THE LIMITS IN OPEN CODE: REGULATORY STANDARDS AND THE FUTURE OF THE NET

By Lawrence Lessig [†]

ABSTRACT

This essay considers the effect of the open source software movement on government's ability to regulate the Net. Its claim is that an increase in open source software within the application space of the Internet decreases the government's power to regulate.

This is an essay about standards in the future of the Internet's governance. I begin with a distinction between two types of standards, and then continue with a reminder of a bit of history of the evolution of thought about regulation in cyberspace. I then draw upon this distinction and this history to suggest a question about the future of the Net's regulation. This question relates to the place of open source software in the future of the "application space" of the Internet. My argument is that open source software will make regulating cyberspace more difficult than it otherwise would be.

I. STANDARDS

Distinguish between two sorts of standards: coordinating and regulating. A coordinating standard is a rule that facilitates an activity that otherwise would not exist. A regulating standard restricts behavior within that activity, according to a policy set by the regulators. A coordinating standard can be imposed from the top down, or emerge from the bottom up; a regulating standard is ordinarily imposed only from the top down. Driving on the right side of the road is a coordinating standard. A speed limit is a regulating standard. Coordinating standards limit liberty (drive on the right) to make an activity possible (driving); regulating standards limit liberty within that activity (speeding) to advance a regulatory end (safety or

[†] Jack N. and Lillian R. Berkman Professor for Entrepreneurial Legal Studies, Harvard Law School. This essay is drawn from a lecture. Thanks to Susan Freiwald and Joel Reidenberg for pointing out the implications of this argument for regulation internationally. Thanks to Karen King for her research support.

fuel conservation). We understand why an individual would want to deviate from a regulating standard; it is (often) hard to make sense of a desire to deviate from a coordinating standard.

Standards on a computer network are similarly coordinating and regulating. TCP/IP is a coordinating standard—it is a convention that makes exchange of information over the Internet possible.[1] Space allocation on a network server is a regulating standard—it limits the storage space assigned to a particular user to allow many users to use the same storage resource.

Most of the important Internet standards to date have been coordinating standards—standards such as TCP/IP, FTP, and HTML. The Internet community has demonstrated well its ability to develop and deploy coordinating standards; this is the genius in organizations such as the Internet Engineering Task Force ("IETF").[2] But in the future, most of the most significant debates about standards will be debates about regulating standards—about standards that allow the government to carry its policy choices into effect, whether or not those choices are the choices of bottom-up organizations like the IETF.

The Net's success with standards in the future, then, depends upon the standards at stake. And its success with coordinating standards will not necessarily entail a similar success with regulatory standards.

II. REGULABILITY

That's the distinction; now the history. It's important that we remark how the debate about the regulation of cyberspace has changed. Three years ago the world was techno-libertarian. Frustrated sorts from our bureaucratic age looked to cyberspace as a place where regulation would not work, and hence as a place where people would be free. "Free" had two senses for these sorts—first, life in cyberspace was free from *any* regulation, and second, life there was free from regulation by *government*. Life

[1] *See generally* Charles L. Hendrick, *Introduction to the Internet Protocols* (July 3, 1987) <http://www.shiva.com/prod/techinfo/ip-intro.html> (giving history as well as explanation of TCP/IP).

[2] The IETF is the single most important Internet standards body, though it functions in a very different manner from ordinary standards bodies. Membership of the IETF is open, and standards get adopted only if implemented. *See* Internet Engineering Task Force, *Overview of the IETF*, (visited Apr. 1, 1999) <http://www.ietf.org/overview.html>. *See also* Scott Bradner, *The Internet Engineering Task Force*, *in* OPEN SOURCES: VOICES FROM THE OPEN SOURCE REVOLUTION 47 (Chris DiBona et al. eds., 1999).

in cyberspace, libertarians promised, was unregulated and *unregulable*. Behavior there was beyond the government's reach.

