DESIRABLE INEFFICIENCY

Paul Ohm[*] *and Jonathan Frankle*[**]

Abstract

Computer scientists have recently begun designing systems that appear, at least at first glance, to be surprisingly, wastefully inefficient. A stock exchange forces all electronic trades to travel through a thirty-eight mile length of fiber-optic cable coiled up in a box; the Bitcoin protocol compels participants to solve difficult yet useless math problems with their computers; and the iPhone locks users out for many painful seconds after a mistyped password, a delay that increases with each subsequent mistake. We draw these examples and others together into a common, emerging, and underappreciated approach to digital system design, which we name "desirable inefficiency." Designers have turned to desirable inefficiency when the efficient alternative fails to provide or protect some essential human value, such as fairness or trust. Desirable inefficiency is an example of a design pattern that engineers have organically and voluntarily adopted to make space for human values. Regulators should study these emergent engineering responses and actively impose design patterns like desirable inefficiency to protect values important to society.

---

* Professor of Law, Georgetown University Law Center. Faculty Director, Georgetown Center on Privacy and Technology. This work is supported by the AXA Award on Big Data, Privacy, and Discrimination, from the AXA Research Fund.

** Ph.D. candidate in Computer Science, Computer Science and Artificial Intelligence Laboratory, Massachusetts Institute of Technology. The authors would like to thank participants of the Privacy Law Scholars Conference, the Roundtable on Computer Science and the Law at the University of Pennsylvania, the faculty of the Georgetown, American University, Loyola L.A., and Washington University law schools, and the students and faculty of the Cardozo Law IP/IL Colloquium. We thank in particular Fred Bloom, Chris Buccafusco, Michael Carroll, Bryan Choi, Julie Cohen, Lorrie Cranor, Pierre De Vries, Deven Desai, Christine Farley, Nick Feamster, Brett Frischmann, Brian Galle, Erik Gerding, James Grimmelmann, Scott Jordan, Margot Kaminski, Pauline Kim, David Lehr, Bill McGeveran, Arvind Narayanan, Scott Peppet, Neil Richards, Jennifer Rothman, Bryant Walker Smith, Neel Sukhatme, David Super, Aaron Wright, Felix Wu, and Christopher Yoo. Thanks to John Douglass for research assistance.

INTRODUCTION

What values do software developers promote and protect with their code? Ten years ago, the answer would have been straightforward: the value that typically mattered most was efficiency. From the first moment

of undergraduate instruction in universities around the globe, computer-scientists-to-be are taught always to optimize for efficiency.[1] In similar fashion, twenty-first-century consumers have been conditioned to yearn for faster performance, higher bandwidth, and greater storage.[2]

This has all started to change. Today's developers have been asked to consider coding their systems to protect a broader range of human values such as fairness, autonomy, liberty, and legitimacy, to name only a few. These calls to consider values beyond efficiency reflect an increasing awareness of the fundamental role software plays today in the ordering of society and regulation of human behavior.

Nearly two decades ago, Lawrence Lessig observed that "code is law."[3] This vital observation has never been as important as it is today, as code—software and algorithms—begins to "regulate" spheres of human behavior that we once protected primarily through law. Software increasingly runs the world, giving rise to new problems from invasions of privacy,[4] to discrimination,[5] to product safety scares,[6] to surveillance.[7] Many legal scholars, increasingly worried that the laws designed for the

---

1. *Cf.* THE JOINT TASK FORCE ON COMPUTING CURRICULA ET AL., COMPUTER SCIENCE CURRICULA 2013 50 (2013), http://www.acm.org/education/CS2013-final-report.pdf (mentioning the importance of a student developing the "mathematical maturity" to analyze algorithmic efficiency).

2. *See generally* JAMES GLEICK, FASTER: THE ACCELERATION OF JUST ABOUT EVERYTHING (1st ed. 1999) (describing how the use of time-saving devices and time-saving strategies has led to a human condition focused on speed).

3. LAWRENCE LESSIG, CODE: AND OTHER LAWS OF CYBERSPACE 6 (1st ed. 1999).

4. *See generally* JULIE E. COHEN, CONFIGURING THE NETWORKED SELF: LAW, CODE, AND THE PLAY OF EVERYDAY PRACTICE (2012) (discussing the lack of restrictions on the flow of personal information through technology); DANIEL J. SOLOVE, THE DIGITAL PERSON: TECHNOLOGY AND PRIVACY IN THE INFORMATION AGE (2004) (explaining the unchecked creation of digital dossiers by electronic databases, which consist of personal data gathered from consumers).

5. *See* Pauline T. Kim, *Data-Driven Discrimination at Work*, 58 WM. & MARY L. REV. 857, 857 (2017) (arguing that "[a]lgorithms built on inaccurate, biased, or unrepresentative data can produce outcomes biased along lines of race, sex, or other protected characteristics); Solon Barocas & Andrew D. Selbst, *Big Data's Disparate Impact*, 104 CALIF. L. REV. 671, 677 (2016) (discussing the mechanisms that render data mining discriminatory).

6. *See* FDA, MOBILE MEDICAL APPLICATIONS: GUIDANCE FOR INDUSTRY AND FOOD AND DRUG ADMINISTRATION STAFF 8 (2015), https://www.fda.gov/downloads/MedicalDevices/.../UCM263366.pdf (explaining how it intends to apply its oversight authority to mobile medical applications); Danielle Keats Citron, *Reservoirs of Danger: The Evolution of Public and Private Law at the Dawn of the Information Age*, 80 S. CAL. L. REV. 241, 245 (2007) (describing the tension between "the social goals of economic growth and individual safety").

7. *See generally* LAURA K. DONOHUE, THE FUTURE OF FOREIGN INTELLIGENCE: PRIVACY AND SURVEILLANCE IN A DIGITAL AGE (2016) (discussing the U.S. government's collection of massive amounts of data from individuals); JENNIFER STISA GRANICK, AMERICAN SPIES: MODERN SURVEILLANCE, WHY YOU SHOULD CARE, AND WHAT TO DO ABOUT IT (2017) (explaining how surveillance law has fallen behind while surveillance technology has grown exponentially).

pre-digital world are ill-equipped to control the new reach of code, have recently taken a hard turn toward the study of regulating software system design.[8] They argue that we can regulate code better if we replace some of our legacy rules with new ones focused on shaping the design of software and digital systems.[9]

Policymakers have joined these calls, casting increasingly suspicious eyes at how those who write code wield power over speech and decisionmaking.[10] It seems likely that the next decade will bring a wave of new laws regulating code and coders.

But the shift to consider values other than efficiency comes not only from regulators. Software companies and the developers who work for them have repeatedly identified the need to create systems that are trustworthy, fair, unbiased, and respectful. Some have been motivated by fear of consumer backlash or government regulation; others more nobly by the desire to do well and to serve society.

We must study closely the means by which developers have regulated themselves in the absence of the pressure of a law mandating them to do so. How have developers tried to create systems that were fair or

---

8. Most notably, there is an emerging academic literature around what is called "privacy by design." WOODROW HARTZOG, PRIVACY'S BLUEPRINT: THE BATTLE TO CONTROL THE DESIGN OF NEW TECHNOLOGIES (2018); Michael Birnhack et al., *Privacy Mindset, Technological Mindset*, 55 JURIMETRICS 55, 55 (2014); Deirdre K. Mulligan & Jennifer King, *Bridging the Gap Between Privacy and Design*, 14 U. PA. J. CONST. L. 989, 989 (2012); Neil M. Richards & Jonathan H. King, *Big Data Ethics*, 49 WAKE FOREST L. REV. 393, 430–31 (2014); Ira S. Rubinstein & Nathaniel Good, *Privacy by Design: A Counterfactual Analysis of Google and Facebook Privacy Incidents*, 28 BERKELEY TECH. L.J. 1333, 1333 (2013); Ira S. Rubinstein, *Regulating Privacy by Design*, 26 BERKELEY TECH. L.J. 1409, 1410 (2011). The embrace of this concept has led to parallel calls for "by design" regulation to address other persistent problems. Andrew Guthrie Ferguson, *The Internet of Things and the Fourth Amendment of Effects*, 104 CALIF. L. REV. 805, 879 (2016) (discussing "security by design"); Woodrow Hartzog & Frederic Stutzman, *Obscurity by Design*, 88 WASH. L. REV. 385, 385 (2013) (stating that "[d]esign-based solutions to confront technological privacy threats are becoming popular with regulators"); Rebecca Wexler, *Warrant Canaries and Disclosure by Design: The Real Threat to National Security Letter Gag Orders*, 124 YALE L.J. F. 158, 160 (2014) (predicting that "companies increasingly will design canary-like alerts embedded into technology to notify users when their data suffers a security breach").

9. HARTZOG, *supra* note 8.

10. Elizabeth Dwoskin & Hamza Shaban, *In Silicon Valley, the Right Sounds a Surprising Battle Cry: Regulate Tech Giants*, WASH. POST (Aug. 24, 2017), https://www.washingtonpost.com/business/economy/in-silicon-valley-the-right-sounds-a-surprising-battle-cry-regulate-tech-giants/2017/08/24/818a6518-8832-11e7-961d-2f373b3977ee_story.html; Kenneth P. Vogel & Cecilia Kang, *Senators Demand Online Ad Disclosures as Tech Lobby Mobilizes*, N.Y. TIMES (Oct. 19, 2017), https://www.nytimes.com/2017/10/19/us/politics/facebook-google-russia-meddling-disclosure.html; *America's Tech Giants Have No Political Party to Protect Them*, ECONOMIST (Oct. 26, 2017), https://www.economist.com/news/united-states/21730652-they-could-eventually-receive-kind-scrutiny-banks-received-after-financial.

trustworthy, or code that respected the autonomy or liberty of other people? Interestingly, this is not simply a matter of legal scholars learning what computer scientists already know; we perceive that the computer scientists who have started to spearhead innovative approaches of generating values-by-code are neither systematically talking to one another, nor synthesizing what they are creating, nor seeing their work as part of an emerging new subdiscipline of research. In this Article, we call for a new, interdisciplinary research agenda investigating how values can be embedded into code. We inaugurate this agenda by identifying a specific new trend that not even the participants have recognized.

The trend we have unearthed is perhaps best symbolized by a shoebox-sized metal receptacle sitting in a server room in northern New Jersey.[11] This box contains Wall Street's best hope for counteracting the undesirable side-effects of high-frequency trading,[12] behaviors blamed for market instability and flash crashes[13]: a coil of fiber-optic cable thirty-eight miles long.[14] Signals propagate through this conduit at two-thirds the speed of light, meaning that it takes data 350 microseconds—millionths of seconds—to travel from one end of the cable to the other.[15] All trading communication with IEX, a new stock exchange designed around this hallowed length of wire, must first traverse the cable, enduring a delay longer than that to any other exchange on Wall Street.[16]

Before the recent shift in software from coding for efficiency to other values, a box like this would have made no sense. The IEX shoebox, which imposes an artificial delay on all communication, represents a strike against the single-minded law of efficiency.[17] IEX argues that its

---

11. MICHAEL LEWIS, FLASH BOYS: A WALL STREET REVOLT 177–78 (1st ed. 2014).

12. *Id.* at 178.

13. *See* Andrei A. Kirilenko et al., *The Flash Crash: The Impact of High Frequency Trading on an Electronic Market*, 72 J. FIN. 967, 967 (2017).

