11-1-2005

Once and Future Copyright

James Gibson

Follow this and additional works at: http://scholarship.law.nd.edu/ndlr

ONCE AND FUTURE COPYRIGHT

James Gibson*

INTRODUCTION

By the time this paragraph comes to an end, it will be protected by copyright law. This would likely come as a surprise to the layperson, who probably assumes that to secure copyright protection one must do something more than put pen to paper. But the fact is that copyright asks only two things of an author before bestowing its benefits: compose some original expression and fix it in a tangible medium.[1] Despite its pedestrian phrasing and amateur attempts at alliteration, this paragraph satisfies the originality standard, and its appearance in print means it is sufficiently fixed. And that's it. Copyright law requires nothing more. No magic incantations need be uttered, no orphic symbols need appear. I do not have to file papers with some government agency, or deposit a copy with the Library of Congress, or even affix the totemic c-in-a-circle to the printed page. In fact, even if no journal had published this paragraph, and it languished forever in a dusty file cabinet, copyright law would still protect it.

Securing copyright protection was not always so simple. For most of copyright's history, authors had to meet certain formal requirements if they wanted copyright to protect their works. Copyright did not cover works that were unpublished, or were not registered and deposited with the government, or failed to include a notice indicating their protected status.[2] Over the last century, however, these formalities withered away, generally without any rigorous analysis of

---

1   17 U.S.C. § 102(a) (2000).

2   For an account of the rise and fall of these formalities, see Christopher Sprigman, *Reform(aliz)ing Copyright*, 57 STAN. L. REV. 485, 491–99 (2004). Sprigman's informative article also provides a detailed statistical analysis of the historical usage of the formalities, *id.* at 494–523, evaluates their constitutional stature as "traditional contours" of copyright, *id.* at 528–39, and suggests ways in which they could be resurrected without running afoul of international copyright protocols, *id.* at 545–68. The Ninth Circuit is also currently considering the constitutional question. *See* Amended Opening Brief for Appellants, Kahle v. Ashcroft, No. 04-17434 (9th Cir. Feb. 1, 2005)

whether we were better off without them and with little protest from scholars.[3] Today they play no role in earning copyright protection, and we are left only with the undemanding threshold requirements of originality and fixation.

In an era when print predominated, this low threshold was not particularly problematic. Mass distribution of information was an expensive process rarely undertaken by those with no plans to profit from their creativity. The law could therefore reasonably assume that authors who bothered to disseminate their writings wanted copyright's protection against unauthorized copying, without requiring them to further demonstrate their desire by filling out a form or jumping through any other regulatory hoop. In the print era, then, copyright could grant its exclusive property rights to everyone who met the two threshold requirements, confident that in the vast majority of cases those rights were fulfilling their constitutional role as an incentive to create and share expression with the public.[4]

It approaches cliché to observe that computers have changed everything. But computers have changed everything. Digital architecture has so drastically reduced the cost of creating and distributing expression that today we can all be authors *and* publishers, a development that empowers the individual and promises to enrich and democratize the content of our culture. And in some respects our new interconnectedness makes copyright more important than ever, because there are more authors who can take advantage of its incentivizing effects and earn a living from sharing their creativity with the rest of us.

Yet as we try to fit the digital peg of computer technology into the analog hole of copyright law, we have seen a host of seemingly intractable problems arise. Because copyright protection attaches the instant an original thought is expressed in fixed form, this new generation of authors never has a chance to affirmatively decide whether to invoke the law's protection for its creativity. It has no opportunity to decline copyright's kind invitation. The website of the U.S. Copyright Office has answers to over one hundred Frequently Asked Questions about copyright law—including such gems as "How do I protect my sighting of Elvis?" and "Can I register a diary I found

---

(seeking reversal of district court ruling that absence of formalities in digital era does not violate First Amendment and Intellectual Property Clause).

   3   Sprigman, *supra* note 2, at 488 & n.13.

   4   U.S. CONST. art. I, § 8 (giving Congress power to "promote the Progress of Science and useful Arts, by securing for limited Times to Authors and Inventors the exclusive Right to their respective Writings and Discoveries").

in my grandmother's attic?"[5]—but tells us nothing about how to voluntarily forgo copyright protection. It provides dozens of different forms for registering a work,[6] but not one for donating it to the public domain. Copyright is like a well meaning but ultimately bothersome friend, eager to help but nearly impossible to get rid of.

And the public is poorer for it. Because copyright protection arises by operation of law, authors who have no intention of disseminating their works gain the protection of a legal regime that assumes they will do so. At the other end of the spectrum, millions of works whose authors want to share their creativity but have no interest in copyright's exclusive rights see their works automatically propertized, even though they would be willing to give them away for free. In short, copyright's failure to adjust at a structural level to the development of digital architecture hinders wholly unobjectionable and socially enriching uses of creative material.

At the same time that copyright limits technology's potential, technology undermines copyright's goals. Copyright exists not to enrich authors, but to enrich the culture. Its exclusive rights are merely a means to this end, an incentive for the creation and public dissemination of creative expression.[7] Defining the rights thus involves a careful balancing act. Too much private control over copying and dissemination of creative works denies the public access to valuable goods and to the raw materials needed for further innovation. Too little control results in an insufficient impetus to produce the works in the first place.[8]

Copyright law uses a number of public privileges to strike this balance between private incentive and public benefit: a finite term of protection,[9] limited control over the aftermarket,[10] the immediate

---

5   U.S. Copyright Office, Frequently Asked Questions About Copyright, http://www.copyright.gov/help/faq (last visited Sept. 8, 2005).

6   U.S. Copyright Office, U.S. Copyright Office Forms, http://www.copyright.gov/forms (last visited Sept. 8, 2005).

7   Twentieth Century Music Corp. v. Aiken, 422 U.S. 151, 156 (1975) ("[P]rivate motivation must ultimately serve the cause of promoting broad public availability of literature, music, and the other arts."); United States v. Paramount Pictures, Inc., 334 U.S. 131, 158 (1948) ("[R]eward to the author or artist serves to induce release to the public of the products of his creative genius."); Fox Film Corp. v. Doyal, 286 U.S. 123, 127 (1932) ("The sole interest of the United States and the primary object in conferring the [copyright] monopoly lie in the general benefits derived by the public from the labors of authors.").

8   Brief for Law Professors et al. as Amici Curiae Supporting Petitioners at 6, Metro-Goldwyn-Mayer Studios Inc. v. Grokster, Ltd., 125 S. Ct. 2764 (2005) (No. 04-480), 2005 WL 176448, at *6.

9   17 U.S.C. § 302 (2000).

dedication of a work's ideas to the public domain,[11] and a fair use defense for infringements that are unlikely to harm the work's market or that provide some greater social benefit.[12] Each of these important safeguards, however, is merely a legal entitlement that presumes certain "architectural" conditions—i.e., makes certain assumptions about the state of the physical world.[13] And digital technology threatens the existence of these conditions. Copyright's limited duration is all well and good, but copy protection software does not expire. Copyright's first sale doctrine gives me the right to share my copy of a work with you, but streaming video never provides me with a copy to share. The ideas buried in computer source code may as well be owned by the programmer, for the public never sees them. And fair use is fairly useless when access restrictions refuse entrance to the fair user and pirate alike.

Nowhere are the challenges that digital architecture poses for copyright more apparent than in the regulation of software. Software's fight to be included under copyright's umbrella is over; software has won that legal battle.[14] But the war to control access to and use of computer programs continues on the architectural front, where a quirk of technology allows software developers to hide from the public the very expression that earns their products copyright protection in the first place. Copyright's longstanding assumption that those who create expressive works will share them with the public is thus particularly inapplicable to software.

To add insult to injury, the unsuccessful marriage of copyright and technology has given birth to a mischievous offspring, which I have elsewhere dubbed "technolegical" regulation—i.e., the *legis*lative regulation of *techno*logical behavior.[15] For example, the legal entitlements of trade secret law reinforce software's technological ability to keep its creative expression from reaching the public. The Digital Millennium Copyright Act puts its considerable legal weight behind the technological access restrictions and copy protection that copy-

---

10   *Id.* § 109(a).

11   *Id.* § 102(b).

12   *Id.* § 107.

13   This use of "architectural" originates in Lawrence Lessig, *The Law of the Horse: What Cyberlaw Might Teach*, 113 HARV. L. REV. 501, 507 (1999), and describes a behavioral constraint imposed by "the physical world as we find it, even if 'as we find it' is simply how it has already been made."

14   *See infra* notes 28–32 and accompanying text.

15   James Gibson, *Re-Reifying Data*, 80 NOTRE DAME L. REV. 163, 167 (2004). If you feel compelled to say this word out loud, resist the compulsion. If resistance proves futile, however, I suggest pronouncing it like "technological" except with "ledge" in the middle rather than "lodge"— i.e., tek-no-LEDGE-i-cull.

right owners use to control their digital goods.[16] The Federal Com-
munication Commission imposed a legal obligation on electronics
manufacturers to incorporate a "broadcast flag" into television receiv-
ers, so that antipiracy technologies built into digital broadcasts will be
more effective.[17] To the extent that these legal measures simply rein-
force technologies that promote copyright's goals, they are unobjec-
tionable, even praiseworthy. But when they overreach—as they often
do—they cause more harm than good.

This is not to say that regulating technology in the service of copy-
right law is always a bad thing. To the contrary, "technolegical" regu-
lation is necessary to give meaning to the balance between private
incentive and public benefit in the information age. If digital archi-
tecture goes wholly unregulated, then this balance will be cast aside in
favor of a technological Wild West in which the better technologist
prevails.[18] We will move from a world of legal haves and have-nots to a
world of technological cans and cannots, with no assurance that this
new world will be any more attentive to copyright's goals than the old
one.

Healing the wounds that digital architecture and copyright inflict
on each other will therefore require us to strike a second balance,
between authorial innovation and technological innovation. We need
to ensure both that digital architecture fulfills its promise without de-
stroying a copyright system that has served us well for almost three
hundred years and that technological quirks do not serve to expand
copyright beyond its proper place. And to strike this second balance,
technolegical regulation is vital.

In this Article, I explore how best to strike this balance and sug-
gest a purposely retrograde approach that solves our twenty-first cen-
tury problems using the nineteenth century's technolegical tools:
copyright's formalities. Resurrecting publication, notice, registration,
and deposit as threshold requirements for copyright protection pre-
vents authors and publishers from achieving technologically what they
do not merit legally, while at the same time ensuring that copyright
does not apply in contexts where it is neither necessary nor useful.
What's more, the formalities are tried-and-true copyright concepts;
they would not impede further innovation in information architec-
ture, can easily adjust to changes in legal entitlements, and are rela-

---

16   17 U.S.C. § 1201(a).

17   47 C.F.R. § 73.8000–.9009 (2004). A federal court of appeals recently ruled
that the FCC lacked the authority to promulgate the broadcast flag regulation. Am.
Library Ass'n v. FCC, 406 F.3d 689 (D.C. Cir. 2005).

18   Gibson, *supra* note 15, at 169–70.

tively impervious to the whims of technology and the cycle of obsolescence that could derail a more high-tech solution. In the end, then, the proverbial layperson's expectations have a lot to offer. Reviving the formalities of the past will preserve copyright for the future.

Part I of this Article describes one unwelcome effect of digital architecture, known as closed code, and explains the serious problems that it poses for copyright law. Part II explores possible solutions to the closed-code problem, including reverse engineering and open-source licensing, and concludes that to the extent that these high-tech approaches have something to contribute, it is their implicit recognition of the value of the low-tech tradition of copyright formalities. Finally, Part III paints a broad-brush picture of other ways in which reviving these seemingly archaic remnants of the print era can solve pervasive copyright problems—i.e., increased opportunities for censorship, an insufficiently democratized culture, and the threat of technological hegemony—that digital architecture has caused.

I. SOFTWARE, SOURCE CODE, AND SECRECY

Most copyrighted works reveal their expression as a matter of course. Indeed, the whole point of copyright protection is to allow authors to market their expression to the whole world without fear of unauthorized appropriation. An author who stashes the next great American novel in a locked drawer not only makes no money, but also lets the copyright clock tick down towards its ultimate expiration.

Software, however, is unique among all forms of copyrighted goods in that its expressive content can be kept secret from the public without impairing its marketability. In fact, software can simultaneously claim the protection of both copyright law and trade secret law. Copyright protects this secret content because it is expressive, and trade secret protects it because it is secret. And this dual protection results not from any conscious policy decision on the part of lawmakers, but from a quirk of digital architecture. In the following discussion, I explain this curious state of affairs and show why the public at large (and to some extent software developers as well) would benefit from a regulatory regime that protects software under copyright law or trade secret law, but not both.

A. Object Code, Source Code, and a Technological Quirk

To understand how and why both copyright and trade secret law protect software, one must first understand how computers work. At the core of every computer is a vast collection of on/off switches, known as circuits. By using these circuits in combination with one

another, a computer is able to perform millions of calculations per second and thus execute the operations that make it so useful. Moreover, in addition to providing raw processing speed, complex circuitry allows a computer to perform a wide variety of tasks without any reconfiguration of its physical structure. In other words, a computer can act as a word processor, and then a web browser, and then an accounting ledger, and then a jukebox, all without altering its nuts and bolts—its "hardware"—in any way.

How can a single machine be so versatile? In a word, programming. Users can program a computer—tell it to perform a given function—by feeding it a set of instructions regarding which circuits to turn on, and which to turn off, and when. When users want a computer to be a word processor, they feed it one set of programming instructions. When they want it to be a web browser, they feed it a different set. These sets of instructions are known as software.[19] At its most basic level, then, software comprises a related series of on/off commands to the computer's circuits that tell the computer what function to perform at a given time. Because software's commands to a circuit can relate only one of two messages ("on" or "off"), the language in which commands are expressed is binary—i.e., it has just two characters: the number one, which represents "on," and the number zero, which represents "off." These long strings of ones and zeros that a computer executes are known as a program's "object code."

In the 1950s, when programmers sat down to write the software for the first programmable computers, they wrote their programs in object code.[20] But they soon found it very burdensome to compose any but the most basic of programs in this simplistic language. So in 1959, IBM developed FORTRAN, a computer language in which a programmer could write much more easily.[21] Other programming languages soon followed, such as BASIC, Pascal, and C. Computer code written in one of these languages, known as "source code," is still mostly unintelligible to the untrained eye, but it resembles English much more than it resembles object code and is accordingly easier to write and comprehend. For example, in object code, a program that commands the computer to print the phrase "Hello, world!" would look like a long, unrevealing string of hundreds or even thousands of

---

19 In some contexts, the term "software" means something different from "computer program," see 1 DAVID BENDER, COMPUTER LAW § 2.06[1], at 2-115 (2005), but the differences are not relevant here. I will therefore use the two terms interchangeably.

20 Peter S. Menell, *Envisioning Copyright Law's Digital Future*, 46 N.Y.L. SCH. L. REV. 63, 71 (2003).

21 *Id.* at 71–72.

ones and zeros, but in the source code of the popular C++ programming language it looks like this:

```
#include <iostream>
int main()
{ std::cout << "Hello, world!\n";
}22
```

Although writing programs in source code instead of object code makes a programmer's job much easier, it presents its own problem. Computers only understand the on/off commands of object code. The high-level language of source code looks like gibberish to a computer, just as the ones and zeros of object code look like gibberish to you and me. Special programs are therefore needed to translate source code into object code so the computer can execute it.[23]

Here then we see the technological quirk: the hard, creative work of programming is done in source code, but the final, usable product exists in object code. As a result, the norm in the software industry is to market products in object code format only. The source code remains with the developer, hidden from the public. This "closed code"[24] is fine with the average consumer. Purchasers of WordPerfect or TurboTax don't care what the source code looks like; they care about how the program functions in its executable object code format. But as we will now see, a software developer's ability to keep its source code secret has major implications for the way in which intellectual property law protects its products.

B.   Copyright, Trade Secret, and a Legal Quirk

As the emergence of high-level programming languages and the increasing popularity of personal computers made software a major industry, programmers became concerned about an architectural feature of computer programs that threatened to hinder their commodification: their   susceptibility   to   cheap,   unauthorized reproduction. Because computers excel at processing—and thus reproducing—ones and zeros, any computer user with a copy of a program's object code could easily make and distribute countless

---

22   JOHN R. HUBBARD, PROGRAMMING WITH C++, at 2 (2d ed. 2000).

23   Converting source code into a form that a computer can actually process is a two-step process. First a compiler translates the source code into an intermediate-level language known as assembly code. Then an assembler translates the assembly code into object code. Glenn J. MacGrady, *Protection of Computer Software—An Update and Practical Synthesis*, 20 HOUS. L. REV. 1033, 1036 (1983).

24   The use of the term "closed code" to refer to this technological quirk originated in Lawrence Lessig, *The Limits in Open Code: Regulatory Standards and the Future of the Net*, 14 BERKELEY TECH. L.J. 759, 765 (1999).

additional copies without any permission from or payment to the programmer. It was therefore no surprise that soon after the introduction of FORTRAN and other high-level programming languages, software developers began to seek legal protection against unauthorized copying of their programs. Both copyright law and trade secret law came to their rescue.

In some ways, copyright law is a natural fit for software. Computer code consists of letters and numbers, like the "Writings" that the Constitution refers to in the Intellectual Property Clause,[25] the "books" that were the subject of the first copyright statute Congress ever passed,[26] and the "literary works" covered by the current copyright act.[27] On the other hand, the letters and numbers of code are not expressive in the traditional copyright sense. Books and poems talk to their readers; they stimulate thoughts in our brains. In contrast, software communicates its "meaning" not through literary discourse, but through commanding a machine to operate in a certain way. Software "talks" to a computer in the sense that railroad tracks "talk" to trains. And that type of conversation, although admittedly valuable, is not one in which copyright law has traditionally taken an interest.

Nevertheless, despite a heated debate about whether copyright *should* cover software,[28] today copyright clearly *does* cover any computer program that meets the statute's familiar threshold requirements: fixation and originality.[29] Fixation is easy enough; a programmer fixes a program simply by recording it in some permanent medium—paper, a CD-ROM, a computer's hard drive, and so forth. And satisfying the originality requirement merely requires "independent creation plus a modicum of creativity,"[30] a standard easily met by the subjective choices that a programmer makes in writing source code of any complexity.

The ship has also sailed on whether the copyright in a computer program covers both its source code and its object code; the case law

---

25   U.S. CONST. art. I, § 8.

26   Act of May 31, 1790, ch. 15, § 1, 1 Stat. 124, 124.

27   17 U.S.C. § 101 (2000) (defining "literary work").

28   For cogent arguments against indiscriminately applying copyright law to computer programs, see, for example, Peter S. Menell, *Tailoring Legal Protection for Computer Software*, 39 STAN. L. REV. 1329 (1987); Pamela Samuelson, *CONTU Revisited: The Case Against Copyright Protection for Computer Programs in Machine-Readable Form*, 33 DUKE L.J. 663 (1984).

29   *See, e.g.*, Apple Computer, Inc. v. Franklin Computer Corp., 714 F.2d 1240, 1246–49 (3d Cir. 1983).

30   Feist Publ'ns, Inc. v. Rural Tel. Serv. Co., 499 U.S. 340, 346 (1991).

makes it clear that any creativity in source code provides copyright protection for the corresponding object code as well,[31] notwithstanding creditable arguments in favor of a contrary conclusion.[32] This is an important point, because it is object code that really needs copyright's protection against the unlicensed copyist. Recall the technological quirk identified above: a programmer's creative efforts are manifest in the program's source code, but the program's purchaser only encounters its functional object code. Because only object code is usually made available to the public, only object code is susceptible to unauthorized copying and distribution.

For source code, the more important protection is found in trade secret law, a regulatory regime predicated on maintaining the secrecy of valuable commercial information, such as formulas or manufacturing processes.[33] It is easy to understand how source code fits into this mold; it is, in essence, a formula or industrial process used to manufacture a functional computer program, namely software in its object code incarnation. Therefore, as long as a software developer takes reasonable steps to keep source code secret from the prying eyes of competitors and the public, trade secret law provides legal remedies for its unauthorized and improper appropriation.[34]

The combination of these seemingly reasonable applications of the law produces an intellectual property paradox. Copyright law—the regulatory regime meant to preserve and promote creativity—provides the main protection for object code, which on its face manifests no creative expression, but which does have a functional application. Meanwhile, trade secret law—which has traditionally safeguarded functional, industrial know-how—provides the main protection for source code, in which one *can* find creative expression.

This paradox exists only for software. The technological quirk that makes the paradox possible does not apply to any other kind of copyrighted work, because copyrighted works generally wear their creativity on their sleeve. One cannot sell a book or a painting or a film without disclosing to the public the creative expression that made the work copyrightable in the first place—the words on the page, the

---

31   Computer Assocs. Int'l, Inc. v. Altai, Inc., 982 F.2d 693, 702 (2d Cir. 1992) ("It is now well settled that the literal elements of computer programs, i.e., their source and object codes, are the subject of copyright protection.").

32   *E.g.*, Samuelson, *supra* note 28, at 745.

33   UNIF. TRADE SECRETS ACT § 1(4) (1985), *available at* http://www.law.upenn. edu/bll/ulc/fnact99/1980s/utsa85.pdf (defining "trade secret").

34   Menell, *supra* note 20, at 74.

brush strokes on canvas, the images on celluloid.[35] And this conver-
gence of disclosure and marketability is no coincidence. Rather, it is
part and parcel of copyright law, which exists not to reward authors
for their creative efforts but to encourage the public dissemination of
creative expression.[36] The ability of authors to profit from their crea-
tive efforts is purposefully tied to their willingness to market their
works to the public while their copyright lasts.