These were the ideas that defined first-generation thought about cyberspace and law. Law such as copyright was dead, lyricists such as John Perry Barlow sang.[3] Law was fundamentally threatened, lawyers such as Post and Johnson warned.[4] The Net would be a world where freedom reigned, and in some techno-Marxist way, governments would have no choice but to wither away.

These ideas did not go unchallenged. Rather, there were a few "crazies" around at the time who thought quite differently about regulation on the Net. I met two at a conference at Emory Law School three years ago, where they were busy challenging these then-commonplace ideas about the unregulated life of cyberspace.

One was then an assistant professor from Fordham: Joel Reidenberg. About the claim that life in cyberspace was free—unregulated at all—Reidenberg had a very different view. Life in cyberspace, Reidenberg argued, was regulated as any form of life was. This regulation, however, was built into the code.[5] This form of regulation he called *lex informatica*,[6] and this *lex*, he maintained, defines what behavior is possible in cyberspace and what values cyberspace will uphold.[7] Whether these are values of anonymity or privacy or free speech or access, it is this law that makes those values possible.

But the *lex informatica*, he argued, was not a law that was fixed.[8] The architectures of cyberspace could be changed. The values that cyberspace embraces could be different. There is no nature to the way that cyberspace

[3] *See, e.g.*, John Perry Barlow, *Keynote Address, Symposium on "Fundamental Rights on the Information Superhighway" at the New York University School of Law*, 1994 ANN. SURV. AM. L. 355 (1994); John Perry Barlow, *The Economy of Ideas*, WIRED, Mar. 1994, at 84.

[4] *See, e.g.*, David R. Johnson & David Post, *Law and Borders—The Rise of Law in Cyberspace*, 48 STAN. L. REV. 1367, 1375 (1996).

[5] By "code" I mean generally the software and hardware that constitutes cyberspace as it is. That code might be divided between the basic net protocols of TCP/IP, and the applications that run on those protocols. As I explain more below, it is the application space that is the most important target of regulation.

[6] *See* Joel R. Reidenberg, *Governing Networks and Rule-Making in Cyberspace*, 45 EMORY L.J. 911, 929 (1996). *See also* Joel R. Reidenberg, *Lex Informatica: The Formulation of Information Policy Rules Through Technology*, 76 TEX. L. REV. 553 (1998) [hereinafter Reidenberg, *Lex Informatica*].

[7] *See* Reidenberg, *Lex Informatica*, *supra* note 6, at 568-73.

[8] *See id.* at 579-81.

is built—no nature, simply code. This code could be made to be very different from what it currently is. It could be made, that is, to embrace a very different set of values.

The other crazy was Pam Samuelson, then a professor at the University of Pittsburgh. Samuelson challenged the second idea—that cyberspace could not be regulated by government. For as Samuelson saw it, the law was already threatening an important regulation of life in cyberspace.[9] Not directly, of course, but indirectly—through a series of changes threatened by the Administration's White Paper on Intellectual Property.[10] These changes, designed to increase the law's protection for intellectual property, threatened to fundamentally queer the architectures of cyberspace. Laws would have their effect, if only indirectly, by inducing changes in the *lex* that Reidenberg spoke of.

Time works changes. The views of these two crazies have now become mainstream. Everyone now gets how the architecture of cyberspace is, in effect, a regulator. Everyone now understands that the freedom or control that one knows in cyberspace is a function of its code. Cookies[11] mean less privacy; choice about cookies means more privacy. A world without P3P[12] is a world with less control over privacy; a world with P3P is a world with more control over privacy. A world with PICS[13] is a world where speech is less free; a world without PICS is, well, let's say, nice.[14] The differences in these worlds are differences in the code of these worlds. Different code, different regulation, different worlds.

[9] *See, e.g.*, Pamela Samuelson, *Intellectual Property Issues Raised by the National Information Infrastructure*, 454 PLI/PAT 43 (1996).

[10] Information Infrastructure Task Force, Working Group on Intellectual Property Rights, *Intellectual Property and the National Information Infrastructure: The Report of the Working Group on Intellectual Property Rights* (Sept. 1995) <http://www.uspto.gov/web/offices/com/doc/ipnii/>.