14. LEWIS, *supra* note 11, at 177.

15. Light travels through a vacuum at approximately 300 million meters per second, *NIST Reference on Constants, Units, and Uncertainty*, PHYSICSNIST.GOV, http://physics.nist.gov/cgi-bin/cuu/Value?c (last visited Aug. 13, 2016), or just over 671 million miles per hour, equivalent to 186,000 miles per second. It moves more slowly through fiber-optic cables, in the range of about 124,000 miles per second according to a networking company marketing to traders. Brian Quigley, *Speed of Light in Fiber - The First Building Block of a Low-Latency Trading Infrastructure*, BLOG.ADVAOPTICAL.COM (Apr. 7, 2011), http://blog.advaoptical.com/speed-light-fiber-first-building-block-low-latency-trading-infrastructure. Dividing the thirty-eight miles of IEX's fiber-optic cable by this speed, we find that it would take light just over 300 microseconds to travel from one end of the cable to the other. Since a wide variety of fiber-optic cables with different performance profiles are available on the market, IEX likely uses a brand through which light moves at a slightly slower speed than in our calculations.

16. Jeremy Kahn, *Brad Katsuyama's Next Chapter*, BLOOMBERG (Aug. 23, 2015, 7:00 PM), http://www.bloomberg.com/news/features/2015-08-23/brad-katsuyama-s-next-chapter.

17. *Id.*

system protects fairness, a value it deems lacking in other systems.[18] High-frequency traders disagree, arguing that IEX is the one acting unfairly.[19]

This is but one instance among many of systems that quietly depend on inefficiency to run properly in the name of providing or protecting a non-efficiency value. This Article closely analyzes this family of examples, which is bound together by a formerly unarticulated design principle we call *desirable inefficiency*. Computer scientists have, in a variety of situations, turned to desirable inefficiency in ways that initially might seem as ludicrous as forcing photons to loop through thirty-eight miles of cable in a metal shoebox. From Bitcoin's "proof of work,"[20] to proposed solutions to combat spam,[21] to the time delays built into the iPhone passcode lock,[22] inefficiency has been intentionally injected into systems, dialing back the raw speed and power our information-age conditioning instinctually reveres—all as a means for promoting non-efficiency values.

The values these systems provide run a wide gamut. IEX's fiber coil creates the conditions for fairness among traders, which engenders trust in their market, in ways that will be described in greater detail later.[23] Many of the examples we will examine provide one or both of these two vital human values, fairness or trust, or make systems more compatible with other values such as legitimacy or respect for individual autonomy.[24] In an age of great anxiety over the rise of black box computer systems that dole out government benefits,[25] drive cars,[26] identify potential

18. *Id.*

19. *Id.*

20. SATOSHI NAKAMOTO, BITCOIN, BITCOIN: A PEER-TO-PEER ELECTRONIC CASH SYSTEM 3, https://bitcoin.org/bitcoin.pdf (last visited Apr. 22, 2018).

21. Cynthia Dwork & Moni Naor, *Pricing via Processing or Combatting Junk Mail*, *in* ADVANCES IN CRYPTOLOGY - CRYPTO '92 139, 139–40 (1992).

22. *Infra* note 201and accompanying text.

23. *Infra* Section II.C.

24. *Infra* Section II.C (discussing values instantiated through desirable inefficiency).

25. *See generally* FRANK PASQUALE, THE BLACK BOX SOCIETY: THE SECRET ALGORITHMS THAT CONTROL MONEY AND INFORMATION (2015) (discussing black box computer systems and its effect on privacy of persons); Danielle Keats Citron & Frank Pasquale, *The Scored Society: Due Process for Automated Predictions*, 89 WASH. L. REV. 1, 28 (2014) (discussing automated systems and their effect on benefit programs).

26. Sven A. Beiker, *Legal Aspects of Autonomous Driving*, 52 SANTA CLARA L. REV. 1145, 1146 (2012); *see also* Harry Surden & Mary-Anne Williams, *Technological Opacity, Predictability, and Self-Driving Cars*, 38 CARDOZO L. REV. 121, 121 (2016) (explaining how autonomous vehicles function, the problems with autonomous vehicles, and potential legal solutions for them).

criminals,[27] and fire missiles,[28] we think the study of desirable inefficiency as a method for injecting human values into code might point the way toward a more balanced coexistence with our machines.

To be clear, we are using the words "efficient" and "inefficient" as a computer scientist might rather than as an economist would.[29] The systems we study in this Article each solve two problems simultaneously: a basic technical problem—providing a digital market for trading stocks—and an enhanced, value-oriented problem that defies quantification—providing a digital market for stocks that achieves "fairness" by filtering out difficult-to-define high-frequency trading. The systems are viewed from the vantage point of the basic problem that the solution to the enhanced problem appears inefficient and wasteful (perhaps even bizarre). No designer would impose an "artificial" 350-microsecond delay in a simple digital market without the need to solve the enhanced problem as well.

To lay a foundation for the study of desirable inefficiency as a goal for regulation, we identify four design patterns in desirably inefficient systems.[30] First, desirably inefficient systems tend to act as filters for conduct and activity, making it possible to separate and segregate actions into "good" and "bad." Second, as with IEX's fiber loop, desirable inefficiency is sometimes implemented through hardware rather than software.[31] We delve deeply into this choice, as it seems to violate the disciplinary dogma that software and hardware are interchangeable,[32]

---

27. CLARE GARVIE, ALVARO BEDOYA & JONATHAN FRANKLE, GEO. L. CTR. ON PRIVACY & TECH., THE PERPETUAL LINE-UP: UNREGULATED POLICE FACE RECOGNITION IN AMERICA 1 (2016), https://www.perpetuallineup.org/report; Andrew Guthrie Ferguson, *Predictive Policing and Reasonable Suspicion*, 62 EMORY L.J. 259, 261 (2012); Michael L. Rich, *Machine Learning, Automated Suspicion Algorithms, and the Fourth Amendment*, 164 U. PA. L. REV. 871, 871 (2016).

28. EXEC. OFFICE OF THE PRESIDENT NAT'L SCI. & TECH. COUNCIL COMM. ON TECH., PREPARING FOR THE FUTURE OF ARTIFICIAL INTELLIGENCE 37–38 (2016), https://obamawhitehouse.archives.gov/sites/default/files/whitehouse_files/microsites/ostp/NSTC/preparing_for_the_future_of_ai.pdf.

29. We make this choice as computer scientists speaking, at least in part, to other computer scientists. Economists will likely understand what we are referring to as "inefficiency" as merely the efficient solution to a redefined welfare function.

30. The concept of "design patterns" borrows from the work of Christopher Alexander, a scholar of architecture. CHRISTOPHER ALEXANDER ET AL., A PATTERN LANGUAGE: TOWNS, BUILDINGS, CONSTRUCTION x–xi (1977). This work has already escaped use in architecture and has been especially influential in the study of software engineering, where it has been embraced as a metaphor for object-oriented software design. *See* ERICH GAMMA ET AL., DESIGN PATTERNS: ELEMENTS OF REUSABLE OBJECT-ORIENTED SOFTWARE 1–2 (1994). Other legal scholars have adapted Alexander's method beyond architecture. Erik F. Gerding, *Contract as Pattern Language*, 88 WASH. L. REV. 1323, 1326–27 (2013).

31. Kahn, *supra* note 16.

32. Paul Ohm & Blake Reid, *Regulating Software When Everything Has Software*, 84 GEO. WASH. L. REV. 1672, 1676 (2016).

while also offering a counterexample to the idea that software's dynamism and generativity are always better than hardware's rigidity.[33] Third, desirably inefficient systems tend to be tunable. Fourth, desirably inefficient systems often rely upon, and sometimes provide the preconditions for, decentralized systems, providing a replacement for systems that require centralized intermediaries to operate.

Following the lead of software designers, regulators should consider mandating desirable inefficiency in order to address various concerns about information systems. As one example, we repurpose IEX's shoebox to address information privacy problems. Information privacy sometimes (but not always) depends on the speed with which data can be transmitted and processed, so we think mandatory time delays might sometimes offer a new source of privacy protections. Perhaps we can respond to claims of violation of the "right to be forgotten" by forcing Google to hold back some search results for a few minutes.[34] Perhaps we can offset the possibility that a child predator will use a facial recognition system to gain the trust of a child on a playground by placing all such systems behind a one-hour delay.[35] It would be like requiring Google or the facial recognition service to place its "server on Mars," the evocative label we give to this new approach to regulation.

We do not claim that mandated desirable inefficiency such as the server on Mars will be appropriate in every situation; this is no panacea. But under the right conditions, we think desirable inefficiency will be better than any alternative approach to regulation, including prohibitions, mandates, nudges, or taxes. Unlike most command-and-control prohibitions or mandates, desirable inefficiency will operate less like a binary switch and more like a tunable dial. If the delay proves to be too short, the regulator can force Google or the facial recognition service to metaphorically move its server to Neptune (an eight-hour delay). If the delay is too long, the regulators can allow the server to move to the Moon (1.3 seconds). Unlike nudges, mandated desirable inefficiency is much more interventionist and direct.[36] Rather than tweaking choice

33. JONATHAN ZITTRAIN, THE FUTURE OF THE INTERNET—AND HOW TO STOP IT 15 (2008).

34. EUROPEAN UNION, GENERAL DATA PROTECTION REGULATION, REGULATION 2016/679 ART. 17 (Apr. 5, 2016), https://publications.europa.eu/en/publication-detail/-publication/3e485e15-11bd-11e6-ba9a-01aa75ed71a1/language-en (providing right to erasure); Case C-131/12, Google Spain SL v. Costeja, 2014 E.C.R. I-317, http://eur-lex.europa.eu/legalcontent/EN/TXT/?uri=CELEX%3A62012CJ0131 (holding that Google should be required to remove or hide plaintiff's personal data so that it does not appear in Google search results). *See* MEG LETA JONES, CTRL+Z: THE RIGHT TO BE FORGOTTEN 27 (2016) (discussing the *Costeja* case).

35. *See* GARVIE, BEDOYA & FRANKLE, *supra* note 27, at 25.

36. RICHARD H. THALER & CASS R. SUNSTEIN, NUDGE: IMPROVING DECISIONS ABOUT HEALTH, WEALTH, AND HAPPINESS 6 (2008).

architectures in order to account for human cognitive biases, desirable inefficiency is most effective when the harms are great and the regulator wants to dictate behavior forcefully. In some cases, mandated desirable inefficiency operates a bit like a tax. One proposed solution to spam imposes a cost, in the form of computation, on every email transmitted across the network.[37] But unlike a tax, desirable inefficiency can be laid over a decentralized network, which explains why proof of work sits at the heart of the decentralized Bitcoin protocol.[38]

* * *

Whether as a tool for regulators or as an approach for private actors, we think that desirable inefficiency heralds an important advance at the juncture where code meets human values.[39] It is a useful design strategy for building systems that are fairer and more trustworthy than those they replace and thus more likely to complement rather than disrupt human expectations and institutions. Enabling a human-centric approach to control the encroachment of technology into every area of human life could help this complementary goal.

At a higher level, we believe that our method of analysis in this Article offers a path for regulators increasingly concerned about human values like bias and fairness in technical systems. We (1) observe a situation in which many engineers have, in uncoordinated fashion, voluntarily adopted a design pattern to achieve certain values; (2) formally analyze the properties of the design pattern to understand how it enables these values; and (3) suggest situations in which regulators should consider imposing it. Although desirable inefficiency is our focus in this Article, it is just one example of a range of other technical remedies with which engineers are experimenting as values increasingly influence the design of technical systems.

This Article proceeds in four parts. Part I surveys the importance of efficiency in computer science and documents the underappreciated emergence of systems that resist this trend. Part II defines desirable inefficiency precisely and investigates how it has served to inject human values into software. Part III catalogs four important design patterns of these systems. Finally, Part IV applies these principles to regulation,

37. Dwork & Naor, *supra* note 21, at 139–40.

38. NAKAMOTO, *supra* note 20, at 3.

39. We connect this work to the growing literature on values-in-design. *See generally* DEBORAH G. JOHNSON & HELEN NISSENBAUM, COMPUTERS, ETHICS, AND SOCIAL VALUES 4–6 (1995) (discussing ethical and social values related to a growing computer environment as well as professional responsibility concerns for programmers); Batya Friedman et al., *Value Sensitive Design and Information Systems*, *in* HANDBOOK OF INFORMATION AND COMPUTER ETHICS 69 (K.E. Himma & H.T. Tavani eds., 2008) (explaining value sensitive design through case studies).

surveying what a design-focused, desirably inefficient approach can do that ordinary approaches to the regulation of technology cannot.