In other words, copyright law intends to force a choice on an
author: either keep the entire work secret and forgo profits, or market
it and allow the public to see the protected expression. Only with
software may authors have their cake and eat it too.[37] The technologi-
cal quirk thus leads to a legal quirk: copyright law makes possible the
successful marketing of a computer program without requiring the
programmer to reveal that which earned it copyright protection. As
the following Part will show, these two quirks produce unwelcome
consequences for copyright policy.

C. Consequences of Closed Code

Over the last twenty-five years, software developers have fought
and won the battle for copyright protection of software. Rather than
launch a new salvo in an old battle, then, I will take software's legal
entitlement as my starting point and inquire into how (as opposed to
whether) copyright should protect software. How does software's
unique architectural characteristic—closed code—fit into copyright's
mechanisms for balancing private incentive against public interest?
As we will see, granting copyright protection to software without re-
quiring public disclosure of its source code runs contrary to copy-
right's purposes in a number of ways. It contravenes the quid pro quo
inherent in copyright law. It impoverishes the public domain in ways
that hurt both consumers and innovators. It chills the exercise of priv-

---

35   This does not mean that the entire creative process is laid bare; we can't neces-
sarily tell how Hemingway went about drafting his stories simply by reading them, or
what studio tricks the Beatles used to produce a certain sonic effect in their music
simply by listening to it. But in these examples, much of what makes the work deserv-
ing of copyright protection can be readily perceived. A musical performance, for
example, provides the listener with a great deal of insight regarding the written score
it embodies, even if the written work itself is kept secret. In contrast, a computer
program in executable object code form effectively conceals those aspects of its
source code that might be called creative. *See infra* Part II.A.

36   *See* cases cited *supra* note 7.

37   Samuelson, *supra* note 28, at 710; Mathias Strasser, *A New Paradigm in Intellec-
tual Property Law? The Case Against Open Sources*, 2001 STAN. TECH. L. REV. 4, ¶ 54,
http://stlr.stanford.edu/STLR/Articles/01_STLR_4/article_pdf.pdf.

ileges that copyright law reserves to the public. It encourages hidden regulation of free expression. And it presents a host of practical problems for the efficient administration of intellectual property entitlements.

1. Impoverishing the Public Domain

The Constitution tells us that copyright's ultimate goal is the promotion of "Progress of Science and useful Arts."[38] The exclusive rights that copyright law grants to private persons exist only to serve this ultimate goal, and they are accordingly subject to a number of safeguards designed to further it. The most well known safeguard is the finite duration of copyright: copyright rights expire after a set term of years, giving the author time to secure a return on his or her investment in creating the work, and afterwards donating it to the public domain.[39] But other safeguards—often of greater practical importance—apportion certain aspects of a work to public use even before the copyright expires.[40] Copyright's treatment of software undermines these safeguards and thus impoverishes the public domain.

Consider one such safeguard, known as the idea/expression dichotomy. Copyright protects a work of original authorship upon its fixation in a tangible form, but not all aspects of the work fall within that protection. Rather, the *ideas* within the work pass immediately into the public domain—"even if they are highly original, extraordinarily ingenious, and exceedingly valuable, and even if they took the author many years of effort to develop."[41] Copyright law only concerns itself with unauthorized use of the author's *expression*, the particular way in which the author chooses to set forth the work's ideas.[42]

The idea/expression dichotomy is a time-honored part of copyright's balance between private incentive and public benefit.[43] It al-

---

38   U.S. CONST. art. I, § 8.

39   *See* 17 U.S.C. §§ 301–305 (2000 & Supp. II 2002) (providing rules for calculating duration of copyright rights).

40   I am using the term "public domain" in a "high granularity" sense; I include in the public domain not just complete works whose copyrights have expired, but also ideas and other uncopyrightable aspects of a work, whose copyrights have not yet expired. And, as will become apparent, I view the public domain as substantive, not formal; regardless of legal entitlements, an idea or work is not in the public domain unless the public can actually access it. *See* James Boyle, *The Opposite of Property?*, 66 LAW & CONTEMP. PROBS. 1, 30 (2003).

41   Samuelson, *supra* note 28, at 707.

42   17 U.S.C. § 102(b) (2000).

43   As the Supreme Court observed over one hundred years ago, "The very object of publishing a book on science or the useful arts is to communicate to the world the useful knowledge which it contains. But this object would be frustrated if the knowl-

lows an author to demand compensation from anyone who wants to replicate the particular words (or musical notes, brush strokes, etc.) he or she has chosen to use, but permits others to freely adapt the more abstract aspects of the work in new creative endeavors of their own. This is a worthwhile tradeoff: copyright grants authors exclusive rights that they would not otherwise enjoy, and in exchange allows the rest of us free and immediate use of any ideas found in the authors' works as raw materials for our own innovation. Thus is the Progress of Science and useful Arts encouraged. The idea/expression dichotomy also helps mediate the tension between free speech principles and copyright's regulation of expression; it promotes democratic discourse by allowing others to communicate and discuss the ideas and facts in an author's work without incurring liability.[44] This safeguard has accordingly garnered support not only from those who oppose expanded copyright[45] but also those who support it.[46]

For the idea/expression dichotomy to work, however, the copyrighted content of a work must be made available. One's *legal* right to develop a work's ideas is useless if one has no *architectural* access to those ideas in the first place. As already discussed, this is not a problem for most copyrighted works. Authors cannot typically profit from the expression they create without revealing it—and the ideas it contains—to the public. It's hard to sell a book without giving the reader access to both its words and its ideas. Of course, an author could choose not to sell the book at all, keeping it instead in a locked drawer. But in such an instance neither the author nor the public receives the benefit of copyright's bargain—no profits for the author, no new public domain materials for the public. Authors therefore tend to market their expression and accept the loss of control over their ideas.

---

edge could not be used without incurring the guilt of piracy of the book." Baker v. Selden, 101 U.S. 99, 103 (1879).

44   Eldred v. Ashcroft, 537 U.S. 186, 219–20 (2003); Harper & Row, Publishers, Inc. v. Nation Enters., 471 U.S. 539, 556 (1985).

45   *E.g.*, Jessica Litman, *The Public Domain*, 39 EMORY L.J. 965 (1990) [hereinafter Litman, *The Public Domain*] (arguing that ideas and other public domain aspects of expressive works are a vital part of copyright's viability). Litman would, I think, happily admit to being an opponent of expansive copyright. *See, e.g.*, Jessica Litman, *Revising Copyright Law for the Information Age*, 75 OR. L. REV. 19, 40–41 (1996) [hereinafter Litman, *Revising Copyright*] (suggesting "recasting copyright as an exclusive right of commercial exploitation").

46   *E.g.*, Christopher S. Yoo, *Copyright and Product Differentiation*, 79 N.Y.U. L. REV. 212 (2004) (arguing for broader copyright scope and narrower fair use doctrine but recognizing that the idea/expression dichotomy enables optimal market entry).

With software, however, we have a copyrighted work whose unique architecture allows its author to profit without revealing either its creative expression or its ideas to the purchaser. The software developer thus receives the benefit of copyright protection—the right to sue anyone who engages in unauthorized reproduction or adaptation of the program—without conferring the corresponding benefit on the rest of us.[47] Whatever ideas exist in the creative source code of a computer program remain with the developer; all the public encounters is an impenetrable and unrevealing string of ones and zeroes.

This architectural singularity is particularly troublesome because software's functional nature places not just its ideas but also many of its other constituent parts outside copyright's protection. For example, programmers may try to write a program using as few lines of code as possible, so the computer will not have to work as hard to perform the desired task. This desire for efficiency is admirable from a functional standpoint, but it limits the expressive qualities of the program. Courts have thus consistently held that those aspects of software "dictated by considerations of efficiency" are not within copyright's protection.[48] The same is true for programming decisions that result not from originality on the programmer's part, but from a need to conform to external constraints, such as hardware compatibility, interoperability requirements, manufacturers' design standards, or the demands of the target audience.[49] Other facets of the work, such as any facts it contains, are also free for the taking.[50]

In theory, then, software developers donate all of these unprotected elements of software to the public domain in exchange for their copyright in the remaining elements. But in practice, the source code remains hidden from public view. The irony is striking: the law tells us that software comprises more public domain elements than other copyrighted works, but the architecture of closed code protects software more thoroughly than any of its copyrighted counterparts. Third party programmers are unable to borrow from or even see the unprotected code and thus remain ignorant of the lessons in efficiency and compatibility that the law entitles them to learn. Software developers get the benefit of copyright protection without paying the prescribed price to the public.

---

47 *See* LAWRENCE LESSIG, THE FUTURE OF IDEAS 252–53 (2001); Samuelson, *supra* note 28, at 705–06.

48 Computer Assocs. Int'l, Inc. v. Altai, Inc., 982 F.2d 693, 707 (2d Cir. 1992).

49 *Id.* at 709–10.

50 Feist Publ'ns, Inc. v. Rural Tel. Serv. Co., 499 U.S. 340, 344 (1991).

One might argue that copyright has it wrong here—e.g., that the law should more thoroughly protect code and donate less of software's elements to the public domain. The case law on this issue, however, is clearly here to stay[51] and thus cannot be ignored in balancing private incentive against public benefit. Moreover, it is not even clear that software developers would benefit from commodification of the ideas and other public domain aspects of a program. Copyright places ideas into the public domain to perpetuate a self-sustaining state of enrichment in the realm of creative expression; when today's innovators have access to yesterday's ideas, they can build on what came before without reinventing the wheel.[52] For programmers, then, the widespread availability of ideas and other public domain aspects of software would lower development costs, perhaps significantly. Developers' tendency to conceal source code may accordingly result from a kind of prisoner's dilemma; they might be happy to share the ideas in their code if only they could trust their competitors to do the same.[53]

There are two obvious responses to this point. First, some public domain aspects of source code may be evident in the way the program behaves when it is up and running—i.e., in its onscreen output, how it processes data, and so forth.[54] But this is only the case with regard to certain programs and certain characteristics; studying a program's dynamic structure to ascertain its static structure is not an across-the-board solution to the closed-code problem.[55] Second, revealing ideas and other public domain aspects of a program would necessarily entail revealing protected expression as well, and would thus present an

---

51   *Altai*'s holdings have become the prevailing standard. 4 MELVILLE B. NIMMER & DAVID NIMMER, COPYRIGHT § 13.03[F], at 13-125 to -127 (2004).

52   *See* Litman, *The Public Domain, supra* note 45, at 968.

53   I am indebted to Doug Lichtman for this point. Unlike commercial software developers, Doug is happy to give others access to his ideas.

54   *See, e.g.,* Anthony L. Clapes, *Confessions of an Amicus Curiae: Technophobia, Law, and Creativity in the Digital Arts,* 19 U. DAYTON L. REV. 903, 943 (1994); Arthur R. Miller, *Copyright Protection for Computer Programs, Databases, and Computer-Generated Works: Is Anything New Since CONTU?,* 106 HARV. L. REV. 977, 1029 (1993); *see also* Computer Assocs. Int'l, Inc. v. Altai, Inc., 775 F. Supp. 544, 559 (E.D.N.Y. 1991) (distinguishing between software's "static structure"—i.e., its code—and its "dynamic structure"—i.e., the way it behaves when it is up and running), *aff'd in part and vacated in part,* 982 F.2d 693.

55   *See Altai,* 775 F. Supp. at 559 ("The static structure and dynamic structure of a program can be quite different; indeed from dealing with the behavior of a program, i.e., operating it, one can tell virtually nothing about its text."); Julie E. Cohen, *Reverse Engineering and the Rise of Electronic Vigilantism,* 68 S. CAL. L. REV. 1091, 1122 n.163 (1995) (criticizing Clapes's distinction between access to static code and access to dynamic operation).

increased risk of piracy. I address this concern in the next Part, where I discuss the costs of hidden expression.

In short, software developers should not be able to use a quirk of digital architecture to withhold from the public domain those aspects of a program that copyright apportions to it; this is getting something for nothing from copyright law. As we will see, we could correct this error by resurrecting copyright's formalities. But first, let's examine some other casualties of closed code.

2. The Costs of Hidden Expression

The foregoing discussion focused on the disadvantages that arise from denying access to the public domain aspects of computer code— i.e., the ideas and other legally unprotected elements of a program. There are, however, other disadvantages to closed code, which result from concealment of the legally protected expression rather than from impoverishment of the public domain. Recall that it is not only the public that cannot see the source code that earns a computer program its copyright protection; even those who purchase lawfully made copies of programs do not get access. This unavailability of copyrighted expression not only harms the public at large, but may also disserve the software industry itself.

First, consider the effect on the public. In a world without copyright formalities, we are all expected to recognize copyrighted goods on our own. There is no government registry to consult, no compulsory notice to alert us to a work's protected status. So to decide whether we may copy or modify a writing or other copyrighted work without seeking anyone's consent, we must each examine the expression that it contains and determine whether it satisfies copyright's originality requirement.

Making such a legal determination obviously presupposes architectural access to the work's expression in the first place. With proprietary software, however, the public—indeed, even the purchaser—has no access to the protected expression. It is therefore impossible for the public to know its rights with regard to a given program. We are left to assume that because *some* software *can be* protected by copyright, *all* software *is* protected by copyright. Indeed, this is the operative assumption at the U.S. Copyright Office, which will register computer programs based on examination of object code alone, even as it recognizes that its lack of access to source code precludes a valid copyrightability determination.[56]

---

56   *See* 37 C.F.R. § 202.20(c)(2)(vii)(B) (2004) (stating that software registration application accompanied only by program's object code will be registered under "rule

Of course, there are certain goods that by their very nature and complexity bespeak the requisite modicum (or more) of creative expression. If you know that a new film lasts for two hours or that an album includes fourteen never-before-performed songs, then you may not need to see the film or hear the album to reasonably conclude that it contains sufficient original expression to merit copyright protection. But the same cannot be said of software, at least not as a general matter; as discussed above, even complex programs are composed of largely uncopyrightable elements,[57] and many programs are not particularly complex. What we are left with, then, is a copyright regime that not only fails to ensure the availability of software's public domain aspects, but also precludes ex ante evaluation of its copyright status.

Next, consider again the effect of closed code on those who develop software. Receiving the benefit of copyright protection without revealing a work's expression sounds like a good deal for developers—and in the end it may be a good deal—but it imposes some costs on them as well. Foremost among these is that the development of value-added uses for a proprietary program is limited by the economics of the firm.[58] In other words, because no one outside the company that develops the program has access to its source code, only the company's own employees can readily conceive of value-added modifications.[59] Outside parties with an interest in licensing the right to develop an improvement are shooting in the dark; to propose an improvement, they must incur significant transaction costs—e.g., persuading and presumably paying the original developer to reveal its secrets—without having first seen the code, and thus without knowing whether their contemplated improvements are likely to succeed.

In contrast, if source code were available to the public at large, programmers everywhere could readily envision and propose innovative improvements to it. Of course, a third party could not actually prepare and market any improvement without a license from the orig-

---

of doubt" despite recognition that "no determination has been made concerning the existence of copyrightable authorship").

57   *Altai*, 982 F.2d at 707–10.

58   For the seminal explanation of the economics of the firm, see Ronald H. Coase, *The Nature of the Firm*, 4 ECONOMICA 386 (1937). *See also* Dan L. Burk, *Intellectual Property and the Firm*, 71 U. CHI. L. REV. 3 (2004) (examining relationship between economics of the firm and intellectual property).

59   *See* A. Samuel Oddi, *An Uneasier Case for Copyright than for Patent Protection of Computer Programs*, 72 NEB. L. REV. 351, 426 (1993) ("The common industry practice of distributing programs only in object code may be having a far more inhibitory effect on program improvement than may be attributed to patents on program inventions.").

inal developer, because copyright law prohibits the development of unauthorized derivative works.[60] But a prospective collaborator would be able to assess its improvement's potential much earlier in its development process if it had access to source code from the start. Closed code thus imposes costs on both the public and programmers, who not only are forced to reinvent the wheel rather than using public domain code, but who also cannot conceive of and propose licensed collaborations without first incurring significant and unnecessary transaction costs. Both licensor-developers and licensee-collaborators could benefit from a different approach.

The obvious drawback to making source code's expression available to the public—and particularly to prospective collaborators in the software industry—is the increased threat of infringement. Revealing source code clearly invites some copying that could not have taken place in a closed-code regime. One must remember, however, that copyists do not need access to source code to engage in copyright law's main target (and the software industry's greatest threat): outright piracy. To make and distribute a verbatim copy of a program, one needs only the object code, which is available even in a closed-code world.[61]

So what new infringements would result from revealing source code? If the reverse engineering case law (discussed in more detail below) is any indication, those most interested in a program's source code rarely intend to create a directly competing product.[62] The few cases in which direct competition does result tend to involve more traditional manufactured products that happen to include a software component.[63] Infringement might also occur when someone tries to

---

60   17 U.S.C. § 106(2) (2000).

61   The exception is software-as-service enterprises, discussed below. *See infra* notes 164–66 and accompanying text. These firms currently do not make their source code or object code available and therefore might see a significant increase in infringement if they were forced to do so. But such enterprises also have little need to rely on copyright in the first place, because they do not market their goods in a readily copyable form. They are more likely to find that trade secret law satisfies their intellectual property needs and could thus opt out of formalities-based copyright. *See* Strasser, *supra* note 37, ¶ 41.

62   Pamela Samuelson & Suzanne Scotchmer, *The Law and Economics of Reverse Engineering*, 111 YALE L.J. 1575, 1613–14 & n.182 (2002).

63   *Id.* at 1613 n.182 (citing Alcatel USA, Inc. v. DGI Techs., Inc., 166 F.3d 772 (5th Cir. 1999) (defendant reverse engineered operating system software for plaintiff's phone switches so it could compete with plaintiff in market for expansion cards), and Secure Servs. Tech., Inc. v. Time & Space Processing, Inc., 722 F. Supp. 1354 (E.D. Va. 1989) (defendant reverse engineered plaintiff's "handshake protocol" software so it could compete with plaintiff in market for secure fax machines)).

tailor the source code to his or her own specific needs or to fix bugs—
a violation of copyright, to be sure, but one that is of less concern than
wholesale, verbatim copying. The biggest threat would be outright pi-
rates, those who alter source code to produce a competing program
whose object code incarnation looks different enough from the origi-
nal to escape detection as an infringing copy.[64] But if any of these
parties wanted to procure copyright protection so as to mass-market
their modifications, they would first have to reveal their own source
code, which would expose their infringement to public view and thus
invite litigation.

In the end, one cannot know the scale of infringement that
would result from making source code public.[65] Any viable solution
to the closed-code problem must simply remain attentive to this legiti-
mate concern and remain adaptable enough to accommodate it as it
plays out. As we will see, a new formalities regime possesses character-
istics—such as its opt-in nature and flexible access rules—that serve us
well in this regard.

3. Practical Disadvantages of Closed Code

The discussion so far has identified a gap between the goals that
copyright law espouses and the manner in which it approaches regula-
tion of software, and has suggested that public availability of copy-
righted source code would make copyright more faithful to its own
principles. The drawbacks of closed code are not all so grounded in
lofty principle, however. The inaccessibility of source code also brings
with it certain practical disadvantages in the administration of intellec-
tual property entitlements and the market for software.

For example, closed code may make it harder to access and thus
appropriate a hardworking programmer's creative expression, but it
also aids the dishonest software developer who does somehow manage
to get its hands on someone else's copyrighted source code. Because
that developer can market pirated copies without revealing their
source code, it can engage in piracy without significant risk of detec-

---

64   A few simple changes to source code can make two substantially similar pro-
grams appear different at the object code level. *See* OFFICE OF TECH. ASSESSMENT, U.S.
CONGRESS, FINDING A BALANCE: COMPUTER SOFTWARE, INTELLECTUAL PROPERTY, AND
THE CHALLENGE OF TECHNOLOGICAL CHANGE 147 (1992); STEERING COMM. FOR INTELL.
PROP. ISSUES IN SOFTWARE, NAT'L RESEARCH COUNCIL, INTELLECTUAL PROPERTY ISSUES
IN SOFTWARE 11, 78 (1991).

65   Lawrence Lessig, for one, thinks that "the remedy (no source code) is worse
than the harm"—i.e., less complete protection for software. LESSIG, *supra* note 47, at
253.

tion.[66] A few simple changes in source code can make two substantially similar programs look different at the object code level,[67] and the aggrieved programmer cannot tell whether the remaining operational similarities result from illegal copying unless it has and invests the considerable resources necessary to file suit and obtain discovery.

Another practical disadvantage of the closed-code system is that it decreases the chances of preserving programs for posterity and future adaptation. Because source code is not normally released to purchasers of software, customers are often very dependent on software developers for modifications to and updates of a program. If a developer goes out of business, its customers are left high and dry; the only part of the program available to them is the inherently unmodifiable object code.[68] Some mechanisms have emerged for handling this problem, such as source code escrow arrangements[69] and a federal bankruptcy provision that allows a purchaser to access source code upon the developer's bankruptcy if an executory contract so provides,[70] but they carry their own costs and their efficacy is unproven.

Even software firms that remain in business can fail to preserve their source code. Computer programs tend to lose their value quickly, and firms have little commercial incentive to maintain obsolete software. Yet there are uses to which enterprising parties could put source code even after its commercial utility as a mass-market product has withered away, such as adaptation by those who cannot afford the newer version of a program.[71] Two factors nonetheless discourage the preservation of old code. First, of course, copyright's restrictions on unauthorized reproduction mean that third party copying is a chancy undertaking, even for preservation purposes,[72] and these restrictions last much longer than necessary given software's

---

66   *Cf.* Jonathan Zittrain, *Normative Principles for Evaluating Free and Proprietary Software*, 71 U. CHI. L. REV. 265, 285–86 (2004) (discussing difficulty in determining whether proprietary software has stolen code from open-source software).

