrow Host with such documentation, as necessary to operate and maintain the Mirror System in material conformity with the Specifications. Vendor shall, or shall enable Escrow Host to: (a) update the Mirror System's software, hardware, and other infrastructure, and the Customer Data it contains, so that they are materially the same as that of the System, no less often than _______ [every second, day, week]; and (b) maintain Mirror System so that it can replace the System and continue operations in material conformity with the Specifications within _________ [seconds, hours, days] of a Release Condition. The parties shall pay for the components of the Mirror System, and pay the Escrow Host, as follows: ____________________.

The clause box above offers the heart of a cloud services pseudo-escrow. You'll also need a contract with the escrow host, including instructions about going live after a release condition. You may also need terms about providing the host with staff members capable of running the system. The escrow host will also need a license from the vendor to the system's software. On top of all that, vendors should consider confidentiality requirements for both the host and the customer, protecting any secret features of the system—possibly all the software—as suggested in Subchapter 2 above. And vendors should restrict the escrow host’s use of the system, both before a release condition and after, to make sure the host doesn’t compete with the vendor or its successors.

In other words, a meaningful cloud services escrow probably involves a large project.

T. Financial Stability

When one company comes to rely on another, it often wants assurances of the other's financial stability. Technology agreements provide those assurances through reporting clauses and insurance clauses. These financial stability terms appear most often in cloud services agreements, but they're not limited to those contracts. Usually, it's the customer that wants assurances of the vendor's stability. That's the case with the examples in this chapter, but there's no reason the vendor can't get financial stability assurances from the customer.

Many customers also want assurances about their vendors' technology, particularly the ability of cloud services to protect data. Customers ask for third party audits of those systems, like "SOC 1" or "SSAE 16" audits, or "SOC 2" audits. Those audits and related work by the auditors may provide some insight into the vendor's financial stability, too. This book addresses them in Subchapter II.H.3 ("Data Security").

1. Financial Reporting

Reporting clauses give the customer early warning of trouble. If the vendor faces financial instability, it could tumble into bankruptcy, and it could lose the will or ability to perform vital services, like keeping sensitive data safe. But if the customer sees that trouble far enough in advance, it might be able to protect itself—by terminating the contract, retaining another vendor, taking back the data, etc.

Financial Reporting

Once per _________, on reasonable advance notice, Customer may review Vendor's financial books and records, including a current balance sheet, a statement of income and losses for the preceding 12 months, and a statement of cash flow for such 12 months. Customer's review will be conducted in Vendor's offices, and Customer will have no right to retain copies of any books or records. All books and records and all information contained therein will be Vendor's Confidential Information pursuant to Section __ (*Nondisclosure*).

The example in the clause box above gives the customer the right to review the vendor's key books and records. Obviously, vendors should hesitate before granting anyone that kind of access. The clause makes the most sense when the vendor is a small company without a track record of stability, and a valuable customer plans to rely on it for vital services. The clause wouldn't make sense for a publicly traded vendor, which discloses its financial performance to the public every quarter.

The clause box provides that the vendor's reports and records are "Confidential Information" pursuant to the contract's nondisclosure clause. If the contract lacks such a clause, the vendor should insist on a separate NDA.[52]

2. *Insurance*

An insurance clause requires that the vendor maintain various insurance policies. That helps ensure enough money to keep the vendor operating and to compensate the customer for damages the vendor causes.

The example in the following clause box calls for a relatively broad spectrum of insurance policies. But some contracts add another type of coverage: insurance for damage caused by technology-related errors and omissions. And many IT contracts require less, limiting themselves to "commercial general liability insurance" (covered in Subsection (a) below).

Insurance

During the term of this Agreement and for _______ thereafter, Vendor shall maintain in full force and effect: (a) commercial general liability insurance covering personal injury and property damage, including without limitation contractual liability, with limits of at least $_______ per occurrence and $_______ in the aggregate; (b) business automobile liability insurance for all vehicles, including those owned or rented by Vendor or its employees, covering personal injury and property damage, with a limit of at least $_______ per occurrence; (c) worker's compensation and employer's liability insurance, with limits of at least $_______; and (d) cyber liability insurance covering ____________, with a limit of at least $_______ per occurrence. Vendor shall maintain all such insurance with carriers rated __ or better by _____________. The insurance policies required pursuant to this Section __ will stipulate that they are primary insurance and that no insurance policy or self-insurance program of Customer will be called upon to contribute. Before commencement of Services, and from time to time thereafter upon renewal of any such policy of insurance, Vendor shall provide Customer with certificates of insurance evidencing the above coverages and naming Customer as certificate holder entitled to 30 days' written notice following any cancellation, reduction, or change in coverage.

In the example above, the insurance policies' limits ($2 million, $5 million, etc.) vary with the parties' needs. The same goes for the minimum financial rating of the insurance carriers, as well as the source of that rating. Many clauses call for "insurance with carriers rated A- or better by A.M. Best Company." [53]

The coverage period varies, too. Some contracts call for insurance during the term of the agreement and no longer. But if claims affecting the customer could come in after contract termination, the clause should require that insurance coverage continue for some set period, as in the first sentence of the example in the clause box.

Subsection (d) in the clause box calls for cyber liability insurance. That's not a well-defined concept, so the subsection provides a blank to fill in the required coverage. The policy could protect against losses from data breaches, denial of service and other network attacks, IP claims, website defacement, and even online blackmail. Obviously, the more detail about the policies required, and not required, the better.

U. Alternate Dispute Resolution

Alternate dispute resolution clauses aim to keep the parties out of court. They provide alternative procedures for resolving arguments.

The following clause box provides three alternate dispute resolution (ADR) clauses: escalation, mediation, and arbitration. Your contract might include any or all of these.

Dispute Resolution

In the event of dispute, either party may call for escalation by written notice to the other. Within __ business days of such notice, each party shall designate an executive with authority to make commitments that would resolve the dispute (a “Senior Manager”). The parties’ Senior Managers shall meet in person or by telephone (“Dispute Conference”) within __ business days of their designation and shall negotiate in good faith to resolve the dispute. Except to the extent necessary to prevent irreparable harm or to preserve rights or remedies, neither party shall initiate arbitration or litigation until 10 business days after the Dispute Conference.

• • • •

If the parties cannot themselves resolve a dispute arising out of or related to this Agreement, they shall attempt to resolve such dispute through mediation under the auspices of ___________ [ADR association], in __________ [city], with the parties sharing equally the costs of mediation. Except to the extent necessary to prevent irreparable harm or to preserve rights or remedies, neither party shall initiate arbitration or litigation until 30 days after the first mediation conference, unless the other party has materially breached its obligations set forth in the preceding sentence.

• • • •

Any claim arising out of or related to this Agreement, including without limitation claims related to the parties’ negotiations and inducements to enter into this Agreement, shall be submitted to mandatory, binding arbitration under the auspices of ___________ (the “ADR Association”), in __________ [city], with the parties sharing equally the costs of arbitration. Arbitration will proceed according to the commercial rules of the ADR Association. This Section __ does not limit either party’s right to provisional or ancillary remedies from a court of competent jurisdiction before, during, or after the pendency of any arbitration, and the exercise of any such remedy does not waive either party’s right to arbitration. Judgment on an arbitration award may be entered by any court with competent jurisdiction.

Escalation is the least formal alternate dispute resolution procedure. It calls on the parties to bump the argument up to executives, as in the first example in the clause box above. An escalation clause should provide that no one can file a lawsuit until the senior executives have met and tried to resolve the dispute.

Mediation is a bit more formal. In a mediation clause, like the second example in the clause box, the parties agree to work with a third party on resolving the dispute. The mediator tries to broker a deal, including by helping each party see the disadvantages of litigating.

Arbitration is probably the most common ADR clause in IT contracts. In provisions like the third example in the clause box, the parties agree to let a third party *decide* their dispute. This third party arbitrator acts like a judge. Usually, the contract provides that arbitration is "mandatory and binding." "Mandatory" means that if one party wants to arbitrate, the other can't escape and go to court. "Binding" means the arbitrator's decision is final. There's no appeal to a court or anyone else (unless the arbitrator does something clearly improper, like accepting a bribe or ignoring obvious legal rules). In other words, arbitration creates a private "people's court,"[54] with power to act like a real court.

Mediation usually involves one "neutral" helping the parties settle the dispute, while arbitration may involve one or three arbitrators. Three-person arbitrator panels often appear in bigger deals: "Each party shall select an arbitrator who has no financial or family relationship with such party and identify him or her in writing within 10 days of the demand for arbitration. The two arbitrators shall select a third arbitrator within 10 days of written identification of both." Three-person panels cost more, but with more arbitrators, you have less chance of error, bias, or random craziness.

You can retain mediators and arbitrators through dispute resolution companies, like the American Arbitration Association or JAMS.[55] Those companies usually hire lawyers and retired judges, though in some cases they bring in specialists with technical expertise. Some trade groups offer ADR services, too, as do many lawyers. Many ADR companies and organizations provide language for mediation and arbitration clauses, so check to see if your chosen association has language you might prefer to the examples in the clause box above.

ADR associations may provide their own private rules of civil procedure —rules for conducting discovery and otherwise running an arbitration—as well as rules for selecting arbitrators. Your clause should specify the rules you'll use, particularly if your chosen association has more than one set.

The third example in the clause box above calls on the ADR association's "commercial rules," but your association might have something more specific, like "information technology rules." If you pick an arbitrator or association without standard rules, your ADR clause should address discovery and other procedural issues, so the parties don't find themselves litigating about procedures. At the very least, confirm your arbitrator's power to define the rules: "The arbitrator shall resolve any disputes regarding procedure for conducting the arbitration, including discovery."

In theory, arbitration is cheaper and faster than litigation because it involves less formal procedures and less crowded dockets. Some lawyers, however, doubt arbitration's value as a time or money saver. The best I can say is that arbitration is *often* cheaper and faster. But arbitration does have three other clear advantages. First, in an arbitration, you might be able to choose decision makers with appropriate technical expertise. Juries of regular folks can make a mess of technical cases, and so can judges.

Second, an arbitration clause can require *expedited* arbitration. For instance: "The parties shall submit briefs within 3 business days of selection of the arbitrator; the arbitrator shall hold a hearing within 3 business days of submission of briefs; and the arbitrator shall issue the decision within 5 business days of the hearing." Also, ADR associations often provide procedures for expedited arbitration. So the clause might read: "In the event that the arbitration relates to release of source code for the Mission Critical Application, arbitration will proceed pursuant to Geriatric Judges' *Expedited Arbitration Rules*, and the parties shall take all required actions as promptly as reasonably possible."

Third, in many foreign countries, it's easier to enforce an arbitrator's ruling than an American court's. That's because of the 1958 UN Convention on the Recognition and Enforcement of Foreign Arbitral Awards, also known as the New York Convention. The New York Convention is a treaty signed by most industrial nations, including India, China, Russia, Japan, Canada, most European countries, and the United States. Each nation promises to enforce foreign arbitrators' decisions—even though it might not always enforce foreign *courts'* decisions. Imagine you want disputes resolved in the United States, but one of the parties is Indian and has few American assets. If you litigate and win in the United States, you might need an Indian court to enforce the judgment. In most cases, you'll face fewer obstacles if that judgment came from an American arbitrator, as opposed to a U.S. court, thanks to the New York Convention. So if your contract does involve foreign interests, you should consider an arbitration clause for that reason alone. And if you do choose such a clause, your contract should mention the Convention: “This Agreement is subject to the operation of the 1958 United Nations Convention on the Recognition and Enforcement of Foreign Arbitral Awards.”

Finally, note that in nonnegotiable form contracts, it’s usually a good idea to “call out” the arbitration clause. The point is to warn the other party, in clear, bold text, that the contract requires arbitration. Many form contracts provide a warning on the top line: “Terms of Service: CONTAINS ARBITRATION CLAUSE.” Some online contracts go even further: the link to the contract includes an arbitration warning, and a Web surfer has to click on the warning to see the contract. The reason is that courts sometimes won’t enforce an arbitration clause without clear, undeniable notice, particularly against consumers—despite the clause’s “mandatory” language.[56]

• • •

Another type of alternate dispute resolution clause crops up in IT contracts, though it’s rare. Sometimes, the parties waive their right to a jury trial. Disputes go to court, but the judge decides the facts. (In jury trials, the jury decides the facts while the judge interprets the law.) Juries tend to be more emotional than judges and less capable of grasping technical issues, so trial by judge has some advantages. Here’s a sample jury waiver clause—in caps, to make sure no one fails to notice this waiver of constitutional rights: “EACH PARTY HEREBY WAIVES ITS RIGHT TO A TRIAL BY JURY FOR DISPUTES ARISING OUT OF OR RELATED TO THIS AGREEMENT, INCLUDING WITHOUT LIMITATION COUNTERCLAIMS REGARDING SUCH DISPUTES, CLAIMS RELATED TO THE PARTIES’ NEGOTIATIONS AND INDUCEMENTS TO ENTER INTO THIS AGREEMENT, AND OTHER CHALLENGES TO THE VALIDITY OR ENFORCEABILITY OF THIS AGREEMENT. THE WAIVER IN THE PRECEDING SENTENCE APPLIES REGARDLESS OF THE TYPE OF DISPUTE, WHETHER PROCEEDING UNDER CLAIMS OF CONTRACT OR TORT (INCLUDING NEGLIGENCE) OR ANY OTHER THEORY.”

V. Term and Termination

Termination clauses address four issues. First, when, if ever, will the contract expire? What is its *term* or duration? Second, when can a party terminate for cause? Third, when, if ever, can a party terminate for convenience: for any reason or no reason? Fourth, what happens after termination?

Some contracts don't need term and termination provisions. In a simple purchase agreement, for example, termination might not make sense. The vendor provides the goods, the customer pays, and then the deal is done. There's nothing to terminate. Term and termination clauses make the most sense in contracts with continuing rights or obligations—like most IT contracts.

1. Term

The term is the period during which the contract operates: its duration. The word implies something temporary, like a senator's "term" of office.

Term

This Agreement will continue until terminated by either party as specifically authorized herein.

• • • •

This Agreement will terminate on Customer's acceptance of the Final Deliverable (as defined in Section __, *Deliverables*).

• • • •

This Agreement will remain in effect for _____ years from the date of execution by both parties. Thereafter, it will renew for successive 1-year periods, unless either party refuses such renewal by written notice 30 or more days before the end of the current term.

• • • •

This Agreement will continue until terminated by either party as specifically authorized herein. Vendor will provide Maintenance for a period of _______, starting on the Effective Date. Thereafter, Maintenance will renew every _____, unless Customer notifies Vendor of its intent not to renew 30 or more days before any renewal date. After Maintenance has renewed ______ times, Vendor may refuse any subsequent renewal by written notice 30 days before the next renewal date.