I. EFFICIENCY AND OTHER VALUES IN COMPUTING

Efficiency is among the most important values in computer science.[40] Its centrality is so deep-seeded that it is rarely verbalized in the computational community, tantamount to the emphasis medical professionals place on preserving human life. There are sound economic reasons for pursuing this value: efficient code can do more work in less time using fewer resources, a benefit that directly translates into financial rewards.[41]

In recent years, however, a series of independent technical efforts have simultaneously arrived at a shared design principle that seemingly violates this disciplinary doctrine in the name of protecting values other than efficiency. Inefficiency, a property that generations of researchers have toiled to stamp out, is sometimes necessary or useful, providing for values such as fairness, autonomy, or legitimacy. For this reason, we connect this work to those engaged in studying "values in design," "privacy by design," and "security by design."[42]

We christen the shared mode of thought that unites our examples *desirable inefficiency*, a conceptual framework that remains loosely articulated within the computational community and virtually unknown beyond it. For computer scientists and policymakers alike, there is a tremendous deal to learn from closely inspecting this emerging paradigm. To fully appreciate the implications of this trend, however, we must understand the importance of efficiency in computer science.

A. *Efficiency: Computing's Cardinal Virtue*

Computational efficiency can be weighed on a number of axes—the running time of a program, the amount of electricity it consumes, the storage space it occupies, the accuracy the underlying algorithm,[43] or

40. *See* Harry R. Lewis & Christos H. Papadimitriou, *The Efficiency of Algorithms*, 238 SCI. AM. 96, 96 (Jan. 1978) (discussing the importance of algorithmic efficiency).

41. THOMAS H. CORMEN ET AL., INTRODUCTION TO ALGORITHMS 12–13 (3d ed. 2009) (discussing the importance of algorithms).

42. *See supra* note 8 and accompanying text.

43. The accuracy of a machine learning algorithm can be interpreted as the efficiency with which it is able to classify examples. Alternatively, the only value more precious than efficiency is perhaps *correctness*—whether a program came to the right answer. Traditional computer programs are typically assumed to be correct for the purposes of efficiency analysis. In contrast, machine learning algorithms, which aim to make predictions about unseen data, rarely come close to perfect correctness. In this sense, the accuracy of a machine learning algorithm is a consideration even more fundamental than efficiency.

even the amount of time necessary to develop it.[44] Each of these quantities is scarce, expensive, and, therefore, extraordinarily precious. Resource limitations were particularly severe on primitive computers in the early days of computer science, the formative era that shaped the discipline's psyche.[45] Today, courses studying canonically efficient algorithms and techniques for designing new ones lie at the heart of computer science curricula.[46]

Economically, this emphasis is unsurprising. Even in an age of ubiquitous computing devices performing billions of calculations per second, superfluous work can be costly. In 2010, for example, Google's datacenters—huge warehouses of servers that drive its search, video, email, and other offerings—consumed as much electricity as 200,000 homes.[47] A marginally more parsimonious algorithm that improved power-efficiency by a fraction of a percent would save the company millions of dollars each year.[48] Conveniently, this enhancement would also advance purposes beyond Google's bottom line, diminishing the environmental externalities of the vast computing infrastructure necessary to power the modern web.[49] It is little wonder, then, that when hiring, tech companies eschew traditional interviews for rigorous, multi-hour onslaughts of algorithmic puzzles in need of efficient solutions.[50]

In many contexts, efficiency represents far more than a mere luxury for conserving resources and reducing expenditures. Rather, it is often the difference that makes solving difficult computational problems achievable. Many tasks continue to remain hopelessly beyond the reach of even the world's most powerful supercomputers, but each passing year

---

44. CORMEN ET AL., *supra* note 41.

45. *See* C.G. Bell et al., *The Evolution of the DECsystem10*, 21 COMMS.ACM 44, 44–45 (1978) (discussing the then state-of-the-art PDP-10 computer and the evolution of its predecessor models, which were priced in the tens or hundreds of thousands of dollars and had on the order of kilobytes of memory).

46. *See* THE JOINT TASK FORCE ON COMPUTING CURRICULA ET AL., *supra* note 1, at 37.

47. James Glanz, *Google Details, and Defends, Its Use of Electricity*, N.Y. TIMES (Sept. 8, 2011), http://www.nytimes.com/2011/09/09/technology/google-details-and-defends-its-use-of-electricity.html.

48. Jack Clark, *Google Cuts Its Giant Electricity Bill with DeepMind-Powered AI*, BLOOMBERG TECH. (July 19, 2016, 3:54 PM), https://www.bloomberg.com/news/articles/2016-07-19/google-cuts-its-giant-electricity-bill-with-deepmind-powered-ai ("Saving a few percentage points of electricity usage means major financial gains for Google.").

49. *See Google Green*, GOOGLE, https://www.google.com/green/efficiency/datacenters/ (last visited Aug. 14, 2016) (discussing energy efficiency measures in Google's datacenters).

50. A cottage industry has arisen of books purporting to help prospective employees with this style of interview. *E.g.* GAYLE LAAKMANN MCDOWELL, CRACKING THE CODING INTERVIEW 2 (6th ed. 2015).

sees another formidable problem tamed into tractability by a computer scientist who has developed a cleverly efficient algorithm.[51]

This preoccupation with efficiency extends to the sphere of hardware, where engineers strive to build ever-faster circuitry. Gordon Moore, the founder of Intel, famously predicted in 1965 that the number of transistors that could fit on a processor—and thereby increase the raw performance of the chip—would double every one to two years as technology improved and components shrank.[52] Until recent years (when transistor sizes began to near the atomic level[53]) this trend continued unabated in what became known as Moore's Law, doubling computer speeds at least biennially.[54] Moore conceded in 2003 that transistor technology was nearing the boundaries of feasible miniaturization,[55] inspiring a corresponding renaissance in software efficiency among programmers who must contend with this new hardware constraint.

As the dividends of these electronics advancements have dramatically amplified computational capacity while decreasing accompanying prices, engineers have begun to focus on the efficiency of a different resource: themselves. In an employment market where demand far outstrips supply, software developers are at a premium.[56] Any measure that improves a programmer's productivity is thus of enormous fiscal consequence to her employer. In deference to the precious time of their colleagues, professional programmers are intimately concerned with style—formatting and structural conventions that keep code consistent, predictable, and comprehensible across an engineering organization.[57] In large technical outfits, these rules are codified in tomes of legalistic edicts governing proper indentation, line length, and capitalization. Google's style guide for the Java language, for example, comprises 5,000 words

51. *See, e.g.*, Adrian Cho, *Mathematician Claims Breakthrough in Complexity Theory*, SCI. (Nov. 10, 2015, 5:45 PM), http://www.sciencemag.org/news/2015/11/mathematician-claims-breakthrough-complexity-theory.

52. Gordon E. Moore, *Cramming More Components onto Integrated Circuits*, 38 ELEC. 114, 115 (1965).

53. Max Schulz, *The End of the Road for Silicon?*, 399 NATURE 729, 730 (1999) (predicting the eventual limitations of silicon-based transistors); Tom Simonite, *Intel Puts the Brakes on Moore's Law*, MIT TECH. REV. (Mar. 23, 2016), https://www.technologyreview.com/s/601102/intel-puts-the-brakes-on-moores-law/ (confirming Schulz's prediction).

54. Gordon E. Moore, *No Exponential Is Forever: But "Forever" Can Be Delayed!*, IEEE INT'L SOLID-STATE CIRCUITS CONFERENCE (Feb. 10, 2003), https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=1234194.

55. *Id.*

56. BUREAU OF LAB. STATS., *Occupational Outlook Handbook: Software Developers*, http://www.bls.gov/ooh/computer-and-information-technology/software-developers.htm (last updated Dec. 17, 2015).

57. *See generally* BRIAN W. KERNIGHAN & P. J. PLAUGER, THE ELEMENTS OF PROGRAMMING STYLE (2d ed. 1978) (discussing different elements of style to improve programming).

dictating the expected formatting of every character that a programmer might type.[58] These onerous requirements serve a valuable purpose for efficiency—an extra hour spent writing compliant code pays for itself many times over when inevitable readers can quickly surmise its design and behavior.[59]

It is not enough to call efficiency an important value in computer science. It is a tacit presumption, not only taken for granted but implicitly regarded as nearly inviolable.

B. *The Value of Inefficiency: Cryptography*

In a few corners of computer science, the central preoccupation with efficiency is turned on its head. A major contribution of this Article is to begin to draw this disconnected group of research and ideas into a coherent category of methods. In these circles, problems that lack efficient algorithms and cannot be "broken" by clever insights are considered extraordinarily valuable.[60] We highlight this contrarian community and delineate the unspoken design values that underpin its unorthodox priorities. These principles remain exotic even to mainstream computer scientists, let alone legal scholars, but we believe that both disciplines can draw important insights from the unheralded moments when the central dogma of computer science fails.

The most mainstream and prominent members of this unconventional subfield are cryptographers, whose duty is to make extracting encrypted information as challenging as possible.[61] Ideally, doing so should be so difficult that an adversary (as security researchers call their imaginary foes) constrained by the laws of time, physics, and reasonable efficiency should find the task impossible.[62] Information thus locked away will remain permanently secure from anyone lacking the proper credentials. Cryptographers represent our first example of a unique breed of computer scientists who engineer with inefficiency. Their practice is extreme, however, designing inefficiency with the same zeal that their peers might display when designing a faster algorithm. Efficient algorithmic

58. *See* Google, *Google Java Style Guide*, GITHUB, https://google.github.io/styleguide/javaguide.html#s7-javadoc (last visited Aug. 18, 2016).

59. Toward the same end, professional programmers often strategically decide *not* to improve the efficiency of certain code, calculating that this inaction is more cost-efficient than investing expensive engineering-time to marginally improve performance.

60. *See* James Nechvatal et al., *Report on the Development of the Advanced Encryption Standard (AES)*, 106 J. OF RES. OF THE NAT'L INST. OF STANDARDS AND TECH., 511, 524 (2001) (discussing the desire to utilize an encryption algorithm that is secure against analytic shortcut attacks).

61. BRUCE SCHNEIER, APPLIED CRYPTOGRAPHY 21 (2d ed. 1996).

62. C.E. Shannon, *Communication Theory of Secrecy Systems*, 27 BELL SYS. TECHNICAL J. 656, 659 (1948) (describing the concept of "perfect secrecy" in cryptography).

techniques that weaken these defenses are cause for immense concern, having revealed unforeseen flaws in ostensibly impenetrable mathematical armor.[63]

Although cryptography depends on the inefficiency of adversaries, it should simultaneously minimize the computational burden imposed on the computers of users who possess the proper credentials. In this respect, encryption schemes should continue to meet traditional criteria for efficiency.[64] Ciphers robust enough to tolerate billions of secure web connections each day must deftly navigate this careful tension between remaining impregnable to adversaries and nearly invisible to users. In this light, cryptographers treat efficiency and inefficiency, not as values to blindly pursue in every circumstance, but as tools to thoughtfully deploy according to need and circumstance.

C. *Introducing Desirable Inefficiency*

Between the extreme inefficiency cryptography imposes on its adversaries and the mainstream pursuit of algorithmic performance is a delicate middle ground where inefficiency is judiciously calculated and balanced. Many of these systems encompass what we define as *desirable inefficiency*. The essence of desirable inefficiency, which we will define in Part II, is a connection between apparently inefficient code and human values. Engineers turn to desirable inefficiency when they conclude that conventionally efficient solutions are unlikely to advance human values that cannot be coded into the technical substance of the system.