[11] Cookies allow web sites to track users over multiple visits. *See generally* David Whalen, *The Unofficial Cookie FAQ, Version 2.51* (visited Apr. 1, 1999) <http://www.cookiecentral.com/faq/index.shtml>.

[12] "The Platform for Privacy Preferences Project (P3P) enables Web sites to express their privacy practices and enables users to exercise preferences over those practices." World Wide Web Consortium, *Platform for Privacy Preferences (P3P) Syntax Specification* (working draft) (July 2, 1998) <http://www.w3.org/TR/WD-P3P10-syntax-19980702>.

[13] The Platform for Internet Content Selection (PICS) is a protocol for facilitating the rating and filtering of content on the Internet. *See* World Wide Web Consortium, *Platform for Internet Content Selection (PICS)* (last modified Jan. 3, 1998) <http://www.w3.org/pics>.

[14] *See* Lawrence Lessig, *Tyranny in the Infrastructure*, WIRED, Jul. 1997, at 96.

And so too do most now see how government might have a role in this regulation. Smart governments will regulate, but not by directly regulating the *behavior* of people in cyberspace. Smart governments will instead regulate by regulating the *code* that regulates the behavior of people in cyberspace. Cyberspace's code will become the target of regulation.[15] The future will be littered with examples of government trying to intervene to assure that cyberspace is architected in a way to protect government's interests. Whether those interests will be interests against copyright management circumvention[16] or interests in favor of encryption control,[17] the government will increasingly see that the most efficient target of regulation is not people but binary code. Enslave the code while telling the world that you are leaving the space free[18]—this is the formula for taming the liberty that cyberspace now provides.

Two important conclusions follow from the arguments of these two crazies. First, if code is a kind of law, then we should focus, as we do with real-space law, on the freedoms and the constraints built into this code, and on how these freedoms and constraints are changing. And second, if governments regulate code, then we should think about the limits that should constrain government's power to regulate. For our constitutional tradition is one which limits governmental power by limiting government's direct legislative action; yet the future of the government's regulation of the Net is a future where government regulates by indirect legislative action. Constitutional values should constrain both indirect and direct regulation; so far it is not clear that they do.[19]

[15] For a great example of regulation of the code, see the Oxley-Manton Amendment to the Security and Freedom Through Encryption (SAFE) Act, H.R. 695, 105th Cong. (1997), which would have regulated the type of permissible encryption to be just that which provided the required governmental access.

[16] *See* Digital Millennium Copyright Act of 1998, Pub. L. No. 105-304, § 1201, 112 Stat. 2860, 2863-2872 (codified at 17 U.S.C. § 1201) (1998). For a critique of this anti-circumvention provision, see Pamela Samuelson, *Intellectual Property and the Digital Economy: Why the Anti-Circumvention Regulations Need to be Revised*, 14 BERKELEY TECH. L. J. 519 (1999).

[17] *See supra* note 15.

[18] *See, e.g.*, WILLIAM J. CLINTON AND ALBERT GORE, JR., A FRAMEWORK FOR GLOBAL ELECTRONIC COMMERCE (1997), *available at* <http://www.iitf.nist.gov/eleccomm/ecomm.htm>.

[19] My favorite example is *Rust v. Sullivan*, 500 U.S. 173 (1991), in which the Supreme Court upheld an indirect means of discouraging abortion, something the government cannot do directly. *See* Lawrence Lessig, *The New Chicago School*, 27 J. LEGAL STUD. 661, 670, 690-91 (1998).

III. LIMITS ON REGULABILITY

That's the history: Now something about the future. I want to focus on a new wrinkle in this debate about regulating cyberspace. We are just beginning to understand this new wrinkle, yet it may become the most important question about the future of cyberspace that we have yet seen.