In some contracts, the parties' rights and obligations continue indefinitely. In that case, use an open-ended term, continuing until someone terminates, as in the first and last examples in the clause box above. The parties can also select a more definite end date—one year from signing, for instance, or upon completion of services—as in the second and third examples in the clause box.

Term clauses often let the agreement renew over and over. If the term does renew automatically, the clause will give one party or both the right to refuse the next renewal. (That's the point of multiple renewing terms, as opposed to a single long term.) See the third example in the clause box above. No one should accept permanent renewal without the right to get out at some point. So if only one party has the right to refuse renewal at first, the other should get that right eventually. See the fourth example in the clause box above (regarding termination rights for maintenance).[57]

A term clause may address the duration of certain rights and obligations, rather than the duration of the whole contract. In software licenses, for instance, the customer's license rights may continue indefinitely while the vendor's maintenance obligations eventually end, or at least could end. In that case, the agreement has an open-ended term, but maintenance doesn't. See the last example in the clause box above.

2. *Termination for Cause*

Termination for cause happens when something's gone wrong, particularly when someone has breached the contract.

Termination for Cause

Either party may terminate this Agreement for the other's material breach by written notice, effective in 30 days unless the other party first cures such breach.

• • • •

Either party may terminate this Agreement for the other's material breach on 30 days' written notice, unless the other party cures such breach before the effective date of termination; provided termination will become effective immediately upon such notice, without opportunity to cure, if: (a) this Agreement provides that "times is of the essence" with regard to the performance subject to the breach; or (b) the injury caused by the breach cannot be remedied by performance after notice of termination.

• • • •

Either party may terminate this Agreement for cause by written notice, without opportunity to cure, in the event that: (a) the other party fails to function as a going concern; (b) a receiver, trustee, or other custodian for the other party or its assets is appointed, applied for, or consented to; (c) the other party becomes insolvent or unable to pay its debts as they mature in the ordinary course; (d) the other party makes an assignment for the benefit of creditors; (e) the other party is liquidated or dissolved; or (f) any proceedings are commenced by or against the other party under any bankruptcy, insolvency, or debtor's relief law and not dismissed within 60 days.

The usual *cause* for termination is breach of contract. Most clauses clarify that the breach must be "material," as in the first and second examples in the clause box above. That means a minor breach, like delivering software a day late, won't authorize termination. The law generally reaches the same conclusion, but clarity help.

Often, termination for breach clauses require advance notice—30 days is common—and an opportunity to cure, as in the first two examples in the clause box. If the breaching party fixes the breach during the cure period, the contract isn't terminated. But a cure period isn't required. Nor is advance notice, though the clause should at least require written notice. Also, some contracts allow immediate termination, without an opportunity to cure, for particularly important deadlines, usually through terms saying "time is of the essence" for that particular performance. See Subsection (a) of the second example in the clause box.[58] And some breaches can't be cured, so a notice and cure period wouldn't make sense. If the contract requires that the vendor keep a secret, and the vendor gives the secret to the press, 30 days to cure won't make any difference. See Subsection (b) in the second example in the clause box.

Sometimes the parties need the right to terminate for cause even without a breach. Either party might want to escape if the other goes bankrupt, as in the third example in the clause box above. (Bankruptcy will threaten payment, provision of services, and other obligations.)

3. Termination for Convenience

Termination for convenience is an escape hatch. It lets a party get out for any reason or no reason at all. If you're afraid the contract might become burdensome—including because of some business change you can't predict—consider a termination for convenience clause.

Termination for Convenience

Either party may terminate this Agreement for any reason or no reason on __ days' advance written notice.

• • • •

Customer may terminate this Agreement for convenience upon __ days' advance written notice. On the date of such termination, Customer shall pay Vendor an early termination fee of __ % of the fees for Services not yet performed.

Convenience clauses sometimes authorize termination "for any reason or no reason." See the first example in the clause box above. That's arguably a clearer way of putting it, but I think "convenience" is so widely understood that the two are interchangeable. See the second example.

Most convenience clauses require a notice period, as in both examples in the clause box above. Usually, that period lasts longer than the 30 days typically required in a termination for breach clause. A notice period of 90 days is common.

Many contracts impose a price on termination for convenience. The party terminating has to pay a fee. The fee generally represents a rough guess at the other party's losses caused by termination. For instance, in a professional services contract, the vendor might spend time and money hiring or reassigning staff to serve the customer's needs, at substantial cost. If the customer terminates early, the termination clause could require fees that more or less match that cost.

The second example in the clause box above uses a percent of the unpaid, unperformed services as the early termination fee. But you could also use a fixed fee or calculate the fee in almost any other way.

Don't confuse early termination fees with liquidated damages, covered in Chapter II.Q ("Liquidated Damages"). Termination for convenience is not breach of contract; it's authorized. So the fee isn't a damages calculation. It's the *contract's price* for early termination. Liquidated damages, on the other hand, are *damages for breach of contract*. They have nothing to do with termination for convenience.

4. Effects of Termination

Termination doesn't mean the contract disappears—that it becomes null and void. In fact, termination *triggers* certain clauses. They go into effect the moment the contract terminates. Other clauses continue both before and after termination. So what does termination do? It ends the flow of goods and services—usually the obligations addressed in the prime clauses, described in Part I of this book. In a software license, the customer usually loses its rights to exploit the software. In a cloud services or professional services contract, the vendor no longer has to provide the services. And of course, the customer doesn't have to pay for software or services it's not receiving.

Termination can trigger a variety of obligations. For instance, in some contracts, each party should promise to return the other's property upon termination or shortly after. And as explained in Sub-chapter 3, termination for convenience may trigger an obligation to pay early termination fees. Some of these "triggered provisions" will already appear in other clauses. Nondisclosure clauses, for instance, typically provide: "Upon termination of this Agreement, Recipient shall return all Confidential Information to

Discloser.”[59] The termination clause often lists some of the triggered provisions, too.

Effects of Termination

Upon termination of this Agreement, the licenses granted in Section __ (*Software License*) will terminate, Customer shall cease all use of the Software and delete all copies in its possession or control, and each party shall promptly return any property of the other’s. The following provisions will survive termination of this Agreement: (a) any obligation of Customer to pay for Software used or services rendered before termination; (b) Sections __ (*Nondisclosure*), __ (*Data Management & Security*), __ (*Indemnity*), __ (*Limitation of Liability*), and __ (*Arbitration*); and (c) any other provision of this Agreement that must survive to fulfill its essential purpose.

Certain rights and obligations “survive” termination, though they’re not triggered by termination. What if the data management clause says the vendor can’t ever disclose private information? What if the nondisclosure clause requires protection of confidential information for three years? Those obligations shouldn’t end just because the contract terminates after nine months. Clauses that govern interpretation of the contract generally survive, too. These include provisions about alternate dispute resolution, limitations of liability, definitions, and severability.

Many contracts specify the clauses that survive termination. Often, the clause itself addresses survival: “The provisions of this Section 8 will survive any termination or expiration of this Agreement.” The termination clause may also list the surviving clauses, as in the example in the clause box above. It’s not strictly necessary to specify survival of clauses that govern interpretation of the contract. It should be obvious that a definitions clause or a choice of law clause survives. But the rule isn’t so obvious for survivors that require affirmative action on one party’s part, like indemnity and nondisclosure clauses. In general, the safest course is to specify all the clauses that survive. And just to make sure you didn’t miss anything, throw in “any other provision of this Agreement that must survive to fulfill its essential purpose,” as in Subsection (c) of the clause box.

Software licenses raise some special termination issues. IT professionals generally assume software rights terminate with the contract, but the law isn’t clear. Just to be safe, vendors should make sure the contract specifies license termination and explicitly requires that the customer stop *using* the software—or the distributor stop distributing it. See the example in the clause box above. Of course, sometimes the parties do intend license survival. If the license is perpetual and irrevocable, the termination clause should list the license as one of the clauses that survives termination.[60] Also, in a "value-added" reseller agreement, the distributor may need to keep its rights to reproduce and use the software to the extent necessary to support units of its product already licensed or sold.[61]

W. Everything Else

What have the parties negotiated? Whatever the terms are, write them in clear, simple English. As the Introduction explains, good contracts are customized. Contract drafting is a creative process, like doing business itself, so don't hesitate to blaze new trails. Just think through what you've agreed to say, then write it down.

---

[1]. For more on these five clauses, see Chapters and Subchapters II.I.1 ("Warranty of Function"), II.C ("Maintenance, Including Updates and Upgrades"), I.F.1 ("Defining the Professional Service"), II.F ("Delivery, Acceptance, and Rejection"), and II.B ("Service Level Agreements").

[2]. See Subchapter II.B.3 ("SLA Modification") for modification of service level agreements, which raises similar issues.

[3]. Some SLAs address data security or disaster recovery standards, but those terms often appear in separate clauses. This book addresses them in Chapters II.H ("Data Management and Security") and II.R ("Disaster Recovery").

For more on the overlap between SLAs and maintenance clauses, see Chapter II.C ("Maintenance, Including Updates and Upgrades").

[4]. See Subchapter II.V.3 ("Termination for Convenience").

[5]. Chapter II.B ("Service Level Agreements").

[6]. See Chapters II.F ("Delivery, Acceptance, and Rejection") and II.A ("Technical Specifications").

[7]. See Subchapter II.V.2 ("Termination for Cause").

[8]. See Chapter II.A ("Technical Specifications").

[9]. Nondisclosure clauses also appear in many employment contracts. But be warned: some states restrict employers' rights to impose nondisclosure terms on their employees, particularly where the terms might limit the employee's future job prospects.

[10]. This definition paraphrases the Uniform Trade Secrets Act, adopted in most states.

[11]. The forms library at this book's website includes a sample NDA you can use for the recipient's employees. Please visit http://TechContracts.com.

[12]. See Chapters I.C ("Software Licenses in General") and I.D ("Technology Ownership: Assignment and Work-for-Hire").

[13]. See Chapter II.G ("Nondisclosure/Confidentiality").

[14]. The federal government's key data security laws include the Gramm-Leach-Bliley Act (GLBA), governing financial institutions, the Health Insurance Portability and Accountability Act (HIPAA), the Children's Online Privacy Protection Act (COPPA), and arguably the Sarbanes-Oxley Act (on corporate corruption and financial reporting). Most states have information privacy laws and regulations too. Massachusetts has some of the most far-reaching regulations, codified at 201 CMR 17.00.

[15]. Not a real law.

[16]. SOC 2 reports use the AICPA's Attestation ("AT") standards Section 101. "SOC 3," discussed here too, uses the AT Section 101 standard too.

[17]. SSAE 16" stands for Statement of Standards for Attestation Engagements No. 16. It's the AICPA's standard for conducting SOC 1 reports. SSAE 16 replaces an older standard, SAS 70: Statement on Accounting Standards No. 70.

[18]. Representations and warranties do play different roles in litigation. A false representation gives rise to a claim of misrepresentation, while a false warranty gives rise to a claim of breach of warranty. But the litigation distinction isn't necessarily dependent on whether the contract uses the term "represent" or "warrant." The nature of the claim usually depends on the context, not the choice of words.

[19]. See Chapter II.A ("Technical Specifications").

[20]. For acceptance, see Chapter II.F ("Delivery, Acceptance, and Rejection").

[21]. See Chapter II.B ("Service Level Agreements").

[22]. See Chapter II.K ("Limitation of Liability").

[23]. For more on the differences between copyrights and other forms of IP, see Appendix 1 ("Intellectual Property").

[24]. The American Law Institute (ALI), a prestigious legal advisory body, has suggested that the law includes or should include an implied indemnity against IP infringement. This suggestion is controversial. See ALI's *Principles of the Law of Software Contracts* (ALI 2010), § 3.01. See also Chapter II.J ("Indemnity").

ALI has also suggested that the law includes or should include a non-disclaimable warranty of no "hidden material defects." That suggestion is controversial too. See *Principles* § 3.05(b).

[25]. Of course, if the contract has a strong limitation of liability clause, the vendor still might not be very motivated to go the extra mile. See Chapter II.K ("Limitation of Liability").

[26]. *See* Brocato, "Drafting Indemnification Clauses," *The Transactional Lawyer*, A Publication of the Commercial Law Center, Gonzaga University, Vol.1 (Oct. 2011), p. 1.

[27]. Some indemnitors (particularly IT vendors) do away with both "indemnify" and "hold harmless." For instance: "Vendor shall defend Customer against any Covered Claim and pay any settlements or judgments resulting therefrom." That may mean exactly the same thing as an obligation to "defend and indemnify." Whether it does, or whether the customer loses something, depends on the applicable state laws.

[28]. See Subchapter II.K.4 ("Exceptions: Liability That's Not Limited").

[29]. *The example in this clause box is not complete!* It replaces the first sentence of Subsection (a) of the clause box in Subchapter 1 above. If you use this example, be sure to include the rest of the language from the Subchapter 1 clause box (with "Vendor," "Customer," and "Customer Associates" replacing "Indemnitor," "Indemnified Party," and "Indemnified Associates").

[30]. An IP indemnity covers the same ground as an IP warranty, so some of the same considerations apply. See Subchapter II.I.2 ("Intellectual Property/Ownership Warranty").

[31]. That switches the clause from a *conduct-based indemnity* to a *status-based indemnity*. See Subsection 5 below for the distinction.

Usually these clauses don't extend the indemnity to cases involving *the customer's* negligence, but they could.

[32]. Section 2-312(3) of the Uniform Commercial Code, which forms the basis of many states' contract laws, provides that "a buyer who furnishes specifications to the seller must hold the seller harmless against [a rightful claim of infringement or the like] which arises out of compliance with the specifications." It's not clear whether or how that applies to IT contracts.

[33]. *The examples in this clause box are not complete!* Each replaces the first sentence of the clause box in Subchapter 1 above. If you use these examples, be sure to include the rest of the language from the Subchapter 1 clause box (with "Customer," "Vendor," and "Vendor Associates" replacing "Indemnitor," "Indemnified Party," and "Indemnified Associates").

[34]. For policies regarding customers' use of online systems—acceptable use policies (AUPs)—see Appendix 4 ("Online Policy Documents"). The same appendix addresses the Digital Millennium

Copyright Act, which protects certain online vendors from customers' copyright problems.

[35]. *The example in this clause box is not complete!* Its terms replace the first sentence of the clause box in Subchapter 1 above. If you use this example, be sure to include the rest of the language from the Subchapter 1 clause box (with "Distributor, "Vendor," and "Vendor Associates" replacing "Indemnitor," "Indemnified Party," and "Indemnified Associates").