Desirably inefficient systems are the digital scions to non-digital precursors such as speed bumps and stop signs, which force drivers to limit their pace, not to a complete standstill, but to a rate amenable to the safety of pedestrians and others with whom they share the road. These traffic-calming devices impose a moderate degree of desirable

63. When these "attacks" weaponize efficiency against protocols actively deployed on hundreds of millions of computers, concern rapidly gives way to outright panic. An encryption scheme so fatally damaged by efficiency must swiftly be discarded and replaced, a task of unthinkable logistical proportions in our modern digital world.

To minimize the risks of such a disaster, cryptographic suites endure immense public scrutiny before they can attain certification for general use. Modern protocols are chosen through contests in which groups from academia and industry submit candidate schemes for wider consideration. Teams of rival cryptographers then work to develop efficient algorithms that undermine the proposals, uncovering fatal deficiencies that might otherwise have remained catastrophically dormant. Only after surviving this brutal battery of attempted attacks can a protocol be considered secure, believed with a high degree of confidence to be immune to efficiency. *See* Nechvatal et al., *supra* note 60, at 519–21. Once beyond this security threshold, researchers can weigh a protocol along more conventional axes.

64. *See generally* SCHNEIER, *supra* note 61 (analyzing the efficiency of encryption schemes).

inefficiency on commuting times, balancing a still-acceptable rate of travel with safety.

A number of digital speed bumps and stop signs are deeply embedded in our everyday interactions with technology—even an interface as simple as the numerical or pattern passcodes used to secure mobile phones is an example of desirable inefficiency. On modern devices, a failed unlocking attempt is followed by a mandatory waiting period of a fraction of a second. After several mistaken entries, a phone will refuse submissions for a period of many seconds or minutes.[65] An efficiency-maximizing approach would allow a user struggling to remember a passcode to try every candidate code that comes to mind as quickly as possible. Instead, phone designers use time delays to rate-limit this process in order to thwart rapid guessing by a thief trying to break into the device.[66] This inefficiency, measured in time, balances the inconvenience imposed on a forgetful user or poor typist with the device's security—only after several failed attempts are the delays so burdensome as to become perceptible, let alone frustrating. This penalty structure is designed to distinguish erroneous keystrokes, which are likely to be corrected after one or two attempts, from brute-force attempts to compel entry.[67]

A second example is the IEX stock exchange. High-frequency traders make their living by exploiting the physical geography of the many exchanges that mediate stock transactions.[68] The time light takes to journey the miles between ordinary exchanges, registering in single digits of microseconds, is significant enough to create momentary pricing discrepancies that high-frequency traders convert into profits.[69] Armed

65. APPLE, IOS SECURITY: IOS 11, 12–15 (2018), https://www.apple.com/business/docs/iOS_Security_Guide.pdf.

66. Kim Zetter, *Apple's FBI Battle is Complicated. Here's What's Really Going On*, WIRED (Feb. 18, 2016, 1:15 PM), https://www.wired.com/2016/02/apples-fbi-battle-is-complicated-heres-whats-really-going-on/.

67. *Id*. This feature was at the heart of the recent Apple/FBI debate over unlocking the San Bernardino shooter's iPhone. The FBI demanded from Apple the ability to guess passwords in an arbitrary rate. *In re* the search of an Apple iPhone, No. ED 15-0451M (E.D. Cal. 2016) (order compelling Apple, Inc. to assist agents in search), *available at* https://assets.documentcloud.org/documents/2714005/SB-Shooter-Order-Compelling-Apple-Asst-iPhone.pdf.

68. LEWIS, *supra* note 11, at 177–78.

69. This phenomenon is not unique to stock exchanges. Any network of computing devices suffers from miniscule communication delays that create information lags between members of the system. The time delays among stock exchanges are relatively small since they are all located in the New York metropolitan area. The Internet, as a global network, faces far larger delays, while the individual components inside a computer experience the same phenomenon on a far smaller scale. These so-called *propagation delays* are responsible for a pernicious computational problem called *clock synchronization*, where networked computers struggle to keep the same time of day as one another with high precision. Messages traveling between computers with out-of-

with the fastest connections that money can buy, they pounce before the exchanges—let alone other traders—have time to react.[70] This so-called "race to zero" has become an arms race among traders, who battle for the ever smaller communication delays that make this style of trading profitable.[71] After repeatedly moving their servers closer to the exchanges and even co-locating them in the same rooms,[72] some companies are going so far as to experiment with laser arrays mounted across the northern New Jersey skyline.[73]

The thousands of dollars spent shortening these distances and investing in faster hardware shaved mere millionths of seconds off of this propagation time, but the advantage in doing so was dramatic. High-frequency traders were effectively assessing a small tax on large stock transactions by capitalizing on asynchrony in asset prices between exchanges before the markets themselves had time to come to consensus.[74]

Consider, as an illustration, a high-frequency trading technique known as front-running.[75] Brokers typically fulfill large stock orders by splitting them into smaller trades across many exchanges which, collectively, have the buying or selling capacity necessary to complete the transaction.[76] A high-frequency trader could front-run some of the components of the larger trade by listening for a single fragment on one exchange and exploiting its faster connections to beat the remaining pieces to others.[77] In the process, the high-frequency trader will enjoy the price increase that

---

sync clocks can sometimes seem to go backwards in time, arriving before they were sent and wreaking havoc on systems that were not designed with this seemingly impossible contingency in mind.

70. Matt Levine, *The 'Flash Boys' Exchange is Still Controversial*, BLOOMBERG VIEW (Dec. 22, 2015, 4:14 PM), https://www.bloomberg.com/view/articles/2015-12-22/the-flash-boys-exchange-is-still-controversial.

71. Eric Budish et al., *The High-Frequency Trading Arms Race: Frequent Batch Auctions as a Market Design Response*, 130 Q. J. OF ECON. 1547, 1548–49 (2015); Christopher Steiner, *Wall Street's Speed War*, FORBES (Sept. 9, 2010, 10:00 AM), http://www.forbes.com/forbes/2010/0927/outfront-netscape-jim-barksdale-daniel-spivey-wall-street-speed-war.html.

72. Geoffrey Rogow, *Colocation: The Root of All High-Frequency Trading Evil?*, WALL ST. J. (Sept. 20, 2012, 1:57 PM), http://blogs.wsj.com/marketbeat/2012/09/20/collocation-the-root-of-all-high-frequency-trading-evil/.

73. Scott Patterson, *High-Speed Stock Traders Turn to Laser Beams*, WALL ST. J. (Feb. 11, 2014, 11:00 PM), http://www.wsj.com/articles/SB10001424052702303947904579340711424615716.

74. Concept Release on Equity Market Structure, 75 Fed. Reg. 3594, 3608 (proposed Jan. 21, 2010) (to be codified at 17 C.F.R. pt. 242).

75. LEWIS, *supra* note 11, at 176.

76. *Id.*

77. Concept Release on Equity Market Structure, 75 Fed. Reg. at 3609.

follows the arrival of the rest of the order.[78] Another strategy, known as stale quote arbitrage, is to observe a buyer seeking a stock at a higher price than that for which a seller elsewhere is offering it.[79] A high-frequency trader can buy the stock from the seller at the lower price and sell it to the buyer at the higher price before the exchanges themselves have time to synchronize, keeping the difference as profit.[80]

The thirty-eight-mile fiber loop eliminates these sorts of strategies by delaying all communication with IEX by 350 microseconds.[81] This delay ensures that no front runner can find out about a transaction until after it has time to propagate to every other exchange.[82] This fiber loop creates inefficiency—transactions arrive at IEX far more slowly than they would in its absence, ending the race to zero.[83] But this inefficiency is desirable: it eliminates an unwanted behavior by subverting high-frequency trades.[84] The inefficiency, then, trades off speed for a human value that cannot be expressed in the economic vocabulary of the underlying system: fairness.[85]

D. *Proof of Work*

Several key examples of desirable inefficiency fall within a category of algorithms known as proof of work. Proof of work problems are computational puzzles that require a tangible, regimented amount of energy to be expended in order to arrive at a solution.[86]

Proof of work has long been a curious tool in need of an application.[87] It was first suggested by Cynthia Dwork and Moni Naor in the early 1990s as a solution to the pernicious problem of spam.[88] In Dwork and Naor's proposal, a computer sending an email would have to complete a proof of work puzzle in order to convince the recipient server to accept the message.[89] This task would consume a small but appreciable amount

78. Levine, *supra* note 70.
79. Kahn, *supra* note 16.
80. *Id.*; Concept Release on Equity Market Structure, 75 Fed. Reg. at 3608.
81. Levine, *supra* note 70.
82. *Id.*
83. *Id.*
84. *Id.*
85. *Id.*
86. Markus Jakobsson & Ari Juels, *Proofs of Work and Bread Pudding Protocols*, *in* 23 SECURE INFORMATION NETWORKS 258, 258 (Bart Preneel ed., 1999) (coining the term "proof of work").
87. Dwork & Naor, *supra* note 21, at 1.
88. *Id. But see* Ben Laurie & Richard Clayton, "*Proof of Work" Proves Not to Work*, THIRD ANN. WORKSHOP ON ECON. OF INFO. SECURITY 8 (2004) (discussing the problems with employing proof of work to fight spam).
89. Dwork & Naor, *supra* note 21, at 4.

of computation time, akin to paying the cost of postage before sending a letter.[90] To the average user transmitting a few dozen emails a day, this minor expense would hardly be noticeable.[91] To a spammer unleashing thousands of messages each minute, however, this cumulative work would be so burdensome as to undermine the entire business model.[92] The key insight underlying this proposal is that proof of work serves as a digital speed bump, permitting email to proceed only on a human timescale.[93]

Ronald Rivest, Adi Shamir, and David Wagner applied similar methods to the problem of digitally sealing data for a set period of time, devising a technique called "time-release cryptography."[94] Public figures, for example, might request that their diaries remain encrypted for 50 years before being made available for historical study.[95] Some have characterized this need as the digital equivalent of burying a time capsule.[96] Since computation has no inherent connection to "wall-clock time" (as computer scientists describe our physical temporal perceptions),[97] securing information in this manner with the mathematical rigor cryptographers demand is deceptively vexing.

The typical cryptographic toolkit, which seeks to maximize inefficiency and render decryption permanently impossible, is far too imprecise for this task. Instead, with time-release cryptography, a message is encrypted weakly enough that sustained computational exertion will yield its contents.[98] The proposal takes into account Moore's Law and other trend lines of technological progress to render encrypted messages susceptible to brute force cracking, but only with technology predicted to be available a few decades from now.[99] The level of

90. *Id.*

91. *Id.*

92. *Id.*

93. *Id.*

94. Ronald L. Rivest et al., *Time-Lock Puzzles and Timed-Release Crypto*, MIT TECHNICAL RPT. 1–2 (1996), http://dl.acm.org/citation.cfm?id=888615.

95. *Id.* at 1. Such a request might be governed by statute. For example, the Presidential Records Act of 1978, 44 U.S.C. § 2204(b) (2012), exempts the records of Presidents from disclosure through Freedom of Information Act (FOIA), 5 U.S.C. § 552 (2012), for five years, after which the presumption flips to public access. 44 U.S.C. § 2204(b); *see also id.* § 2204(a) (providing a longer, twelve-year period of Freedom of Information Act exemptions for Presidential records in certain categories).

96. Ronald L. Rivest, *Description of the LCS35 Time Capsule Crypto-Puzzle*, MIT COMPUTER SCI. & ARTIFICIAL INTELLIGENCE LAB (Apr. 4, 1999), https://people.csail.mit.edu/rivest/pubs/Riv99b.lcs35-puzzle-description.txt.