You might think it follows from the commonplace views of our day—from those views once held by the crazies only, but now considered mainstream by most—that government is capable of effectively regulating the Net. If government can regulate the code, then government can require codewriters to build the standards that the government needs into the code. The future of regulatory standards under this view, then, would simply be a future where the government tells codewriters how to architect their code so as to incorporate governmental regulatory standards.

But in fact, the story is interestingly more complicated. In fact, this power of government depends upon a feature of the code—application space code[20]—that has only recently become salient. This feature is its *ownership*. Whether government can regulate code depends in part upon who controls that code. If the code is closed—controlled by private for-profit organizations—then government's power is assured. But if the code is open—outside of the control of any particular private for-profit organization—then the government's power is threatened. The more application space code is open code, the less government can regulate that code.

The reason is straightforward. Open code is software in plain view. It is software that comes bundled with its source code as well as its object code. Object code is the code that the computer reads. If you display it on your machine, it will appear as gibberish. But source code is code that programmers can read.[21] It is this code that allows a programmer to open an open source software project and see what makes it tick. By being able to see what makes it tick, open source software makes transparent any control that the code might carry. For example, if the code carries a government-mandated encryption routine, that routine will be apparent to open source coders. And because it is apparent, open source coders can then choose whether or not to adopt that portion of an open code project. For by its nature, and by the promises that it comes bundled with in the

[20] I define this more *infra* in the text accompanying note 29.

[21] Compilers can read it as well, but compilers simply turn this code from source code into object code.

form of licenses,[22] any open code software project remains available for adopters to modify or improve, however the adopters think best.

Closed code functions differently. It does not come bundled with its source, which means that its code is hidden under a hood that won't open.[23] Thus adopters or users of closed code cannot as easily detect what makes closed code tick. They can't as easily see whether it carries within it a given encryption routine or systems for collecting private data or technologies for monitoring and reporting usage. Clever adopters can try to work it out through reverse engineering[24] or hacking. But no matter how clever the adopter, closed code will be harder to monitor, and harder to change than open code. An adopter of open source code who doesn't like a module can simply substitute another; an adopter of closed code has no equivalently simple choice.[25]

This difference is critical to the question of regulability. For if the application space is built with closed code, then the ability of adopters to change that code is less than it would be if the application space were comprised of open code. If it is harder for adopters to change code, then it

[22] *See, e.g.*, Free Software Foundation, *GNU General Public License Version 2* (June 1991) <http://www.gnu.ai.mit.edu/copyleft/gpl.html>; Ira V. Heffan, Note, *Copyleft: Licensing Collaborative Works in the Digital Age*, 49 STAN. L. REV. 1487, 1508 (1997). ("The GNU GPL gives users permission to copy, modify, and distribute GNU software conditioned on the user's agreement to license all derivative versions under the same terms. Further, users must agree (1) not to establish proprietary rights in the software; (2) to provide the source code to anyone to whom they give the object code; (3) to include in the software notice of the applicability of the GNU GPL; and (4) to accept the software without warranties of any kind.") (footnotes omitted).

[23] The idea is stolen (but can an idea be stolen?) from Austin Bunn, *Under the Hood*, FEED (visited Apr. 1, 1999) <http://www.feedmag.com/oss/ossintro.html>.

[24] Though many licenses expressly forbid reverse engineering. *See, e.g.*, Mark A. Lemley & David McGowan, *Legal Implications of Network Economic Effects*, 86 CALIF. L. REV. 479, 528 (1998) ("Microsoft has argued that each of the 100 million-plus copies of object code it sells are limited distributions of trade secret information subject to a 'shrinkwrap license' agreement that prevents reverse engineering, and therefore that no one can obtain a copy of Microsoft's operating systems without 'agreeing' not to reverse engineer it."). I am with those who believe that copyright's rules about reverse engineering should be read to trump the contract promise to the contrary. *See, e.g.*, Mark A. Lemley, *Beyond Preemption: The Law and Policy of Intellectual Property Licensing*, 87 CALIF. L. REV. 111 (1999); Mark A. Lemley, *The Economics of Improvement in Intellectual Property Law,* 75 TEX. L. REV. 989 (1997).