[36]. *The example in this clause box is not complete!* It replaces the first sentence of the clause box in Subchapter 1 above. If you use this example, be sure to include the rest of the language from the Subchapter 1 clause box, including the definition of "Indemnified Associates."

[37]. *The example in this clause box is not complete!* It replaces the first sentence of the clause box in Subchapter 1 above. If you use this example, be sure to include the rest of the language from the Subchapter 1 clause box, including the definition of "Indemnified Associates."

Subchapter 2 above discusses terms similar to this clause box. It suggests an indemnity based on the vendor/indemnitor's status as the operator of computers that leaked data. That's status-based, though the text of Subchapter 2 doesn't use that term, except in footnote 31.

[38]. The line between direct and consequential damages depends on the context, and it's not always clear. Direct damages *can* run high, but usually consequential damages run higher.

[39]. See Chapters II.J ("Indemnity") and II.Q ("Liquidated Damages"). Customers should avoid provisions that exclude the vendor's obligation to *indemnify* the customer but not its obligation to *defend* the customer. As Chapter II.J explains, both are crucial. That's why the first example in the clause box above references the entire indemnity section, which usually includes both.

[40]. See Chapter III.G ("Government Restricted Rights").

[41]. For more on trademarks, see Appendix 1 ("Intellectual Property").

[42]. It's hard to define "extensive." If in doubt, get a license—or talk to a trademark attorney.

[43]. During recent decades, a creature called the "business process patent" has blurred the line between invention and idea. Business process patents increase the vendor's risk from doing without a feedback license, but not much.

[44]. California's notoriously strict pro-competition laws actually play a role in the success of Silicon Valley. California forbids non-competes altogether, except in the sale of a business or dissolution of a partnership, and strictly limits employee non-solicits. (California Business & Professions Code Division 7, Part 2.) As a result, staff can move freely between competing companies in IT and other industries, facilitating sharing of ideas. That gives California a fertile environment for innovation.

[45]. See Chapter II.G ("Nondisclosure/Confidentiality"). Trade secrets law will lend a hand, too. See Appendix 1 ("Intellectual Property").

[46]. This book's first edition did offer a sample in a clause box (along with warnings about pro-competition laws). Since then, I've changed my mind about the value of a general example for the restriction on direct competition.

[47]. Some businesses ask one of the parties' lawyers to serve as escrow agent. That's a bad idea because it creates a conflict of interest. A lawyer should be loyal to his or her client, while an escrow agent should be neutral.

[48]. If the contract has an employee non-solicit (no poaching) clause, it might interfere with the customer's plan to hire the employees. So you might need to draft an exception. See Subchapter II.O.2 ("Employee Non-Solicit").

[49]. See Chapter I.A ("Standard End User Software License") and Subchapter I.C.1 ("Copyright License Rights").

[50]. For additional language addressing licenses and bankruptcy, see Chapter III.M, "Bankruptcy Rights."

[51]. See Chapter II.G ("Nondisclosure/Confidentiality").

[52]. See Chapter II.G ("Nondisclosure/Confidentiality").

[53]. Standard & Poor's and Moody's appear in many of these clauses, too.

[54] In fact, most TV court shows are really arbitrations. The TV “judge” is an arbitrator. He or she has the power to resolve the dispute because the parties have signed an arbitration agreement.

[55] JAMS originally stood for Judicial Arbitration and Mediation Service, but now the company’s name is just JAMS.

[56] For more on execution of form contracts, see Appendix 3 (“Clickwraps, Browse-wraps, and Other Contracts Executed without Ink”).

[57] The third and fourth examples in the clause box are automatic renewal or “evergreen” clauses. Some states won’t enforce these clauses against consumers in certain circumstances. In those cases, the contract might be considered month-to-month after the first term; if the vendor wants longer commitments, it needs to sign a new contract.

[58] See Chapter III.C (“Time Is of the Essence”).

[59] See Chapter II.G (“Nondisclosure/Confidentiality”), Subchapter II.G.3 (“Injunction, Termination, and Retention of Rights”).

[60] A perpetual license should survive the term of the contract, but it ends if someone terminates. An irrevocable license should survive any termination, but it ends when the term ends. And a license that is both perpetual and irrevocable should survive the term and any termination. See Subchapter I.C.2 (“Scope Terms”).

[61] See Subchapter I.B.1 (“Distribution”).

PART III

Boilerplate Clauses

The boilerplate clauses include a set of terms usually placed at the end of a contract, as well as introductory material like recitals and definitions. These clauses usually trigger less debate than the general clauses or prime clauses. But that doesn’t necessarily make them less important. You can’t tell what issue will crop up in a business relationship, so you never know when one of these boilerplate clauses will become vital.

A. Introduction and Recitals

A contract’s introduction generally identifies the parties and the contract itself. The recitals explain why the parties are doing business and sometimes give a little of their history. Neither introduction nor recitals is absolutely necessary. The contract’s first line could read: “BluntCo, LLC will provide the following services to Laconic Industries, Inc. . . .” But introductory language makes contracts easier to understand.

Introduction and Recitals

This Software License and Customization Agreement (this “Agreement”) is entered into as of ___________, 20__ (the “Effective Date”) by and between ____________, a ___________ (“Customer”), and ______________, a _________ (“Vendor”).

RECITALS

Vendor provides a software application known as ___________ (the “Base Application”), and the parties have agreed that Vendor will modify the Base Application to fit certain needs of Customer and license the modified software to Customer. The parties have also agreed that Vendor will provide maintenance and support services related to such modified software. Therefore, in consideration of the mutual covenants, terms, and conditions set forth below, including those outlined on Attachments __ through __ (which are incorporated into this Agreement by this reference), the adequacy of which consideration is hereby accepted and acknowledged, the parties agree as follows.

TERMS AND CONDITIONS

[Insert everything else here.]

The introduction and recitals should include no operative clauses: no promises, no rights, and no obligations. They might define a few terms, but otherwise they’re just introductory, with one exception. In the clause box above, the last sentence of the recitals does include some almost operative language. It states that the contract has adequate “consideration.” Consideration means something is exchanged: the document doesn’t record a one-way transaction, like a gift. A court won’t enforce an agreement without consideration; it's not a contract. A recital's claim of consideration doesn't guarantee a court will agree, but it can help.[1]

Many recitals start with "whereas" and end with "now, therefore." For instance: "WHEREAS, Vendor provides copy-editing software that makes modern writing read like a Victorian treaty; and WHEREAS Customer wishes to add some flair to its otherwise dull online content; NOW, THEREFORE, in consideration for the promises set forth below, the parties hereto agree as follows . . . ." These "whereas" and "therefore" terms actually do serve a purpose. They identify the recitals, so the reader knows they're introductory as opposed to operative clauses. But there's an easier way. Just slap the word "Recitals" above the recitals, and the words "Terms and Conditions," or just "Terms," above the rest of the contract, as in the clause box above.

B. Definitions

If a contract uses a defined term in a single section, it’s fine to define the term in that section. But if several sections use a defined term, readers might have trouble finding it if it’s buried in Section 9. So it’s often a good idea to collect all the defined terms used in more than one section and put them in a single definitions clause. That clause is usually Section 1 and lists the defined terms in alphabetical order.

C. Time Is of the Essence

If a contract says “time is of the essence,” even the slightest delay by either party is a material breach, at least in theory. In reality, almost no one intends such a harsh rule, so the clause’s meaning requires guesswork. Which of the parties’ deadlines absolutely do need to be met, and which don’t? Because the clause is so vague, you shouldn’t apply it to the whole contract. Instead, use time clauses for individual time-sensitive obligations.

Time Is of the Essence

Any failure of Vendor to deliver the Mission Critical Module by its due date constitutes a material breach of this Agreement.

• • • •

Time is of the essence with regard to Vendor’s First Priority Deadlines (as defined in Attachment __, *Project Schedule*).

The two examples in the clause box above mean the same thing: failure to meet the deadline constitutes breach. The first example is clearer. If you use the second example, add language to your termination clause, confirming that “time is of the essence” means a breach justifies termination, whether or not it’s material. See Subchapter II.V.2 (“Termination for Cause”).

D. Independent Contractors

An independent contractor clause confirms that the parties aren’t partners; they haven’t formed a legal partnership. They also aren’t principal and agent or employer and employee. The point is to avoid the tax implications and other legal consequences that can flow from “dependent” relationships, and to make sure neither party can make legal commitments on the other’s behalf.

In professional services deals, the independent contractor clause should also confirm that the vendor’s employees won’t be considered employees of the customer and that the vendor remains responsible for their benefits and pay.

Independent Contractors

The parties are independent contractors and will so represent themselves in all regards. Neither party is the agent of the other, and neither may make commitments on the other’s behalf. The parties agree that no Vendor employee or contractor will be an employee of Customer. Vendor shall be responsible for all employment rights and benefits of Vendor employees, including without limitation: (a) federal, state, and local income and employment taxes and social security contributions; (b) workers’ compensation, health benefits, vacation pay, holiday pay, profit sharing, retirement, pension, disability benefits, and other health and welfare benefits, plans, or programs; and (c) insurance.

E. Choice of Law and Courts

A choice of law clause picks the state whose laws will govern the contract. The clause usually also picks the courts with jurisdiction over disputes. The latter choice may become important because fighting a case downtown is easier than fighting a thousand miles away. In most cases, each party tries for the courts closest to home. Ideally, you'll pick the same state's law, because that's easiest to manage. But you can pick one state's laws and another's courts.[2]

If you don't make a choice in the contract, the courts may choose for you—usually selecting the law and court with the closest connection to the deal. If the parties hail from different states, it's often hard to say which one the courts will choose. So a choice of law clause reduces uncertainty.

Choice of Law and Jurisdiction

This Agreement will be governed solely by the internal laws of the State of ____________, without reference to: (a) any conflicts of law principle that would apply the substantive laws of another jurisdiction to the parties' rights or duties; (b) the 1980 United Nations Convention on Contracts for the International Sale of Goods; or (c) other international laws. The parties consent to the personal and exclusive jurisdiction of the federal and state courts of __________ [city or county], _________ [state].

Some choice clauses disqualify two particular laws: the 1980 United Nations Convention on Contracts for the International Sale of Goods (the UN Convention) and the Uniform Computer Information Transactions Act (UCITA). The UN Convention governs certain contracts between parties in different nations. Clauses like the example above exclude it because the parties want Wyoming law or California law or whatever, and they don't want to think about whether the UN Convention applies. UCITA, on the other hand, is a purely American affair. It's a proposed law that had a lot of people up in arms a few years back. Customers in particular felt UCITA unfairly limited their rights and remedies. Only Maryland and Virginia have adopted UCITA, but many customers and vendors exclude it, just in case their state ever does adopt it—though further spread is unlikely—or in case their deal somehow comes under a UCITA state's jurisdiction. The clause box above doesn't address UCITA, but here's sample language: "This Agreement will not be governed by the Uniform Computer Information Transactions Act as adopted in any jurisdiction."

A final note. When each party wants its own state, negotiators often compromise by naming a "neutral forum": a state with no connection to the deal. I think that's a bad idea (at least for choice of jurisdiction/courts) because if the deal has no connection to the neutral state, its courts may reject the case. Courts are busy, and no state runs an almost free arbitration system. So by choosing a neutral forum, you may lose your chance to choose at all. In general, I don't recommend "forum shopping"; I recommend naming a state connected to the deal, like one party's location or the site of services—and using that same state's laws *and* courts.[3]

F. Notices

A notices provision provides an address for each party to receive notices related to the contract. With a notices clause, the parties have a legally effective address for a termination or other actions requiring notice. If the party receiving notice moves without telling the other party, and a notice doesn't get through, the termination or whatever is still effective. The failure to communicate is the recipient's own fault.

Notices

Notices pursuant to this Agreement shall be sent to the addresses below, or to such others as either party may provide in writing. Such notices will be deemed received at such addresses upon the earlier of (a) actual receipt or (b) delivery in person, by fax with written confirmation of receipt, or by certified mail return receipt requested.

For Vendor: ____________________________________.
For Customer: __________________________________.

Many notice provisions, like the clause box above, provide a system for notices. Notices will be considered received, whether they actually were or not, if sent by certified mail or whatever other mechanism the notice clause chooses.

G. Government Restricted Rights

Federal regulations give the U.S. government some surprising rights to software. If a federal agency has software developed, or receives it under certain other circumstances, it gets “unlimited rights” or, in some cases for defense agencies, “government purpose rights.” When the agency gets unlimited rights, it can make as many copies as it wants, share the software with other federal agencies and with the public, etc. Government purpose rights restrict use of the software to a government purpose, but the agency can still make unlimited copies and distribute to other agencies. Obviously, either type of government rights could cost the vendor a lot of sales. So vendors of commercial software, or of any software not created for the feds, often include a clause addressing federal rights. The clause clarifies that the software is *not* subject to unlimited or government purpose rights. Rather, it’s commercial software, licensed with “restricted rights.”

Government Restricted Rights [4]

The Software is provided with Restricted Rights. Use, duplication, or disclosure for or by the government of the United States, including without limitation any of its agencies or instrumentalities, is subject to restrictions set forth, as applicable: (i) in subparagraphs (a) through (d) of the *Commercial Computer Software-Restricted Rights* clause at FAR 52.227-19; or (ii) in similar clauses in other federal regulations, including the NASA FAR supplement. The contractor or manufacturer is ______________ [Vendor]. Customer shall not remove or deface any restricted rights notice or other legal notice appearing in the Software or on any packaging or other media associated with the Software. Customer shall require that its customers, distributors, and other recipients of the Software agree to and acknowledge the provisions of this Section __, in writing.

Government restricted rights clauses sometimes appear in standard contract forms—forms used for both government and nongovernment deals. So even if you’re not working on a federal contract, you may run into terms like the clause box above. If you’re the customer and you don’t plan to share the software with the feds, the clause shouldn’t restrict you in any important way. (The clause box does require that you leave various notices in the software and on its packaging.) And if you’re the vendor and you see *no* chance of distribution to the federal government, you don't need the clause. Some vendors, however, include it just in case.

If you're a vendor and you *are* licensing to the federal government, or to distributors who might do so, add a government restricted rights clause. But first consider consulting an attorney who specializes in government contracting and also knows IT. (A government contracts attorney can also help you place proper government restricted rights notices in your software and on its packaging.)

H. Technology Export

U.S. law restricts the export of certain technologies, and the same restrictions apply to what you might call “virtual export”: technology accessed through the cloud. To avoid liability to the government, tech vendors often require that customers promise to obey those laws.