67   *See* sources cited *supra* note 64.

68   LESSIG, *supra* note 47, at 253.

69   *See* Jonathan L. Mezrich, *Source Code Escrow: An Exercise in Futility?*, 5 MARQ. INTELL. PROP. L. REV. 117 (2001).

70   11 U.S.C. § 365(n)(1) (2000); S. REP. NO. 100-505, at 9–10 (1988) (stating that § 365(n)(1) applies to right to acquire source code).

71   *See, e.g.*, Rob Pegoraro, *Creaky Operating Systems Show Their Age*, WASH. POST, Feb. 27, 2005, at F7 (discussing programs as recent as Microsoft's Windows 95 whose developers no longer support them and that do not work with recent applications and hardware).

72   Copying code for preservation or research might in some circumstances be fair use, but for reasons discussed *infra* Part II.A, fair use is an inadequate safeguard for this and other issues.

short shelf life.[73] Second, and more pertinent to the issue of closed code, even after copyright expires and a program theoretically enters the public domain its source code is often not architecturally available because it was never released to the public in the first place. At best, it remains locked in some cabinet in the developer's warehouse, and copyright law provides no incentive to release it.

Keeping source code secret also imposes costs on the administration of patent law, an intellectual property regime that has become increasingly friendly to protecting software and thus increasingly important to software development.[74] Before a patent issues, the U.S. Patent and Trademark Office conducts an inquiry to determine inter alia whether the innovation is novel and nonobvious in light of "prior art"—i.e., previous advances in the field. A patent examiner's ability to examine prior art is therefore crucial to deciding whether an innovation merits patent protection. But because software grew up as a creature of copyright and trade secret law, and was only recently embraced by patent, there is no centralized patent registry that provides a record of past software developments.[75] Nor is there any tradition in the software industry of publishing innovative developments in trade journals.[76] (This should come as no surprise; one reason to keep source code private and market only object code is to hide one's innovations from others in the industry.) So when the patent examiner is trying to determine whether a software invention is novel and nonobvious, he or she lacks the necessary resources on the state of prior art. Indeed, the lack of an established reservoir of prior art for computer programs was reportedly one of the reasons that the Patent and Trademark Office initially eschewed accepting them as patentable material.[77]

---

73 Copyright endures for seventy years after the death of the author, or at least ninety-five years when the author is an institution rather than an individual. 17 U.S.C. § 302 (2000).

74 Julie E. Cohen & Mark A. Lemley, *Patent Scope and Innovation in the Software Industry*, 89 CAL. L. REV. 1, 8–11 (2001) (describing Johnny-come-lately nature of software patents).

75 *See* LESSIG, *supra* note 47, at 260.

76 *See* Cohen, *supra* note 55, at 1178 (noting that the PTO traditionally examines only "previously issued patents and previous scholarly publications" when trying to determine prior art and that "much that qualifies as prior art lies outside" those areas); Stephen M. McJohn, *The Paradoxes of Free Software*, 9 GEO. MASON L. REV. 25, 50 (2000) (noting that "[c]omputer programming . . . has had much less systematic archiving of knowledge" than other industries and that much of the knowledge is "in informal form" or "intentionally kept out of the public domain").

77 Cohen & Lemley, *supra* note 74, at 8–9.

Closed code also impedes competition in the software market. Because source code is hidden, software is an "experience good"—i.e., even a sophisticated programmer cannot know what a program does until he or she takes it home, loads it up, and tries it out. If source code were visible, however, software would assume some characteristics of a "search good"—i.e., a product whose features could be ascertained without first using it. Search goods promote efficiency: if a purchaser can obtain information about a product early in the transaction, his or her search costs are lower and the market for that class of product becomes more competitive. Of course, one would have to be an experienced programmer to evaluate programs based on their source code, but increased efficiency for programmers alone would still be a welcome development, and some programmers would share their observations with the rest of us.[78]

There are other practical disadvantages to closed code. In certain circumstances, closed code may make it more difficult to identify and remedy a program's vulnerability to malicious viruses.[79] The unavailability of source code has even been cited as a defense against secondary liability in the online copyright wars, with file-sharing defendant Grokster claiming that it lacked the power to filter copyrighted works from its network "because it was a mere licensee without access to the underlying 'source code' for the peer-to-peer software that is the backbone of its system."[80] In short, from piracy to preservation to patent, closed code causes a host of practical problems that underscore the need for regulatory reform.

4. Free Expression and Freeing Expression

Finally, keeping source code hidden has subtle but pervasive antidemocratic effects. Code is part engineering and part speech. Indeed, if code had no expressive qualities, it would not qualify for copyright protection in the first place. Therefore, legal regulation of computer code raises First Amendment concerns and brings to the fore related issues of autonomy, transparency, and accountability—

---

78   *Cf.* Strasser, *supra* note 37, ¶ 105 (making the analogy that "[m]ost people do not understand the normative implications of complex areas of law" but "because there are lawyers and the media, who follow legislative and judicial developments and inform the public of important issues, they are able to make informed decisions").

79   *See* Neal Kumar Katyal, *Digital Architecture as Crime Control,* 112 YALE L.J. 2261, 2267 (2003) (arguing that publicizing code for operating systems makes it less vulnerable but publicizing code for "more specialized applications with few users" may have the opposite effect).

80   Petition for Writ of Certiorari at 7 n.4, Metro-Goldwyn-Mayer Studios Inc. v. Grokster, Ltd., 125 S. Ct. 2764 (2005) (No. 04-480), 2004 WL 2289200.

although the degree to which formal First Amendment protection should apply to code is a matter of much debate.[81] Closed code is particularly susceptible to hidden government regulation simply because programming that the government mandates (or inserts on its own) is hidden from users' view.[82]

This is not an abstract issue, of concern only to law professors and others with too much time on their hands. Computers increasingly govern our interactions with one another and thus influence our daily lives. The code through which digital communication takes place is accordingly rife with regulatory potential. Fortunately, bits and bytes are blind. The Internet's architecture treats the ones and zeros of a pornographic image the same as the ones and zeros of a political tract or party invitation; any filtering that takes place occurs at one end of the transmission or the other, not during the data's journey through the network. But this content-blindness is less a result of a political judgment than of a desire to keep the Internet technologically streamlined, so that it can transmit the greatest amount of data with the least amount of processing.[83] With data processing capacity doubling approximately every eighteen months,[84] however, we can easily conceive of an Internet that could differentiate among its transmissions—blocking some forms of content entirely, tagging others for surveillance, filtering others for certain recipients only, and so forth. Closed code would render such regulation invisible to those it regulates.

In fact, one does not need to look to a parade of hypothetical horribles for an example of the antidemocratic consequences of closed code. Consider the controversy over proprietary software that tabulates votes cast in federal and state elections. Because the software's source code is kept secret, the public cannot readily examine the program for accuracy or for vulnerability to hacking and

---

81   *See, e.g.*, Eduardo Gomez, *"Pure Speech or Expressive Conduct?": The "DeCSS Saga" and the Inconsistent Treatment of Computer Code Under the First Amendment*, 31 AIPLA Q.J. 231 (2003) (arguing that source code often receives more First Amendment protection than warranted); Lee Tien, *Publishing Software as a Speech Act*, 15 BERKELEY TECH. L.J. 629 (2000) (arguing that code usually is speech for First Amendment purposes).

82   *See* Lessig, *supra* note 24, at 764–67.

83   *See* Mark A. Lemley & Lawrence Lessig, *The End of End-to-End: Preserving the Architecture of the Internet in the Broadband Era*, 48 UCLA L. REV. 925, 931 (2001) ("While the e2e design principle was first adopted for technical reasons, it has important social features as well.").

84   This theory, known as Moore's Law, is named for Intel founder Gordon Moore and has proved remarkably accurate since he first articulated it in 1965. *See* Doris J. Yang, *The Lawgiver: Gordon Moore Is Still Chipping Away*, U.S. NEWS & WORLD REP., July 10, 2000, at 38.

election fraud.[85] States have responded to this problem by insisting that the developers give election officials access to the source code for examination,[86] by requiring that a copy of the code be held in escrow,[87] and by allowing individuals to petition for access to it.[88] But as long as the code by default remains off limits to the general public (and particularly to those citizens skilled enough to evaluate it), such measures will be of limited utility.[89] If computer code is to be subject to the same pro-democratic conditions as other expressive content, a different approach is needed.

II.   TECHNOLEGICAL TOOLS FOR OPENING CODE

Closed code causes problems both theoretical and practical, undermines copyright's goals, and imposes costs on programmers and the public alike. The following discussion explores three possible ways to solve these problems. The first, a combination of fair use and reverse engineering, proves too unreliable and inaccessible, but it does demonstrate the importance of ensuring that the solution is "technolegical"—i.e., that it directly regulates technology. The second, open-source licensing, comes with too much legal baggage to work for traditional copyright actors, but it serves to highlight the architectural features necessary for a workable solution. The third and best option is a return to the venerable copyright formalities, technolegical tools flexible enough to open code architecturally without releasing it from copyright's legal protection.

A.   Fair Use and Reverse Engineering

Fair use is copyright's ultimate safeguard, an equitable privilege that can mutate into any form needed to strike copyright's balance between private incentive and public benefit. It should therefore come as no surprise that courts have tried to use the fair use doctrine to right the wrongs of closed code—for example, to excuse technically infringing but merely temporary reproduction of computer code when the copyist's goal was merely to extract the uncopyrightable

---

85   Andrew Massey, Note, *"But We Have To Protect Our Source!": How Electronic Voting Companies' Proprietary Code Ruins Elections*, 27 HASTINGS COMM. & ENT. L.J. 233, 234–35 (2004).

86   *E.g.*, CAL. ELEC. CODE § 19205 (West 2003) (setting forth specifications for testing of code by Secretary of State).

87   *E.g.*, *id.* § 19103(a) (West Supp. 2005).

88   *E.g.*, *id.* § 19202 (West 2003).

89   *E.g.*, *id.* § 19206 (limiting number of expert testers to three).

facts within.[90] The fair use doctrine thus theoretically addresses at least some of the closed-code problems articulated above, in the sense that it provides legal protection for certain steps that might be necessary to access the public domain materials—ideas, facts, etc.—that closed code conceals.

For a number of reasons, however, fair use is not a panacea. First, it is too indeterminate a doctrine to provide a reliable touchstone for future conduct. Deciding whether a given activity constitutes fair use requires review of several complicated, interdependent, and nonexclusive statutory factors,[91] an intimidating and expensive undertaking.[92] In some cases precedent might provide an answer, but clear precedent on fair use is a rare thing in the fast-changing world of digital technology, and thus in many cases the uncertainty of the outcome would undoubtedly have a chilling effect on socially beneficial behavior. For instance, consider the potential third party collaborator described above, who may need to make temporary copies of another developer's source code to determine the viability of a certain value-added adaptation. Although this unauthorized copying might pose little threat to the copyright holder's interests (and indeed could ultimately promote them), it is far from clear that fair use would provide a defense; the question would likely be resolved only after protracted and expensive litigation. The potential collaborator might therefore not bother to undertake the project in the first place. Other examples of a similar chilling effect abound in copyright law.[93]

---

90 Nautical Solutions Mktg., Inc. v. Boats.com, No. 8:02-CV-760-T-23TGW, 2004 WL 783121, at *2 (M.D. Fla. Apr. 1, 2004); Evolution, Inc. v. SunTrust Bank, 342 F. Supp. 2d 943, 956–57 (D. Kan. 2004); *see also* Assessment Techs. v. Wiredata, Inc., 350 F.3d 640, 644–45 (7th Cir. 2003) (noting that fair use allowed defendant to copy copyrighted compilation if that was only way to access uncopyrighted data within); Ticketmaster Corp. v. Tickets.com, Inc., No. CV997654HLHVBKX, 2003 WL 21406289, at *5 (C.D. Cal. Mar. 7, 2003) (finding that temporary copying of copyrighted material in electronic form in order to extract facts was fair use).

91 17 U.S.C. § 107 (2000).

92 LAWRENCE LESSIG, FREE CULTURE 187 (2004) ("[F]air use . . . simply means the right to hire a lawyer to defend your right to create."); Mark A. Lemley, *Dealing with Overlapping Copyrights on the Internet*, 22 U. DAYTON L. REV. 547, 566 (1997) (pointing out that "fair use . . . is hard to predict in advance and that it will be expensive to prove"); Litman, *Revising Copyright, supra* note 45, at 45–46 (" [F]air use is a troublesome privilege because it requires a hideously expensive trial to prove that one's actions come within its shelter.").

93 *See, e.g.*, LESSIG, *supra* note 92, at 95–99 (describing documentary filmmaker's decision not to engage in a seemingly obvious fair use because of fear of litigation and other practical obstacles). For more examples of the chilling effect that fair use's uncertainty produces, see *infra* Part III.A.2 (discussing copyright as censorship).

Second, and more specific to computer code, closed code is by definition unpublished, and unpublished works are presumptively less susceptible to fair use. Because the entry of a copyrighted work into the marketplace often represents the point at which its author stands to profit the most,[94] deciding when to publish a work has long been regarded as uniquely within the author's province (the so-called "right of first publication").[95] Unpublished works have therefore been poor candidates for fair use because their exploitation by an unlicensed user represents a usurpation of this important authorial prerogative. Yet even when the author never intends to release the work to the public—as is the case with most source code—courts maintain this presumption against fair use.[96] Therefore, even if one manages to get one's hands on unpublished source code without running afoul of some other legal restriction,[97] fair use may be a hard argument to make.[98]

Third, and most important, fair use is merely a legal entitlement; it has no effect on one's architectural ability to access a copyrighted work. In the classic terminology of Hohfeld, fair use is a mere "privilege," not a "right," and therefore imposes no Hohfeldian "duty" on software developers to afford the public fair use opportunities or to refrain from keeping their source code secret.[99] Fair use is fairly useless without the means to access the work,[100] and closed code prohibits such access. The doctrine only comes into play as a defense, after the fair user has somehow managed to obtain the source code.[101]

Therefore, even if one were confident that a given use was fair, and had the resources to defend that assertion in court, fair use would solve the closed-code problem only if it worked in tandem with an

---

94   *See, e.g.*, Stephen Breyer, *The Uneasy Case for Copyright: A Study of Copyright in Books, Photocopies, and Computer Programs*, 84 HARV. L. REV. 281, 300 (1970) ("Publishers expect a tradebook to begin to return a profit within a few months of publication if it is ever to do so.").

95   *See* Harper & Row, Publishers, Inc. v. Nation Enters., 471 U.S. 539, 550–53 (1985).

96   *See* sources cited *infra* notes 199–204.

97   For example, the Digital Millennium Copyright Act makes it illegal to access copyrighted work in certain ways, even when the ultimate goal is a lawful use of the work. *See infra* Part III.B.1.

98   *Cf.* McJohn, *supra* note 76, at 61 (making inverse point that open-source software is more susceptible to fair use because it is published).

99   *See* Wesley Newcomb Hohfeld, *Some Fundamental Legal Conceptions as Applied in Judicial Reasoning*, 23 YALE L.J. 16, 30–33 (1913).

100   Gibson, *supra* note 15, at 207.

101   *See* Hohfeld, *supra* note 99, at 35 (observing that even if *X* has a privilege to eat a salad, *A* may still "hold[ ] . . . fast to the dish").

architectural measure—some way of ensuring actual access to the code so as to make it available for fair use. This brings us to a second existing mechanism that can mitigate the disadvantages of closed code: reverse engineering.

Although proprietary software developers keep source code secret, they must release their object code in order to market their products. A clever third party can accordingly try to gain insight into the characteristics of source code through "decompilation"—a reversal of the process by which the original programmer converted the source code into the object code. The first step in decompilation is to translate the object code into an intermediate language known as assembly code, using programs known as disassemblers.[102] Experienced programmers can then get some idea of what source code looks like by examining and experimenting with the assembly code.

With reverse engineering, then, we have an architectural complement to the legal fair use privilege. If one wants to engage in a socially beneficial use of closed source code, one can use decompilation to secure architectural access to the code and then employ the fair use privilege to ward off any ensuing infringement suit. In tandem, then, reverse engineering and fair use constitute a technolegical tool; they serve copyright by dismantling both legal and technological barriers to achievement of its goals.

Courts have recognized and embraced this complementary use of reverse engineering and fair use, most prominently in cases in which a software developer wished to make its program compatible with another developer's hardware or operating system but lacked the interoperability specifications necessary to do so. In *Atari Games Corp. v. Nintendo of America Inc.*,[103] for example, Atari wanted to manufacture video games that would play on Nintendo's gaming console but was prevented from doing so by a Nintendo "lock-out" program. Atari accordingly extracted the lock-out program's object code from a computer chip and then disassembled it so as to understand the unprotected ideas and methods of operation necessary to achieve interoperability for Atari's games.[104] Although this process involved the making of unauthorized copies of the program, the court held that such copying was fair use: "An author cannot acquire patent-like protection by putting an idea, process, or method of operation in an

---

102   *See* Jessica Litman, *Copyright and Information Policy*, 55 LAW & CONTEMP. PROBS. 185, 197 (1992). For an in-depth discussion of disassembly and decompilation, giving examples of source, assembly, and object code, see OFFICE OF TECH. ASSESSMENT, *supra* note 64, at 7, 147–50.
103   975 F.2d 832 (Fed. Cir. 1992).
104   *Id.* at 842–44.

unintelligible format and asserting copyright infringement against those who try to understand that idea, process, or method of operation."[105] The Ninth Circuit used the same reasoning in the similar case of *Sega Enterprises Ltd. v. Accolade, Inc.*,[106] holding that where a party has a legitimate interest in accessing the unprotected ideas and functional elements of a program, and disassembly is the only means of gaining such access, any intermediate copying that occurs during the disassembly process is privileged as a fair use.[107]

Most scholars have applauded these decisions,[108] and other courts have readily adopted their reasoning.[109] And although the facts of *Atari* and *Sega* pertained only to interoperability concerns, the courts spoke more generally about the propriety of accessing and using public domain elements embedded in closed code. The *Atari* court espoused the broad principle that the fair use doctrine generally "permits an individual in rightful possession of a copy of a work to undertake necessary efforts to understand the work's ideas, processes, and methods of operation."[110] And the Ninth Circuit observed in *Sega* that distribution of programs in object code "defeats the fundamental purpose of the Copyright Act—to encourage the production of original works by protecting the expressive elements of those works while leaving the ideas, facts, and functional concepts in the public domain for others to build on."[111] Several subsequent cases have accordingly allowed copying of copyrighted content in the course of trying to extract the unprotected material within.[112]

Nevertheless, even in tandem, fair use and reverse engineering are not a comprehensive technolegical solution to the closed-code problem described above, because any such solution must be both dependable and accessible. But decompilation is not dependable: it is

---

105   *Id.* at 842.

106   977 F.2d 1510 (9th Cir. 1992).

107   *Id.* at 1520.

108   *See* Samuelson & Scotchmer, *supra* note 62, at 1609 n.163 (collecting sources); *cf.* Maureen A. O'Rourke, *Toward a Doctrine of Fair Use in Patent Law*, 100 COLUM. L. REV. 1177, 1230–35 (2000) (proposing fair use defense for patent law that would cover reverse engineering of object code for interoperability purposes).

109   *E.g.*, Sony Computer Entm't, Inc. v. Connectix Corp., 203 F.3d 596, 608 (9th Cir. 2000); Bateman v. Mnemonics, Inc., 79 F.3d 1532, 1539 n.18 (11th Cir. 1996); Mitel, Inc. v. Iqtel, Inc., 896 F. Supp. 1050, 1056–57 (D. Colo. 1995), *aff'd on other grounds*, 124 F.3d 1366 (10th Cir. 1997).

110   975 F.2d at 842.

111   977 F.2d at 1527.

112   *See* sources cited *supra* note 90.

often prohibitively difficult and inaccurate[113]—indeed, the high costs of reverse engineering are one reason that trade secret law views it as permissible, as the required investment of resources acts as a built-in disincentive to undertake it[114]—and even when it is successful it does not reveal valuable commentary buried in the code.[115] Moreover, the decompiler's chances of success bear no necessary relation to intellectual property law's desired balance of legal entitlements. Nor is decompilation accessible: it takes a skilled technologist to even attempt it, and there is no reason to expect that decompilers would target those programs in which the technologically unskilled public is interested, or would choose to share the results of their efforts with the unwashed.[116]

Fair use is not dependable or accessible either. As already discussed, the fair use doctrine works best as a backward-looking defense

---

113   Menell, *supra* note 20, at 74 ("It is very difficult and time consuming to reverse engineer a computer program from its object code."); Samuelson & Scotchmer, *supra* note 62, at 1587 n.50 ("Some trade secrets may have been serendipitously developed at low cost yet are difficult to reverse engineer, while other expensive and time consuming innovations may be impossible to hide in the final product."); Pamela Samuelson et al., *A Manifesto Concerning the Legal Protection of Computer Programs*, 94 COLUM. L. REV. 2308, 2336 (1994) ("[I]n the current state of the art, decompilation is a painstaking and time-consuming process."); Strasser, *supra* note 37, ¶ 8 n.18 (noting that "[t]he practical obstacles to reverse engineering are not always fully appreciated in the legal literature" and that "[t]he truth is that in today's world, reverse engineering commercial software is virtually impossible"). Nor can we count on the science of reverse engineering improving in the future. *See* Samuelson et al., *supra*, at 2341–42 & n.115.