97. Eric Raymond, *Wall Time*, JARGON FILE (Version 4.4.7), http://www.catb.org/jargon/html/W/wall-time.html.

98. Rivest et al., *supra* note 94, at 2.

99. *Id.*

encryption (and thereby inefficiency) can be finely regulated to adjust the time necessary to extract the message.[100] Just as with Dwork and Naor's solution to spam, Rivest, Shamir, and Wagner harnessed inefficiency in order to attach the abstract difficulty of breaking cryptography back to a human notion of time.[101]

It is the meteoric rise of cryptocurrencies, however, that has elevated the stature of proof of work as a design method.[102] Most prominent among hundreds of new cryptocurrencies is Bitcoin, whose economy is currently valued at more than $127 billion.[103] Each Bitcoin user maintains one or more accounts on the larger Bitcoin network, a collection of thousands of Internet-connected computers (called nodes) that collaboratively facilitate and validate transactions.[104] To send bitcoins[105] from one account to another, a user announces the transaction to every node on the Bitcoin network.[106] These nodes, in turn, each maintain individual ledgers of all the Bitcoin transactions they have ever witnessed.[107] If an account lacks sufficient funds or a transaction otherwise appears invalid, nodes simply ignore it.[108]

Unfortunately, these nodes sometimes disagree with each other when announcements are lost or arrive out of order, a reasonably frequent occurrence built into the underlying architecture of the Internet.[109] Messages take time to propagate across the Internet, so nodes can grow to have vastly different views of the Bitcoin world even when all participants are acting in good faith.[110] Worse, some users might attempt to exploit these inconsistencies to spend money twice, subverting Bitcoin's fundamental ability to function as a currency.[111] Without a way to keep nodes in synchrony, Bitcoin would devolve into a confused tangle of contradictory transactions and uncertain account balances.

---

100. *Id.*

101. *Id.*

102. ARVIND NARAYANAN ET AL., BITCOIN AND CRYPTOCURRENCY: A COMPREHENSIVE INTRODUCTION 1 (2016).

103. *Cryptocurrency Market Capitalizations*, CASHBET COIN, https://coinmarketcap.com (last visited July 20, 2018).

104. NAKAMOTO, *supra* note 20, 3–4.

105. According to the official Bitcoin website, the name should be capitalized "when describing the concept of Bitcoin, or the entire network itself." It should be left uncapitalized when "used to describe bitcoins as a unit of account"—that is, when referring to the currency itself. Vocabulary, BITCOIN, https://bitcoin.org/en/vocabulary#bitcoin (last visited Mar. 9, 2018).

106. NAKAMOTO, *supra* note 20, 3–4.

107. *Id.*

108. *Id.*

109. *Id.*

110. *Id.*

111. NARAYANAN ET AL., *supra* note 102, at 34–35.

At the core of Bitcoin's strategy to remedy this inconsistency is proof of work, making Bitcoin the intellectual descendant of Dwork and Naor's research on spam.[112] Bitcoin divides its massive ledger into a list of blocks, each of which contains approximately ten continuous minutes' worth of transactions that the entire network has already agreed upon.[113] In aggregate, this structure is known as the blockchain.[114] To update the blockchain, a node collects the transactions it has witnessed since the last block was created, assembling them into a candidate block that it hopes the entire network will eventually come to accept.[115] Before it is allowed to propose its block to the network, however, the node must complete a proof of work problem based on the block's contents.[116] The difficulty of this problem is calibrated so that, after ten minutes on average, one node on the entire Bitcoin network should have arrived at a solution for its candidate block.[117] The node that does so sends its block and solution to the entire network, and any node able to validate these items adopts it onto its view of the blockchain, restarting the process.[118]

Proof of work plays a key role in setting the rigid tempo of block creation that delays updates to the blockchain long enough for consensus to form.[119] Ten minutes provides sufficient time for word of a newly-generated block to reach the entire network, ensuring that all nodes share the same view of the transaction ledger before more new blocks appear.[120]

---

112. NAKAMOTO, *supra* note 20, at 3.
113. NARAYANAN ET AL., *supra* note 102, at 41–42.
114. *Id.*
115. NAKAMOTO, *supra* note 20, at 3.
116. *Id.*; NARAYANAN ET AL., *supra* note 102, at 41–42.
117. NARAYANAN ET AL., *supra* note 102, at 41–42.
118. NAKAMOTO, *supra* note 20, at 3.
119. NARAYANAN ET AL., *supra* note 102, at 41–42.
120. Simon Barber et al., *Bitter to Better—How to Make Bitcoin a Better Currency*, *in* FINANCIAL CRYPTOGRAPHY AND DATA SECURITY 399, 411 (2012) ("Another approach is to fundamentally reduce the transaction confirmation delay by re-parameterizing the computational puzzles to reduce the average block creation interval from 10 minutes to 10 seconds. However, this would increase the forking propensity on slow communication networks, which could become a concern."); *see* NARAYANAN ET AL., *supra* note 102, at 43 ("Why do we want to maintain this 10-minute invariant? The reason is quite simple. If blocks were to come very close together, then there would be a lot of inefficiency, and we would lose the optimization benefits of being able to put a lot of transactions in a single block."); *see also* Aviv Zohar, *Bitcoin: Under the Hood*, 58 COMMUNICATIONS OF THE ACM 104, 107 (2015) ("The difficulty of the computational problem is automatically adjusted so blocks are created only once every 10 minutes in expectation throughout the entire network. This period of time is sufficiently long to make conflicts extremely rare.").

Were the time set much lower, competing blocks could simultaneously surface, threatening to inflict further chaotic discord. Were it set much higher, transactions would take so long to settle on the blockchain that the humans involved—merchants and shoppers waiting for payments process—would grow impatient and frustrated, possibly abandoning the currency entirely. NARAYANAN ET AL., *supra* note 102, at 37–38.

As with Dwork and Naor's solution to spam and Rivest et al.'s approach to time-release encryption, Bitcoin's proof of work process uses inefficiency to advance qualitative ends beyond pure performance optimization. All three designs tie computational mechanisms back to human notions of time and speed that that would otherwise dissolve in a technical system left to its own devices.

Proof of work serves an additional purpose as "proof of existence" for nodes on the Bitcoin network.[121] The block mining process is like a voting procedure, where each node gets a "vote" with which to weigh in on its view of the Bitcoin world.[122] Bitcoin uses proof of work to ensure that each node that "votes" represents a real computer with tangible processing capabilities.[123] In other words, proof of work is the metric by which metaphorical voting rights are distributed across the Bitcoin network, a concept crystallized by Bitcoin's mysterious creator Satoshi Nakamoto in the phrase, "one CPU, one vote."[124]

The measured application of desirable inefficiency is an overlooked design technique with implications that extend well beyond the obscure margins of computer science. Computer scientists and legal scholars alike should draw numerous lessons from this collective engineering response. By altering their designs to trade away sought-after efficiency for the sake of advancing other values, computer scientists have blazed a trail that we will later encourage regulators to travel.

E. *Computing and Human Values*

Desirable inefficiency is ultimately a case study of how to inject important human values into services and products. Values like fairness and trust do not exist in the vocabulary of programming languages, nor does any distinction between "good" and "bad" uses of a technology.

---

121. NAKAMOTO, *supra* note 20, at 3.

122. *Id.* In practice, nodes exercise this voting power by mining for blocks and deciding whether newly-created blocks mined by others are, in fact, valid. If this voting analogy were the way Bitcoin truly operated, participants could rig the vote by artificially creating armies of fake nodes for the sole purpose of stuffing the ballot box. *Id.* at 3–4.

123. NAKAMOTO, *supra* note 20, at 3–4.

124. NAKAMOTO, *supra* note 20, at 3. Bitcoin was created under the pseudonym Satoshi Nakamoto, a figure whose identity has remained a mystery since Bitcoin's inception in 2009. An Australian computer scientist named Craig Wright recently announced that he was Satoshi Nakamoto. *Craig Wright Reveals Himself as Satoshi Nakamoto*, ECONOMIST (May 2, 2016), http://www.economist.com/news/business-and-finance/21698060-craig-wright-reveals-himself-as-satoshi-nakamoto. However, Wright declined to provide proof of his identity in the form of moving bitcoins that definitively belong to Nakamoto, casting doubts on his claim. *Wright's Wrongs*, ECONOMIST (May 7, 2016), http://www.economist.com/news/finance-and-economics/21698294-quest-find-satoshi-nakamoto-continues-wrightu2019s-wrongs.

Programmers have conjured desirable inefficiency as the means to encourage ends like fairness and trust.

1. Motivations

This Article builds on earlier work about Values-in-Design, an interdisciplinary approach that investigates socio-technological systems to discover the way system designers and other stakeholders advance human values through technical design.[125] In particular, we are interested in revealing the hidden connections between human values and inefficient design. Given the zeal with which some computer scientists adhere to the goal of efficiency, we find it important and revealing to study the work of those who have instead turned to inefficiency.

The second major foundation stone for this work is Julie Cohen's conception of semantic discontinuity.[126] Semantic discontinuity is a rich prescription Cohen offers as a "structural condition for human flourishing" in digital, intermediated systems.[127] She defines semantic discontinuity as "interstitial complexity within the institutional and technical frameworks that define information rights and obligations and establish protocols for information collection, storage, processing, and exchange."[128] Part of this is what she calls "boundary management" or the creation and maintenance of "gaps" in information systems.[129]

We also build upon less theoretical, more practice-oriented work on the need to inject values into software development practices. There is a rich emerging literature about "Privacy by Design" or "Security by Design."[130] Consistent with these disciplines, we think privacy or security can sometimes be implemented only through desirable inefficiency.

In addition to values-in-design, the other major strand of literature this Article embraces begins where Lawrence Lessig's classic work, *Code: And Other Laws of Cyberspace*, leaves off.[131] It starts with Lessig's maxim that "code is law."[132] This assertion has primarily been used by other scholars as a descriptive observation of the role software plays in delineating what is possible and impossible online—in establishing the physics of cyberspace.[133] In today's world, software directs and channels

125. *E.g.*, Friedman et al., *supra* note 39, at 69; JOHNSON & NISSENBAUM, *supra* note 39, at 1.

126. COHEN, *supra* note 4, at 239.

127. *Id.*

128. *Id.*

129. *Id.* at 248.

130. HARTZOG, *supra* note 8; Rubinstein, *supra* note 8, at 1409; *see* ANN CAVOUKIAN, PRIVACY BY DESIGN 1 (2009), https://www.ipc.on.ca/wp-content/uploads/Resources/7foundationalprinciples.pdf.

131. LESSIG, *supra* note 3.

132. *Id.* at 3.

133. *Id.* at 6.

society much like concrete and steel did once before.[134] Software is rule-like and shapes human activity in ways that used to be the sole province of architecture and regulation.[135]

Too many scholars have interpreted Lessig as doing little more than issuing a license to imagine that anything is possible online, falling into a "science fiction trap." Too rarely do they consider the process of *how* code ends up the way it does (let alone how regulators can make use of this process), leaving a significant void in the utility of this body of work.[136]

We help fill this gap by focusing our attention on a specific technology trend—the emergence of desirable inefficiency—that sees engineers engaging in a sort of self-imposed, design-based regulation, integrating desirable inefficiency when engineers themselves see a deficit of an important human value.

Finally, this work builds on a long tradition in computer engineering of studies that document the patterns of software development. This is an important body of work to the computer science community which legal scholars have underutilized for too long.[137] It can be divided into two categories: one focused on coders and the other focused on code. Some of these studies dissect the social dynamics and human capital of software development.[138] We admire this important work immensely, but this Article is more closely aligned with the latter group: studies of the changing properties of software.[139] Our method is to recognize and document a new trend in the way software is being written, paying less

134. *Id.* at 6–7.

135. *Id.*

136. This is not to say that nobody has explored this territory before us. Two notable early works are R. Polk Wagner, *On Software Regulation*, 78 S. CAL. L. REV. 457, 461 (2005) (stating that whether law and software can coexist in the cyberspace context "will and should drive the policy decisions that shape the emerging regulatory infrastructure of cyberlaw"), and Tim Wu, *When Code Isn't Law*, 89 VA. L. REV. 679, 680 (2003) (discussing "[t]he idea that computer code may be emerging as a meaningful instrument of political will").