Technology Export

Customer shall not: (a) permit any third party to access or use the System in violation of any U.S. law or regulation; or (b) export the Software or otherwise remove it from the United States except in compliance with all applicable U.S. laws and regulations. Without limiting the generality of the foregoing, Customer shall not permit any third party to access or use the System in, or export the Software to, a country subject to a United States embargo (as of the Effective Date, Cuba, Iran, North Korea, Sudan, and Syria).

The clause box above offers simple export terms. The customer promises to obey all U.S. export laws and regulations—including laws forbidding sending technology to embargoed countries: nations the U.S. government considers dangerous, like Iran, North Korea, and (bizarrely) Cuba. If the customer has sublicensees or its own customers receiving tech access, consider a more restrictive clause: “Customer shall require that all its sublicensees, customers, and other recipients of the Software or of access to the System sign a written agreement promising to comply with all applicable U.S. laws and regulations related to export.” Some vendors go even further and require an indemnification from the customer. The customer indemnifies the vendor against any government or other lawsuit alleging a customer violation of export laws.[5]

Of course, foreign laws also address technology export. If the vendor is concerned about a particular country’s laws, it should have an attorney licensed there review the clause. An international “catchall” clause might also provide some protection: “Customer shall not export or transmit the Software across any national boundary, or provide international access to the System, except in compliance with all applicable laws and regulations, including without limitation the export laws and regulations of the originating country.”

Vendors of data security systems and other sensitive technology shouldn’t rely on contract clauses for protection. They should become familiar with export laws and seek technical advice on steps to prevent illegal export.

I. Assignment

An assignment clause governs whether a party can transfer the contract—with all its rights and duties—to someone else. The clause can forbid all assignments, or let only one party assign the contract, or permit any assignment.[6]

The clause box below permits assignment only in case of a merger or acquisition, and that's common.

Assignment

Neither party may assign this Agreement or any of its rights or obligations hereunder without the other's express written consent, except that either party may assign this Agreement to the surviving party in a merger of that party into another entity or in an acquisition of all or substantially all that party's assets. An assignment authorized pursuant to the preceding sentence shall not become effective unless and until the assignee agrees in writing to be bound by all the assigning party's rights and obligations set forth in this Agreement. Except to the extent forbidden in this Section __, this Agreement will be binding upon and inure to the benefit of the parties' respective successors and assigns.

Some assignment clauses rule out assignments to the other party's competitor, even in case of a merger or acquisition. "Notwithstanding the provisions of Section __ above, neither party may assign this agreement to a Competitor of the other (as defined in Section __)." The question is, what's a competitor? Many contract drafters leave it undefined. That's a mistake, because it leaves the clause impossibly unclear. Is a database provider the competitor of a SaaS vendor? What if the SaaS system includes a database, so its customers don't need to license one? What if the database provider someday acquires a company with software arguably similar to the SaaS application? Or is a toy manufacturer the competitor of a construction company? What if they both develop project management software and decide to sell it? What if they're each part of a conglomerate, now or in the future, and both conglomerates include a sub selling beauty products?

Subchapter II.O.1 ("Non-Compete") provides guidance on defining "competitor," starting on page 164.[7]

J. Force Majeure

A force majeure clause releases the parties from their contractual obligations if “acts of God” or other forces out of their control interfere. A court might reach the same conclusion, but the clause increases certainty. Usually force majeure clauses benefit both parties, but sometimes they benefit only one—usually the party that drafted the vendor.

> ***Force Majeure***
>
> No delay, failure, or default, other than a failure to pay fees when due, will constitute a breach of this Agreement to the extent caused by acts of war, terrorism, hurricanes, earthquakes, other acts of God or of nature, strikes or other labor disputes, riots or other acts of civil disorder, embargoes, or other causes beyond the performing party’s reasonable control.

Most force majeure clauses don’t excuse the obligation to pay, and that’s the case with the clause box above.

Many courts won’t enforce a force majeure clause unless it specifically lists the events that excuse performance. In other words, the clause might not work if it merely excludes performance blocked by “events beyond the performing party’s reasonable control.” So think through the events that should excuse performance and list them. The clause box above has a typical list: war, acts of nature, strikes, etc.

If the clause excuses the other party’s performance, whether it excuses yours or not, you should ask if it’s too broad. Should the other party perform even if a hurricane strikes? Customers should also ask which force majeure events should excuse SLA or disaster recovery obligations, if any. Subchapter II.R (“Disaster Recovery”) addresses this last issue in greater detail, starting on page 175.

K. Severability

A severability clause limits the impact of unenforceable terms. It confirms that the parties intend the contract to operate as written to the maximum extent possible. They don’t want the deal to fall apart if one or more clauses can’t be enforced.

Terms like the clause box below also try to preserve the contract, as written, by having each party waive any legal rights that would prevent full enforcement. The waiver itself might not be enforceable, but it’s worth a shot.

Severability

To the extent permitted by applicable law, the parties hereby waive any provision of law that would render any clause of this Agreement invalid or otherwise unenforceable in any respect. In the event that a provision of this Agreement is held to be invalid or otherwise unenforceable, such provision will be interpreted to fulfill its intended purpose to the maximum extent permitted by applicable law, and the remaining provisions of this Agreement will continue in full force and effect.

L. No Waiver

Companies sometimes fail to notice when someone's breached one of their contracts, particularly big companies. So they often include a no waiver clause. These terms provide that a party's failure to sue quickly won't waive its right to do so—and failure to respond to one breach doesn't waive the right to sue for another.

> ***No Waiver***
>
> Neither party will be deemed to have waived any of its rights under this Agreement by lapse of time or by any statement or representation other than by an authorized representative in an explicit written waiver. No waiver of a breach of this Agreement will constitute a waiver of any other breach of this Agreement.

No waiver clauses also tend to require that any waiver be in writing, as in the clause box above. The point is to make sure a careless oral statement by an employee doesn't waive an important contract right. But prohibitions against oral waiver aren't always enforceable, so it's a good idea to add a second layer of protection. The no waiver clause should provide that only an authorized representative can waive a contract right, again as in the clause box. "Authorized representative" usually isn't defined, but at least it pretty reliably rules out very junior staff. (For more on what that term means, see Chapter III.S, "Amendment.")

M. Bankruptcy Rights

If a company goes bankrupt, it can escape most of its contractual obligations. Promises to provide services may be out the window, along with warranty obligations, indemnity obligations, and almost everything else. Software licensing customers, however, can choose to preserve their license rights. That's because Section 365(n) of the U.S. Bankruptcy Code[8] gives them that right. A bankruptcy rights clause confirms that the license is subject to Section 365(n). Confirmation isn't always necessary—Section 365(n) should protect the customer either way—but it helps to be sure.

If the contract has a separate license section for source code or other technology in escrow, or for anything else, the bankruptcy rights clause should refer to that section, too, as in the clause box below.[9]

Bankruptcy Rights

The rights and licenses granted to Customer in Sections __ (*Software License*) and __ (*Escrow Materials License*) of this Agreement (the "License Provisions") are licenses to "intellectual property" rights, as defined in Section 365(n) of the United States Bankruptcy Code (11 U.S.C. Sections 101, *et seq.*). If Vendor is subject to any proceeding under the United States Bankruptcy Code, and Vendor as debtor in possession or its trustee in bankruptcy rejects this Agreement, Customer may, pursuant to 11 U.S.C. Section 365(n)(1) and (2), retain any and all rights granted to it under the License Provisions to the maximum extent permitted by law. This Section __ will not be construed to limit or restrict any right or remedy not set forth in this Section __, including without limitation the right to retain any license or authority this Agreement grants pursuant to any provision other than the License Provisions.

N. Conflicts among Attachments

Many contracts include multiple documents, with some attached to the contract's main body. What happens if the terms of those attachments contradict each other or contradict the main body? What if the main body says the vendor owns the deliverables, and Attachment B says the customer owns them?

You'd think a little care would eliminate the risk. But you can't predict every possible interpretation of the contract. Plus, some attachments could be forms attached to the contract without a lot of thought or a chance to edit. Others, like statements of work or orders, may be drafted after contract execution—possibly by someone who didn't read the contract. So a conflicts clause can prevent a lot of trouble.

Most contract drafters prefer that project-specific documents take precedence over more general ones. In other words, a statement of work or order governs if it conflicts with the contract's main body (the "Master Services Agreement" or whatever). That's how the first example works in the clause box below. Among other advantages, that lets the parties change their minds about negotiated terms just by executing a new statement of work, without formally amending the contract. Still, I think it's a mistake.

Providing that project-specific terms overrule general ones is like saying laws overrule the constitution. The whole point of a master agreement (or constitution) is to create an agreed-on structure for future work. If a statement of work or order takes precedence over the contract's main body, that structure won't be consistent for all projects. Plus, companies often dedicate more legal expertise—lawyers, contract managers—to the main body than to attachments, particularly statements of work or orders signed long after the main body. Sometimes the main body gets more executive attention, too. Why spend money and legal hours to get the terms right and then leave them open to unintended revision in a future statement of work by a junior employee? Yes, changing the contract is more burdensome if you can't do it through a statement of work or order. But that's the point: it should be burdensome. Attempts to amend negotiated terms should raise a red flag, so that they get appropriate attention. That’s why the main body governs in the second two examples in the clause box below.

Conflicts among Attachments

In the event of conflict with the main body of this Agreement, a Statement of Work or Addendum will govern, but only with respect to the subject matter of such Statement of Work or Addendum.

• • • •

In the event of conflict with an attachment to this Agreement, this main body of this Agreement will govern. In addition, no Statement of Work or other attachment incorporated into this Agreement after execution of this main body of this Agreement will be construed to amend this main body unless it specifically states its intent to do so and cites the section or sections amended.

• • • •

In the event of any conflict among the attachments to this Agreement and this main body, the following order of precedence will govern, with lower numbers governing over higher ones: (1) this main body of this Agreement; (2) Attachment __; (3) Attachment __; and (4) any Statement of Work, with more recent Statements of Work taking precedence over later ones. No Statement of Work or other attachment incorporated into this Agreement after execution of this main body will be construed to amend this main body or any earlier attachment (subject to the preceding sentence’s order of precedence) unless it specifically states its intent to do so and cites the section or sections amended.

In the second two examples above, the main body takes precedence over attachments. And because courts generally let a later-signed document amend an earlier one, both examples provide that future attachments won’t serve as amendments unless they specifically state their intent to do so. That helps avoid accidental amendment, though it doesn’t entirely eliminate the risk. (A court still might hold that a later-signed document amends an earlier one.)[10]

Most clauses just address conflicts between the contract’s main body and the attachments, as in the first two examples above. That doesn’t address conflicts among multiple attachments, so some clauses go further and create a multi-document order of precedence, like the third example in the clause box above.

Don’t let the conflicts clause become an excuse for sloppy drafting. You should still review every attachment and do your best to eliminate conflicting terms.

O. Execution in Counterparts

Often, the parties can’t get together in person to sign a printed contract. So they sign separate copies of the signature page, or of the whole document, and exchange them by fax, mail, e-mail attachment, etc. The counterparts clause says that these separate copies form a single contract.

Execution in Counterparts

This Agreement may be executed in one or more counterparts. Each counterpart will be an original, but all such counterparts will constitute a single instrument.

If you execute in counterparts, make sure each party signs exactly the same document—and make sure you can prove it.

P. Construction

Courts generally construe unclear terms against the party that wrote them. In other words, if one party wrote the contract (e.g., a clickwrap or other nonnegotiated form), a court will interpret vague terms in a way that favors the *other* party. Construction clauses seek to make sure that rule doesn't apply to a contract the parties actually did negotiate, even if one party still did most of the drafting or determined most of the terms. The parties agree that there will be no favoritism in construction.

Construction

The parties agree that the terms of this Agreement result from negotiations between them. This Agreement will not be construed in favor of or against either party by reason of authorship.

Q. Internet-Related Boilerplate

If you're an online vendor, federal and state statutes may require that you include certain disclosures in your contract or on your website. For example, the federal Communications Decency Act requires that you include something like the following in your contract: "Pursuant to 47 U.S.C. Section 230(d), Vendor hereby notifies Customer that parental control protections (such as computer hardware, software, or filtering services) are commercially available that may assist in limiting access to material that is harmful to minors. Information regarding providers of such protections may be found on the Internet by searching 'parental control protection' or similar terms." Other required disclosures may address topics like data privacy[11] and consumers' right to submit complaints to the vendor or to the state government.

These disclosure requirements vary from industry to industry and state to state, so if you're doing business online, you should do some research or consult with an appropriate attorney.

R. Entire Agreement

The entire agreement or “integration” clause confirms that all terms the parties meant to include are *in* the agreement. It voids any letters, discussions, side agreements, or anything of that kind existing before the contract was signed or at the time of signature.

Entire Agreement

This Agreement sets forth the entire agreement of the parties and supersedes all prior or contemporaneous writings, negotiations, and discussions with respect to its subject matter. Neither party has relied upon any such prior or contemporaneous communications.

It’s important to clarify that neither party relied on any earlier or same-day communication, to help defeat any claim that one party tricked the other into signing (aka “fraud in the inducement”).

S. Amendment

An amendment clause lays down rules for amending the contract. This chapter addresses both typical and “unilateral” amendments. A typical amendment is essentially a new contract changing the original terms, executed by both parties. In a unilateral amendment, the vendor imposes new terms, without active participation from the customer.

Amendment
This Agreement may not be modified except in a written agreement signed by authorized representatives of both parties.

The clause box above has typical terms. Its requirement for a written, signed amendment reduces the risk of accidental amendment through dumb oral promises. But prohibitions against oral amendment aren’t always enforceable, so the clause box adds further protection by providing that only the parties’ authorized representatives can amend the contract. That still isn’t foolproof against oral amendment, but it helps.

Few contracts define “authorized representative.” If a court ever rules on who’s authorized, it will look at the amending party’s conduct: at who *looks* authorized, based on title, role, or other factors. But some clauses eliminate that uncertainty through a definition: “A party’s ‘Authorized Representative’ is any employee of that party with a title of Director or Corporate Counsel, or with a title more senior than either of those.” (This assumes you have a traditional title structure so it’s easy to determine seniority.) You can also choose a particular officer or officers, but then you have to address what happens if those positions are vacant. The same problem arises if you designate an authorized representative by name.

Designation of authorized representatives doesn’t guarantee that no one else will be able to amend the contract. If a company lets an employee act like an authorized representative, a court may treat him or her that way. But designation in a contract will help.

• • •

Vendors who use standard, nonnegotiable contracts often want the right to amend those contracts without getting every customer's consent. That calls for a unilateral amendment clause: terms giving the vendor the right to change the agreement even if the customer doesn't sign anything or click "I agree" on a new version.