114   *See* Samuelson & Scotchmer, *supra* note 62, at 1582.

115   Some of a program's most useful ideas may be found in the explanatory commentary that the programmer inserts into the source code, which does not compile into object code and thus cannot be reverse engineered through decompilation. *See* Andrew Johnson-Laird, *Software Reverse-Engineering in the Real World*, 19 U. DAYTON L. REV. 843, 896 (1994); MacGrady, *supra* note 23, at 1065 n.148. If copyright were to condition its protection for software on the disclosure of source code, as proposed below, it would probably have to impose something akin to patent law's "best mode" requirement, 35 U.S.C. § 112 (2000), so that developers would include in the disclosed source code whatever commentary they used in their own internal development of the software. (I am indebted for this point to Alaric Fox, Associate Editor at JURIMETRICS JOURNAL.)

116   Of course, those interested in the ideas in a program will tend to be programmers themselves, and therefore more skilled than the average citizen. But not all programmers have the skills and resources to reverse engineer software, and in some instances the unskilled general public will have an interest in accessing other public domain aspects of a program, such as the facts it contains. *See, e.g.*, Assessment Techs. v. Wiredata, Inc., 350 F.3d 640, 644–45 (7th Cir. 2003) (allowing copying of copyrighted compilation in order to access uncopyrighted public tax assessment data it contained).

against proven infringement, not as a forward-looking guide for future conduct. While the case law has been friendly to reverse engineering in pursuit of interoperability, it is impossible to know whether courts will be equally willing to give their blessing to reverse engineering for other purposes. What of the curious computer science student who decompiles software not for interoperability purposes, but to view (not copy, just view) its protected expression—e.g., to learn how its developer dealt with a tricky programming issue?[117] Would the *Sega* court consider that a legitimate justification for making the unauthorized intermediate copies that the decompilation process requires? Accessibility is even more suspect: would our presumably penurious student ever have the resources necessary to bring the fair use argument to a court's attention?

Finally, even if reverse engineering and fair use were cheaper, more reliable, and more widely available, they are unlikely to provide the best solution to the closed source problem. Ad hoc acts of decompilation would not mitigate the practical disadvantages of closed code described above—the loss of unsupported programs, the difficulty of assessing prior art, and so forth. Most important, being forced to employ decompilation and defend one's conduct in court begs the question: why should the public have to jump through these hoops at all, when part of copyright's quid pro quo is to require authors of creative works to make their works' ideas and other unprotected elements available? Seen in this light, the costs of reverse engineering and the fair use defense—no matter how low—represent mere waste, the expenditure of resources on something to which one is entitled free of charge.[118] Indeed, developers also know that reverse engineering is permitted, so they too engage in wasteful conduct by making their products unnecessarily complex and thus more resistant to decompilation.[119]

---

117   *Cf.* LESSIG, *supra* note 47, at 253 ("Thus, while an English Department gets to analyze Virginia Woolf's novels to train writers in better writing, the Computer Science Department doesn't get to examine Microsoft's operating system to train its students in better coding.").

118   *See* Mark A. Lemley & David W. O'Brien, *Encouraging Software Reuse*, 49 STAN. L. REV. 255, 260 (1997) ("Rough estimates place the annual aggregate cost of reinvention at between $2 billion and $100 billion."); Christina M. Reger, *Let's Swap Copyright for Code: The Computer Software Disclosure Dichotomy*, 24 LOY. L.A. ENT. L. REV. 215, 235 (2004) ("Reinvention, in the software context, is a model of inefficiency.").

119   Cohen, *supra* note 55, at 1099–101 (pointing out that both *Sega* and *Atari* involved reverse engineering of "lock-out" programs explicitly designed to make interoperability more difficult); *see also* Douglas Lichtman, *How the Law Responds to Self-Help*, 1 J.L. ECON. & PUB. POL'Y 215, 236–37 (2005) (discussing the "arms race" that reverse engineering produces).

Thus we need to do more than tinker with the feasibility and legality of reverse engineering. We need to make it entirely unnecessary, by making public access to a work's expressive content part of the copyright scheme. But how do we get there? How do we make source code available without all the waste and uncertainty that accompany the reverse-engineering-*cum*-fair-use approach? As it happens, the programming community has for some time been experimenting with an alternative means of software distribution, one that illuminates this question but stops short of answering it completely: open-source software.

B.   Lessons from the Open-Source Movement

1.   Freeing Software and Opening Source

Closed code may be the norm for commercial software developers, but it is not the only approach that the larger programming community uses. Beginning in the 1960s and 1970s, a group of academics and software engineers from loosely-controlled corporate research environments routinely shared the source code to their programs on an informal basis and allowed others to freely make use of it.[120] This sharing ethos not only allowed each institution to tailor programs to its own needs, but also provided for cooperative development. For example, a researcher at MIT might study an operating system written by a programmer at Bell Labs or Xerox, devise ways to improve its performance, and share those improvements with the program's originator and others in their cooperative community.

In the early 1980s, however, some of the institutions whose programmers formed this fellowship began to favor the use of proprietary software, and the freedom to share and adapt source code suffered as a result.[121] Particularly galling to the collaborative programmers was the commercialization of Unix, an operating system to which many of them had contributed: in 1982, AT&T began marketing a version of Unix without making its source code available.[122] In direct response to this trend toward commodification of software, Richard Stallman of MIT founded the Free Software Foundation, an organization devoted to creating an environment conducive to the widespread development

---

120   Josh Lerner & Jean Tirole, *Some Simple Economics of Open Source*, 50 J. INDUS. ECON. 197, 200–01 (2002).

121   RICHARD M. STALLMAN, *The GNU Project*, in FREE SOFTWARE, FREE SOCIETY 15, 15–16 (Joshua Gay ed., 2002).

122   *See* LESSIG, *supra* note 47, at 53.

and distribution of a variety of freely adaptable computer programs.[123]

The Free Software Foundation wanted to avoid a repeat of the AT&T/Unix experience, in which programmers had openly contributed to the improvement of a program only to see their efforts co-opted by a commercial firm. So its first order of business was to develop some means of ensuring that cooperative software projects remained cooperative. Its solution was an innovative copyright licensing scheme, which Stallman dubbed "copyleft."[124] Under the copyleft approach, each copy of shared source code includes a license that defines the permissible uses of the program: the recipient can run the program, modify the program, and distribute the program in its original or modified form—but any such distributions must be governed by the same licensing terms and are thus subject to the same constraints.[125] The license therefore binds itself to the software like a virus to a host, accompanying it through theoretically infinite distributions to and modifications by downstream programmers. Despite the term "free software," the license does not prohibit charging a fee for copies of licensed programs, modified or not—Stallman has famously analogized the "free" in free software to the "free" in free speech, not free beer[126]—but one may not distribute them except on the same copyleft terms.

Although today a number of popular programs use copyleft licenses,[127] Stallman and the other copyleft devotees originally focused on developing one particular program, an operating system compatible with Unix. Stallman named this new program "GNU," a recursive acronym that stood for "GNU's Not Unix."[128] In 1991, programmer Linus Torvalds developed the final piece of the GNU puzzle, known as the Linux kernel, and the operating system (which came to be known as GNU/Linux) was complete.[129] The particular copyleft license

---

123 STALLMAN, *supra* note 121, at 21; Lerner & Tirole, *supra* note 120, at 201.

124 STALLMAN, *supra* note 121, at 20–21.

125 *Id.* at 20.

126 RICHARD M. STALLMAN, *Free Software Definition, in* FREE SOFTWARE, FREE SOCIETY, *supra* note 121, at 41, 41.

127 Some of the most well known are the Apache web server, the Perl programming platform, and the Sendmail e-mail transfer agent. Yochai Benkler, *Coase's Penguin, or, Linux and the Nature of the Firm*, 112 YALE L.J. 369, 371–72 (2002).

128 STALLMAN, *supra* note 121, at 17. The use of recursive acronyms like GNU was a tradition in Stallman's programming community. *Id.* Notwithstanding the pun in the title of this Article's next subsection, GNU is pronounced "guh-NEW." RICHARD M. STALLMAN, *Free Software: Freedom and Cooperation, in* FREE SOFTWARE, FREE SOCIETY, *supra* note 121, at 155, 162.

129 STALLMAN, *supra* note 121, at 26.

under which it was distributed was called the GNU General Public License (GNU GPL)—the first of what became known as the open-source licenses.[130]

Over time, as the open-source community expanded, it developed other licenses, some of which were much more liberal than the GNU GPL in the downstream uses they permitted. For example, a spin-off group of programmers formed the Open Source Initiative (OSI), which will certify licenses that allow a downstream programmer to incorporate the licensed software into a proprietary program that he or she then releases under a different, potentially more restrictive license—e.g., one that allows no access to or modification of the source code.[131] The OSI is thus less doctrinaire in its views than the copyleft devotees; it sees shared source code and a nonproprietary approach to software as advantageous on a pragmatic level and reconcilable with commercial interests, whereas Stallman and his colleagues view their ethos as morally compulsory and incompatible with commodification.[132]

2. Something Old, Something GNU

The open-source approach to software development has inspired much commentary on the challenge it poses to software's established commodification model and to the traditional copyright assumption that exclusive property rights are a sine qua non of the creative process.[133] For present purposes, however, what is most interesting about the open-source model is its innovative, homegrown approach to solving the closed-code problem. To fully appreciate what this model does and does not have to offer in this regard, one must first understand the intricacies of its licensing scheme.

---

130 For the text of the GNU GPL, see Free Software Found., GNU General Public License (1991), *available at* http://www.fsf.org/copyleft/gpl.html. "Open source" has become the conventional term used to describe the entire community of cooperative programmers, including the copyleft devotees—much to Stallman's dismay. *See* RICH-ARD M. STALLMAN, *Why "Free Software" Is Better than "Open Source," in* FREE SOFTWARE, FREE SOCIETY, *supra* note 121, at 55, 55–56.

131 Lerner & Tirole, *supra* note 120, at 202.

132 Zittrain, *supra* note 66, at 275–76; *see also* ROD DIXON, OPEN SOURCE SOFTWARE LAW 7–8 (2004) (discussing OSI's origins and attitudes); STALLMAN, *supra* note 130, at 55 (setting forth Stallman's argument on the superiority of his approach); Open-Source.org, History of the OSI, http://www.opensource.org/docs/history.php (last visited Sept. 8, 2005) (explaining OSI's more pragmatic approach).

133 *E.g.,* Benkler, *supra* note 127; Lerner & Tirole, *supra* note 120; David Mc-Gowan, *Legal Implications of Open-Source Software,* 2001 U. ILL. L. REV. 241.

Start with the notion that copyright licenses do not have to be contractual in nature; copyright imposes restrictions on everyone, not just on those with whom the copyright owner can claim contractual privity.[134] Federal statute, not contract law, forbids us to do certain things with a copyrighted work—copy it, modify it, distribute it, perform or display it publicly—without the permission of the copyright owner.[135] So the law's default rule with regard to such conduct favors prohibition, not freedom. In order to procure the legal freedom to engage in these otherwise forbidden activities, one usually needs to contact the copyright owner and obtain his or her consent. Sometimes this transaction will be contractual, as when the copyright owner demands payment in exchange for the permission. But sometimes the copyright owner will simply grant a license to copy or modify—a "permission to commit some act that would otherwise be unlawful"[136]—without requiring any consideration or establishing any contractual privity.[137]

Open-source licenses follow this model. Suppose a programmer adds his or her own code to the GNU/Linux operating system and then releases the resulting product in object code format only. By distributing the modified program without making its source code available, the programmer has infringed the GNU/Linux copyright, because the license that accompanies the GNU/Linux software (the GNU GPL) grants permission to make and distribute modified versions only if those versions are distributed under the same terms as the original program, and those terms require accessible source code in downstream modifications.[138] And if the programmer tries to es-

---

134 *See* DIXON, *supra* note 132, at 24 (pointing out that contractual privity tends not to be an issue in open-source licensing because end-users usually have no right to copy or modify code absent a license).

135 17 U.S.C. § 106 (Supp. II 2002). The statute does not actually use the term "modify"; it refers instead to the preparation of "derivative works" based on the original copyrighted work. *Id.* § 106(2).

136 BLACK'S LAW DICTIONARY 931 (7th ed. 1999).

137 For example, refer to the license set forth on the first page of this article. It gives individuals permission to make and distribute copies of this Article for educational purposes, subject to certain attribution and downstream licensing requirements. Students who see this license and make copies of this Article for use in class will therefore not be liable for copyright infringement. But if one were to exceed the terms of this license by, say, substantially altering my text and selling the modified copies, I could sue for infringement, because that conduct implicates a copyright prerogative—modification—for which my permission has not been given. (Perhaps one could also characterize the transaction proposed in my first footnote as contractual, but one would not have to do so in order to use the remedy that copyright provides.)

138 Free Software Found., *supra* note 130. Not all open-source licenses contain these same restrictions, but the GNU GPL is by far the most popular of those licenses.

cape infringement liability by arguing that the GNU GPL does not apply, then he or she has just abandoned the only argument that would have made the modification legal in the first place; without the GPL's protection, the modification is unlicensed and infringing.[139]

Open-source licenses thus cleverly enlist copyright law to promote their own unorthodox purposes. Despite the movement's occasional anticopyright rhetoric,[140] copyright is vital to its success. That aspect of open-source with the best claim at being revolutionary—its explicit, uncompensated invitation to downstream users to adapt and improve the product—is achieved through a viral sequence of traditional licenses that draw their power from the same copyright rights that the movement seeks to dilute.[141] If those exclusive rights did not exist, computer programs would be in the public domain, and any programmer could modify source code and distribute the resulting program in object code format only, without securing any license—contrary to the copyleft ethos. The GNU GPL and similar new, innovative licensing schemes thus promote the availability of source code, but only because of copyright law's old, established exclusive rights in expression.

So is open-source licensing the solution to the closed-code problem? It is certainly true that if all programmers release their programs under the GNU GPL or a similar license, then source code will always be exposed for all to see. The ideas, facts, and other public domain aspects of the work will be available for the taking. Potential collaborators could view the protected expression and assess whether their improvements are worthwhile. Unauthorized copies and adaptations will be easier to detect. Patent examiners will have a rich reservoir of

---

MICHAEL A. EINHORN, MEDIA, TECHNOLOGY AND COPYRIGHT 169 (2004) (estimating that 29,000 open-source projects out of total of 40,000 use the GNU GPL).

139   *See* Welte v. Sitecom Deutschland GmbH, No. 21 O 6123/04, at 8 (LG München I May 19, 2004), *available at* http://www.jbb.dc/urteil_lg_muenchen_gpl. pdf (enforcing GPL under German law); *see also* McGowan, *supra* note 133, at 289 ("[A] downstream user who claimed the GPL did not bind her would merely open herself to an infringement action.").

140   *E.g.,* RICHARD M. STALLMAN, *Science Must "Push" Copyright Aside, in* FREE SOFTWARE, FREE SOCIETY, *supra* note 121, at 87, 87–88.

141   James Boyle, *The Second Enclosure Movement and the Construction of the Public Domain,* 66 LAW & CONTEMP. PROBS. 33, 65 (2003); Michael J. Madison, *Reconstructing the Software License,* 35 LOY. U. CHI. L.J. 275, 276–77 (2003); R. Polk Wagner, *Information Wants To Be Free: Intellectual Property and the Mythologies of Control,* 103 COLUM. L. REV. 995, 1031–32 (2003).

prior programming art to draw on.[142] In short, all the disadvantages of the closed-code system will disappear.

But open source has one obvious drawback: it does not work for software developers who want to sell their products for a profit. Indeed, even if a for-profit software firm were willing to provide a copy of its source code with every copy of object code that it sells, it could not adopt the viral open-source approach, because the latter makes it impossible to control the acts of copying and modification that allow a firm to generate revenue. The closest a for-profit firm could come to viral distribution would be the shareware model, which relies on voluntary compliance with an inherently unenforceable pricing scheme, or the bundling model, in which software is given away for free but support services or other extras cost money. These models have met with some success but have not succeeded in replacing the proprietary closed-code approach.[143]

Even the less restrictive open-source licenses that focus more on making source code available and less on ridding the software world of copyright entitlements will not do the trick. Consider the Berkeley Software Distribution (BSD) license, the most venerable of the more permissive licenses. Unlike the GNU GPL, the BSD license does not require downstream modifiers to release the source code to their modifications.[144] Code initially developed under the license has thus been appropriated into commercial closed-code projects, such as the Macintosh OS X operating system.[145] The BSD license does, however, grant permission to make copies and modify the code—an activity

---

142   Indeed, the open-source community has already made some efforts to establish a centralized collection of prior art. *See* DAN RAVICHER, OSRM POSITION PAPER: MITIGATING LINUX PATENT RISK 4 (2004), *available at* http://www.osriskmanagement.com/pdf_articles/linuxpatentpaper.pdf (discussing open-source community's efforts to build "arsenal of prior art" in order to provide PTO with basis to reject software patent applications); *see also* Cohen, *supra* note 55, at 1179 (discussing Software Patent Institute's similar efforts); Strasser, *supra* note 37, ¶ 46 (proposing providing patent examiners with "powerful, constantly updated databases" of source code).

143   For example, the wildly popular closed-code computer game *Doom* was distributed as shareware; the public could copy, distribute, and play the first part of the game for free but had to pay to access the rest. *See* James Boyle, *A Politics of Intellectual Property: Environmentalism for the Net?*, 47 DUKE L.J. 87, 104 (1997). And Red Hat, Inc. makes money by bundling documentation and support services with open-source software. Susan Scafidi, *Intellectual Property and Cultural Products*, 81 B.U. L. REV. 793, 834 (2001).

144   For a copy of the BSD license, see OpenSource.org, The BSD License, http://www.opensource.org/licenses/bsd-license.php (last visited Sept. 8, 2005).

145   David S. Evans & Bernard J. Reddy, *Government Preferences for Promoting Open-Source Software: A Solution in Search of a Problem*, 9 MICH. TELECOMM. & TECH. L. REV. 313, 353 (2003).

that would otherwise violate the original programmer's copyright. One therefore cannot use the BSD license to make source code available without simultaneously surrendering some of the very legal rights that allow developers to charge for their efforts.

Seen in this light, it is the very ingenuity of the open-source scheme that makes it inappropriate for achieving the more modest goal of solving the closed-code problem. The genius of open-source licensing is its viral nature—the absence of any need to contact the copyright owner directly for permission to copy, adapt, and so forth as long as one stays within the terms of the license. But if a "license" provides only the right to view the source code (rather than copy or modify it), then there is no license at all, because no copyright permission is ever needed simply to look at copyrighted content. There is no license if there is no legal right forsaken.

3.   Something Borrowed, Something Not Borrowed

By definition, then, open-source licenses do more than open up source code. They also require software developers to surrender legal entitlements, such as the right to control (and thus profit from) the making of copies and downstream modifications. Indeed, that is the whole point of copyleft's overtly dogmatic approach to software development: it sees copyright law—at least as applied to software—as a fundamentally unsound and unneeded theory of propertization, like Kafka's cage that went in search of a bird.[146] The open-source movement uses the law only to encourage programmers to contribute to a collective effort that transcends any individual's self-interest.

Self-interest, however, has proven to be a good motivator over hundreds of years of copyright history. The fact that it might not be necessary in every circumstance or indispensable to every industry does not justify its wholesale abandonment as copyright's legal underpinning.[147] Requiring commercial software firms to surrender these legal entitlements is therefore unfair; expecting them to do so voluntarily is unrealistic.[148] On the other hand, the architectural achieve-

---

146   "Ein Käfig ging einen Vogel suchen." FRANZ KAFKA, HOCHZEITSVORBER-EITUNGEN AUF DEM LANDE UND ANDERE PROSA AUS DEM NACHLAß 41 (Max Brod ed., 1953), *translated in* WEDDING PREPARATIONS IN THE COUNTRY AND OTHER POSTHUMOUS PROSE WRITINGS (Ernst Kaiser & Eithne Wilkins trans., Secker and Warburg 1954).

147   *See, e.g.,* McGowan, *supra* note 133, at 284–87 (discussing circumstances under which open-source development might not be as successful as its alternatives).

148   This is not to say that commercial firms will never embrace open-source licenses; several have done so, or have even released their code into the public domain. *E.g.,* Jonathan Krim, *IBM To Help Open-Source Developers,* WASH. POST, Jan. 11, 2005, at E5 (detailing IBM's decision to allow hundreds of its patents to be used in open-

ment of the open-source approach—making code public—is something that copyright law should embrace.

What we need to do, then, is separate the legal from the architectural. A programmer cannot modify a program without architectural access to its source code, even if he or she has legal permission to do so. So without architectural access to the source code, a license to modify a program is useless. But the reverse is not true: one could release the source code of a program to the public, and thus permit architectural access thereto, without also granting the public any legal right to copy, modify, or distribute it. The closed-code problem can therefore be resolved by making the code architecturally accessible without forcing programmers to surrender the legal entitlements that copyright provides. In other words, copyright should embrace the architectural freedom that open source promotes but reject the radical legal freedoms that make open-source licensing effective.[149]

So what architectural tools does the open-source movement use? It turns out that in this respect, as in its legal aspects, open source represents a return to traditional copyright principles rather than a revolution against them. In fact, it harkens back to copyright concepts that proprietary firms would consider quite antiquated: most notably the formality of publication, but also the notice formality.