[25] Technically, this is misleading. Programmers of closed code do publish application program interfaces (APIs) that enable others to "plug in" to a closed application. In principle, if these were fully transparent, closed code would be closer to open code. The significant difference is that closed code APIs still cannot be modified.

is easier for governments to regulate through that code. Say the government has a standard it wishes to impose on some aspect of the application space. To the extent the regulatory standard gets imposed on closed code, it is more likely to be adopted by users than the same regulation imposed on open code. If the adopters don't like the regulatory standard (which, given the nature of many regulatory standards, is not unlikely), adopters can more easily swap out the regulated code if they use open code than if they use closed code.

An example offered by Peter Harter at this conference makes the point well. Netscape has turned its code for Netscape Communicator over to a version of the open source software movement. Its code is controlled by an organization called Mozilla, but its source is open. When Mozilla releases a new version, adopters around the world are permitted to download the source code, and adopt it or modify it as they wish.[26]

The French government didn't get this idea. They wanted Netscape to modify the SSL standard[27] to enable decryption of SSL transactions, and so they asked Netscape to implement the request. But as Netscape reportedly told the French, there is really very little that Netscape can do to enable the cracking of SSL, and it is easy to see why.[28] Even if Netscape built a French version of SSL, enabling the French to spy whenever the French government wants, whether that version got *used* depends upon whether it is *adopted*. And even if Netscape put the French SSL version into the code of Netscape Communicator, there is no reason to expect that adopters of the code wouldn't simply substitute a different version of SSL for the French spy-enabled version. Whether the SSL code is adopted is a decision that rests with the users, not with Netscape.

[26] *See* The Mozilla Organization, *Our Mission* (visited Apr. 1, 1999) <http://www.mozilla.org/mission.html> (announcing that Netscape Communicator and its source code would be available free of charge, and describing mozilla.org's role as a "clearing-house for the newly-available Netscape source … to collect changes, help authors synchronize their work, and periodically make new source releases which incorporate the best work of the net as a whole").

[27] SSL, the Secure Sockets Layer, is a security protocol developed by Netscape which "provides data encryption, server authentication, message integrity, and optional client authentication for a TCP/IP connection." Netscape Communications Corp., *Secure Sockets Layer* (visited Apr. 7, 1999) <http://home.netscape.com/security/techbriefs/ssl.html>.

[28] Technically, there are two reasons why there is little that Netscape can do here. One is that SSL is an open standard, which Netscape doesn't control. But the second is the reason I am focusing on here: Even if it could control it, its "control" depends upon whether the code is open or closed.

Harter's example is an instance of my more general point: to the extent that code remains open, it is harder for government to regulate; to the extent it is closed, it is easier. Had the French demanded a change in a part of Netscape's code before Netscape had given its code to Mozilla, then it would have been much harder for adopters to identify and disable that code. But after the code is in the commons, governments' power is less. Thus my point: the regulability of the application space turns in part on whether the application space is open.

That's the claim, but it requires some qualifications.

First, my argument turns upon the nature of the "application space" code. This is not the distinction between operating systems and applications, but rather the distinction between the basic Internet protocols and the applications (or "ends") that depend upon or use these protocols. It is the design philosophy of the Net to keep the protocols simple and general, and to build sophistication and complexity into the ends.[29] It is possible to imagine the government trying to regulate the Internet's basic protocols. But because these are coordinating standards that effect very little substantive control on the content of the Net, they are unlikely to be the source of any powerful or significant regulation. Regulation, or regulatory standards, if they are to be effective, would have to be embedded in the application space code.

Second, my argument is not that a world with open code, or mostly open code, couldn't be regulated. In my view, there could be relatively small shifts in the architecture of the Net—in the functionality built into the application space—that would fundamentally enable state regulation, even if that application space were open code.[30] If the Internet became "certificate rich"—meaning that many people carried and used digital certificates[31] while "on" the Net—local government's power to regulate the Net could fundamentally increase, whether or not the basic certificate architectures were open or closed code.