Unilateral amendment plays a role in many cloud services contracts and other online agreements. Obviously, it's not good for customers. They should avoid it in negotiated contracts. If the vendor can change the contract later, why bother with negotiations? But for cloud services with standard, nonnegotiable terms, customers usually can't avoid unilateral amendment.

If you're a vendor with a standard contract, the good news is that unilateral amendment can work; you don't have to sit down and negotiate amendments with every customer. But that doesn't mean you can force new terms on an unwilling customer. An amendment requires *notice and consent*: the customer has to know about the new terms and accept them. (The same requirements apply to execution of a contract in the first place.)

Many cloud services contracts include something like the following: "Vendor may amend this Agreement by posting a new version at the Website, and such amendment will become effective upon posting thereof. Customer shall periodically review the Website for amendments, and Customer's continued use of the System after an amendment will indicate consent thereto." It's amazing how often vendors use terms like that because they very well may not work. Customers don't actually check vendor websites for contract revisions, and many courts consider such requirements unrealistic—particularly in agreements with consumer customers. So the clause likely fails the notice and consent test.

Unfortunately, it's not clear what kind of notice and consent an amendment clause needs, because the law isn't well settled. The consequences of a mistake aren't clear either. I suspect most legal scholars would say a bad amendment clause fails to create an effective and speedy system for contract amendment, leaving uncertainty about which version of the contract applies. But at least one court has suggested a worse outcome for the vendor: a bad clause invalidates the whole contract.[12] If that's right, a bad clause would cost the vendor its limitation of liability, disclaimers of warranties, payment terms, etc.

The clause box below offers better unilateral amendment terms.

Unilateral Amendment

Vendor may amend this Agreement from time to time by posting an amended version at its Website and sending Customer written notice thereof. Such amendment will be deemed accepted and become effective 30 days after such notice (the "Proposed Amendment Date") unless Customer first gives Vendor written notice of rejection of the amendment. In the event of such rejection, this Agreement will continue under its original provisions, and the amendment will become effective at the start of Customer's next Term following the Proposed Amendment Date (unless Customer first terminates this Agreement pursuant to Section __, *Term & Termination*). Customer's continued use of the Service following the effective date of an amendment will confirm Customer's consent thereto. This Agreement may not be amended in any other way except through a written agreement by authorized representatives of each party. Notwithstanding the foregoing provisions of this Section __, Vendor may revise the Privacy Policy and Acceptable Use Policy at any time by posting a new version of either at the Website, and such new version will become effective on the date it is posted.

• • • •

Vendor may amend this Agreement from time to time by posting an amended version at the Website and sending Customer written notice thereof. Such amendment will become effective 30 days after such notice (unless Customer first terminates this Agreement pursuant to Section __, *Term & Termination*). Customer's continued use of the Service after such 30-day notice period will confirm Customer's consent to such amendment. This Agreement may not be amended in any other way except through a written agreement by Authorized Representatives of each party.

The examples in the clause box above assume the contract is posted online. If not, replace the "posting" provisions with something like: "Vendor may revise this Agreement from time to time by sending a new version to Customer." (But you don't see a lot of unilateral amendment clauses in offline contracts.)

Each example in the clause box requires concrete *notice* of contract amendments, like an e-mail or letter.[13] The first example also has concrete *consent* terms. It gives the customer 30 days to review the amendment. If the customer accepts the change, it does nothing, and the amendment becomes effective after the 30 days. If the customer rejects the changes, it notifies the vendor and the contract continues under its original provisions. But at the start of the customer's next term, the amendment goes into effect whether the customer likes it or not. That assumes the customer can terminate the agreement at the start of the next term, so it can reject the amendment by walking away.[14] In other words, the first example provides for both notice and consent.

Vendors could instead give customers the right to terminate the agreement if they don't like the amendment. But most vendors prefer to minimize customer termination rights. The first example above gives the customer a meaningful choice—which should make the clause enforceable —but doesn't allow early termination. Plus, in most online cloud services, the vast majority of customers will do nothing—they won't respond to the amendment notice—so the amendment usually *will* become effective in 30 days.

The second example in the clause box does *not* specifically grant the customer a chance to reject the amendment. But if the customer can get out of the relationship before the amendment becomes effective, the clause still provides a meaningful choice. The term could be month-to-month, for instance, or another section could let the customer terminate for convenience (without any early termination fees or other adverse consequences).[15] In either case, the clause should be enforceable. Just make sure the amendment doesn't kick in until at least a few days after the customer's next chance to terminate. If the customer has to give 30 days' notice to terminate for convenience, or the contract is month-to-month, the clause should require a 40- or 45-day notice period before amended terms go into effect.

The first example in the clause box above ends with terms that might be considered an exception to the unilateral amendment rule. The vendor can change its privacy policy and acceptable use policy (AUP) without going through the amendment procedure: without giving the customer meaningful notice and consent. I think unilateral amendment works for standard policy documents that don't have a major impact on the customer's rights and obligations, though the law isn't clear. Privacy and acceptable use policies usually work like internal policies, which companies generally change at will, rather than like traditional contract clauses. At least, they work that way if the vendor keeps fundamental contract provisions out of these policy documents and sticks to their natural subject matter—privacy procedures, acceptable use, responses to copyright claims, etc.—and if the amendment isn't too surprising, like switching from a typical privacy policy to a total disclaimer of privacy-related obligations.[16] I think tech buyers and sellers accept the idea of amendment without advance consent for online policy documents, and the courts will follow their lead. Nothing is certain, though, so vendors should consider reducing their risk through a notice requirement: "Notwithstanding the foregoing provisions of this Section __, Vendor may amend the Privacy Policy and Acceptable Use Policy at any time by posting a new version of either at the Website *and sending Customer written notice thereof*. Such new version will become effective 10 business days after such notice.” (Emphasis added.)[17]

As noted above, unilateral amendment makes the most sense with ongoing service relationships, like cloud services deals where the customer has continuous access to the vendor’s system. Unilateral amendment makes little sense for relationships that involve a new contract every time the parties interact. If the customer has to click “I agree” and accept the contract every time it uses the system or accesses the website, the vendor doesn’t need unilateral amendment terms. If that’s the plan, the vendor should make sure its online contract leaves no doubt. “THIS AGREEMENT GOVERNS A SINGLE INSTANCE OF ACCESS TO THE SERVICE [*or* A SINGLE DOWNLOAD OF THE SOFTWARE]. TO ACCESS THE SERVICE AGAIN, CUSTOMER MUST AGREE SEPARATELY TO VENDOR’S THEN CURRENT TERMS OF SERVICE, WHICH MAY NOT MATCH THIS AGREEMENT. CUSTOMER IS ON NOTICE THAT THIS AGREEMENT DOES NOT GOVERN FUTURE ACCESS TO THE SERVICE AND THAT CUSTOMER MUST REVIEW AND ACCEPT VENDOR’S THEN STANDARD TERMS OF SERVICE FOR FUTURE ACCESS.” Terms like that play a particularly important role in “browsewrap” contracts.[18]

---

[1]. If you’re concerned that your deal lacks consideration—lacks a two-way exchange—consult an attorney who specializes on contracts, ideally IT contracts.

[2]. If you have an arbitration clause, the choice of law tells the arbitrator what law to apply. And the jurisdiction part of the clause determines what court will enforce the arbitrator’s decision. See Chapter II.U (“Alternate Dispute Resolution”).

[3]. Contract drafters who do forum-shop often favor Delaware law and courts and avoid California. If I recommended forum shopping, I’d actually recommend the reverse: avoid Delaware and choose California. Delaware has special advantages related to *corporate* law—shareholders’ rights—but that has nothing to do with tech agreements, which involve IP and commercial law. Delaware isn’t big and doesn’t have a large IT industry, so you wouldn’t expect particularly tech-savvy laws or courts there. California, on the other hand, is the nation’s largest state and hosts the tech industry’s international hub: the Silicon Valley. So you would expect tech-savvy laws and courts. California gets a bad rap for pro-consumer and pro-employee laws. But that’s exaggerated and anyway doesn’t keep companies like Apple and Google from choosing California law, including for their consumer contracts. (Full disclosure: I live and practice in California.)

[4]. In the clause box, “FAR” refers to the Federal Acquisition Regulations.

[5]. See Chapter II.J (“Indemnity”).

[6]. This chapter discusses assignments of an entire contract, not of intellectual property. For IP assignments, see Chapter I.D (“Technology Ownership: Assignment and Work-for-Hire”).

[7]. Any restriction related to competitors could raise antitrust and unfair competition law concerns. If you do restrict assignment to competitors, consider consulting an attorney who knows the relevant laws, particularly state laws.

[8]. 11 U.S.C. § 365(n).

[9]. See Chapter II.S ("Technology Escrow"), Subchapter 2 ("License and Confidentiality").

[10]. See also Chapter III.S ("Amendment").

[11]. See also Appendix 4 ("Online Policy Documents").

[12]. *Harris v. Blockbuster*, 622 F.Supp.2d 396 (N.D. Tex. 2009). The *Harris* court implied that a clause saying the vendor can change terms at will, with no customer consent, renders the whole contract "illusory."

[13]. E-mail notice is far more effective than notice through a website. Letters are even better because they're not subject to spam-filters or anti-spam laws, but they're not always practical. RSS feeds are practical and effective, and so are Twitter messages (tweets), so vendors should consider combining either or both of those with e-mail notice. ("RSS" stands for "really simple syndication" or "rich site summary," depending who you ask.) In general, vendors should consider multiple means of notice—whatever's necessary to make sure customers get the message.

[14]. See Subchapter II.V.1 ("Term").

[15]. See Subchapter II.V.3 ("Termination for Convenience"). The right to terminate won't necessarily be called "termination for convenience."

[16]. See Appendix 4 ("Online Policy Documents").

[17]. Subchapters II.A.4 ("Specifications as a Moving Target") and II.B.3 ("SLA Modification") suggest another exception, arguably, to the unilateral amendment rule: terms allowing the vendor to change technical specifications and service level agreements with limited notice and consent. In each case, though, the chapter suggests terms meant either to reduce the impact on the customer's fundamental rights, so that the change works like the policy amendments discussed above, or to give the customer notice and a right to terminate.

[18]. See Appendix 3 ("Clickwraps, Browsewraps, and Other Contracts Executed without Ink").

Appendices

APPENDIX 1

Intellectual Property

This appendix provides a brief explanation of intellectual property (IP), a field of law important to many IT transactions, particularly software licenses.

Intellectual property law lets creators monopolize certain products of their intellect. Because of IP law, you can *own* an invention you’ve developed. You can also own software or a story you’ve written, a photograph you’ve taken, music you’ve written or recorded, a sculpture you’ve carved, a name or logo you’ve used in commerce, and various other types of intangible property. And your ownership covers more than just physical products. If you build a better mousetrap and get a patent, you can monopolize the invention so that no one else can build mousetraps like yours. If you write a story, you own more than the copies you print; you can keep anyone else from printing and selling your story, because of your copyright. In other words, if you have a patent or copyright, or one of the other forms of IP, you can monopolize the intangible products of your brain.

Companies can own intellectual property, too. They usually own IP their employees create within the scope of their duties. And, of course, they can buy IP.

Intellectual property plays a central role in the IT industry, but it’s important not to exaggerate its reach. Many IT professionals assume someone owns every innovation. That’s wrong.[1] In fact, you should view intellectual property as the exception, not the rule. In U.S. law, ideas and innovations are free as the birds. If I come up with a great idea and tell you about it, you’re almost always free to exploit it. You can take my idea and make millions, or become famous, leaving me behind. It’s only in rare instances that I can monopolize the product of my brain. IP law creates those rare instances—those exceptions to the rule.

IP consists principally of copyrights, patents, trade secrets, and trademarks.

1. **Copyright** applies to any original work of authorship fixed in a tangible medium of expression—like a story written in a notebook or software written on a disk. As Chapter I.C ("Software Licenses in General") explains, a copyright holder has the exclusive right to reproduce (copy), modify, distribute, publicly perform, and publicly display his or her work. Software licenses generally address those rights, because copyright applies to almost all software. Copyright comes into existence at the moment of authorship, so if you write original software, you automatically own the copyright. You can register a copyright with the U.S. Library of Congress, and registration helps enforce your rights. But registration isn't necessary for ownership. Duration of copyright varies, but generally copyright lasts 70 years from the death of the author, or 120 years from creation in the case of a corporation authoring software through its employees.[2]

   Copyright applies to expression—to written words, written software, etc.—not to the underlying ideas or inventions. So if you write a book about computer repair, copyright gives you the exclusive right to reproduce and distribute that book. But it doesn't give you exclusive rights to your book's computer repair *ideas*. An imitator could read your book and use your ideas to start a competing computer repair business. Your imitator could even write his own computer repair book, laying out ideas learned in your book—so long as he doesn't reproduce the actual words of your book. The same goes for software. An imitator can read your source code—the human-readable version of your software—and write her own software using your ideas and techniques. That's generally perfectly legal, so long as the imitator doesn't copy any of the actual code. If you want to protect a technique or process built into your software, copyright won't help you. For that, you need to keep the technique secret, which does the trick all by itself and might also give you trade secret protection. Or you need a patent.

2. A **patent** is a government grant of certain rights to a device or process—to an invention. To get a patent, you have to apply to the U.S. Patent and Trademark Office (the PTO). The PTO will only grant the patent if the invention meets certain criteria. It has to be "novel, useful, and nonobvious." If you do get a patent, you have exclusive rights to make the invention or have it made and to use, market, sell, offer for sale, and import it. Those monopoly rights generally continue for 20 years.[3]

   Some software includes patented processes—patented inventions. But end users and distributors generally don't need patent licenses for that software. A copyright license will grant all the necessary rights. That's why software contracts rarely address patent rights.

3. A **trade secret** is information that's (1) valuable because it's not widely known or easily discovered by people who could use it and (2) subject to reasonable efforts to maintain secrecy. The recipe for Coca-Cola is a famous trade secret. So is the algorithm for Google's search engine. Trade secret law comes from the states, rather than the federal government, but it's pretty consistent across the country, since most states have adopted a model law called the Uniform Trade Secrets Act.[4] A trade secret holder doesn't exactly have a monopoly over its use. A competitor can use a trade secret obtained legally—through independent discovery or conception, for instance. But the holder can prevent use of a trade secret discovered through misappropriation: industrial espionage, employees breaching NDAs, etc. Unlike copyrights and patents, trade secrets have no expiration date.