Take publication first. The central tenet of the open-source ethos is that source code must accompany every copy of the program—no more distributing software in object code alone. This is also the essence of the publication formality, the traditional notion that making creative expression available should be part of the distribution process that copyright promotes. Indeed, until 1976, federal copyright law did not protect a work until it was published (i.e., pub-

---

source programs). These firms may have realized that in industries characterized by incremental innovation and in which significant revenue is expected to come from path-dependent developments, publishing research may sometimes make more sense than seeking intellectual property protection. *See* Oren Bar-Gill & Gideon Parchomovsky, *The Value of Giving Away Secrets,* 89 VA. L. REV. 1857 (2003). Or they may support open source so as to prevent a rival from cornering an important market. *See* Robert P. Merges, *A New Dynamism in the Public Domain,* 71 U. CHI. L. REV. 183, 191–93 (2004). Yet such instances remain the exception rather than the rule.

149   *See* Strasser, *supra* note 37, ¶ 57 (noting that these two aspects of open source "are not the same, and, as a conceptual matter, it is important to keep them separate"). *But see* Reger, *supra* note 118, at 237–43 (advocating use of legal aspects of open-source approach to achieve architectural goals); Massey, *supra* note 85, at 240 & n.38 (assuming that providing source code to public necessarily involves "loss of intellectual property protection").

licly distributed in a fixed form).[150] Even after 1976, copyrighted
works could not as a practical matter be marketed without simultane-
ously revealing their creative content, at least to the purchaser—ex-
cept, of course, in the case of software. By structuring software
transactions so that source code is always available, then, open-source
licensing simply brings software back into the copyright fold.[151] Seen
in this light, open-source, even in its most doctrinaire copyleft incar-
nation, is decidedly old-school.[152] If we reinstate publication as a
threshold requirement for copyright protection, then, we extract the
architectural advantage that open source provides without paying
open source's radical legal price—the surrender of a copyright enti-
tlement. Thus is the closed-code problem solved.

Of course, characterizing this proposal as less than radical may
seem strange to the developer of proprietary software, who has long
been accustomed to benefitting from copyright protection while keep-
ing source code hidden. But by conditioning copyright protection on
the publication of source code, the law puts software developers in no
worse a position than authors of other expressive works in digital
form. Copyright evolved under the assumption that authors could
profit from their creativity only by sharing it with the public; the law
accordingly granted them exclusive rights over their creative expres-
sion in return for public access to that expression, and to the ideas
and facts it contained. Returning to that model cures—rather than
causes—a problem with how copyright handles software.

This is not to say that the software developer has nothing to fear
from exposing its previously hidden source code. Publishing source
code will undoubtedly provide new opportunities for infringement—
although, as discussed above, these opportunities might not be as
threatening as they first appear.[153] For certain programs, publication

---

150   17 U.S.C. § 10 (1976) (repealed 1976). Because the definition of "publication"
encompassed only those works provided to the public in a tangible form, there was an
exception to the publication requirement for works that could be made available to
the public intangibly—such as plays, lectures, and musical performances; they could
secure copyright protection through registration and deposit alone. *See id.* § 12 (re-
pealed 1976); REGISTER OF COPYRIGHTS, 87TH CONG., COPYRIGHT LAW REVISION: RE-
PORT OF THE REGISTER OF COPYRIGHTS ON THE GENERAL REVISION OF THE U.S.
COPYRIGHT LAW 39 (Comm. Print 1961) [hereinafter 1961 REPORT].

151   *Cf.* DIXON, *supra* note 132, at 22 (arguing that open-source approach adheres
more closely than closed-code industry to copyright's constitutional goal of progress).

152   Indeed, the copyleft approach might even be viewed as the most traditional
open-source licensing permutation, in that it compels downstream programmers to
reveal their source code, whereas more permissive licenses like the BSD license simply
allow or encourage them to do so.

153   *See supra* Part I.C.2.

of source code might also increase vulnerability to hacking and vi-
ruses.[154] The publication formality should therefore be carefully tai-
lored to balance its benefits against these drawbacks.

One way to strike this balance is to adopt another architectural
feature of open-source licenses, namely the requirement the license
itself feature prominently in any distribution of the software it gov-
erns. This provision exposes downstream users to the license when
they first obtain the software so they can immediately understand what
rights they do and do not have with regard to it. In other words, be-
cause open-source licensing depends not on the bilateral nature of
contract law but on unilateral permission to use the code in a certain
way, its operative precondition is purely architectural: can the down-
stream user *see* the license? Open-source licenses thus not only regu-
late copyright entitlements but also serve a *notice* function, and it is
this indispensable notice function that makes the viral nature of open
source possible. The notice allows downstream users to copy, modify,
redistribute, etc.—whatever the license allows—without having to seek
out the copyright owner to determine what conduct is and is not
permitted.

As with publication, using notice to inform the public of its rights
and responsibilities with regard to expressive works is nothing new in
copyright law. In fact, it is quite old. It has been a part of copyright
law since the first Congress,[155] and until 1989 the failure to affix a
notice to a published copyrighted work meant that the work entered
the public domain.[156] The familiar c-in-a-circle, followed by the year
of publication and the name of the copyright owner, served to inform
the public that it was dealing with something laden with legal restric-
tions, that there were certain things it could and could not do with the
work. What the open-source licensing scheme has done is resurrect
this quaint formality and cleverly adapt it so that an individual
programmer can use it to further his or her sharing ethos.

---

154   *See* Katyal, *supra* note 79, at 2267 (arguing that publicizing code for operating
systems makes it less vulnerable but publicizing code for "more specialized applica-
tions with few users" may have the opposite effect).

155   *See* H.R. REP. No. 94-1476, at 143 (1976).

156   For works published between 1978 and 1989, a copyright owner had five years
to cure its failure to affix the notice. 17 U.S.C. § 405(a) (2000). Before 1978, the
failure was instantly fatal to the work's copyright. Stewart v. Abend, 495 U.S. 207, 233
(1990); H.R. REP. No. 94-1476, at 143 (1976). The current act retains a notice provi-
sion, but affixing a notice no longer affects a copyright's validity; notice merely makes
it harder for an infringer to reduce damages by arguing that the infringement was
innocent. 17 U.S.C. § 405(b).

Combining notice and publication results in a more flexible technolegical tool that can assuage some of the software developer's concerns. For example, resurrecting publication as a precondition to copyright protection would not necessarily mean that each copy sold would contain both the object code and the source code; most purchasers would probably not care about the latter. Instead, the law could use a notice formality to trigger the publication obligation: each copy of object code could bear a notice informing the purchaser how to contact the developer to obtain a copy of the source code, just as the traditional copyright notice identified the party to be contacted if a license was desired. This approach would limit possession of the code to authorized purchasers, at least in the first instance—as is the case with other copyrighted works. The developer would also know who had obtained copies of its code and could draw up a list of suspects if infringement or hacking ensued.

An alternative approach would be to use copyright's two other traditional formalities, registration and deposit, as threshold requirements for procuring a software copyright. Current law contains registration and deposit provisions, but they are of secondary importance at best. The U.S. Copyright Office will register software without even seeing its source code,[157] and registration of any work is unnecessary unless the copyright owner intends to file an infringement action.[158] And although today's copyright regulations do require deposit of published works with the Library of Congress, the lack of consistent enforcement and the toothless consequences of noncompliance render these regulations ineffective.[159]

Resurrecting these two formalities would result in a larger copyright bureaucracy, but it has the benefit of giving policymakers the option of opening source code to parties other than purchasers of the program, in effect establishing a national archive of code. Access to the archive could be completely unregulated if the benefits to the public were deemed important enough—the electronic voting machine example comes to mind—or could be restricted to privileged groups, such as patent examiners researching prior art or those who

---

157 *See supra* note 56.

158 17 U.S.C. § 411 ("[N]o action for infringement of the copyright in any United States work shall be instituted until registration of the copyright claim has been made in accordance with this title."). As its language clearly indicates, this provision applies only to U.S. works; works that originate abroad do not have to register even as a precondition to bringing suit.

159 *See* 2 NIMMER & NIMMER, *supra* note 51, § 7.17[B], at 7-190.1 to -.3.

establish their fair use bonafides.[160] Or the archive copy could remain off limits to all comers until expiration of its copyright, which would address (at least in part) the threat that closed code poses to the preservation of obsolete or unsupported programs.[161] Moreover, if registration were a precondition to earning the right to affix a notice and involved substantive examination of the source code for originality, then those citizens who lack the skill to evaluate the code themselves (as most of us do) would have some assurance that a program bearing a notice was copyrighted.

There are tradeoffs inherent in all of these examples. Some increase the chances of infringement in order to overcome the drawbacks of closed code, while others have the opposite effect. The goal here is not to settle on one option as optimal, but to demonstrate the adaptability inherent in this approach to closed code. Using copyright's traditional formalities as technolegical tools both focuses policymakers on copyright's intended balance and gives them the freedom to craft more radical or less radical solutions, as they see fit.

Yet whenever policymakers use architectural means to promote a legal goal, they must take care not to make unwarranted assumptions about the current and future state of digital architecture and the technology of software creation and distribution. For instance, the traditional definition of publication is tied to the notion that copyrighted works will mainly be marketed in individual, stand alone copies.[162]

---

160   One commentator suggests an opt-in, two-tiered system of copyright protection for source code, with a short period of true exclusivity followed by a period in which the code is filed with a government agency and is available for inspection by anyone. Anthony J. Mahajan, Note, *Intellectual Property, Contracts, and Reverse Engineering After ProCD: A Proposed Compromise for Computer Software*, 67 FORDHAM L. REV. 3297, 3331–35 (1999). If you object to a government agency's deciding who can and who cannot access library materials (or to its tracking of patrons' comings and goings), keep in mind that such an approach would serve as a supplement to—not a substitution for—traditional fair use entitlements and other public prerogatives. For example, one would still be permitted to obtain a copy of a work in the market and engage in fair use without accessing the deposit copy. *See infra* notes 286–88 and accompanying text.

161   *See* LESSIG, *supra* note 47, at 253 (proposing deposit of escrow copy of software as condition of copyright protection). Of course, copyright's duration far exceeds the useful life of any computer program, so the benefit of preserving a copy of a program until the end of its copyright term is speculative at best. But as I will discuss later, formalities can solve this problem as well, and not just for software. *See infra* Part III.A.3.c.

162   *See* 17 U.S.C. § 101 (defining publication as "the distribution of copies or phonorecords of a work to the public"). This definition generally codifies the longstanding meaning of the term in copyright law. 1 NIMMER & NIMMER, *supra* note 51, § 4.04, at 4-20.

Thus a work is not considered "published" if it is merely performed or displayed publicly.[163] But in an age of digital networks and high bandwidth, information goods are likely to be distributed more and more through real-time, on-demand transmission and less and less in fixed, tangible copies. The most familiar example is the streaming audio and video known to most Internet users. The software industry is also beginning to experiment with selling its product as a pay-per-use service—rather than as a freestanding good—by housing application programs in a central server and using high-bandwidth connections to provide them to customers on a real-time basis.[164] In that circumstance, even the object code would remain hidden, let alone the source code.[165] The definition of publication would therefore have to adjust to these new distribution models so as to ensure revelation of their source code in some accessible form.[166]

Of course, like anything that intellectual property law protects, source code is nonrival, and therefore no technolegical tool will be completely effective in keeping the genie in the bottle. Resurrecting the formalities, however, would not diminish the impressive array of legal entitlements that copyright has always provided and will continue to provide. *Looking* at creative content does not implicate copyright's entitlements at all; indeed, it's part of what copyright is supposed to promote. (By way of example, the Copyright Office must make available for *inspection* the copies that it has retained from the registration process,[167] but it restricts *copying* them.[168]) The resurrected formalities would thus promote public access to source code, with all its attendant benefits, but allow software developers to exercise their legal prerogative to bar copying of expression and other

---

163   17 U.S.C. § 101 ("A public performance or display of a work does not of itself constitute publication.").

164   *See* Leslie Walker, *Selling the Appeal of Renting out Business Software*, WASH. POST, Feb. 5, 2004, at E1.

165   *See* Strasser, *supra* note 37, ¶ 41 (concluding that even object code could be protected as trade secret in that circumstance).

166   This reveals an error in the reverse engineering case of *Atari Games Corp. v. Nintendo of America Inc.*, 975 F.2d 832, 843–44 (Fed. Cir. 1992), in which the court held that decompiling a program can constitute fair use only when the user is in authorized possession of a copy of the work. This ruling made sense given the facts of that case—Atari had acquired Nintendo's program through a fraud on the U.S. Copyright Office, *id.* at 841–42,—but relying on estoppel or unclean hands to deny Atari's claim of fair use would have been preferable to establishing a bright-line rule that assumes that lawful copies will always be available.

167   17 U.S.C. § 705(b).

168   37 C.F.R. § 201.2(d)(2) (2004).

wrongful appropriative acts.[169] And unlike the open-source approach, it would address the closed-code problems identified above without forcing the software developer to surrender the means to seek redress for downstream infringement.

Nor would the formalities eliminate other useful tools in the software developer's toolbox, such as the ability to control their products contractually and architecturally. Many developers already make it technologically impossible to use their programs absent consent to an end-user license agreement. They also employ copy protection technologies and similar means to inhibit the unlicensed copying and distribution of their software. The formalities would have no effect on these significant protections. (In fact, the only legal obstacle to this use of copyright formalities is our obligations under international copyright protocols;[170] if the other signatory nations fail to see the merit in reviving the formalities, Christopher Sprigman has already devised inventive methods for reconciling them with the treaties' provisions.[171])

Finally, even under a regime of revived formalities, publishing source code will never be compulsory for software developers. They can always choose to forgo copyright protection entirely, keep their source code secret, and protect their efforts using other means, both legal (e.g., contract and trade secret law) and technological (e.g., copy protection and access controls). Software developers, like other technological innovators, already face this kind of decision when they consider whether to seek patent protection for a program: they can patent their innovation, thus revealing its workings to the public but gaining a valuable exclusive right, or they can maintain the program as a trade secret, thus exploiting its value without revealing it to the public but taking the risk that a competitor will independently develop or reverse engineer the innovation.[172] Copyright has long im-

---

169   *See* Samuelson et al., *supra* note 113, at 2407 ("A legal regime for protecting applied know-how in programs from market-destructive appropriations should not block access to that know-how, but merely regulate its use."); *cf.* Samuelson & Scotchmer, *supra* note 62, at 1651 (noting in reverse engineering context that "it may be more sensible to regulate post-reverse-engineering activities than to regulate reverse engineering as such").

170   *See* Agreement on Trade-Related Aspects of Intellectual Property Rights, Apr. 15, 1994, Marrakesh Agreement Establishing the World Trade Organization, Annex 1C, Legal Instruments—Results of the Uruguay Round, 33 I.L.M. 81 (1994); Berne Convention for the Protection of Literary and Artistic Works art. 5(2), July 24, 1971, 102 Stat. 2853, 1161 U.N.T.S. 3.

171   *See* Sprigman, *supra* note 2, at 545–68.

172   Which option to choose depends on a number of factors, such as whether the innovation can be effectively exploited as a trade secret, its susceptibility to reverse

posed a similar choice on other authors: keep your creative expression secret and forgo mass marketing, or profit from it but share certain aspects of it with the public. It should impose the same choice on those who seek copyright protection for software.

III.   A Broader Role for Formalities

No copyrightable work other than software can profit from copyright's protection without revealing the expression that earns it that protection. Closed code thus poses an important but narrow question. Yet the technolegical tools that answer this question can address other, seemingly unrelated challenges that digital technology presents for copyright law as well. I focus below on two such challenges: the folly of maintaining copyright's automatic propertization of all original and fixed expression, and the danger that technological protections will replace copyright's careful balance of legal entitlements. As we will see, these two problems are a result of high-tech innovations, but they are best addressed through the resurrection of copyright's low-tech formalities.

A.   Curing Copyrightis

If intellectual property were a medical field, doctors would conclude that the body politic suffers from an inflammation of the copyright, characterized by swelling of the code, pain in the author and user regions, and a significant loss of function. The illness can be traced to an irritant that, if left untreated, threatens to become a full-blown infection that may kill the copyright corpus: the ubiquity of computer technology and digital networks. No single medication can cure this case of copyrightis, but we can start by trimming the fat—by ensuring that copyright applies only where it is needed.

Recall how easy it is to create a copyrightable work. Copyright's protection attaches as soon as original expression is fixed in a tangible medium. A copyrightable work is therefore a copyrighted work; copyright, as one astute commentator has observed, is not a transitive verb.[173] In an era when print predominated, this low threshold may have been reasonable. Distribution of information goods was an expensive process rarely undertaken by those with no desire to profit from their creativity. So although copyright might have protected, for

---

engineering, how long it is likely to be profitable, and whether it is likely to meet patent's strict threshold requirements. *Cf.* Samuelson & Scotchmer, *supra* note 62, at 1615–20 (discussing pros and cons of making application programming interfaces public versus keeping them private).

173   MacGrady, *supra* note 23, at 1049.

example, a personal letter from one correspondent to another, that protection was of little consequence because there would rarely be a circumstance in which the letter would be subject to unauthorized copying, adaptation, public distribution, or other conduct that might implicate its copyright. The architecture of information was such that mass distribution required reification—i.e., the reduction of creative content to a fixed *res* in multiple copies, with all the attendant expense that process implied.[174]

The advent of broadcasting posed a challenge for this print paradigm; for the first time, creative content could be delivered to the public at large in a de-reified form. The resulting growing pains were perhaps most evident in the Sony Betamax case, in which copyright owners challenged the legitimacy of a technology that gave their audience the ability to record and reproduce individual broadcasts.[175] But broadcasting did not completely transform the copyright landscape, because it required even greater resources and technological sophistication than its print predecessor and was therefore similarly restricted in practice to those with a mind to profit from their efforts. And despite its otherwise revolutionary nature, broadcasting perpetuated print media's stark division between creator and audience. It was a one-way street.

Copyright's commodification model therefore managed to survive the broadcasting revolution because the cast of content characters did not change significantly; only the medium was different. In contrast, ubiquitous digital networking represents a watershed technological development that does more than continue broadcasting's tradition of largely de-reified content. It changes the copyright equation along two other dimensions as well: it makes the creation and dissemination of content affordable to all, and it turns content delivery into an interactive two-way street. (See Table 1.) Digital architecture invites everyone into the copyright tent, and the traditional monied actors must accordingly make room for the amateurs. To accommodate these divergent interests, we need a more sophisticated—and limited—model of propertization.

Two developments have had a particularly important impact on how copyright works, or doesn't work, in a digital world. First, as already mentioned, the widespread accessibility of computers and the networks that connect them means that the production and widespread dissemination of creative expression is no longer a privilege restricted to those who can attract the attention of a publishing house

---

174   *See* Litman, *Revising Copyright, supra* note 45, at 37.
175   Sony Corp. of Am. v. Universal City Studios, Inc., 464 U.S. 417 (1984).

TABLE 1. EVOLUTION OF COPYRIGHT TECHNOLOGIES

	Medium		
	Print	Broadcasting	Digital
Need for fixed *res*	High	Low	Low
Cost of dissemination	High	High	Low
Interactivity	Low	Low	High

or media conglomerate. Combine this with copyright's low threshold requirements, and the result is the constant and ubiquitous creation and online dissemination of millions upon millions of copyrighted works.

Second, the everyday use of computer technology routinely results in unauthorized reproduction, adaptation, distribution, performance, and display of digitized content. Several courts have held, for example, that loading a program or file into a computer's active RAM memory constitutes copying for the purposes of copyright law,[176] despite legislative history that seems to support a contrary result[177] and vociferous dissent from copyright commentators.[178] Under these holdings, every time a copyrighted work is so much as viewed on a computer screen, the viewer needs either the permission of the copyright holder or the protection of a privilege—even if the disk or file from which the image is summoned was made with the copyright owner's permission and was lawfully purchased.[179] Even if RAM copies do not implicate copyright's exclusive rights, a host of other common computer activity does, from forwarding e-mail, backing up data, and printing a hard copy of an online document to caching fre-

---

176   The seminal case is *MAI Systems Corp. v. Peak Computer, Inc.*, 991 F.2d 511, 518–19 (9th Cir. 1993), but subsequent cases have reached the same result. *E.g.*, *In re* Indep. Serv. Org. Antitrust Litig., 910 F. Supp. 1537, 1541 (D. Kan. 1995); Advanced Computer Servs. of Mich., Inc. v. MAI Sys. Corp., 845 F. Supp. 356, 362–64 (E.D. Va. 1994).

177   H.R. REP. No. 94-1476, at 53 (1976), *as reprinted in* 1976 U.S.C.C.A.N. 5659, 5666 ("[T]he definition of 'fixation' would exclude from the concept purely evanescent or transient reproductions such as those . . . captured momentarily in the 'memory' of a computer.").

178   *E.g.*, James Boyle, *Intellectual Property Policy Online: A Young Person's Guide*, 10 HARV. J.L. & TECH. 47, 56 (1996); Lemley, *supra* note 92, at 551; Litman, *Revising Copyright*, *supra* note 45, at 21–22.

179   There exist some such privileges, such as the privilege to make such copies as are necessary to use the program, 17 U.S.C. § 117(a)(1) (2000), but none that would excuse the full range of seemingly legitimate uses of a lawfully acquired program or file.

quently accessed files, cataloging Internet sites, and webcasting one's travels.[180]

In tandem, these two developments produce a world replete with copyrighted content that is constantly being infringed. Although the copyright protection is often unneeded, and the infringement is often unobjectionable, the legal consequences are considerable. They include a poorer public domain, the conversion of copyright from an instrument of progress to an instrument of censorship, a failure of digital technology to realize its democratizing potential, and a woeful and regrettable lack of understanding of and respect for copyright law. Fortunately, the same technolegical tools that fixed the closed-code problem provide a solution here. Returning to copyright's tried-and-true formalities can help restore copyright to its intended, socially beneficial purposes.