137. A notable, praiseworthy, and recent exception published in a legal journal is authored, unsurprisingly, primarily by computer scientists. *See generally* Joshua A. Kroll et al., *Accountable Algorithms*, 165 U. PENN. L. REV. 633 (2017) (discussing the auditability of software). We ally our work with this important recent article.

138. *See generally* FREDERIC P. BROOKS, JR., THE MYTHICAL MAN MONTH: ESSAYS ON SOFTWARE ENGINEERING (Anniversary ed. 1995); E. GABRIELLA COLEMAN, CODING FREEDOM: THE ETHICS AND AESTHETICS OF HACKING (Princeton Univ. Press ed., 2013). For more popular writing along these lines, see SCOTT ROSENBERG, DREAMING IN CODE: TWO DOZEN PROGRAMMERS, THREE YEARS, 4,732 BUGS, AND ONE QUEST FOR TRANSCENDENT SOFTWARE (2007); *see also* KATIE HAFNER & MATTHEW LYON, WHERE WIZARDS STAY UP LATE: THE ORIGINS OF THE INTERNET (1996); TRACY KIDDER, THE SOUL OF A NEW MACHINE (1981).

139. *See generally* BEAUTIFUL CODE: LEADING PROGRAMMERS EXPLAIN HOW THEY THINK (Andy Oram & Greg Wilson, eds. 2007); STEVE MCCONNELL, CODE COMPLETE (2d ed. 2004).

attention to the individuals and social forces that bring about the code. Although we emphasize the technical over the social, however, we are not trying to naturalize software development, treating it as a product of evolution or nature. Software is and will remain the product of human engineering and organization.[140]

2. Values Implicated

Consider how computer systems and networks challenge human values. We are not attempting to provide a complete list but instead focus on some of the problems that designers have addressed with desirable inefficiency.

The problem of fairness is perhaps most important. Questions about the fairness of algorithms have generated an already-massive literature, much of it falling under the interdisciplinary banner of "FAT*," for fairness, accountability, and transparency.[141] This community has interrogated the fairness of using machine learning (ML) models to make important decisions about human beings,[142] the accountability of systems that can put human life or safety at risk,[143] and the interpretability of black box systems.[144] Although none of our examples focus on ML systems, we hope to see future work at the intersection of ML and desirable inefficiency.[145]

Our primary example about fairness is, once again, the problem of front-running in high-frequency trading. Notice how the notion of fairness requires system designers to think beyond the bare technical requirements of the system. There is nothing intrinsically "unfair" in allowing a front-running trader who has invested resources in order to secure a privileged position in the network from driving up the price of

---

140. *Cf.* Langdon Winner, *Do Artifacts Have Politics?*, *in* THE WHALE AND THE REACTOR: A SEARCH FOR LIMITS IN AN AGE OF HIGH TECHNOLOGY 19 (1986) (discussing the connection between technology and politics).

141. An emerging "FAT*" community convene an academic conference. *Fairness, Accountability, and Transparency*, FAT* CONFERENCE, https://www.fatconference.org/ (last visited Nov. 2, 2017) [hereinafter FAT*].

142. Barocas & Selbst, *supra* note 5, at 671–72; *see* Julia Angwin et al., *Machine Bias: There's Software Used Across the Country to Predict Future Criminals. And It's Biased against Blacks*, PROPUBLICA (May 23, 2016), https://www.propublica.org/article/machine-bias-risk-assessments-in-criminal-sentencing.

143. Surden & Williams, *supra* note 26, at 132.

144. *See generally* Pasquale, *supra* note 25 (questioning whether algorithmic applications, whose values and prerogatives are hidden within black boxes, are fair). For the program of a recent NYU Law conference on "Algorithms and Explanations," see NYU Law, *Algorithms and Explanations*, http://www.law.nyu.edu/centers/ili/events/algorithms-and-explanations.

145. Machine learning could perhaps be integrated into the desirable inefficiency paradigm by considering classification accuracy as a metric of efficiency.

another trader's pending trade.[146] In fact, the notion that these trades are unfair has itself been contested by the front-running traders.[147] From their point of view, what is unfair is redesigning a system to deny them the rents they have earned from their architectural investment.[148]

Many other values are implicated, in addition to fairness and trust. Spam can be characterized as intruding upon the liberty of people to be free from unwanted messages. It might also be characterized as disrupting the autonomy of email users.

Automated web scrapers or crawlers are regarded as problems of legitimacy. By harvesting information at a speed or frequency much faster than a human, they consume resources and "cut in line" in front of legitimate customers.[149] Another challenge to legitimacy comes from hackers and thieves who try to guess passwords or passcodes on hosted accounts or devices, in order to gain access to the information of the rightful owner. And the creators of digital currencies have confronted legitimacy concerns of a much larger scale, puzzling over how to prevent a holder of a digital "coin" from spending it twice or otherwise hiding transactions.[150]

This is but one list of values that designers have tried to inject into services and products using desirable inefficiency: fairness, trust, liberty, autonomy, and legitimacy. Later we will elaborate precisely the role desirable inefficiency plays in protecting or encouraging these values.[151] We do not think this list is complete; desirable inefficiency will likely be useful for protecting other values as well.

3. The Rise of Values as a Design Constraint

Efficiency has enjoyed a long reign atop the list of priorities for computer scientists, because software has not often engaged expressly with a broad range of human values, aside from correctness and speed.

We are starting to see more examples of desirable efficiency because software and computer networks have slipped the boundaries of business processing to become an important part of the fabric of our social, political, and aesthetic lives.[152] People use software to communicate and collaborate, build and maintain connections to other people, organize and

146. Levine, *supra* note 70.

147. *Id.*

148. *Id.*

149. *Id.*

150. Sudhir Khatwani, *What is Double Spending & How Does Bitcoin Handle it?*, COINSUTRA (Feb. 2, 2018), https://coinsutra.com/bitcoin-double-spending/.

151. *Infra* Section II.C.

152. Ohm & Reid, *supra* note 32, at 1676.

remember.[153] Software helps people express themselves, find love and sex,[154] and bring beauty and inspiration to their lives and the lives of others. With software, people vote, agitate and protest, and topple preexisting power structures.[155] In some societies, it helps determine who gets fed, housed, killed, and healed.[156]

As software becomes an increasingly important part of these non-economic, non-business, non-corporate parts of our lives, it raises difficult questions about human values such as fairness, due process, equality, privacy, and liberty more than ever before. From the Facebook emotional contagion study[157] and trending topics controversies,[158] to OKCupid's manipulation of dating results,[159] to Apple's battles with the FBI,[160] to the roles Cambridge Analytica and Facebook appear to have played in the election of Donald Trump,[161] it is hard to disagree that software does much more today than merely power the economy. And when the decision is made to attend to values like these in software, desirable inefficiency is an increasingly popular path.

There might be other mechanisms that operate like desirable inefficiency—mechanisms that provide an interface between code and values. We hope that this Article serves as a prototype for analyses that seek to identify and leverage other techniques that engineers have organically developed to make space for values in technical systems.

Finally, we predict that the importance of desirable inefficiency (and other engineering responses to the need for values) is likely to grow. As software continues to collide with values, we expect developers and

153. LESSIG, *supra* note 3, at 4–5.

154. Karen E.C. Levy, *Intimate Surveillance*, 51 IDAHO L. REV. 679, 679–80 (2015).

155. ZEYNEP TUFEKCI, TWITTER AND TEAR GAS: THE POWER AND FRAGILITY OF NETWORKED PROTEST x-xi (2017).

156. Danielle Keats Citron, *Technological Due Process*, 85 WASH U. L. REV. 1249, 1249–50 (2008).

157. Adam D. I. Kramer et al., *Experimental Evidence of Massive-Scale Emotional Contagion through Social Networks*, 111 PROC. NAT'L ACAD. SCI. 8788, 8788 (2014); *see* Gregory S. McNeal, *Facebook Manipulated User News Feeds to Create Emotional Responses*, FORBES (June 28, 2014, 1:10 PM), http://www.forbes.com/sites/gregorymcneal/2014/06/28/facebook-manipulated-user-news-feeds-to-create-emotional-contagion/.

158. Ian Sherr, *Facebook's Trending Topics Hits Turmoil Over Fake News Story*, CNET (Aug. 29, 2016, 2:26 PM), https://www.cnet.com/news/facebook-trending-topics-megyn-kelly-fake-news-story-controversy/.

159. Cheryl V. Jackson, *Why Dating Site OKCupid Performed Secret Experiments on its Users*, CHI. TRIB. (Apr. 22, 2015, 3:05 PM), http://www.chicagotribune.com/bluesky/originals/chi-christian-rudder-okcupid-bsi-20150422-story.html.

160. *Breaking Down Apple's iPhone Fight with the U.S. Government*, N.Y. TIMES (Mar. 21, 2016), http://www.nytimes.com/interactive/2016/03/03/technology/apple-iphone-fbi-fight-explained.html (explaining the legal battle between Apple and the FBI).

161. Matthew Rosenberg et al., *How Trump Consultants Exploited the Facebook Data of Millions*, N.Y. TIMES, Mar. 17, 2018 at A1, https://www.nytimes.com/2018/03/17/us/politics/cambridge-analytica-trump-campaign.html.

system designers to look outside the efficiency toolkit with increasing frequency.

II. CHARACTERIZING DESIRABLE INEFFICIENCY

In Part I, we singled out several systems that use inefficiency in ways counter to the tendencies of computer science. In this Part, we systematically organize these examples into a rigorous characterization of desirable inefficiency. By understanding the goals that these systems achieve through inefficiency, we can understand the choices available to system designers and regulators who hope to wield inefficiency in a bid to tame code. Amidst this definitional and theoretical work, we introduce even more examples of desirable inefficiency, particularly a large and exciting family of systems that utilize so-called proof of work algorithms, most notably cryptocurrencies like Bitcoin and Ethereum.

Finally, we connect our formal definitions and examples to the human values that have served as the central motivation for these developments. Systems of desirable inefficiency will again be shown to promote and protect values such as fairness, autonomy, and legitimacy.

A. *Defining Desirable Inefficiency*

We define desirable inefficiency to describe a digital system that solves a given problem in an inefficient manner when compared to a much more efficient solution to a closely related problem. It is crucial to our analysis to define the necessary characteristics of the paired problems that fulfill this definition, particularly to make this concept comprehensible to economists, who do not use the term *inefficiency* in the same way.

1. Core Definitions

We begin our analysis by defining what we mean by *efficiency*, which we define in computer scientific rather than economic terms:

> The efficiency of a system is the extent to which it minimizes the consumption of time, energy, space, or cost in satisfying a specification of correctness for a given problem.

First and foremost, efficient systems should be correct—they need to actually achieve the purpose they were tasked with addressing.[162] For example, a system must "put numbers in sorted order," "find the fastest driving directions to Boulder, Colorado," or "process all valid credit card transactions."

An efficient system must be more than just correct, however; it must

162. Zohar Manna & Amir Pnueli, *Axiomatic Approach to Total Correctness of Programs*, 3 ACTA INFORMATICA 243, 243 (1974).

also minimize resource consumption in the process of solving its problem. A file must download as quickly as possible or a car must consume as little gas as possible. Although we specifically name time, energy, space, and cost in our definition, since they are among the most common aspects of a computational system to be optimized, many other qualities (throughput, latency, transistor density, etc.) could appear in their place. A system could even optimize several of these quantities at once, balancing the running time of a program with the cost of developing it.

An *inefficient system* is defined as the converse of an efficient system:

> An inefficient system fails to minimize the consumption of time, energy, space, or cost in satisfying a specification of correctness for a given problem.