   IT vendors often protect software techniques by keeping source code secret, and that can give those techniques trade secret protection. Nothing the vendor distributes to the market can be a trade secret, of course, but many techniques aren't visible from the way the software performs. To grasp them, you'd have to read the software—and the vendor only distributes the object code version, which no one can read.

4. A **trademark** is a name, logo, word, or phrase that announces the source of goods or services—that tells the market who provides

them. “Trade dress” is a type of trademark too, covering a product’s design or shape, including the on-screen look and feel of some software applications. The name *Coca-Cola* is a trademark, and so is *Apple* (as applied to computers but not to fruit). The classic shape of a Coke bottle is trade dress, and so is the basic design of an iPhone. That little four-colored square flag you see on Microsoft products is a trademark too—a logo—and so is the phrase “Intel Inside.”

The owner can keep others from using a trademark if it’s distinctive and it’s used in commerce. Like copyrights, trademarks don’t need registration for legal protection, but registration helps enforce trademark rights. You can register a trademark with a state government, but more owners use federal registration, through the U.S. Patent and Trademark Office, since it provides broader protection.[5] Like trade secrets, trademarks don’t have expiration dates.

APPENDIX 2

Open Source Software Licenses

An open source license grants the customer access to the software's source code. It also permits modification of the software and redistribution of both the original and the new, modified version. At least, that's a thumbnail explanation of a complex concept. This appendix explains open source licenses. You'll understand this appendix better if you first read Chapter I.C ("Software Licenses in General").[6]

This appendix refers to the parties in an open source license as the "licensor" and the "recipient." That's because "vendor" and "customer" don't fit well, since often both parties qualify as software developers and distributors.

"Open source" is often confused with "free," but open source software isn't necessarily free. Licensors can charge for software and still be considered open source providers, so long as they don't charge any additional fee or royalty for source code or for the rights to modify and redistribute. But because each recipient can redistribute the software—at any price or no price—market forces usually keep licensors from charging a lot. If the price is high, recipients can just get the software from someone else: from another recipient. As a result, much open source software actually is free or very inexpensive. Most of the licensors don't care about making money directly from their software. They generate revenues from related professional services, like support and consulting, or from related technology that isn't open sourced. Or they're just creative individuals who write software for the fun of it (or the glory).

"Open source" is also often confused with "public domain," and that's also a mistake. If software is in the public domain, it's entirely free of copyright and other intellectual property restrictions. No one needs a license to copy it, and we’re all free to modify it, distribute new versions, etc. Open source software, on the other hand, is licensed by definition. So it’s governed by copyright law. The license just grants freedoms that aren’t common in commercial licenses—along with some unique restrictions, in many cases.

The key advantage of open source software is *evolution*. When the source code circulates widely and developers can fix bugs, add new modules, and otherwise revise, the software improves. If the various improvements circulate, recipients will choose the best ones and use and redistribute those. So software quality increases through natural selection. Compare that to typical commercial in-house development. A small group of developers tests and improves the software during a limited period, possibly with help from some beta customers. No one else’s input finds its way into the code.

But the open source model comes with some disadvantages for commercial software vendors. As noted above, each recipient can redistribute the software and compete with the original licensor, limiting license revenues. Also, source code often includes sensitive techniques, and revealing it shares those techniques with competitors. Finally, some open source licenses include additional restrictions that cause problems for commercial developers. See below for these “copyleft” restrictions. As a result, many software developers won’t invest a lot of resources in open source software.

Licensors rarely draft their own open source licenses. They license software under standard forms widely used in the open source community. Among other reasons, open source recipients prefer familiar terms, as opposed to licenses they’d have to scrutinize. So this appendix doesn’t offer proposed language for open source licenses or contracts. It does, however, refer you to some of the best-known open source license forms.

The rest of this appendix explains the two types of open source software licenses: copyleft and permissive. It also briefly describes the non-IP terms that generally appear in open source licenses.

Copyleft Licenses

"Copyleft" is a flavor of open source licensing that turns copy*right* protection around. Copyleft license terms require that any recipient of the software *also* use the open source model if it redistributes the software. In fact, copyleft requires that *all* future/downstream recipients use the open source model if they redistribute.

Imagine a recipient that's also a software developer. It creates a "derivative work" of copyleft software: it modifies the software to create a new program or adds the software to a larger program. If the developer distributes one of these derivative works, it has to use the open source model. It has to distribute the new program with source code and with the right to modify and redistribute. As a copyleft hater might put it, if copyleft software gets into another program, it "infects" that program, turning it into open source software. That's why some in the IT industry call copyleft licenses "viral."

Copyleft creates a nightmare for some software developers. An engineer working for the developer includes a small amount of copyleft-licensed software in a massive system, without telling anyone. Now the whole system is "infected": it's all open source software. Even worse, the developer's customers and distributors face the same problem. If they create a derivative work by combining the "infected" software with any of their own systems, they catch the copyleft "virus" too and have to treat their own system as open source software. The result could be the loss of millions of dollars invested in commercial software that wasn't supposed to be open sourced.

At least, that's the concern. No one's entirely sure how far copyleft restrictions can be enforced in court. But few software developers want to serve as the test case. So the developer community has spent a lot of money trying to find copyleft-licensed software in their systems and remove it.

The best-known copyleft license form is the GNU General Public License (the GPL), provided by the Free Software Foundation, a nonprofit dedicated to the open source model. You can find the most current edition of the GPL, version 3, online at http://www.gnu.org/licenses/gpl.html. The key copyleft licensing provisions appear in Sections 2, 4 through 6, and 10. But version 3 is relatively new, and lots of software licensed under the GPL uses version 2, found at http://www.gnu.org/licenses/gpl-2.0.html. In version 2, look in particular at Sections 1 through 3. Whatever version you review, beware: the GPL is not for the faint of heart. The language confuses many lawyers, including me, so don't be surprised if you have to read it over and over.

If you're planning to provide software under the open source model—to become an open source licensor—use a copyleft license if you want to make sure all your recipients use the open source model. It requires that your recipients—immediate and downstream—use open source licensing for any modified version *they* distribute.

If you're a potential open source recipient, copyleft can help you or hurt you, depending on your plans for the software.

If all you want to do is *use* the software—if you're just a recipient and never a vendor for that software—copyleft shouldn't worry you. You'll probably benefit from the software evolution encouraged by copyleft—from the fact that other recipients contribute new code to a community of developers, for use by all. You shouldn't worry about copyleft if you're a software vendor, either, if you don't mind sharing your source code and letting others redistribute your software. Copyleft doesn't present a problem.

But if you're a traditional software vendor and you *don't* want to provide your software under the open source model, copyleft creates a problem. If you plan to distribute a software product and you want to keep its source code to yourself, or if you want to make sure no one else can distribute it, you've got to remove any copyleft-licensed software before you distribute a single copy. That's no easy task with a massive system. It may call for help from an open source audit firm—a company that detects buried copyleft-licensed software.

Finally, if you're a cloud services vendor and you don't want to use the open source model, copyleft software in your system *might* create a problem. As explained in Chapter I.E ("Subscription for Cloud Services"), SaaS and other cloud services vendors don't give their customers copies of their software. They don't *distribute* software—or even license it if they draft their contracts right—but rather give their customers remote access.

It's distribution that triggers most copyleft clauses, not access, so copyleft doesn't apply to cloud systems—most of the time.

Cloud services vendors do face two lingering concerns. First, they become software licensors if they let their customers host the software: if they provide a copy for the customer to run on its own server. That counts as distribution, so copyleft would apply. Suddenly, anyone with a copy can distribute your software and has a right to source code. Second, another license from the Free Software Foundation, called the Affero General Public License (AGPL), *does* try to extend copyleft requirements to cloud-hosted software.[7] AGPL accounts for a lot less software than the other copyleft licenses, but the risk remains. (It's possible copyleft stands on shakier legal ground in the cloud than in the software licensing world, but again, who wants to serve as the test case?)

If you're a developer—cloud services or traditional—and you don't plan to use the open source model, consider requiring that your own software licensors/vendors do everything possible to exclude copyleft software from *their* products, and protect you from the consequences if some slips in anyway. Among other remedies, you should consider an open source warranty in your contracts with licensors, as well as a strong intellectual property indemnity.[8]

Permissive (Less Restrictive) Open Source Licenses

Most open source licenses lack copyleft provisions. They let the recipient include open source software in a larger program and then distribute that program under just about any terms it wants, with or without the open source model.

If you plan to provide open source software—if you're the licensor—consider a less restrictive open source license if you don't care whether your recipients themselves use the open source model if they redistribute your software. If you're the recipient, less restrictive licenses probably won't cause you any trouble, whether or not you plan to redistribute the software or provide it to your customers as a cloud service. But, of course, you should still read your license terms.

The BSD-type licenses are probably the least restrictive open source contracts. The category is named for the Berkeley Software Distribution of Unix, an open source operating system. These licenses place no IP-specific restriction on redistribution of software.

The BSD category has several licenses, including the BSD License itself, as well as the MIT License, named for the Massachusetts Institute of Technology. You can find current versions of both the BSD and MIT licenses online at the website of the Open Source Initiative, another open source nonprofit. Both licenses are short and easy to read. The BSD License appears at http://www.opensource.org/licenses/bsd-license.php, and the MIT License appears at http://www.opensource.org/licenses/mit-license.php.

The Mozilla Public License is slightly more restrictive but still widely viewed as friendly to traditional software developers. It's maintained by the Mozilla Foundation, yet another open source nonprofit (and the source of the Firefox browser). Like copyleft forms, the Mozilla license requires that modified versions of the software be distributed under the open source model: with source code and with the right to modify and redistribute. But that restriction is far narrower than copyleft. It doesn't apply to all derivative works, just to “modifications” of the original code: individual files that contain the original software. So if the recipient puts the open source software into a larger program, it hasn’t “infected” the whole program. The recipient has to use the open source model for the files containing the original open source software if it redistributes, but it can distribute the rest of the program under any IP terms it likes.

You can find the Mozilla Public License at http://www.mozilla.org/MPL/MPL-1.1.html. It’s longer than the BSD and MIT licenses, but the language is relatively manageable. In version 1.1, see in particular Sections 1.9, 2, 3.1, 3.2, 3.6, and 3.7.

Other Terms of Open Source Licenses

Open source licenses generally include various disclaimers and notice requirements, protecting the licensor. A disclaimer of functionality and other warranties is almost universal. Open source software is usually provided “as is,” without warranties (express or implied).[9] And warranty disclaimers go hand in hand with notice requirements. The licensor doesn’t want downstream recipients—those who get the software directly or indirectly from the original recipient—to expect warranties either. So when the recipient redistributes the code, it’s required to include the warranty disclaimer. Generally, recipients also have to include copyright notices, informing downstream customers that the software is subject to copyright and identifying the copyright holder.

Other terms are less universal. Open source contracts may include limitations of liability, patent licenses (with “open” terms similar to the copyright licenses discussed above), and choice of law clauses, among other provisions.[10]

APPENDIX 3

Clickwraps, Browsewraps, and Other Contracts Executed without Ink

This appendix covers contracts that aren't signed in ink: usually nonnegotiable standard agreements prepared by the vendor of products or services, or of a website. The main issue is enforceability. If you're the vendor, you've got to make sure your customer executes the contract in a way that confirms both notice of the terms and consent. Without notice and consent, you don't have a contract, and a court won't enforce your terms.

We traditionally confirm notice and consent through an ink signature. Courts enforce signed contracts because we're all expected to know what a signature means. If you signed on the dotted line and didn't realize you'd consented to the terms above that line . . . well, that's your problem.

But an ink signature isn't always practical for a form contract. That's where shrinkwrap contracts come in, along with their more modern descendants: clickwraps and browsewraps.

A *shrinkwrap* contract is a printed form accompanying a product container. It might be printed on the outside of a box of software, or it might be a paper form held against the box by a plastic wrapper. The name "shrinkwrap," in fact, comes from the clear plastic shrink-to-fit wrappers traditionally used with these contracts. A shrinkwrap generally begins with something like: "By opening this box, you agree to the contract terms below." The customer confirms notice of the terms and consent by opening the box. If the customer reads the terms and doesn't consent, he or she doesn't open the box and can return the product for a full refund.

Courts have accepted the logic of notice and consent via shrink-wrap, so they'll usually enforce the contract. But notice and consent have to be meaningful. If the contract isn't clearly visible—if the customer could open the box without noticing—a court might refuse enforcement. Courts might also refuse if the customer can't easily return the product after reading the full contract.

A *clickwrap* contract is an electronic form posted online or on a start-up screen of a software product. It says something like: "By clicking 'I agree' below, you agree to the following terms." The customer can't install the software or use the service until he or she clicks "I agree." And some clickwraps go further: the customer can't click "I agree" until he or she scrolls all the way through the contract. (Some IT professionals call these "scrollwraps.")

If you use a clickwrap, make sure the "click" comes before payment—or at least make sure the customer can easily decline the service or return the software if he or she decides not to click. Make sure also that the signature click says something like "I agree" or, even better, "I agree to the Terms and Conditions"—rather than just "Next" or "Sign Up." In other words, the contract might not be enforceable if you just note somewhere on the page that by clicking "Next," the customer accepts the contract. (That would create what one court has called a "sign-in-wrap.")[11] The button itself should clearly, unequivocally state that the customer accepts the contract by clicking. If you do all that, a clickwrap should work. The customer can't persuasively argue that he or she didn't notice the contract or understand that consent was necessary.

A *browsewrap* contract is an online form without a click-to-agree feature. For instance, many websites provide a "terms of service" link at the bottom of each page. The customer doesn't have to click "I agree" or take any other action to use the site or the products or services provided there.

You'll find browsewraps harder to enforce than clickwraps or shrinkwraps, because it's harder to establish notice and consent. The customer might not notice the contract or realize it applies to him or her. So you should avoid browsewraps, if possible, and use clickwraps.

Sometimes, however, a clickwrap isn't practical. No one clicks "I agree" to surf a website. If you have to use a browsewrap, give the customer clear notice of the terms and of the fact that they apply to him or her. Ideally, you'd splash big flashing letters across the top of each webpage: "BY USING THIS WEBSITE, YOU CONSENT TO THE TERMS AND CONDITIONS POSTED HERE." That's almost never practical, but keep it in mind as the ideal, and make your notice as visible as possible. Also, if you’re offering a product or service that requires some action by the customer, the notice should immediately precede the action point. For instance, just above a software download button, provide a link reading: “DO NOT CLICK THE DOWNLOAD BUTTON BELOW UNLESS YOU AGREE TO THE TERMS AND CONDITIONS POSTED HERE.” Of course, once you’ve gone that far, you might as well use a clickwrap.