1. The Gift and Curse of Widespread Authorship

In the digital age, we can all be authors, and we can all be publishers. Anyone with a computer and an Internet connection can command the attention of the wired world as long as he or she has something to say, or draw, or perform. This is a good thing. Empowering individual citizens in this unprecedented fashion promises to enrich our culture and democratize its content[181]—a development of particular value in a time when traditional media outlets are undergoing a bout of consolidation.[182] (Consider, for example, the valuable role that amateur weblogs play as a check on and rival to mainstream news reporting.) And in one respect this widespread empowerment makes copyright even more important, in that more authors can take advantage of copyright's incentives and commodify their creativity.

But only if they want to. And therein lies the problem. Because copyright protection attaches the instant an original thought is expressed in fixed form, these newly empowered authors never have a chance to affirmatively decide whether they want that protection. Of course, if digital authors who have no desire to profit from their crea-

---

180   For a thoughtful discussion of these and other ways in which routine use of a computer can be said to implicate copyright rights, see Lemley, *supra* note 92, at 554–63. On the implications of caching, see I. Trotter Hardy, *Computer RAM "Copies": Hit or Myth? Historical Perspectives on Caching as a Microcosm of Current Copyright Concerns,* 22 U. DAYTON L. REV. 423 (1997).

181   LESSIG, *supra* note 92, at 42.

182   *See* Yochai Benkler, *Intellectual Property and the Organization of Information Production,* 22 INT'L REV. L. & ECON. 81, 93–95 (2002) (arguing that strong intellectual property entitlements lead to increased concentration and homogenization of information production).

tivity are sufficiently organized and attuned to copyright concerns, they can manufacture their own legal escape hatches; open-source licensing is the most obvious example. But because modern-day acts of creative expression are so ubiquitous and unremarkable—writing e-mails, posting to message boards, creating websites, sharing photos online—we cannot expect a diffuse public to take the time to prioritize the enrichment of the public domain or the digital commons.

For example, suppose that a regular Joe takes a break from his two jobs and four children to write a movie review and send it to an electronic mailing list of film afficionados. The listserv technology that this community uses automatically makes hundreds of copies of his review—one copy appears in the e-mail inbox of every member of the list, others reside (at least for a time) on intermediate servers that facilitate the transmission of Internet communications, another appears in an archive that the list moderator maintains, and so forth. Now, one might reasonably argue that Joe authorized these copies by sending his review in the first place, and that copyright law accordingly works fine here. But what is the legal consequence when a listserv member decides to print out his review, or forward it to a friend, or place it on his or her own website?

The right answer, in all likelihood, is "Who cares?" Not Joe, certainly. But copyright law cares, because copyright law protects that movie review from the moment Joe saves it to his hard drive—or even from the moment he types it into the active memory of his word processing program.[183] We saddle Joe with an entitlement he probably cares nothing about and thereby saddle the public with three unappealing options: invest the time and resources necessary to trace the provenance of online content and secure its author's permission, break the law by making and distributing unauthorized copies, or forgo a wholly unobjectionable and socially enriching use of creative material. We need a fourth option.

2. Copyright as Censorship

This is not to say that no injury ever results from the unauthorized copying and distribution of personal extemporanea. But it tends not to be a cognizable copyright injury. Consider the example of investment banker Peter Chung, whose overly creative e-mail to friends about his decadent lifestyle cost him his dignity, and eventually his job, when it left the private circle of intended recipients and made

---

183   *See supra* note 176 and accompanying text.

the public rounds of the Internet.[184] Chung and others surprised to
see their private affairs laid bare to the online world can certainly ar-
ticulate an injury. The harm, however, is not the economic loss that
results from the usurpation of the right to profit from the copying and
distribution of a creative work. It is instead more akin to a violation of
the author's privacy. Concerns of that type have traditionally been the
province not of federal copyright, but of state law and its now mori-
bund system of common-law copyright.[185] The federal copyright
model grants alienable and descendible property rights to facilitate
the economic regulation of materials intended for widespread public
dissemination. In contrast, as Samuel Warren and Louis Brandeis
pointed out over one hundred years ago in their seminal article *The
Right to Privacy*,

> where the value of the production is found not in the right to take
> the profits arising from publication, but in the peace of mind or the
> relief afforded by the ability to prevent any publication at all, it is
> difficult to regard the right as one of property, in the common ac-
> ceptation of that term.[186]

These concerns sound in privacy, not property, and are thus not the
province of federal copyright at all.[187]

Laws governing privacy should therefore provide any relief
deemed appropriate for this kind of injury. But because federal copy-
right now covers unpublished works, the state law standby that would
have addressed the privacy concerns in the days of Warren and Bran-
deis—i.e., common-law copyright—is not available. Federal copyright
law has taken its place. Thus we see federal copyright employed to
keep information from reaching the public. Rap music star Eminem
enjoins a magazine from publishing the lyrics to racist songs that he
wrote in his youth.[188] The Church of Scientology shields its scripture

---

184   Editorial, *How Those Cyber Slip-Ups Could Get You the Sack*, IRISH INDEP. (Dublin),
Sept. 29, 2004, at 16 (recounting Chung's and similar e-mail mishaps).

185   Although state copyright law was sometimes statutory in nature, the accepted
practice is to refer to it as common-law copyright. 1 NIMMER & NIMMER, *supra* note 51,
§ 2.02, at 2-18 & n.1.

186   Samuel D. Warren & Louis D. Brandeis, *The Right to Privacy*, 4 HARV. L. REV.
193, 200–01 (1890).

187   *See* Harper & Row, Publishers, Inc. v. Nation Enters., 471 U.S. 539, 554 (1985)
(citing *The Right to Privacy* for proposition that "common-law copyright was often en-
listed in the service of personal privacy"); Warren & Brandeis, *supra* note 186, at 205
("The principle which protects personal writings and all other personal produc-
tions . . . against publication in any form, is in reality not the principle of private
property, but that of an inviolate personality.").

188   Shady Records, Inc. v. Source Enters., Inc., 351 F. Supp. 2d 64, 69 (S.D.N.Y.
2004) (finding magazine in contempt for violating temporary restraining order that

from public examination by invoking copyright protection.[189] The famously reclusive author J.D. Salinger prevents the publication of a biography containing excerpts and paraphrases of his writings.[190] Dunkin' Donuts uses the threat of copyright infringement to facilitate its purchase of an online forum for complaints about the company, claiming that it could "more effectively capture the comments and inquiries that are being submitted by our customers"—yet the original website no longer exists, and the company's own site provides no consumer forum.[191]

In none of these examples did the authors intend to publish the works on their own; the goal was to keep them from ever reaching the public at large. Copyright is ill suited to deal with such cases. Its constitutional goal is to promote enlightenment, not retard it—"to increase and not to impede the harvest of knowledge."[192] The property rights it establishes must "ultimately serve the cause of promoting broad public availability of literature, music, and the other arts."[193] Copyright law thus operates under the assumption that the author wishes to disseminate his or her work to the public for a fee, and that the only thing standing in the way is the threat of unauthorized (i.e., uncompensated) copying.

When the opposite is true—when the goal is privacy, not profits—copyright often gets it wrong. Because copyright law assumes that authors' desire for control derives from their desire to market their works, courts presume an irreparable injury whenever infringement takes place.[194] This makes it remarkably easy for a copyright owner to secure an injunction, a prior restraint on speech that would be ex-

---

barred it from publishing lyrics). *But see* Shady Records, Inc. v. Source Enters., Inc., No. 03 Civ. 9944, 2005 WL 14920, at *19 (S.D.N.Y. Jan. 3, 2004) (sending case to trial on fair use issue).

189   Religious Tech. Ctr. v. Lerma, CIV.A. No. 95-1107-A, 1996 WL 633131, at *1 (E.D. Va. Oct. 4, 1996); *see also* Bridge Publ'ns, Inc. v. Vien, 827 F. Supp. 629, 635–36 (S.D. Cal. 1993) (finding no fair use of secret Scientology texts where defendant used them in for-profit course).

190   Salinger v. Random House, Inc., 811 F.2d 90 (2d Cir. 1987).

191   Fara Warner et al., *Holes in the Net*, WALL ST. J., Aug. 30, 1999, at A1 (describing Dunkin' Donuts suit).

192   *Harper & Row*, 471 U.S. at 545; *see also* U.S. CONST. art. I, § 8 (giving Congress copyright power in order to "promote the Progress of Science and useful Arts").

193   Twentieth Century Music Corp. v. Aiken, 422 U.S. 151, 156 (1975).

194   *E.g.*, Hasbro Bradley, Inc. v. Sparkle Toys, Inc., 780 F.2d 189, 192 (2d Cir. 1985); Apple Computer, Inc. v. Formula Int'l, Inc., 725 F.2d 521, 525 (9th Cir. 1984); Apple Computer, Inc. v. Franklin Computer Corp., 714 F.2d 1240, 1254 (3d Cir. 1983); Atari, Inc. v. N. Am. Philips Consumer Elecs. Corp., 672 F.2d 607, 620 (7th Cir. 1982).

ceedingly difficult to get outside the copyright context.[195] When copyright's operating assumption is correct, the prior restraint is not particularly troubling because it is merely a short-term sacrifice in pursuit of the greater goal of encouraging the creation and dissemination of the expression at issue. Where the author never intends to disseminate his or her expression, however, copyright serves not as "the engine of free expression"[196] but as an instrument of speech suppression. The laudable "right of first publication" becomes a lamentable "right of no publication."

This serves as a rebuttal to an issue that has been looming over my litany of complaints about "copyrightis": if online authors generally don't care about the entitlement that copyright automatically grants them, and their online audience also routinely ignores the entitlement and proceeds to copy and distribute at will, then why is copyrightis a problem at all? We now see one answer: it allows those who want to keep their works private to use copyright as an instrument of censorship.

One might expect copyright law's main First Amendment stand-in, the fair use doctrine, to mediate this dispute between privacy and public interest.[197] Fair use jurisprudence, however, has labored under the same erroneous assumption about authors' desire to profit. Courts usually view one of the four fair use factors, the nature of the copyrighted work,[198] as weighing against the copyist whenever the work is unpublished—regardless of whether the author ever intends to publish.[199] Likewise, when it comes to the fourth and most impor-

---

195   *See, e.g.,* Bantam Books, Inc. v. Sullivan, 372 U.S. 58, 70 (1963) ("Any system of prior restraints of expression comes to this Court bearing a heavy presumption against its constitutional validity.").

196   *Harper & Row,* 471 U.S. at 558 ("[T]he Framers intended copyright itself to be the engine of free expression.").

197   *See* Eldred v. Ashcroft, 537 U.S. 186, 219–20 (2003) (listing fair use among copyright's "built-in First Amendment accommodations").

198   17 U.S.C. § 107(2) (2000).

199   *E.g.,* Salinger v. Random House, Inc., 811 F.2d 90, 97 (2d Cir. 1987) (finding that second factor weighs heavily in author's favor when work is unpublished even absent intent to publish); Religious Tech. Ctr. v. Lerma, CIV.A. No. 95-1107-A, 1996 WL 633131, at *7–8 (E.D. Va. Oct. 4, 1996) (rejecting argument that "where a copyright owner intends never to exploit the right of first publication, the need to protect that right diminishes and the scope of fair use correspondingly expands"). *But see* Online Policy Group v. Diebold, Inc., 337 F. Supp. 2d 1195, 1203 n.13 (N.D. Cal. 2004) (attaching no weight to "first publication right" when author "clearly has indicated that it never intended to publish"); WILLIAM A. LANDES & RICHARD A. POSNER, THE ECONOMIC STRUCTURE OF INTELLECTUAL PROPERTY LAW 131–42 (2003) (arguing that unpublished works whose author never intends to publish are particularly good candidates for fair use).

tant fair use factor, the use's effect on the market for the work,[200] courts again ignore whether the author ever intends to publish the work and thus find a market effect when there is no real market to be affected.[201] This case law and the already uncertain nature of a fair use defense produce a chilling effect: even those who seem to have a good shot at winning a fair use argument cannot afford to take the issue to court.[202]

Another of copyright's free speech mechanisms, the idea/expression dichotomy,[203] has played a role in the copyright-as-censorship cases, but no more successfully than fair use. The author's argument here is that the copyist could have conveyed the same information through paraphrasing rather than verbatim copying, and it is an argument that some courts have accepted.[204] But even paraphrasing does not always save the defendant,[205] and in any event it misses the point. Copyright exists to incentivize the public dissemination of information. The copyist should not have to conform to the particularities of copyright's idea/expression dichotomy to escape liability when the author intends to withhold his or her expression from the public. Instead, copyright should play no role at all.

---

200   17 U.S.C. § 107(4).

201   *E.g., Salinger*, 811 F.2d at 99 ("[T]he effect on the market for Salinger's letters is not lessened by the fact that their author has disavowed any intention to publish them during his lifetime.").

202   *See, e.g.*, LESSIG, *supra* note 92, at 95–99 (describing documentary filmmaker's decision not to engage in a seemingly obvious fair use because of fear of litigation and other practical obstacles); Richard Byrne, *Silent Treatment*, CHRON. HIGHER EDUC., July 16, 2004, at A14 (describing recall of book due to mere threat of lawsuit despite arguable fair use defense for excerpting ninety-four lines of material).

203   "[C]opyright's idea/expression dichotomy 'strike[s] a definitional balance between the First Amendment and the Copyright Act by permitting free communication of facts while still protecting an author's expression.'" Harper & Row, Publishers, Inc. v. Nation Enters., 471 U.S. 539, 556 (1985) (quoting Harper & Row, Publishers, Inc. v. Nation Enters., 723 F.2d 195, 203 (2d Cir. 1983), *rev'd*, 471 U.S. 539) (second alteration in original); *see also* Eldred v. Ashcroft, 537 U.S. 186, 219 (2003) (listing idea/expression dichotomy among copyright's "built-in First Amendment accommodations").

204   *See, e.g.*, N.Y. Times Co. v. United States, 403 U.S. 713, 726 n.* (1971) (Brennan, J., concurring) (distinguishing copyright cases from other cases involving prior restraints because "copyright laws, of course, protect only the form of expression and not the ideas expressed").

205   *Salinger*, 811 F.2d at 97 ("[P]rotected expression has been 'used' whether it has been quoted verbatim or only paraphrased.").

3.   Resurrecting the Formalities

Copyright's current role as an instrument of censorship is ironic, given that it was a deliberate renunciation of censorship that gave birth to modern copyright law almost three hundred years ago. In 1710, the British Parliament passed the Statute of Anne,[206] which rejected the existing copyright system under which the Tudor monarchs and their successors had granted royal monopolies to printing houses as a means to control the content of books.[207] Parliament focused instead on a new notion—i.e., that copyright should provide a property right to authors, not publishers, with the goal being "the Encouragement of Learned Men to Compose and Write useful Books."[208]

Today, the ubiquity of digital architecture serves to Encourage an unprecedented number of Learned Men (and Women) to Write useful Books, and more. Yet as information technology has advanced, copyright has regressed, becoming too omnipresent and too intrusive. Fortunately, we can find an answer to this dilemma in copyright's past as well: the venerable formalities.

a.   Publication

If copyright should play no role in regulating the copying and distribution of a work whose author never intends to publish it, then the obvious solution is to resurrect publication as a threshold requirement for federal copyright protection. Copyright should ensure, rather than assume, that publication is the author's goal by withholding its exclusive rights from those who have not published their works or who have no demonstrable plans to publish. The "right of first publication" would thus remain a copyright matter, but the "right of no publication" would revert to privacy law.

Of course, the law is not the primary regulator of behavior in the privacy realm. That honor belongs not to legal modalities, but to architectural modalities. Consider the enormous extralegal, real world advantage that privacy-minded plaintiffs possess: if they do not want their writings made public, they can simply lock them up. Indeed, the law has traditionally made such architectural efforts a precondition to its grant of certain entitlements. Privacy torts apply only to disclosure

---

206   An Act for the Encouragement of Learning, by Vesting the Copies of Printed Books in the Authors or Purchasers of such Copies, during the Times therein mentioned, 1710, 8 Ann., c. 19 (Eng.) [hereinafter Statute of Anne].
207   Howard B. Abrams, *The Historic Foundation of American Copyright Law: Exploding the Myth of Common Law Copyright*, 29 WAYNE L. REV. 1119, 1134–42 (1983).
208   Statute of Anne pmbl.

of that which the plaintiff has not left open to public inspection.[209] Whether you can sue for trespass or ejectment may depend on whether you built and maintained a fence to keep out unwanted traffic.[210] The repo man can take a debtor's car from the driveway, but not from the padlocked shed behind the house.[211]

Intellectual property law routinely uses similar preconditions to separate privacy-minded innovators from those who want to market their creations using an alienable property right. Trade secret protection of an unpublished innovation is conditioned on the inventor's desire to keep the innovation private and on the architectural and legal steps he or she takes to realize that goal.[212] If the inventor chooses instead to market the innovation, and thus reveal it to the public, then a legal regime designed to facilitate commodification intervenes—patent law—and substitutes a property right for trade secret's privacy interest.[213] Likewise, a private citizen may bring a common-law tort claim for invasion of privacy when his or her likeness is used in trade without consent.[214] But when the injury derives from uncompensated publicity rather than unwanted publicity—think of professional athletes or actors who choose to market their celebrity—the cause of action is based on the property entitlement found in rights of publicity, not the personal injury underpinnings of privacy law.[215]

Resurrecting publication as a copyright requirement simply applies this sensible and seasoned intellectual property distinction to the protection of works of authorship. If an author prefers privacy to publication, then a legal regime designed to protect privacy should govern—i.e., privacy torts and other state law premised on architectural

---

209 RESTATEMENT (SECOND) OF TORTS § 652D cmt. b (1977) ("[T]here is no liability for giving further publicity to what the plaintiff himself leaves open to the public eye.").

210 *E.g.*, Rice v. Miller, 238 N.W.2d 609 (Minn. 1976) (holding that realty owner successfully defeated claim of prescriptive easement by physically blocking path); *see also* Lofland v. Truitt, 260 A.2d 909 (Del. Ch. 1969) (finding prescriptive easement where realty owner failed to maintain impediments to trespass).

211 Butler v. Ford Motor Credit Co., 829 F.2d 568 (5th Cir. 1987) (holding that Mississippi law allows secured creditor to repossess debtor's vehicle from debtor's driveway); Henderson v. Sec. Nat'l Bank, 140 Cal. Rptr. 388 (Ct. App. 1977) (holding that repossession of debtor's vehicle from locked garage constituted conversion).

212 Gibson, *supra* note 15, at 220.

213 *See* Lichtman, *supra* note 119, at 231 & n.43.

214 *See* RESTATEMENT (SECOND) OF TORTS § 652C & cmt. b.

215 *See* Michael Madow, *Private Ownership of Public Image: Popular Culture and Publicity Rights*, 81 CAL. L. REV. 127, 167–78 (1993) (discussing propertization of right of privacy).

efforts to maintain secrecy.[216] If instead the author wants to market the work and profit therefrom, then a legal regime designed to facilitate that choice and convey a property right should govern: federal copyright law. As mentioned above, the old definition of publication would need some updating so as to cover both the traditional notion of distribution through print copies and latter-day notions of distribution through broadcasting, streaming media, and other more ephemeral means of reaching the public. But once we make this adjustment, we will once again have a copyright system that encourages both creation and dissemination of creative expression.

b. Notice

Of course, resurrecting publication as a sine qua non of copyright will cure only those symptoms of copyrightis that have to do with the specter of censorship. It does nothing for Joe, our amateur movie reviewer who wants to publish yet cares nothing for copyright protection. Here, however, we can call on another venerable copyright formality—notice—to provide the answer. The argument against notice as a copyright prerequisite in the print era was that few authors would incur the expense of publishing a work absent an intent to profit. Why then should the law require them to jump through an additional hoop simply to demonstrate their desire for a copyright? As discussed above, however, it makes little sense in the age of the Internet for copyright to attach upon mere creation, or even publication, of a work when the instruments of creation and publication are both widely available and practically free.

Copyright should accordingly require for-profit authors to declare their commercial intent by displaying a copyright notice on their works, as was the case for all but twenty-eight years of U.S. copyright history.[217] Those many other authors who prefer not to restrict the

---

216   States could revive the standards of common-law copyright to regulate this behavior, but the nascent privacy entitlements that Warren and Brandeis first articulated in 1890 may by now have matured enough to provide relief without invoking the incongruous rubric of "copyright." *See, e.g.,* RESTATEMENT (SECOND) OF TORTS § 652A (restating modern rights of privacy); *cf.* David W. Melville & Harvey S. Perlman, *Protection for Works of Authorship Through the Law of Unfair Competition: Right of Publicity and Common Law Copyright Reconsidered,* 42 ST. LOUIS U. L.J. 363, 367 (1998) (arguing that "creation of rules for artistic works analogous to those governing trade secrets would provide sufficient protection for unfixed works").