In other words, every inefficient system could be replaced by a more efficient alternative that meets the same standard for correctness but needs less time, space, energy, or cost to do so. For example, it would be inefficient in terms of both energy and cost to travel across the country in a gas-guzzling SUV if a smaller hybrid were available. It would be inefficient in terms of time if there were the option to fly. In terms of cargo-carrying capacity, however, both flying and the smaller hybrid would be less efficient than driving an SUV.

As outlined in Part I, according to conventional computer science, inefficient systems are undesirable. Theoreticians toil to develop new algorithms in a never-ending search for slightly smaller running times. Silicon Valley giants pour vast resources into stamping out profit-draining sources of inefficiency.

What circumstances, then, make inefficiency desirable? The hallmark of desirable inefficiency is in the way we define the problem to be solved. A desirably inefficient solution always solves at least two distinct problems, a *basic problem* and an *enhanced problem*. Building on the earlier two definitions, desirable inefficiency can be defined as follows:

> A desirably inefficient solution fails to minimize the consumption of time, energy, or space in satisfying a specification of correctness for a given basic problem in order to address a different, related enhanced problem.

Much rests on defining further the attributes of problems that qualify as basic and enhanced pairs. Before we formalize this part of the definition, which we think addresses some objections we have received from economists, consider some examples. The following table places the examples presented in Part I and other examples described in this Part within the framework of this definition.

System	Basic Problem	Enhanced Problem	Resource Expended	More Efficient Alternative
Smartphone password lockout mechanism	A user can access the data and apps on a smartphone	Thieves (and the FBI) cannot access the data, but forgetful users can	Time	A user could access the data faster after mistyping a password if there were no delays
IEX fiber loop	Information is exchanged between IEX and traders	Unfair trading is prohibited	Time / Latency	Traders could get prices and execute trades faster without the fiber loop
Proof of work for spam	Emails are sent	Spam is blocked	Energy and computation	Emails could be sent without solving a computational puzzle
Time-release crypto	Encrypted data is unlocked	Encrypted data is easy to unlock in the future	Time via computation	Data could require less computation to unlock
Bitcoin's blockchain	Transactions are validated	Transactions are validated fairly (integrity of currency is preserved)	Energy and computation	Proof of work could be easier or removed entirely
Websites with captchas	A user accesses a particular web page	Only human (and not robot) users can access the web page	Time via mental exertion	The captcha could be removed entirely, making web browsing much faster
Internet tromboning	Route packets from point A to point B	Avoid surveillance by the NSA	Time / Latency	A faster route is available

Table 1: Examples of Desirable Inefficiency

2. Basic and Enhanced Problems

The division of problems into basic and enhanced stands as the cornerstone of our definition. Desirable inefficiency can be identified

only relative to an alternative solution, one which addresses a similar but non-identical problem, which we call the basic problem.

At this point, the economist will likely object that there is nothing inefficient about our examples. What we call desirable inefficiency is simply the efficient solution to the enhanced problem.[163] Every one of our examples finds computer scientists and engineers designing an efficient (if clever and counterintuitive) solution to the enhanced problem, one which merely relies on a different definition of the welfare function than the basic problem, so what role is the basic problem serving at all?[164] The fact that a related basic problem exists might be interesting but does not merit misusing the label *inefficiency*, or so the economist will protest.

We think our fellow computer scientists will understand at least the intuition of our word choice, as well as the need to pair enhanced and basic problems. What motivates us is the cognitive dissonance we ourselves experienced when we were first confronted by algorithms crafted to take the inefficient path, for example the first time we considered proof-of-work, time-lock cryptography, or the IEX fiber loop. We argue that the root of this dissonance is the understanding that there is another, related but more basic, problem that can be (or has been) solved, which is measurably more efficient than the solution sitting before us. This ability to identify another, more efficient solution tugs on the primeval, lizard part of our computer-scientist brains, the neural pathways that were forged in our first courses in algorithmic complexity. Good programmers stamp out inefficient code. Why have these programmers—all talented systems designers—abandoned this orthodoxy?

We try to formalize the computer scientist's intuition, but to do so, we must precisely constrain the category of problems that may by definition properly be classified as basic. Without these constraints, the view of the economist would prevail. Every potential enhanced problem could be paired with another related, simpler, purportedly basic problem, which would support the economist's view that desirable inefficiency is simply efficiency redefined.

For example, reconsider the SUV. The SUV is inefficient in fuel consumption but efficient when it comes to cargo-carrying capacity. One might wonder, then, whether the SUV is a desirably inefficient vehicle under our definition, solving the "cargo carrying" enhanced problem while sacrificing the "fuel efficiency" basic problem. We do not think this meets our definition. A simple tradeoff is not the same thing as desirable inefficiency. Solutions that merely sacrifice one measure of

163. RICHARD A. POSNER, ECONOMIC ANALYSIS OF LAW 10 (2d ed. 1977).

164. *Id.*

efficiency for another are arguably inefficient, but they are not desirably inefficient.

To take a more computational example, consider the paradigmatic family of algorithms used for studying efficiency: sorting algorithms.[165] The task is to take a long list of words and return them in alphabetical order.[166] Generations of computer scientists have scrutinized various algorithms for accomplishing this task, and it serves as the primary example used to teach students about efficiency and complexity.[167]

Sorting algorithms can be efficient in terms of time—they can minimize the number of seconds it requires to sort a long list—or space—they can minimize the amount of copies of items in the list that need to be temporarily stored in the computer's memory—or both.[168] But an algorithm that tries to achieve efficiency of both time and space (say, heapsort) might seem inefficient as being slightly slower than an algorithm that does not economize on memory (say, quicksort).[169]

We would not conclude that heapsort is desirably inefficient. "Sort while conserving space and time" is not an enhanced version of "sort while conserving only time." It is merely a tradeoff, akin to the SUV's failure to economize on fuel in order to provide for more cargo capacity.

The key to distinguishing desirable inefficiency from mere efficiency is to add a single mandatory constraint to the way we define the enhanced and basic problems: enhanced problems require human judgment, values, or discretion in the definition of success or failure, while basic problems are reducible to mechanistic metrics of success and failure. Success for the delivery of email is defined mechanistically: did the recipient mail server receive all of the bits comprising the email message without modification within a set threshold of time?[170] In contrast, success for a spam blocker, the subject of a desirably inefficient solution described in the next sub-part,[171] requires a human's judgment: the software succeeds if it blocks spam but allows through non-spam, however each user may define those terms.[172] In fact, because spam blockers are never perfect,

---

165. CORMEN ET AL., *supra* note 41, at 148 ("Many computer scientists consider sorting to be the most fundamental problem in the study of algorithms.").

166. *Id.* at 206.

167. *See id.* at 147–50 (providing a detailed investigation of the algorithmic complexity of various sorting algorithms in a classic text).

168. *Id.* at 148 (discussing running time and ability to sort "in place" of various sorting algorithms).

169. *Compare id.* at 151–69 (analyzing heapsort), *with id.* at 170–90 (analyzing quicksort).

170. RFC 2821: Simple Mail Transfer Protocol (Apr. 2001), https://www.ietf.org/rfc/rfc2821.txt (defining Internet standard for the delivery of email).

171. *Infra* Section II.B.

172. Mehran Sahami et al., A Bayesian Approach to Filtering Junk E-mail 55 (1998), http://erichorvitz.com/ftp/junkfilter.pdf.

the definition of spam blocking requires a second layer of human judgment: a successful spam blocker blocks enough spam while letting through enough non-spam.[173]

The fact that enhanced problems require human judgment, values, or discretion while their corresponding basic problems do not, gives rise to two other corollary features often characteristic, but not necessary, of desirable inefficiency: (1) basic problems lend themselves to quantifiable definitions of correctness, while enhanced problems do not; and (2) basic problems tend to have a long track record of being solved by computer scientists while attempts to solve enhanced problems tend to be newer. Consider each of these in turn.

First, a lack of quantifiability tends to flow from the defining characteristic of human judgment. It is the role of the human that tends to make the enhanced problem so difficult to quantify. Human judgment is unpredictable and inconsistent. Often, success or failure is in the eye of the beholder. What is unwelcome spam for one user may be considered desirable marketing email to another.[174]

Second, basic problems tend to come with historical track records of computer scientists and engineers trying to solve them. We have been trying to transmit email for many more years than we have been designing systems to block spam.[175]

This second characteristic might flow from the changing role of technology in society. Before the rise of the Internet, software tended not to be deeply entwined with our daily lives and interactions. In general, this meant that software designers were primarily addressing industrial goals, the kind that lent themselves to mechanistic problem definitions. In the first decade or two of the commercial Internet, software became closely associated with human communication, which gave rise more often to problems that were difficult to reduce to a number, such as spam.[176] Today, software permeates systems that directly drive every human interaction. Machine learning systems match people for romantic encounters.[177] Internet of Things sensors collect information about our

173. *Id.* at 56.

174. *Id.* at 60.

175. Computer scientists did not attempt to build systems to recognize spam until junk email emerged as an important problem shortly after the commercialization of the Internet in the mid-1990's. BRIAN MCWILLIAMS, SPAM KINGS xi–xv (1st ed. 2005) (describing history of spam on the Internet). Email, on the other hand, was invented decades earlier. HAFNER & LYON, *supra* note 138, at 189.

176. HAFNER & LYON, *supra* note 138, at 214.

177. *The Online Dating Engine That Assesses Your Taste in the Opposite Sex (And Whether They Find You Attractive)*, MIT TECH. REV. (Nov. 18, 2013), https://www.technologyreview.com/s/521826/the-online-dating-engine-that-assesses-your-taste-in-the-opposite-sex-and-whether-they/.

most intimate and sensitive activities.[178] Artificial intelligence systems increasingly choose society's winners and losers and implicate civil rights.[179]

Taking the defining characteristic and the two corollary characteristics into consideration, we can now explain why the SUV and heapsort, while inefficient, are not desirably inefficient. These are both solutions to mechanistically defined problems that do not directly implicate human values or human judgment. Calculating cargo-carrying capacity and measuring the memory used by an algorithm are not the kind of human-driven problems that qualify as enhanced in our definitions.

3. Human Values Reduced to Quantifiable Solutions

Only the primary criterion—human judgment, values, or discretion in the definition of the problem to be solved—is strictly necessary. Although a lack of quantifiability is commonly associated with enhanced problems, it is not necessary. Sometimes enhanced problems lend themselves to quantifiable definitions of correctness. Take, for example, the final entry in Table One, Internet tromboning.[180] Due to the vagaries of Internet routing and the complicated economic relationships between companies that operate routers, traffic will occasionally follow extraordinarily circuitous paths around the globe, a phenomenon known as tromboning.[181] For example, it is possible that a message sent from two countries in Northern Africa could pass through the United States.[182]

Although tromboning is inefficient from a speed perspective (the basic problem), it can be desirable when the goal is to ensure that web traffic avoids countries with particular objectionable policies (the enhanced problem).[183] For example, a user in France concerned about NSA surveillance might prefer that her traffic take a slow, convoluted path outside of the United States rather than a faster, direct route that passes through New York City.[184] A system implemented to satisfy this preference would classify as desirably inefficient.

---

178. Meg Leta Jones, *Privacy Without Screens & the Internet of Other People's Things*, 51 IDAHO L. REV. 639, 641 (2015); Scott R. Peppet, *Regulating the Internet of Things: First Steps Toward Managing Discrimination, Privacy, Security, and Consent*, 93 TEX. L. REV. 85, 129–32 (2014).

179. Barocas & Selbst, *supra* note 5, at 673; Citron & Pasquale, *supra* note 25, at 3.

180. ANNE EDMUNDSON ET AL., CHARACTERIZING AND AVOIDING ROUTING DETOURS THROUGH SURVEILLANCE STATES 2 (2016), http://www.cs.princeton.edu/~annee/pdf/surveillance.pdf.