Some contracts are *hybrids*: both clickwrap and browsewrap. Many websites require that their paying customers sign up online and, in the process, click “I agree” to contract terms. That’s a click-wrap. These sites also provide a link to the same contract at the bottom of each page, for website visitors who never sign up. It’s the same contract, but now it’s a browsewrap. If you use a hybrid, keep in mind that you’re less likely to enforce it against casual visitors—against browsewrap customers—than against clickwrap customers.

Finally, whatever type of “wrap” you use, make sure the customer has some way to save and store the terms. That’s easy enough online, because we can all print or save just about any webpage we come across. But vendors should consider going the extra mile and making it easy. You might include a “click here to print” button, or a “click here to download a copy” button.

Within software (as opposed to online), access to the contract becomes more difficult for the customer. Few of us could track down the clickwrap we executed when we first booted up our computer. So vendors should be sure to include a print or save choice for their offline clickwraps.

APPENDIX 4

Online Policy Documents

This appendix covers acceptable use policies (AUPs), Digital Millennium Copyright Act (DMCA) policies, and privacy policies. Companies doing business online post these documents on their websites to explain their rules and procedures.

AUPs, DMCA policies, and privacy policies aren't exactly contract clauses. They're just statements: information for customers and other users. Still, some online policy documents work like contract clauses. When customers buy products and services online, they often rely on AUPs and privacy policies, and that can make those policies binding on the vendor. Also, some online agreements require that customers comply with certain policies, particularly AUPs, while others fully incorporate all their policies into the contract, transforming them into contract clauses binding on both parties. However this incorporation issue plays out, vendors limit legal risk if they obey all their own online policies.

As you'll see, each policy in the following clause boxes includes a "Date Posted" or "Effective Date." You should include these dates in all your online policies. They help customers determine whether the policy has changed since they last reviewed it (assuming they ever do).

Acceptable Use Policies

AUPs outline user behavior that won't be tolerated. They're most useful for services that enable online communication. Internet service vendors, for instance, should consider AUPs. So should software-as-a-service and other cloud services vendors offering messaging, postings, chat rooms, and other communications systems.

As noted above, the AUP is a policy statement at heart, not a contract clause. But if you're the vendor, there's no reason your contract can't require compliance with your AUP. "Customer shall comply with Vendor's acceptable use policy ("AUP"), posted at www.congenialme.com, as such policy may change from time to time." Just make sure incorporation of the AUP doesn't tie your hands. Retain the right to change the AUP—as authorized in the suggested language above and, usually, in the AUP itself. See part D of the following clause box. And make sure incorporation into the contract doesn't require that you enforce rules you don't always want enforced, or fully enforced. "Neither this Agreement nor the AUP requires that Vendor take any action against any customer or user violating the AUP, but Vendor is free to take any such action it sees fit."

Acceptable Use Policy

Date Posted: _______________

A. Unacceptable Use

Vendor requires that all customers and other users of Vendor's cloud-based service (the "Service") conduct themselves with respect for others. In particular, observe the following rules in your use of the Service:

1)*Abusive Behavior:* Do not harass, threaten, or defame any person or entity. Do not contact any person who has requested no further contact. Do not use ethnic or religious slurs against any person or group.

2)*Privacy:* Do not violate the privacy rights of any person. Do not collect or disclose any personal address, social security number, or other personally identifiable information without each holder's written permission. Do not cooperate in or facilitate identity theft.

3)*Intellectual Property:* Do not infringe upon the copyrights, trademarks, trade secrets, or other intellectual property rights of any person or entity. Do not reproduce, publish, or disseminate software, audio recordings, video recordings, photographs, articles, or other works of authorship without the written permission of the copyright holder.

4)*Hacking, Viruses, & Network Attacks:* Do not access any computer or communications system without authorization, including the computers used to provide the Service. Do not attempt to penetrate or disable any security system. Do not intentionally distribute a computer virus, launch a denial of service attack, or in any other way attempt to interfere with the functioning of any computer, communications system, or website. Do not attempt to access or otherwise interfere with the accounts of other users of the Service.

5)*Spam:* Do not send bulk unsolicited e-mails ("Spam") or sell or market any product or service advertised by or connected with Spam. Do not facilitate or cooperate in the dissemination of Spam in any way. Do not violate the CAN-Spam Act of 2003.

6)*Fraud:* Do not issue fraudulent offers to sell or buy products, services, or investments. Do not mislead anyone about the details or nature of a commercial transaction. Do not commit fraud in any other way.

7)*Violations of Law*: Do not violate any law.

B. Consequences of Violation

Violation of this Acceptable Use Policy (this "AUP") may lead to suspension or termination of the user's account or legal action. In addition, the user may be required to pay for the costs of investigation and remedial action related to AUP violations. Vendor reserves the right to take any other remedial action it sees fit.

C. Reporting Unacceptable Use

Vendor requests that anyone with information about a violation of this AUP report it via e-mail to the following address: ___________. Please provide the date and time (with time zone) of the violation and any identifying information regarding the violator, including e-mail or IP (Internet Protocol) address if available, as well as details of the violation.

D. Revision of AUP

Vendor may change this AUP at any time by posting a new version on this page and sending the user written notice thereof. The new version will become effective on the date of such notice.[12]

The behavior forbidden by an AUP depends on the nature of the vendor's business. The example in the clause box above would work for many cloud services vendors. When you design your own AUP, think about user behavior that could injure you, injure your users, or lead to liability.12

An AUP doesn't have to list consequences for violation, but it's usually a good idea. Your goal is to inform. Generally, consequences for violation should include suspension of the service or termination, as in part B of the clause box above. But make sure you don't limit yourself. For instance, part B says the vendor can take any other actions it sees fit.

The AUP doesn't need a reporting clause either, but you might want watchful users to help police your service. See part C of the clause box.

Finally, AUP prohibitions often include imprecise language—language you'd want to avoid in a contract, if possible. For instance, part A.5 in the previous clause box defines "Spam" as "bulk unsolicited e-mail." What does "bulk" mean? And exactly what do "harass" and "threaten" mean in part A.1? AUPs often deal with hard-to-define concepts—like *respect*, which lies at the heart of every AUP. So it's hard to avoid some imprecision. But do everything you can to make your AUP clear. And when you're dealing with potential AUP violations, give the customer the benefit of the doubt regarding unclear terms. That practice will help if you ever wind up in court, battling over whether an AUP-related termination was justified.

DMCA/Copyright Notice Policies

The Digital Millennium Copyright Act is a U.S. federal law. One of its provisions protects online communications vendors from copyright liability. If an ISP or other vendor follows certain procedures, it's not liable for copyright infringement initiated by its subscribers. In other words, if a subscriber posts a third party's recording or article or other copyrighted work, the *subscriber* might be liable for copyright infringement, but the vendor isn't—even though the vendor's computers made the unauthorized copy and displayed it to the public.[13]

DMCA policies help vendors take advantage of the act's "safe harbor": its protection from liability. The policies are online announcements about the vendor's procedures surrounding copyright infringement. That's why they're sometimes called "copyright notice policies" or "intellectual property infringement policies."

This appendix covers DMCA policies and related contract terms, but it doesn't go into detail about the safe harbor's other requirements. However, since some knowledge of the other requirements will help you understand DMCA policies, here's a summary. First, the safe harbor only applies if you, the vendor, had no knowledge of the copyright infringement and got no financial benefit directly attributable to infringement. Second, to take advantage of the safe harbor, you have to take the following steps:[14]

(1) Make sure your network doesn't block any technical measures copyright holders use to protect their work or to identify infringement.

(2) Designate an agent to receive copyright complaints, and register that agent's contact information with the U.S. Copyright Office.

(3) Post the agent's contact information online.

(4) Enforce a policy calling for termination of subscribers who repeatedly infringe copyrights, and inform subscribers of that policy.

(5) Follow the copyright notice, take-down, and counter-notice procedures discussed below.

Steps 3 and 4 are the only ones that require statements to the public or to subscribers, so they're the only ones you have to address in a DMCA policy. A vendor can address them by posting something like the following clause box.

DMCA Notice

For claims of copyright infringement, please contact ____________________ [name of registered agent and/or his or her department, address, phone number, and e-mail address]. We will terminate the accounts of subscribers who are repeat copyright infringers.

That's it. You don't need more for a DMCA policy. But because the act requires that you notify subscribers of your policy against repeat infringers (step 4 above), you should make sure subscribers *see* the notice. You might include a link in a sign-up screen, or just send a link early in the relationship. Or you might include something like the following in your subscriber agreement: "Subscriber is on notice that Vendor may terminate the accounts of subscribers who are repeat copyright infringers."[15]

Many policies go further. They explain step 5: the act's "notice and take-down procedures." In short, if someone sends the vendor a properly detailed notice of copyright infringement, complaining about materials posted by a subscriber, the vendor promptly removes the accused materials. If the vendor does that, it's not liable for copyright infringement. Of course, removing the materials might create liability to the subscriber (though not if your contract gives you the right to remove suspicious materials). But the DMCA protects the vendor there too. To take advantage of that berth in the safe harbor, the vendor informs the subscriber of the claim. Then, if the subscriber sends a properly detailed "counter-notice," claiming the materials don't infringe copyright, the vendor puts the materials back up. If the vendor does all that, it avoids liability for copyright infringement and for removing the materials. From there, the claimant and subscriber can fight it out in court, and the vendor doesn't have to worry.

Much of the language in the following clause box comes directly from the Digital Millennium Copyright Act. You should stick close to that language. DMCA policies don't call for a lot of creativity.

DMCA Policy

Date Posted: _______________

This policy statement lists our requirements for notice of copyright infringement and for responses to such a notice if you or your materials are accused.

We use the copyright infringement procedures of the Digital Millennium Copyright Act.

A. Notice of Copyright Infringement

To notify us of copyright infringement, please send a written communication to our Copyright Notices Department, at the contact points listed below in Part C. That written communication should include the following:

1)A physical or electronic signature of a person authorized to act on behalf of the owner of an exclusive right that is allegedly infringed.

2)Identification of the copyrighted work claimed to have been infringed, or, if multiple copyrighted works at a single online site are covered by a single notification, a representative list of such works at that site.

3)Identification of the material that is claimed to be infringing or to be the subject of infringing activity and that is to be removed or access to which is to be disabled, and information reasonably sufficient to permit us to locate the material.

4)Information reasonably sufficient to permit us to contact the complaining party, such as an address, telephone number, and, if available, an electronic mail address at which the complaining party may be contacted.

5)A statement that the complaining party has a good faith belief that use of the material in the manner complained of is not authorized by the copyright owner, its agent, or the law.

6)A statement that the information in the notification is accurate, and under penalty of perjury, that the complaining party is authorized to act on behalf of the owner of an exclusive right that is allegedly infringed.

B. Counter-Notice by Accused Subscriber

If you are a subscriber and we have taken down your materials due to suspicion of copyright infringement, you may dispute the alleged infringement by sending a written communication to our Copyright Notice Department, at the contact points listed in Part C below. That written communication should include the following:

1)A physical or electronic signature of the subscriber.

2)Identification of the material that has been removed or to which access has been disabled and the location at which the material appeared before it was removed or access to it was disabled.

3)A statement under penalty of perjury that the subscriber has a good faith belief that the material was removed or disabled as a result of mistake or misidentification of the material to be removed or disabled.

4)The subscriber’s name, address, and telephone number, and a statement that the subscriber consents to the jurisdiction of Federal District Court for the judicial district in which such address is located, or if the subscriber’s address is outside of the United States, the Federal District Court for ______________ [insert vendor’s home district], and that the subscriber will accept service of process from the person who provided notification of copyright infringement or an agent of such person.

C. Agent for Notices

Please send all notices required by this policy to our Copyright Notice Department at ______________ [address, phone number, and e-mail address].

D. Termination of Repeat Infringers

In appropriate circumstances, we will terminate the accounts of subscribers who are repeat copyright infringers.

E. Revision of Policy

We may revise this policy at any time, including by posting a new version at this website.[16]

Whatever your DMCA policy says, you should post it in a prominent place on your website—ideally a page linked to your home page.[16]

As discussed earlier, the act requires that you notify subscribers of your policy against repeat infringers. So if that notice appears in your DMCA policy—as in part D of the clause box above—you should make sure subscribers *see* the policy. See the suggestions above, just after the “DMCA Notice” clause box.

Many cloud services vendors include a long DMCA policy in the body of their subscriber agreement, as a contract clause. I think that’s a mistake—at least if the DMCA policy includes notice and take-down procedures, like parts A and B in the clause box above. If you include the policy in your contract, the notice and take-down procedures are no longer optional. You *have* to follow them. What if you mishandle the procedures? A subscriber could sue you for breach of contract. You’ve turned an optional legal protection into a liability. Of course, you could avoid the problem by stating that the DMCA policy binds the subscriber but not you. But why bother? Including the procedures in the contract won’t increase your safe harbor protection. And most copyright claimants will be third parties, not your subscribers. A contract clause wouldn’t bind them, and if the DMCA policy appears in your contract, rather than a separate webpage, third parties might never stumble across it. That defeats the purpose.

Privacy Policies

In a privacy policy, the vendor of a website or an online service explains what it will and won't do with users' private information.

In general, any website or online service that collects personally identifiable information should have a privacy policy. "Personally identifiable information" (PII) refers to any information that could be used to identify an individual, or to contact or locate an individual. That includes sensitive information, like social security numbers, credit card numbers, bank account numbers, and medical records. But it also includes more widely available information, like e-mail addresses, snail mail addresses, and telephone numbers, as well as the *names* of users. So the only websites that don't need privacy policies are simple signposts—sites that collect no information—and sites that collect nothing but 100 percent anonymous feedback, like survey responses that couldn't possibly be used to identify the responder. Arguably, websites that post their operators' e-mail addresses should have privacy policies, even if they collect no information. Users could send e-mail, thus revealing personally identifiable information: their e-mail addresses.

Where does this privacy policy "requirement" come from? The United States has no unified set of laws governing privacy. Instead, we have a confusing array of federal and state statutes. And when you add European, Canadian, and other foreign regulations—which may govern vendors serving foreign customers—the list of laws and regulations grows. There's a fair chance that a law requiring privacy policies applies to your site or service. Even if not, you should consider a privacy policy because it can limit your liability. If you honestly and fully disclose the use you'll make of private information, users can't easily argue that they never knew about your use or authorized it. Finally, privacy policies make business sense. They make users and business partners comfortable.