217   *See* Sprigman, *supra* note 2, at 491–95; *see also* Douglas Lichtman, *Copyright as a Rule of Evidence,* 52 DUKE L.J. 683, 740 (2003) ("[C]opyright law might want to distinguish authors whose expressive activities are motivated by copyright from authors for whom copyright was an afterthought.").

copying and dissemination of their works could simply refrain from affixing a notice and thereby enrich the public domain. Copyright could even develop intermediate sets of entitlements through more sophisticated and flexible forms of notice, as open source does for software: one notice when the author wants full copyright protection, another when the only desideratum is attribution, another when copying is permitted but not adaptation, and so forth. To some extent, the Creative Commons movement has done this by providing the public with a set of copyright licenses and associated notices that it can use to publish its works.[218] (For example, an author who objects only to unauthorized commercial use of his or her work can affix a Creative Commons "Some Rights Reserved" notice to it, which directs the viewer to the corresponding license.[219]) But because Creative Commons is a voluntary grassroots effort, it can neither compel use of its notices nor be assured of attracting a critical mass of authors—let alone those authors whose cooperation is needed the most: the for-profit, full-copyright crowd.[220] In contrast, a top-down, government mandated requirement would both ensure universal participation in the notice regime and provide a more comprehensive starting point for flexible, intermediate licensing.[221]

A resurrected notice requirement could do more than facilitate intermediate licensing and donations to the public domain. Like its pre-1989 predecessor, the new notice could include the author's name and the year that the work's copyright term began to run—i.e., the year of publication—and could thereby give the public some idea of when the copyright is due to expire. This would work well for institutionally authored works, for which the date of expiration depends in

---

218   *See* Creative Commons, Choosing a License, http://creativecommons.org/about/licenses (last visited Sept. 5, 2005). Christopher Sprigman has suggested using Creative Commons licenses as a model for a flexible, government-backed menu of notices. Sprigman, *supra* note 2, at 564.

219   *See* Creative Commons, A Spectrum of Rights, http://creativecommons.org/about/licenses/comics1 (last visited Sept. 5, 2005).

220   Another problem with a bottom-up system of licensing is the risk that its decentralized nature might lead to a proliferation of confusing and poorly drafted licenses, thus undermining the flexibility that is its greatest advantage. *Cf.* Robert W. Gomulkiewicz, *De-Bugging Open Source Software Licensing*, 64 U. PITT. L. REV. 75 (2002) (calling for standards organization that can keep open-source licenses consistent and comprehensible).

221   *See* Merges, *supra* note 148, at 201–02 (noting that "no private initiative will ever quite match the ability of the statute to channel copyright owners into a uniform, widely understood standard practice" and proposing government-sanctioned intermediate license); Samuelson et al., *supra* note 113, at 2426–29 (exploring use of software depository as centralized exchange for low-cost licensing).

part on when the work was first published.[222] For individual authors, the notice requirement would work less well, because expiration depends on when the author dies, rather than on the date of publication[223]—but even here notice would provide helpful information: the name of the author whose death would trigger the expiration clock. And in both cases, including the name of the copyright owner in the notice would lower search costs for potential licensees.

The main downside of resurrecting the notice formality is that unwary authors will lose their copyrights if they fail to affix a notice. But for almost the entirety of American copyright history, authors faced this consequence—yet the creative culture still flourished. In any event, the alternative is worse: automatically attaching copyright to millions of works whose authors would prefer to see them enter the public domain, or at least be governed by some intermediate, more permissive regime. In other words, *someone* has to comply with a legal formality to achieve their goals: either market-minded authors or altruistic authors. It is the former who seek to profit from a government entitlement, and it therefore makes sense for the burden to fall on them.[224]

Moreover, the law can assuage the harsh consequence of a notice requirement by adopting an evenhanded, equitable approach to forfeit of copyright—e.g., by allowing an author to cure any inadvertent omission of notice within a reasonable period of time after publication. Innocent infringement that took place before the omission was discovered might be subject to injunction only, rather than damages, with the unwitting infringer being allowed to complete any enterprise undertaken in reasonable reliance on the absence of notice. Purposeful removal or alteration of another's notice would be subject to harsh punishment.[225]

---

222 The copyright term for works created by institutional authors today is ninety-five years from publication or 120 years from creation, whichever is sooner. 17 U.S.C. § 302(c) (2000). If publication were revived as a threshold requirement, we could either get rid of the alternative 120-year term (perhaps as an incentive for authors to publish earlier) or keep it. In either event, including the date of first publication within the notice would allow the public to calculate a date after which the copyright would definitely have expired.

223   *Id.* § 302(a).

224   "If welfare recipients can be denied their benefits because they fail to complete a benefits form properly, then I can't see the unfairness in requiring those who demand state support to defend their monopoly similarly by filling out a registration form." LESSIG, *supra* note 47, at 251.

225   Many of these safeguards were part of copyright law before the formalities withered away. *See* 1961 REPORT, *supra* note 150, at 61–67.

Some forms of content distribution might be incompatible with notice. For example, it would be difficult to "affix" a notice to a live musical performance. The same might be said for digitally streamed audio, although the more high-tech the distribution method the more likely that some effective form of notice could be coded into the software and hardware.[226] In these instances, however, the requirement would not be as important because the form of distribution would be inherently resistant to unauthorized copying. It could therefore be waived, and the public would be presumed to know—as it is now—which content is covered by copyright.

c. Registration, Renewal, and Deposit

A final cure for copyrightis can be found in the registration formality. Requiring authors to register their works as a condition of copyright would serve much the same purpose as the notice requirement, in that those with no desire to obtain copyright protection could simply forgo registration and thereby donate their works to the public domain. In some instances registration may be better suited for this role because it would be feasible even when the author's primary means of distribution precludes effective notice. Indeed, under the predecessor to the current copyright act, works distributed in a format that resisted notice—such as plays, lectures, and musical performances—could secure copyright protection through registration and deposit alone.[227]

But registration best aids copyright's cause not as a declaration of intent to secure copyright in the first place, but as a declaration of intent to *continue* one's copyright. The theory here is simple: most works that succeed commercially do so within the first few years of their copyright. Continuing the copyright after that point thus often impoverishes the public domain, with no countervailing benefit to the author. For example, between 1883 and 1964, the copyright term could be doubled from twenty-eight years to fifty-six if the author simply filed for renewal, yet less than eleven percent of copyrights were renewed, even though renewal was a relatively painless and cheap process.[228]

---

226   *See, e.g.,* Creative Commons, Metadata Embedding, http://creativecommons. org/technology/embedding (last visited Sept. 5, 2005) (discussing how to embed Creative Commons licenses in code). *But see infra* Part III.B.2 (arguing that copyright should not rely extensively on high-tech solutions to problems that digital technology presents).

227   *See supra* note 150.

228   William M. Landes & Richard A. Posner, *Indefinitely Renewable Copyright,* 70 U. CHI. L. REV. 471, 473 (2003).

Today copyright lasts much longer (a minimum of seventy years) with no need for renewal. Requiring authors to register their works and then renew the registration after a period of years would therefore release into the public domain works that were once—but are no longer—commercially viable; their copyright owners would not bother going through the trouble of renewing even at a low cost. For this reason, commentators from William Landes and Richard Posner to Lawrence Lessig have proposed reviving the registration/renewal requirement.[229]

Registration provides other benefits as well. For example, registration has always required the deposit of one or more copies of the work.[230] Making these copies available for public inspection[231] or (better yet) for inclusion in the collection of the Library of Congress[232] directly promotes copyright's goal of dissemination of creative expression to the public without significantly affecting authorial incentive. Of course, if every copyright-eligible work were to be archived in this way, storage might become problematic, even for a digital depository. But as already discussed, one of the benefits of making copyright an opt-in system is that many works would not be registered, as many authors would not care about protecting their expression, so not all copyright-eligible works would be archived. In any event, current deposit regulations exclude certain categories of works,[233] a practice that could continue if storage problems arise.

Perhaps most important, registration could help lower troublesome search costs. Consider that a potential licensee of a work must incur the expense of identifying and tracking down the copyright owner before licensing negotiations can even begin. If the copyright owner's name and address are not readily available, these search costs might prove prohibitive, even when the copyright owner would gladly

---

229   LESSIG, *supra* note 47, at 251 (proposing series of fifteen renewable five-year terms); Landes & Posner, *supra* note 228, at 518 (concluding that "a system of indefinite renewals (or one that combines renewals with a maximum duration) may enable society to have its cake and eat it too").

230   1961 REPORT, *supra* note 150, at 71.

231   *See supra* notes 167–68 and accompanying text.

232   Current copyright law ostensibly requires deposit of published works to the collections of the Library of Congress, but the requirement is almost completely toothless. The statute makes it abundantly clear that such deposit is not a prerequisite to copyright protection and cannot lead to forfeiture of one's copyright. The only penalty for noncompliance is a small fine, which is not imposed unless the register of copyrights first makes a written demand that goes unheeded. 17 U.S.C. § 407(a), (d) (2000).

233   *Id.* § 407(c); 37 C.F.R. § 202.19(c) (2004).

have issued the license for a reasonable price, or for free.[234] If the law required authors to include their names in a copyright notice and record any subsequent assignments of copyright in a public registry, these costs could be avoided or significantly reduced. Indeed, the law imposes strict recordation requirements on those who transfer interests in real property, even though the real property often provides sufficient indications of ownership on its face.[235] Why then should transfers of interest in intellectual property—whose provenance is inherently difficult to ascertain—not be subject to equally strict recordation rules?

Here again we can look to old-school copyright for a model. The first federal Copyright Act required registration of each work in the office of local federal district court clerk, with a strict recordation procedure, sealed certificate of title, and announcement of the registration in the newspaper.[236] The first major copyright revision, in 1831, further required the local registries to be sent to the Secretary of State annually,[237] and three years later Congress mandated the recordation of all written assignment instruments transferring copyright ownership.[238] The statutes were unclear as to whether the registries were open to public inspection, but the predecessor to the current act, passed in 1909, cleared up any ambiguity; it explicitly provided for public inspection and also required indexed catalogs of all registrations to be distributed regularly to postmasters and customs officers and made available to the public for a small fee.[239] Even more significant was a provision that voided any assignment of a copyright not recorded with the register of copyrights.[240]

The current statute continues to provide for registration, but it is mandatory only when the copyright owner wants to file suit, and then only for U.S. works.[241] It also allows (but does not require) a party to record any transfer of its interest in copyright, or indeed of any document "pertaining to a copyright"; a sufficiently detailed recordation constitutes constructive notice of the transfer if the work itself is regis-

---

234   Eldred v. Ashcroft, 537 U.S. 186, 248 (2003) (Breyer, J., dissenting) (discussing "search costs that themselves may prevent reproduction even where the author has no objection"); *see also* Orphan Works, 70 Fed. Reg. 3739 (Jan. 26, 2005) (requesting comment for Copyright Office study of works whose owners are difficult to locate).

235   Sprigman, *supra* note 2, at 500.

236   Act of May 31, 1790, ch. 15, § 3, 1 Stat. 124, 125.

237   Act of Feb. 3, 1831, ch. 16, § 4, 4 Stat. 436, 437.

238   Act of June 30, 1834, ch. 157, § 1, 4 Stat. 728, 728.

239   Act of Mar. 4, 1909, ch. 320, §§ 56–58, 35 Stat. 1075, 1086.

240   *Id.* § 44, 35 Stat. at 1084.

241   *See supra* note 158.

tered.[242] (Curiously, Congress has also enacted a recordation scheme specific to shareware.[243]) Copyright Office records are indexed and open to public inspection, and for a fee the Office will itself conduct a search of its records.[244]

Copyright's history thus provides a model for mandatory copyright recordation, and current law provides a modern-day framework. All that remains is to combine the two into a compulsory system. Of course, making any formality mandatory imposes new costs on authors—or, more accurately, revives old costs. In the print era, however, authors managed to bear them without impoverishing themselves or our culture. And today the costs should be even less prohibitive, because we can use the Internet and automated systems to streamline compliance with the registration requirement and other formalities.[245]

4.   Digital Democratization

The foregoing discussion explains how copyright's formalities can address discrete problems in copyright law, such as censorship, a needlessly impoverished public domain, confusion about what entitlements govern what works, and high search costs for potential licensees. If resurrecting the formalities did no more than that, it would be a worthwhile endeavor. When we consider their collective effect, however, a final benefit appears—one that transcends the details of copyright policy and helps fulfill the democratizing promise of the information age.

As already discussed, the development of digital architecture and the Internet represents an important second stage in the democratization of copyrighted content. The initial stage was the development of broadcasting technology, which first enabled content providers to liberate content from a specific *res* and deliver it to everyone in a more ephemeral form. Digital networks have taken the broadcasting revolution one step further by turning it into a two-way street. In an

---

242   17 U.S.C. § 205(a), (c) (2000). Recordation is mandatory only when a party wishes to perfect a security interest in a registered copyright. *See, e.g., In re* World Auxiliary Power Co., 303 F.3d 1120, 1128–32 (9th Cir. 2002).

243   Computer Software Rental Amendments Act of 1990, Pub. L. No. 101-650, § 805, 104 Stat. 5089, 5136; *see also* 37 C.F.R. § 201.26 (2004) (implementing regulations).

244   17 U.S.C. § 705.

245   *See* Sprigman, *supra* note 2, at 546; Robert E. Kahn, *Deposit, Registration and Recordation in an Electronic Copyright Management System,* http://www.cni.org/docs/ima. ip-workshop/Kahn.html (last updated July 3, 2002).

online world, everyone can be both a content provider and an audience member.

The architecture of digital networks is thus inherently egalitarian. Therefore, as Yochai Benkler has cogently argued, nondiscriminatory access to a free and universally available public network can significantly advance collective goals of productivity, democracy, autonomy, and social justice.[246] Yet even if that architectural condition is satisfied—i.e., even if everyone has free access to the Internet—such access would fail to realize its democratizing potential if online content is subject to clumsy default rules that automatically propertize anything that satisfies minimal standards of fixation and originality.

In other words, copyright's current rules keep us from achieving legally what is now possible (and desirable) architecturally. Under these rules, members of the public face two equally objectionable choices when they go online. They can either ignore copyright law, and thus use the expression of others as they see fit, or they can comply with copyright law, and thus refrain from all copying and sharing of anything that looks like creative expression. The former option harms the legitimate interests of for-profit authors. The latter option unacceptably hinders participation in our newly accessible democratic culture.

Small wonder, then, that piracy is rampant and that surveys show the prevailing attitude toward copyright is one of either incomprehension or disrespect.[247] Given copyright's complexity and resistance to the layperson's expectations, can we be surprised at the widespread downloading of copyrighted music (to cite one prominent example)? Those downloaders who don't understand copyright law probably think they are doing nothing wrong. Those who do understand copyright law also understand that millions of acts of online infringement take place every day—the forwarding of e-mails, the caching of web pages, etc.—and that no one cares. Why then should they expect any-

---

246   Yochai Benkler, Professor of Law, Yale Law School, Freedom in the Commons: Towards a Political Economy of Information, Lecture at the Second Annual Meredith and Kip Frey Lecture in Intellectual Property at Duke Law School (Mar. 22, 2002), *in* 52 DUKE L.J. 1245, 1246–47 (2003).

247   One survey found that sixty-one percent of those who download music don't care about its copyright status, and just thirty-one percent say that it's a concern for them. AMANDA LENHART & SUSANNAH FOX, DOWNLOADING FREE MUSIC: INTERNET MUSIC LOVERS DON'T THINK IT'S STEALING 12 (2000), *available at* http://www.pewinternet.org/pdfs/PIP_Online_Music_Report2.pdf. In another survey, more than twice as many respondents thought it was a serious crime to drive forty miles per hour when the speed limit is twenty-five than to make illegal copies of software. RICK HARBAUGH & RAHUL KHEMKA, DOES COPYRIGHT ENFORCEMENT ENCOURAGE PIRACY? 6 (2001), *available at* http://econ.mckenna.edu/papers/2000-14.pdf.

thing more than a wink and a nod when they engage in other forms of digital infringement? The problem thus extends beyond copyright education to copyright substance. Even an educated public cannot be expected to respect and consistently conform to copyright law if copyright owners themselves admit that most digital infringement is inconsequential.

Resurrecting copyright's formalities will bring the reality of copyright closer to the public's reasonable expectations. By inhibiting censorship, reducing the amount of unnecessarily protected content, and attaching a notice to those works that remain copyrighted, the formalities eliminate significant legal obstacles to realizing network technology's architectural promise and are thus likely to improve public respect for, and compliance with, copyright law. If copyright is no longer an impenetrable and counterintuitive morass of rules, the layperson is more likely to comply with its rules and embrace a culture of creativity, both online and offline.[248]

B. Battling Technological Hegemony

1. The Threat

Resurrecting copyright's formalities also addresses one last troublesome issue at the intersection of digital architecture and copyright law: the well documented possibility that information technology will supplant copyright law as the primary regulatory modality for works of creative expression.[249] Because copyright exists not to enrich authors, but to enrich the culture, it provides both private property rights and public privileges. For example, its exclusive rights expire after a set period of time,[250] to allow the public unfettered use of a work and to prevent authors from extracting monopoly rents after they have earned a sufficient return on their investment in creativity. Likewise, the first sale doctrine prevents copyright owners from controlling the aftermarket for copies of their works.[251] The idea/expression dichot-

---

248   Ann Bartow, *Electrifying Copyright Norms and Making Cyberspace More Like a Book,* 48 VILL. L. REV. 13, 18 (2003) ("[I]f individuals could access and use digitalized copyrighted works in old, familiar ways, perhaps they would also restrain themselves to analog levels of unauthorized copyright uses and infringements."); Litman, *Revising Copyright, supra* note 45, at 44 (arguing that "conforming the law more closely to popular expectations" would not only "ease enforcement" but also "make mass education about the benefits of intellectual property law more appealing" (footnote omitted)).

249   *See, e.g.,* Lessig, *supra* note 13, at 507–10 (exploring nonlegal possibilities for regulating intellectual property); Joel R. Reidenberg, *Lex Informatica: The Formation of Information Policy Rules Through Technology,* 76 TEX. L. REV. 553, 568–69 (1998) (same).

250   17 U.S.C. § 302 (2000).

251   *Id.* § 109(a).

omy ensures that some of an author's creativity goes immediately into the public domain.[252] And the fair use defense excuses copyright infringement that is unlikely to harm the work's market or that provides some greater social benefit.[253]

As already discussed, however, each of these important legal safeguards presumes the existence of certain architectural conditions. The expiration of a work's copyright is meaningless if the work is not actually available to the public. The first sale doctrine applies only if the copyright owner has chosen to market the work in freestanding, alienable copies. The idea/expression dichotomy has little effect if those who would use a work's ideas never see them. And fair use is fairly useless if the putative user cannot access the work.

Digital technology puts these architectural conditions at risk and thus threatens to upset the balance that the safeguards strike. For example, suppose the music industry develops a pay-per-listen satellite technology that allows anyone anywhere to hear any song through a small radio receiver. If this were music's predominant distribution model, the first sale doctrine would be useless, because songs would not be available in a freestanding format. The fair user—who has the right to make use of the song without the copyright owner's permission—would be out of luck as well. And the expiration of the copyright in a song would not cure these ills; the song might be in the public domain as a legal matter, but as an architectural matter the copyright owner would still control it.

This "celestial jukebox"[254] may lie some years in the future, but we can see signs of the impending technological hegemony that it exemplifies. Content on demand is already here, with pay-per-view movies and streaming video and audio online. Even when a work is released in freestanding form, encryption technology and copy protection measures can architecturally forestall uses of it that copyright law would permit—e.g., by making it difficult to extract unprotected facts from a copyrighted database or to copy a film excerpt from a DVD for fair use purposes. And the constitutional principle of free speech that undergirds the fair user's privilege is useless against private architectural efforts, as they involve no state action.[255]

---

252   *Id.* § 102(b).

253   *Id.* § 107.

254   "Celestial jukebox" is Paul Goldstein's clever term for any technologically advanced content-on-demand system. *See* PAUL GOLDSTEIN, COPYRIGHT'S HIGHWAY 21–22 (rev. ed., Stanford Univ. Press 2003) (1994); *see also id.* at 170–71 (discussing technological threat to copyright's "safety valves").

255   Tom W. Bell, *Fair Use vs. Fared Use: The Impact of Automated Rights Management on Copyright's Fair Use Doctrine,* 76 N.C. L. REV. 557, 592 (1998); *see also* Jane C. Ginsburg,

What's more, content providers have successfully sought the en-
actment of technolegical regulations—i.e., legal protections for their
restrictive technologies. Thus we have the Computer Fraud and
Abuse Act,[256] which began as a criminal prohibition against hacking
into classified government computers but has since expanded into a
general prohibition, complete with a private civil remedy, that covers
the unauthorized access to and alteration of information in almost
any digital technology over which Congress can exert Commerce
Clause jurisdiction.[257] More famous, or infamous, is the Digital Mil-
lennium Copyright Act (DMCA),[258] which among other things out-
laws   the   removal   or   falsification   of   copyright   management
information,[259] encourages online service providers to adopt techno-
logical measures designed to protect copyrighted material against un-
authorized use,[260] and imposes civil and criminal liability on those
who circumvent technological measures designed to control access to
a copyrighted work.[261]

　　To the extent that these technolegical enactments simply provide
legal backing for technology that protects a copyright owner's exclu-
sive rights, they are unobjectionable. The problem, however, is that
the protection that technology provides is not coextensive with copy-
right's entitlements and safeguards. If a film on DVD has built-in copy
protection—and most of them do[262]—that protection will endure
even after the copyright expires, making the film's entry into the pub-
lic domain merely theoretical. If one wants access to the facts in a
password-protected database, one needs the password, even though
the facts themselves are in the public domain. A fair user who wants

---

*From Having Copies to Experiencing Works: The Development of an Access Right in U.S. Copy-
right Law*, 50 J. COPYRIGHT SOC'Y U.S.A. 113, 130 (2003) ("And if hard copies and
unprotected digital copies do disappear in a brave new pay-per-access world, then the
threat to transformative fair use becomes more than a paranoid fantasy.").