[29] This design is more efficient, for building complexity into the protocols would not necessarily lead to simple ends. *See, e.g.*, Jerome H. Saltzer et al., *End-to-End Arguments in System Design*, *in* INTEGRATED BROADBAND NETWORKS 30 (Amit Bhargava ed., 1991).

[30] *See* Lawrence Lessig, *What Things Regulate Speech: CDA 2.0 vs. Filtering*, 38 JURIMETRICS J. 629 (1998).

[31] Digital certificates "allow verification of the claim that a specific [encryption] key does in fact belong to a specific individual. Certificates help prevent someone from … impersonat[ing] someone else." RSA Laboratories, *FAQ 4.0—Frequently Asked Questions About Today's Cryptography* (visited Apr. 7, 1999) <http://www.rsa.com/rsalabs/faq/html/4-1-3-10.html>.

Third, my argument is also not that a world with more closed code is always a world that is more regulable. Some closed code would not affect the Net's regulability. It matters little whether solitaire programs or certain utility programs are open or closed code, for there is little connection between them and any regulation the government might impose. (So long, that is, as they are as they say they are.) Thus the point about regulability is not a point about necessity; it is instead a point about possibility.

And finally, following from the third: my argument is not a criticism of closed code in general. I don't believe that the best possible world is one where all code is open, any more than I believe that the best possible real world is one where all property is public or part of the commons. There is a mix between open and closed spaces in real space and there should be a similar mix between open and closed spaces in cyberspace. The only enemy is the extremes—either a world that was perfectly propertized (either completely, or selectively if selected well), or a world that permitted no closed development. Whatever economic model might support projects like the GNU/Linux OS,[32] there is no reason to believe the same model would work for every coding project.[33]

* * * * *

To many in the open code movement, this whole argument about the values in open source software might seem quite odd. To them, the real issue with open source software is its power. Its real virtue is its amazing efficiency—its robustness and reliability. And no doubt, if these are its virtues, they are valuable indeed.

But my point is not to question any claim about efficiency. My point is simply that there are other issues at stake as well.[34] The architecture of cyberspace embeds a set of values, as it embeds or constitutes the possible. But beyond the values built into this architecture, there are values that are implicated by the ownership of code. Its ownership can enable a kind of

[32] For a description of Linux, see Linux Online, *What is Linux* (last modified Mar. 16, 1999) <http://www.linux.org/info/index.html>.

[33] *See* Brian Behlendorf, *Open Source as a Business Strategy, in* OPEN SOURCES: VOICES FROM THE OPEN SOURCE REVOLUTION 149 (Chris DiBona et al. eds., 1999).

[34] This, I take it, is the strong and true point that Free Software Foundation founder Richard Stallman makes. *See* Richard Stallman, *Why Software Should Be Free* (Apr. 24, 1992) <http://www.fsf.org/philosophy/shouldbefree.html> (arguing that software ownership is harmful because fewer people use the program, none of the users can adapt or fix the program, and other developers cannot learn from the program, or base new work on it). *See generally* Free Software Foundation, *Philosophy of the GNU Project* (last modified Mar. 27, 1999) <http://www.fsf.org/philosophy/philosophy.html>.

check on government's power—a separation of powers that checks the extent that government can reach. Just as our Constitution embeds the values of the Bill of Rights while also embedding the protections of separation of powers,[35] so too should we think about the values that cyberspace embeds, as well as its structure.

However efficient open code may be, arguments about open source must also consider the questions that these values raise. For in my view, it makes as much sense to promote open source on efficiency grounds alone as it does to promote democracy on grounds of economic wealth alone. It may well be that democracies are more wealthy than other forms of government, just as it may well be that open source software is more robust than others. But it is a thin conception of value that would see wealth or efficiency as the only, or most important, value at stake.

[35] *See* Morrison v. Olson, 487 U.S. 654, 710 (1988) (Scalia, J., dissenting) ("While the separation of powers may prevent us from righting every wrong, it does so in order to ensure that we do not lose liberty.").