Because the United States lacks a unified privacy law, it's hard to say exactly what information should appear in your policy. As I've mentioned before, this book is no substitute for an experienced lawyer's advice. But it is possible to lay out a broad set of privacy policy best practices. In general, your privacy policy should do the following to limit legal risk:

1. Identify the categories of personally identifiable information collected.
2. Identify the categories of third parties that receive PII or have access to it (if any).
3. Describe the ways PII is used.
4. Summarize security measures to protect PII.
5. Tell users how they can review and change their PII (assuming they can).
6. Explain how you notify users of changes to the privacy policy.
7. List the policy’s effective date.
8. Say nothing in the privacy policy that is not absolutely and consistently true.[17]

Is a privacy policy a contract clause? In one way, the answer is academic. If a site or service posts a policy, users will rely on it, and that reliance will probably render the policy binding on the vendor. Various laws might also make the policy binding. Still, if you’re a vendor, there’s no particular reason to incorporate your policy into your contract. It’s at least possible you’d be increasing your potential liability, and it’s also possible incorporation into the contract could limit your right to revise the policy.[18]

The following clause box envisions a vendor that sells products and services online, possibly cloud services, and that has customers create a “sign-in account.”

Privacy Policy

Effective Date: _______________

We collect certain information through our website, located at _____________ (our "Website"), including through the products and services provided at the Website. This page (this "Privacy Policy") lays out our policies and procedures surrounding the collection and handling of any such information that identifies an individual user or that could be used to contact or locate him or her ("Personally Identifiable Information" or "PII").

This Privacy Policy applies only to our Website and to the products and services provided through our Website. It does not apply to any third party site or service linked to our Website or recommended or referred by our Website, through our products or services, or by our staff. And it does not apply to any other website, product, or service operated by our company, or to any of our offline activities.

A. PII We Collect

We collect the following Personally Identifiable Information from users who buy our products or services: name, e-mail address, telephone number, address, and credit card number.

We also use "cookies" to collect certain information from all users, including Web visitors who don't buy anything through our Website. A cookie is a string of data our system sends to your computer and then uses to identify your computer when you return to our Website. Cookies give us usage data, like how often you visit, where you go at the site, and what you do.

B. Our Use of PII

We use your Personally Identifiable Information to create your account, to communicate with you about products and services you've purchased, to offer you additional products and services, and to bill you. We also use that information to the extent necessary to enforce our Website terms of service and to prevent imminent harm to persons or property.

We use cookies so that our Website can remember you and provide you with the information you're most likely to need. For instance, when you return to our Website, cookies identify you and prompt the site to provide your username (not your password), so you can sign in more quickly. Cookies also allow our Website to remind you of your past purchases and to suggest similar products and services. Finally, we use information gained through cookies to compile statistical information about use of our Website, such as the time users spend at the site and the pages they visit most often. Those statistics do not include PII.

C. Protection of PII

We employ the following data security tools to protect Personally Identifiable Information: _________________________ ______________________. Unfortunately, even with these measures, we cannot guarantee the security of PII. By using our Website, you acknowledge and agree that we make no such guarantee, and that you use our Website at your own risk.

D. Contractor and Other Third Party Access to PII

We give certain independent contractors access to Personally Identifiable Information. Those contractors assist us with _____________________. All those contractors are required to sign contracts in which they promise to protect PII using procedures reasonably similar to ours. (Users are not third party beneficiaries of those contracts.) We also may disclose PII to attorneys, collection agencies, or law enforcement authorities to address potential AUP violations, other contract violations, or illegal behavior. And we disclose any information demanded in a court order or otherwise required by law or to prevent imminent harm to persons or property. Finally, we may share PII in connection with a corporate transaction, like a merger or sale of our company, or a sale of all or substantially all of our assets or of the product or service line you received from us, or a bankruptcy.

As noted above, we compile Website usage statistics from data collected through cookies. We may publish those statistics or share them with third parties, but they don't include PII. Except as set forth in this Privacy Policy, we do not share PII with third parties.

E. Accessing and Correcting Your PII

You can access and change any Personally Identifiable Information we store through your "My Account" page.

F. Amendment of This Privacy Policy

We may change this Privacy Policy at any time by posting a new version on this page or on a successor page. The new version will become effective on the date it's posted, which will be listed at the top of the page as the new Effective Date.[19]

The example in the clause box above is bare-bones: it lacks reassuring language about the vendor's respect for privacy. You might consider adding something like: "Your privacy is important to us."

The example also lacks contact information for a privacy officer who can discuss PII. Sometimes it's a good idea to make such a person available and to post his or her contact information in the privacy policy.19

Section D in the clause box says the company won't disclose PII to third parties, except certain contractors under NDA or a third party that acquires the whole business. You should think through whether that's really true. Would you ever need to disclose private information to a business partner? Often the answer is "no": no one needs access to the PII itself, as opposed to statistics about it. But if the answer is "yes," consider broadening Section D.

The law imposes special privacy-related obligations on certain businesses, including businesses that collect medical records, financial data, and information about children.[20] Those obligations may affect the content of your privacy policy. If you might be subject to special privacy regulation, you should get help from a lawyer with privacy expertise.

[1]. This misunderstanding leads to excessive concerns about ownership of "feedback," discussed in Chapter II.N ("Feedback Rights").

[2]. Copyright is governed by the Copyright Act of 1976, 17 U.S.C. §§ 101 *et seq*.

Mask works—a stepsister of copyright—are governed by the Semiconductor Chip Protection Act of 1984, 17 U.S.C. §§ 902 *et seq*. Mask works are three-dimensional patterns involved in the creation of semiconductors (computer chips).

[3]. Patent law is governed by the Patent Act, 35 U.S.C. The same act governs design patents: protection for new, original, ornamental designs on manufactured goods.

[4]. The trade secrets definition above paraphrases the UTSA.

[5]. Federal trademark law comes from the Lanham Act, 15 U.S.C. §§ 1051 *et seq*.

[6]. The open source community doesn't consider its licenses "contracts." A hazy line separates licenses and *contracts that include licenses*. In simplified terms, a license grants rights without any corresponding promises from the recipient, while a contract involves promises from both sides (or at least promises in exchange for actions).

For a more complete definition of "open source," see *The Open Source Definition* provided by the Open Source Initiative: http://www.opensource.org/docs/osd.

[7]. You can find the AGPL at http://www.gnu.org/licenses/agpl-3.0.html.

[8]. See Subchapter II.I.4 ("Other Warranties") and Chapter II.J ("Indemnity").

[9]. See Chapter II.I ("Warranty").

[10]. See, respectively, Chapter II.K ("Limitation of Liability"), Appendix 1 ("Intellectual Property"), and Chapter III.E ("Choice of Law and Courts").

[11]. Berkson v. Gogo LLC & Gogo Inc., No. 14-CV-1199 (E.D.N.Y. April 9, 2015).

[12]. See Chapter III.S ("Amendment"), in the second-to-last paragraph, for contracting concerns surrounding amending online policy documents.

[13]. The DMCA appears in multiple sections of the U.S. Code. This appendix discusses one of several portions of the DMCA, called the Online Copyright Infringement Liability Limitation Act, 17 U.S.C. § 512.

The DMCA refers to customers and other users as "subscribers," so this part of Appendix 4 uses the same term.

[14]. The summary here isn't detailed enough to ensure full compliance. You can find various DMCA guides online, and of course a short consultation with an experienced attorney should tell you all you need. You can also read the statute (which is written in English, believe it or not).

[15]. You could also include the repeat infringer policy in your AUP. But you still have to make sure subscribers see it.

[16]. See Chapter III.S ("Amendment"), in the second-to-last paragraph, for contracting concerns surrounding amending online policy documents.

[17]. Recommendations 1, 2, and 5 through 7 are required by one of the better-known state statutes on privacy policies: California's Online Privacy Protection Act of 2003, Cal. Business and Professions Code §§ 22575 *et seq.*—particularly § 22575(b).

[18]. The law on this point isn't clear. For more on this issue, see Chapter III.S ("Amendment"), in the second-to-last paragraph.

[19] See Chapter III.S ("Amendment"), in the second-to-last paragraph, for contracting concerns surrounding amending online policy documents.

[20] The federal government's key data security laws include the Gramm-Leach-Bliley Act (governing financial institutions), the Health Insurance Portability and Accountability Act, the Children's Online Privacy Protection Act, and arguably the Sarbanes-Oxley Act (on corporate corruption and financial reporting). Most states have information privacy laws, too.

Mini-Glossary

The following five terms and phrases are used repeatedly in this book.

calendar (as in calendar year, calendar quarter, and calendar month): Shorthand for a period defined by the standard calendar. Calendar year is often contrasted with a company's *fiscal year*, which may not start on January 1. September is a calendar month, while "the 30 days following delivery" is a month defined by the contract, not the calendar. Finally, the first calendar quarter consists of January, February, and March, and the remaining three calendar quarters are likewise defined by sets of three consecutive calendar months.

including without limitation: A quick way to say: "The following is an example, or a list of examples, but the fact that we're listing these examples does *not* mean there are no others." For instance, a license clause might read: "Vendor will deliver the Software, including without limitation all EasilyForgotten applications." The fact that the clause lists EasilyForgotten applications doesn't mean other software components aren't required.

object code: A version of software that a computer can read. It's also sometimes called "machine-readable code." (Actually, those two terms don't mean exactly the same thing, but they're close enough for our purposes.) Object code is contrasted with source code.

source code: The version of software that a human programmer can read. In fact, source code is the version a human wrote: the original version of most software. Source code gets "compiled" or translated into object code.

without limiting the generality of the foregoing: A quick way to say: "The preceding text gives a broad, general rule. A specific and narrow example follows, but the fact that the example is specific and narrow does not make the general rule any less broad or general." For instance, a license clause might provide: "Distributor will exercise its best efforts to market and sell the Software. Without limiting the generality of the foregoing, if Distributor fails to achieve gross revenues of $500,000 from Software distribution during any calendar year, Vendor may revoke the license granted in this Section __." The rule about minimum royalties is related to the general *best efforts* rule, but it doesn't limit that rule. So the fact that the distributor hits its revenues number doesn't necessarily mean that it's complied with its "best efforts" obligation.

Index

About the ABA Section of Intellectual Property Law

From its strength within the American Bar Association, the ABA Section of Intellectual Property Law (ABA-IPL) advances the development and improvement of intellectual property laws and their fair and just administration. The Section furthers the goals of its members by sharing knowledge and balanced insight on the full spectrum of intellectual property law and practice, including patents, trademarks, copyright, industrial design, literary and artistic works, scientific works, and innovation. Providing a forum for rich perspectives and reasoned commentary, ABA-IPL serves as the ABA voice of intellectual property law within the profession, before policy makers, and with the public.

ABA-IPL Books Editorial Board

ABA Section of Intellectual Property Law
Order today! Call 1-800-285-2221
Monday-Friday, 7:30 a.m. – 5:30 p.m., Central Time
or Visit the ABA Web Store: www.ShopABA.org

Qty	Title	Regular Price	ABA-IPL Member Price	Total
____	ADR Advocacy, Strategies, and Practice in Intellectual Property Cases (5370195)	$179.95	$114.95	$____
____	ANDA Litigation (5370199)	$239.00	$249.00	$____
____	Careers in IP Law (5370204)	$24.95	$16.95	$____
____	Computer Games and Virtual Worlds (5370172)	$59.95	$55.95	$____
____	Copyright Remedies (5370208)	$89.95	$74.95	$____
____	Distance Learning and Copyright (5370163)	$89.95	$79.95	$____
____	Fundamentals of Intellectual Property Valuation (5370143)	$59.95	$49.95	$____
____	IP Attorney's Handbook for Insurance Coverage in Intellectual Property Disputes, Second Edition (5370210)	$139.95	$129.95	$____
____	IP Protection in China (5370217)	$159.95	$129.95	$____
____	A Lawyer's Guide to Section 337 Investigations before the U.S. International Trade Commission, Second Edition (5370203)	$119.95	$89.95	$____
____	A Legal Strategist's Guide to Trademark Trial and Appeal Board Practice, Second Edition (5370200)	$159.95	$129.95	$____
____	Music & Copyright in America (5370201)	$87.95	$67.95	$____
____	New Practitioner's Guide to Intellectual Property (5370198)	$89.95	$69.95	$____
____	The Patent Infringement Litigation Handbook (1620418)	$149.95	$129.95	$____
____	Patently Persuasive (5370206)	$129.95	$99.95	$____
____	Patent Obviousness in the Wake of *KSR International Co. v. Teleflex Inc.* (5370189)	$129.95	$103.95	$____
____	Patent Trial Advocacy Casebook, Third Edition (5370124)	$149.95	$119.95	$____
____	The Practitioner's Guide to the PCT (5370205)	$139.95	$109.95	$____
____	Practitioner's Guide to Trials Before the Patent Trial and Appeal Board (5370209)	$139.95	$114.95	$____
____	Pre ANDA Litigation (5370212)	$275.00	$220.00	$____
____	Preliminary Relief in Patent Infringement Disputes (5370194)	$119.95	$94.95	$____
____	Right of Publicity (5370215)	$89.95	$74.95	$____
____	Settlement of Patent Litigation and Disputes (5370192)	$179.95	$144.95	$____
____	Starting an IP Law Practice (5370202)	$54.95	$34.95	$____
____	The Tech Contracts Handbook (5370188)	$34.95	$29.95	$____
____	Technology Transfer Law Handbook (5370211)	$220.00	$176.00	$____
____	Trademark and Deceptive Advertising Surveys (5370197)	$179.95	$154.95	$____
____	Trademark Surveys (5370207)	$289.95	$239.95	$____

* Tax
DC residents add 6%
IL residents add 9.50%

Payment
□ Check enclosed payable to the ABA
□ VISA □ Mastercard □ American Express

* Tax	$____
** Shipping/Handling	$____
TOTAL	$____

****Shipping/Handling**

Up to $49.99	$5.95
$50 to $99.99	$7.95
$100 to $199.99	$9.95
$200 to $499.99	$12.95
$500 to $999.99	$15.95
$1,000 and above	$18.95

Name____________________

Firm/Organization____________________

Address____________________

City____________________ State_____ Zipcode__________

Phone____________ E-mail____________
(in case of questions about your order)

Please allow 5 to 7 business days for UPS delivery. Need it sooner? Ask about overnight delivery. Call the ABA Service Center at 1-800-285-2221 for more information.

Guarantee: If – for any reason – you are not satisfied with your purchase, you may return it within 30 days of receipt for a complete refund of the price of the book(s). No questions asked!

Please mail your order to:
ABA Publication Orders, P.O. Box 10892, Chicago, Illinois 60610-0892
Phone: 1-800-285-2221 or 312-988-5522 • Fax: 312-988-5568
E-mail: orders@abanet.org

Thank you for your order!

ABA Section of Intellectual Property Law
AMERICAN BAR ASSOCIATION