256　Counterfeit Access Device and Computer Fraud and Abuse Act of 1984, Pub. L.
No. 98-473, §§ 2101–2103, 98 Stat. 1837, 2190–92 (codified as amended at 18 U.S.C.
§ 1030).

257　Gibson, *supra* note 15, at 199; *see also* United States v. Mitra, 405 F.3d 492, 497
(7th Cir. 2005) (upholding CFAA conviction of student who hacked city's computer-
based radio system for emergency services).

258　Pub. L. No. 105-304, 112 Stat 2860 (1998) (codified as amended in scattered
sections of 17 and 18 U.S.C.).

259　17 U.S.C. § 1202 (2000).

260　*Id.* § 512(i).

261　*Id.* § 1201(a) (substantive provision); *id.* § 1203 (civil liability); *id.* § 1204
(criminal liability).

262　*See, e.g.*, Universal City Studios, Inc. v. Reimerdes, 111 F. Supp. 2d 294, 308
(S.D.N.Y. 2000) (discussing DVD copy protection technology).

to parody a pay-per-listen song must pay to get access to it just like anyone else.

One can understand why technological protections are not more sensitive to copyright concerns. Today's digital architecture is not particularly good at "coding" for copyright—i.e., at discerning whether a given act implicates the amorphous legal rights that copyright comprises. Take fair use, for example. Judges and attorneys have a hard enough time figuring out whether fair use applies ex post. How much harder must it be to write a computer program that can recognize ex ante when someone seeking access to a work is or is not a fair user? Moreover, copyright safeguards by their nature involve activities from which copyright owners do not make any money. One cannot therefore expect either copyright owners or the technologists who work for them to spend much time designing their products to take account of fair use and its similarly unprofitable safeguards—at least, not without some external incentive.

In contrast, one might reasonably expect to find some accommodation of copyright's safeguards in the legal provisions that buttress technological protections. After all, the reach of these technolegical enactments is determined not by technological happenstance, but by the deliberate decisions of legislators. Yet the anticircumvention provisions of the DMCA are as insensitive to copyright's safeguards as the technology they protect: bypassing technologies that restrict access to a copyrighted work constitutes a DMCA violation even if the ensuing use of the work is non-infringing.[263] Indeed, if technological access protections keep the public from even viewing a digitized work, the public is unable to determine whether the work is protected by copyright in the first place; in other words, we may not know whether copyright and the DMCA apply until we circumvent the protections—and then it's too late. Technology therefore gives copyright owners a way to control their works without any concern for the balance that we find (or at least aspire to) in copyright law, and technolegical enactments simply reinforce this capability.

We could combat this technological hegemony by amending technolegical laws so that they supplement copyright owners' entitlements without obstructing the exercise of public privileges. For example, in recent years Congress has considered revising the DMCA to impose liability only when circumvention of access controls results in

---

263   17 U.S.C. § 1201(a). In contrast, circumventing a technology that does not control access—such as copy protection code—is not a DMCA violation.

actual copyright infringement.[264] Even in the absence of such an amendment, courts have grown increasingly dubious of efforts to use the DMCA to protect noncopyright interests.[265] Either legislatively or judicially, then, technolegical regulation could focus more on copyright principles and less on technical infringements, in the same way that the fair use jurisprudence permits the making of unauthorized copies in the course of reverse engineering code to get at the unprotected "ideas" necessary for interoperability[266] and allows the temporary copying of code when the copyist's goal is merely to extract the uncopyrightable facts within.[267] The problem with this approach, however, is that—like the copyright safeguards themselves—it operates on a purely legal level. Amending the DMCA and other overly expansive technolegical enactments is all well and good as far as it goes, but it does nothing to address the purely architectural aspect of the problem, namely that access controls and copy protection would still prevent the public from exercising its privileges.

One response to this architectural conundrum is to recognize that technological protections are not bulletproof. A skilled technologist may be able to evade access controls or copy protection technologies. Indeed, the whole point of the DMCA's anticircumvention provisions is to deter such hacking. One might therefore conclude that as long as we adjust the DMCA to permit hacking done in pursuit of some legitimate objective, we can leave the rest to the hackers; they will free digital content from its technological shackles and copyright's safeguards will once more have practical value.[268]

This thinking, however, rests on two faulty premises. First, it assumes that hackers will consistently succeed in circumventing techno-

---

264   *See* Digital Media Consumers' Rights Act of 2003, H.R. 107, 108th Cong. § 5 (2003).

265   *See, e.g.,* Chamberlain Group, Inc. v. Skylink Techs., Inc., 381 F.3d 1178, 1203–04 (Fed. Cir. 2004) (rejecting DMCA claim that would have inhibited competition in market for garage door openers); Lexmark Int'l v. Static Control Components, Inc., 387 F.3d 522, 537–45 (6th Cir. 2004) (rejecting DMCA claim that would have inhibited competition in market for printer toner cartridges).

266   *See supra* Part II.A.

267   *E.g.,* Nautical Solutions Mktg., Inc. v. Boats.com, No. 8:02-CV-760-T-23TGW, 2004 WL 783121, at *2 (M.D. Fla. Apr. 1, 2004); Evolution, Inc. v. SunTrust Bank, 342 F. Supp. 2d 943, 956 (D. Kan. 2004).

268   *E.g.,* Strasser, *supra* note 37, ¶ 108 ("[I]f a piece of software imposed constraints that ordinary consumers would be unable to notice during the normal operation of the software, chances are that somewhere in the world, a skillful programmer would nevertheless discover the constraints at some point and, assuming they are overreaching or otherwise newsworthy, would disseminate its discovery over the Internet to the world at large.").

logical protections. The conventional wisdom is that no system is secure,[269] and it is true that some nascent attempts at copy protection have proved laughably easy to evade.[270] But we cannot be sure that this will always be the case, any more than we can be sure that decompilation of object code will consistently succeed in revealing interoperability standards. Given the mercurial nature of technological innovation, we should be wary of tying legal doctrine to a current particularity of digital architecture.

Second, and more important, hacking generally only helps hackers. Those who lack the skills to circumvent technological protections must therefore rely on the kindness of strangers, who may or may not share the unskilled public's views regarding which products should be liberated and distributed—if they bother to share the products of their hacking at all. Hackers may occasionally hit upon and distribute a technology that allows the rest of us to evade technological protections without learning how to hack, but for those fearful of technological hegemony that is a slim reed on which to rest.[271] As with reverse engineering, then, hacking would simply substitute a world of technological cans and cannots for a world of legal haves and have-nots—a preferable eventuality, perhaps, but not an optimal one.

In any event, even if hacking does consistently triumph over control and the ability to hack becomes ubiquitous, hacking is not an attractive solution because of the risk of overuse. It works only if hackers limit themselves to content to which they have a legal right. In practice, however, many hackers would undoubtedly use their skills to obtain material that a sensible information policy would require

---

269 "Any electronic on-line system is vulnerable to attack. That is close to an axiom in the field of computer security. So too, therefore, are self-help systems vulnerable." Kenneth W. Dam, *Self-Help in the Digital Jungle*, 28 J. LEGAL STUD. 393, 401 (1999) (footnote omitted); *see also* David A. Sklansky, *Back to the Future:* Kyllo, Katz, *and Common Law*, 72 MISS. L.J. 143, 200–01 (2002) (noting that "each new level of security seems inevitably to be defeated by a loose, worldwide network of computer-savvy adolescents").

270 *E.g.*, Kevin Coughlin, *CD Technology Isn't So Protected*, STAR-LEDGER (Newark, N.J.), Oct. 8, 2003, at 4 (describing how college student defeated copy protection technology merely by holding down Shift key as CD was inserted into computer); Matt Richtel, *Digital Lock? Try a Hairpin*, N.Y. TIMES, May 26, 2002, § 4, at 12 (describing how music CD copy protection could be defeated through use of magic marker).

271 *See* Dan L. Burk & Julie E. Cohen, *Fair Use Infrastructure for Rights Management Systems*, 15 HARV. J.L. & TECH. 41, 82 (2001) ("Even the most user-friendly circumvention technologies will require some threshold level of technological competence."); Julie E. Cohen, Lochner *in Cyberspace: The New Economic Orthodoxy of "Rights Management,"* 97 MICH. L. REV. 462, 526 (1998) (noting that despite existence of devices for circumventing videotapes' copy-protection measures "there is no evidence suggesting that substantial numbers of ordinary consumers use them").

them to pay for. Such piracy might expose them to copyright liability, but copyright owners will likely lack the resources necessary to obtain redress for diffuse acts of individual infringement.[272] Again, legal entitlements alone will not solve the problem.

2. The Solution

A world of unfettered technological competition will therefore not yield a socially optimal outcome. If copyright's balance between private incentive and public benefit is to have its intended effect, we cannot afford for digital architecture to become a "Wild West of technological one-upmanship that favors the better technologist."[273] We need a new form of technolegical regulation, one that is sensitive to both technological and authorial innovation.

One such regulatory device would be a bright-line ban on all technological constraints that might impede the exercise of copyright's public privileges. Under this approach, access controls that insulate both copyrighted and public domain materials would have to go, as would copy protection technologies that are incapable of distinguishing between fair users and others. This approach, however, is clearly a nonstarter. It is unrealistic politically, would stifle beneficial innovation in both the technological and commercial spheres, and would encourage widespread piracy even by those without hacking skills.

We could go to the other extreme and encourage the adoption of high-tech, code-intensive implementations of copyright's safeguards. The common term for this approach is digital rights management (DRM).[274] Kenneth Dam, for example, has argued that digital developers can accommodate public privileges on a voluntary basis and suggests ways in which the first sale and fair use doctrines could plausibly be implemented in code.[275] Although he recognizes that current technology might not seem capable of capturing the complex legal

---

272 Gibson, *supra* note 15, at 196.

273 *Id.* at 169–70; *see also* Jack M. Balkin, *Digital Speech and Democratic Culture: A Theory of Freedom of Expression for the Information Society*, 79 N.Y.U. L. REV. 1, 53–54 (2004) (arguing that law must regulate technology to preserve free speech values).

274 The "digital rights management" term is accurate insofar as it describes the technological implementation of copyright entitlements, but quickly becomes deceptive if used to regulate conduct in a way that does not accord with established legal "rights." *See* Dam, *supra* note 269, at 396–97 (eschewing terms "copyright management systems" and "rights management systems" because "the underlying information need not be copyrighted" but may instead be trade secrets or facts or even pirated content).

275 *Id.* at 400 (discussing first sale); *id.* at 403–05 (discussing fair use).

balance inherent in copyright's safeguards, Dam wisely warns us that "anyone who sells technology short by saying 'it can't be done' has very little experience with technology."[276]

Dam's proposal, however, fails to overcome the obstacle identified above, namely that copyright owners have little reason to voluntarily expend the time and resources necessary to facilitate unprofitable public privileges. It is true that in certain instances a content provider might want to encourage fair use: a software developer may want to give computer magazines free access to a program so that they will publish reviews, or an academic may want to allow others to expand on his or her research as long as credit is given.[277] But the situations in which an author takes the time to accommodate uncompensated uses are more the exception than the rule; one must remember that first sale and fair use owe their origin to a long line of cases in which copyright owners so resented the public's unauthorized conduct that they filed suit.[278]

Of course, DRM does not have to be voluntary. A legislative mandate could give copyright owners the impetus needed to undertake the technological facilitation of copyright safeguards. Dan Burk and Julie Cohen have proposed such a framework for the implementation of fair use, one that would automate certain customary, limited, and therefore predictable aspects of the doctrine but would provide a neutral human gatekeeper to mediate more uncertain claims of privilege.[279] Yet even this proposal shares the shortcomings of any approach that involves detailed technical regulation, in that such regulation would require constant updating to remain current with the ever-changing nature of DRM technology. It is hard to envision such a system working absent a government agency tasked with flyspecking

---

276   *Id.* at 411. For one example of how technology might accommodate copyright's public privileges, see generally Deirdre Mulligan & Aaron Burstein, *Implementing Copyright Limitations in Rights Expression Languages* (detailing species of technological protection that would allow exercise of both producers' and users' legal entitlements), *in* DIGITAL RIGHTS MANAGEMENT 137 (Joan Feigenbaum ed., 2003).

277   Dam, *supra* note 269, at 403–04.

278   *E.g.*, Bobbs-Merrill Co. v. Straus, 210 U.S. 339, 350–51 (1908) (holding that copyright's "right to vend" did not include right "to control all future retail sales"); Sebastian Int'l, Inc. v. Consumer Contacts (PTY) Ltd., 847 F.2d 1093, 1096 (3d Cir. 1988) (referring to first sale's "venerable lineage" that "finds its origins in the common law aversion to limiting the alienation of personal property"); Folsom v. Marsh, 9 F. Cas. 342, 345 (C.C.D. Mass. 1841) (No. 4901) ("[A] fair and bona fide abridgment of an original work, is not a piracy of the copyright of the author."); *see also* H.R. REP. No. 94-1476, at 65–66 (1976) (noting that fair use statute merely codifies long line of case law).

279   Burk & Cohen, *supra* note 271, at 59.

DRM developments and issuing regulations on what technologies sat-
isfied the legislative mandate. The other alternative would be to es-
tablish a private right of action against digital publishers who fail to
adequately accommodate public privileges, but this alternative relies
on the dubious premise that fair users and others would have the re-
sources necessary to bring suit and could produce a court decision
timely enough to give guidance to the next generation of
technologists.

In fact, the more technologically intrusive the regulation, the
more likely it is to stifle private innovation that can serve authors and
the public alike. Digital architecture may pose a threat to copyright's
safeguards, but it obviously holds much promise for copyright's over-
all goal of encouraging the creation and dissemination of expressive
works. Apart from the democratizing effect of computer networks al-
ready discussed, consider the benefits of a digital distribution model
like the celestial jukebox described above. If such a technology ex-
isted—and its day is not far off—music lovers could experience un-
precedented convenience and choice; they could listen to any song, at
any time, without the need to purchase a fixed copy (let alone the
other twelve songs on the album that they don't want). Music publish-
ers could avoid the costs of making and distributing music in fixed
form and could pass those savings to the purchasing public. And even
if the celestial jukebox fails to materialize because of licensing hold-
outs, today's digital networks can help musicians reach their audi-
ences directly, without seeking financing from corporate
intermediaries. This lowers barriers to market entry and thus both
decreases prices and increases the variety of music available.

So we are faced with a paradox: the market is good at adapting to
changes in technology, but the government is not—yet only the gov-
ernment can ensure that private technology accommodates copy-
right's public privileges. The solution is to find a form of
technolegical regulation that would allow innovation to continue
while preserving the public's legal privileges. And once more we call
upon those low-tech copyright concepts, the registration and deposit
formalities.[280]

Here's how it would work. To procure copyright protection, an
author would first have to register with an administrative agency and
deposit an architecturally unfettered version of the work—what I have
elsewhere called a "re-reified" copy.[281] The author would thereafter

---

280 *See* Sprigman, *supra* note 2, at 527 (noting that formalities mediated the ten-
sion between the First Amendment and copyright protections).

281 Gibson, *supra* note 15, at 172.

be free to market the work in whatever form he or she likes; access and copying controls would be wholly unregulated. The re-reified copy, however, would remain in the depository, outside the author's control, with access administered by a neutral gatekeeper. The re-re-ified copy would thus provide a basis for exercise of the public privileges. For example, reviewers, researchers, parodists, and other fair users would be free to access and copy the work, as long as they demonstrated their fair use bona fides. And when a work's copyright expired, all restrictions on copying would disappear, no matter who the copyist was. This purposely retrograde approach would give meaning to copyright's safeguards and allow law to resume its proper place as the primary behavioral constraint on copyrighted goods without hindering valuable innovation in information technology.

This proposal is not without its problems. For one, it does nothing to preserve the first sale doctrine because copyright owners would still be allowed to use technological restrictions on alienability to control the aftermarket for their works. This is less of a drawback than it might seem at first blush, however, because one purpose of the first sale doctrine is to allow low-value users who cannot afford an author's initial price to acquire the book in the aftermarket.[282] When copyright owners subject their works to architectural restraints on access and alienability, however, they restrict opportunities for arbitrage, which allows them to price discriminate—i.e., sell their works at one price to high-value users and at a lower price to low-value users.[283] When this happens, those low-value users who depended on a thriving aftermarket to acquire a work can get the work directly from the publisher instead, making first sale less important.

Another obvious shortcoming is that if the depository is to give true meaning to the public's privileges, it would have to be widely accessible to the public. The only way to achieve this goal would be through a digital network that permits remote access. Once again, then, we are faced with a familiar technological dilemma: how can digital architecture prohibit unprivileged copying but facilitate fair use? In other words, some DRM measures would be necessary even under this supposedly low-tech solution, just as Dam proposes, as would the use of a neutral human gatekeeper, as Burk and Cohen propose. The technological challenge here, however, is not nearly as formidable as in the other proposals, because they contemplate integrating copyright's safeguards into the architecture of every copy of

---

282   *Id.* at 213.
283   *See id.* at 206–08. Price discrimination, however, is not a panacea for the ills of technological hegemony. *See id.* at 212–15.

every digital work. In contrast, the depository would have to fashion a DRM solution for the re-reified copy alone. This narrow focus also allows the depository to adjust more quickly than a dispersed system to changes in the law—e.g., a judicial recalibration of fair use—and to advances in DRM technology.

Two broader problems remain. First, a drawback of any formal requirement for copyright protection is that it converts copyright from a protection that arises through operation of law to an opt-in regime. Those authors that view the re-reification requirement as too burdensome might therefore decide not to create their works at all or at least forgo copyright entirely. One way to minimize this occurrence would be to apply the registration and deposit requirement only to those works published in digital form, or (more narrow still) only to those works published with technological restraints on access, copying, or distribution. Print publishers would not have to register unless and until they later decide to "go digital," and failure to register would forfeit copyright protection only for digital copying and distribution of the work.

But those most likely to opt out would not be print publishers, who need copyright's protection more than anyone. The real risk is that digital publishers would decide that the copyright game is not worth the re-reification candle and would decide to take their chances with wholly technological protections. The instances of digital opt-out would be few, however, because digital publishers must allow access to their works in order to market them; no one will pay for a book that cannot be read.[284] The threat of hacking will thus loom large in a market regulated only by technology, where one uncontrolled copy of a work can doom its commercial prospects. Therefore, although hacking is not a panacea for the problem of technological hegemony, it serves as an excellent deterrent for any author tempted to forgo all legal remedies and rely on technology alone to prevent unauthorized copying and distribution; once the genie is out of the bottle, only copyright has any hope of putting it back in. Moreover, some uses of a work that infringe copyright will likely remain resistant to any form of technological control—e.g., unauthorized public performance of a song.[285] In any event, a world in which authors forgo copyright in favor of architectural protections is still better than a world in which the law grants them the legal entitlement without any accompanying technolegical regulation.

---

284   The exception, of course, is closed-code software. *See supra* Part I.A.
285   Lichtman, *supra* note 119, at 241.

Finally, we have the opposite problem: not that authors would ignore the depository system, but that the public would. Even if re-reified works were widely available, fair users might be reluctant to use them for fear that the use might compromise their anonymity. Of even more concern is the specter of a watchful government agency or its agent acting as gatekeeper for the exercise of fair use rights, which are part and parcel of the right to free expression.[286] Burk and Cohen identify the anonymity concern and propose to resolve it through a key escrow system, which might work here as well.[287] But for both concerns, the best answer is that re-reification would be a supplement to, rather than replacement for, traditional avenues of fair use. Fair users would still be free to obtain a work the old-fashioned way—i.e., by buying a copy or otherwise paying for access. Nor is re-reification a substitute for other measures that give meaning to copyright's safeguards, such as a DMCA amendment that permits unauthorized access for fair use purposes or Cohen's "right to hack," which achieve much the same result.[288] In the end, then, reviving the formalities may be just one step copyright must take to preserve its balance between private incentive and public benefit, but it is a necessary step.

CONCLUSION

The promise and threat that digital architecture represents are not something that copyright law can address by tinkering with legal entitlements. Architectural problems call for architectural solutions. If copyright is to maintain its balance between private incentive and public benefit—and between authorial innovation and technological innovation—it needs "technolegical" tools that can regulate those uses of technology that keep it from achieving its goals.

As it happens, the tools best suited to this task are not new, but old. Indeed, the venerable concepts of publication, notice, registration, and deposit withdrew from copyright's front lines just as the technology that now makes them indispensable was coming into its own. Properly understood, these formalities are more than relics of an antiquated copyright culture; they are timeless embodiments of intellectual property principles—indeed, democratic values—that we

---

286   *See* Eldred v. Ashcroft, 537 U.S. 186, 219–20 (2003) (listing fair use among copyright's "built-in First Amendment accommodations").

287   Burk & Cohen, *supra* note 271, at 63–65.

288   *See* Julie E. Cohen, *Copyright and the Jurisprudence of Self-Help,* 13 BERKELEY TECH. L.J. 1089, 1141 (1998) (arguing that "[i]f the user privileges established by copyright . . . are to mean anything, users must be afforded affirmative rights to protect themselves" including "a right to hack the digital code that implements and enforces the challenged restriction").

have lost sight of: free expression, public enlightenment, and transparent governance. Thus their resurrection can provide the solution to a variety of seemingly disparate copyright conundrums: closed code, overpropertization, copyright as censorship, and technological hegemony.

Regardless of whether getting rid of copyright's formalities was a good idea in the twentieth century, then, their revival is vital in the twenty-first. Resurrecting these retrograde concepts not only solves discrete problems that digital architecture has caused, but also refocuses copyright on its underlying justifications, with the familiar formalities serving as both talisman and touchstone.