181. *Id.*

182. *Id.*

183. *Id.*

184. *Id.*

This proposed solution to the enhanced problem is readily quantifiable: has the communication passed through a server in the United States? Even though this is the kind of precisely defined problem that can be stated mathematically and confirmed programmatically—thereby not meeting the first corollary criterion—this is still desirable inefficiency. Although quantifiable and precise, this specification of correctness is infused with human values: it reflects both a fear of surveillance and a judgment about the likelihood that a country will engage in aggressive surveillance, even though it was reducible to a precise specification, one a machine could measure.[185]

In other words, modern users of the Internet might embrace tromboning because it represents a translation of a values-oriented goal—avoid unchecked surveillance—into a concrete requirement. It is a heuristic, a rule of thumb that approximately but concretely protects, preserves, or gives space for a value.

Time-lock cryptography provides another example of quantifiable desirable inefficiency.[186] A public figure might order that her digital records be sealed upon her death using a time-lock cryptographic system configured to become accessible in forty years.[187] This establishes a precise, quantifiable goal, but it is merely a heuristic reflecting an underlying human judgment, the figure's attempt to balance a public's desire to know more about her life against perhaps a desire to protect the privacy of her heirs or to let the distance of time place her records in more of a historical context.

4. Two Categories Not Covered

It will help sharpen the outer boundaries of our definition to give a few examples of systems we do not consider examples of desirable inefficiency, even though they share some features with the category.

First, we exclude non-digital systems. We think the trend we recount in Part I is intrinsically connected to digital hardware and software. We understand, however, that many of the concepts we discuss apply to pre-digital and non-digital examples. We tend to treat these examples as important analogs rather than members of the class we are defining. In future work, we may broaden our definition to connect our insights to examples beyond computer systems.

---

185. Andrew Clement & Jonathan A. Obar, *Canadian Internet "Boomerang" Traffic and Mass NSA Surveillance: Responding to Privacy and Network Sovereignty Challenges*, *in* LAW, PRIVACY AND SURVEILLANCE IN CANADA IN THE POST-SNOWDEN ERA 13, 30–35 (Michael Geist ed., 2015) (proposing ways to avoid Internet routing paths through the United States for communications between two parties in Canada).

186. Rivest et al., *supra* note 94.

187. *Id.* at 1–2.

Second, we exclude systems that do no more than slow down the operation of a computer to match the speed of human processing systems. For example, a large class of problems we exclude are those related to the fact that human sensory and cognitive abilities cannot match the billions of calculations per second that a modern processor achieves. Just as engineers build printer and USB device drivers to allow a computer to communicate with an idiosyncratic range of typically slower add-on hardware,[188] some inefficient (but not desirably inefficient) solutions can be understood as "drivers" for managing particularly unpredictable and fiddly "human devices" with which many technical systems must coexist. Animation rates and input systems must slow and simplify to the speed and capabilities of human cognition, even if the computer itself can achieve far faster operation. A computer might be able to render a virtual car race in the blink of an eye, but a human player will have considerably more fun if the race takes place slowly enough for her to participate.[189] This outcome may be inefficient, but we do not consider it to fall under the umbrella of desirable inefficiency.

This leaves us in an admittedly nuanced stance with respect to systems that slow down computer processing. IEX's delay qualifies as desirable inefficiency, while the slowness of the video game racecar does not. Once again, we point to the criterion and its two corollaries we use to identify basic problems. IEX's delayed stock exchange trades bear all three hallmarks of basic problems: they help create the hard-to-quantify human value of fairness, and they relate to a problem we have long tried to solve without the delay.[190] In contrast, the slowness of the video game racecar does not satisfy any of these three criteria.

This also distinguishes desirable inefficiency from Julie Cohen's work. Cohen's examples tend to focus on protecting the development of the individual (and to a lesser extent the group) from disruption by powerful information entities including governments and platforms in order to give "breathing room" for people to engage in the "play of everyday practice."[191] With this focus, Cohen might include examples of inefficient system design intended merely to put a computer's processing speed in sync with human capabilities.

B. *Desirable Inefficiency and Human Values*

Let us return to the list of values presented in Part I, which we argue have spurred many of the desirably inefficient systems that have been developed: autonomy, legitimacy, fairness, trust, liberty, and integrity.[192]

188. JONATHAN CORBET ET AL., LINUX DEVICE DRIVERS 38–39 (3d ed. 2005).
189. JASON GREGORY, GAME ENGINE ARCHITECTURE 310–16 (2d ed. 2014).
190. LEWIS, *supra* note 11.
191. COHEN, *supra* note 4, at 22.
192. *Supra* Section I.D.

How have designers implemented desirable inefficiency as a way to inject human values such as these into their systems? What follows is a compendium of felt needs that have, to date, given rise to desirably inefficient systems.

1. A Rich Array of Values

Before diving into a few particularly dominant values, we take this opportunity to survey the diversity of values that our example systems achieve through desirable inefficiency. Several systems use inefficiency to support autonomy. Internet tromboning does so explicitly, routing web traffic away from countries that have reputations for mass surveillance.[193] Dwork and Naor's approach to spam advances autonomy more subtly, protecting an individual's cognitive ability to conveniently sift through relevant correspondence.[194] This value could alternatively be framed as liberty, the ability to maintain one's space for communication unimpeded by mountains of unsolicited advertisements.

Time-release cryptography, which makes it possible to archive information for long periods of time, makes possible eventual transparency and historical continuity while preserving the dignity of those who seek to temporarily shield their private documents from the public eye.[195] These sorts of time capsules tend to also facilitate an element of community, the shared gratification of discovering previously concealed information and receiving a message sent through time from the past. In many cases, captchas make forming communities online possible by filtering out robots that would otherwise fill discussion forums with spam or even overload websites with fake traffic.[196]

Several systems also protect various forms of security. In situations where the risk of traffic falling into the wrong hands is sufficiently high, use of techniques like Internet tromboning could be matters of life and death.[197] Similarly, a smartphone lockout mechanism could keep incriminating information out of the hands of a repressive regime.

We recognize that this accounting is incomplete. Whether these are the correct list of values or the only values connected to those digital systems remains open to debate. Still, this exercise demonstrates the rich array of values for which desirable efficiency can create space. In the

193. EDMUNDSON ET AL., *supra* note 180.

194. Dwork & Naor, *supra* note 21.

195. Rivest et al., *supra* note 94.

196. *See* Brad Yale, *The CAPTCHA: A History, a Problem, Possible Solutions*, INFORMIT (Sept. 10, 2014), http://www.informit.com/blogs/blog.aspx?uk=Why-Are-CAPTCHAs-So-Awful.

197. *See* EDMUNDSON, *supra* note 180 (defining "tromboning" as the situation "whereby an Internet path starts and ends in a country, yet transits an intermediate country").

following Sections, we explore a few of the most prominent values in depth.

2. Legitimacy

Many of the systems we have discussed use inefficiency to evaluate the legitimacy of their participants. The most concrete example of a system with this enhanced goal is a captcha.[198] Captchas exist to distinguish human users from robots by presenting small puzzles that remain difficult for computers to solve.[199] The mere act of being able to decipher smeared text or identify all images that contain cats (as opposed to dogs and hamsters) is evidence that a web-surfer is a legitimate human being.[200]

The smartphone lockout mechanism tests for a different form of legitimacy—whether a user actually knows the phone's password.[201] If a user has simply pressed the wrong button or mixed up a couple of digits, it is likely that he will arrive at the correct password in a few tries, enduring, at worst, a few seconds of wasted time. An intruder with little or no knowledge of the password, however, will have to begin systematically guessing, a strategy rendered virtually impossible by the lockout mechanism's time delays.[202]

On a dramatically larger scale, Bitcoin applies a similar approach to its network of thousands of transaction-processing nodes.[203] As we discussed above, one of the purposes of Bitcoin's proof of work is to ensure that transactions are approved via a loose form of voting.[204] When a new block containing the preceding ten minutes of transactions is added to the end of the blockchain, other nodes express their belief that the block is valid by building on it in their future mining efforts.[205] Rather than explicitly holding an election, which would be easy to rig by creating armies of fake nodes, Bitcoin nodes vote with their computational effort,

198. *See* Yale, *supra* note 196.

199. *See id.*

200. *See* Vinay Shet, *Are You a Robot? Introducing "No CAPTCHA reCAPTCHA"*, GOOGLE SECURITY BLOG (Dec. 3, 2014), https://security.googleblog.com/2014/12/are-you-robot-introducing-no-captcha.html.

201. *See* APPLE, IOS SECURITY: IOS 11, at 15 (Jan. 2018), https://www.apple.com/business/docs/iOS_Security_Guide.pdf.

202. *See* William Enck & Adwait Nadkarni, *What if the FBI Tried to Crack an Android Phone? We Attacked One to Find Out*, CONVERSATION (Apr. 8, 2016, 2:23 PM), http://theconversation.com/what-if-the-fbi-tried-to-crack-an-android-phone-we-attacked-one-to-find-out-56556.

203. *See* NARAYANAN ET AL., *supra* note 102, at 33.

204. *See id.*

205. *See id.*

a substance that is impossible to forge.[206] In this fashion, Bitcoin uses proof of work to ensure that only nodes with legitimate computational capability are able to vote to approve transactions.[207]

3. Fairness, Trust, and Integrity

Many of the values discussed in the preceding Sections are not ends in themselves. Rather, they create the conditions necessary to achieve larger values that inspire confidence among users. Here, we discuss three of the most important of these values: fairness, trust, and integrity.[208]

Fairness often entails establishing a level playing field among participants.[209] By increasing the minimum time necessary to communicate stock transactions, IEX purports to ban a perceived unfair type of behavior: exploiting network architecture as a way to step inside another trader's transaction.[210] Among its many purposes, Bitcoin's proof of work is structured to prevent double spending and transaction cancelling that would give certain users unfair financial power.[211] Even captchas enforce a form of fairness where human users cannot be beaten out by robots with faster reaction times.[212]

Often, fairness serves as an intermediate step toward a still-larger value: trust.[213] In many of our examples, it is trust that engineers ultimately seek when they turn to desirable inefficiency. IEX does not require fairness for fairness's sake. Likewise, participants in Bitcoin do not aim for legitimacy or fairness as an intrinsic end. Instead, these values serve to inspire confidence among users—to buttress the perceived integrity of the overall platform. Both of these examples involve the exchange of financial products, putting users at substantial risk if the

206. *See id.*

207. As a consequence, Bitcoin's "voting rights" are distributed according to computational ability. A node with a faster CPU will receive proportionally greater electoral influence. This influence, in turn, will mirror the financial means that allowed the computer's owner to purchase this faster hardware. *See id.*

208. Our work connects to the growing discipline of Fairness, Accountability, and Transparency. FAT*, *supra* note 141.

209. *Cf.* Lee A. Fennell & Richard H. McAdams, *Fairness in Law and Economics: Introduction* 1, 1 (Coase-Sandor Inst. For Law And Econ., Working Paper No. 704, 2014), https://chicagounbound.uchicago.edu/cgi/viewcontent.cgi?article=2381&context=law_and_economics ("[F]airness preference[]. . . [means that] individuals will, at a cost to themselves, choose to distribute wealth to others, reward those who contribute to public goods, or punish those who free-ride.").

210. *See* Levine, *supra* note 70.

211. *See* NARAYANAN ET AL., *supra* note 102, at 38.

212. *See* Yale, *supra* note 196.

213. *Cf.* Neil Richards & Woodrow Hartzog, *Taking Trust Seriously in Privacy Law*, 19 STAN. TECH. L. REV. 431, 435 (2016) (arguing that "[t]trust in information relationships is necessary for the digital economy not just to function, but to flourish").

systems do not function as advertised. By inspiring trust in the integrity of their systems, the creators of IEX and Bitcoin help to ensure the success of these undertakings.

Many other systems we have described fit into the admittedly inclusive expanse of fairness, trust, and integrity. Dwork and Naor's solution to spam aims to level the playing field between humans and spammers, inspiring trust in the provenance and value of messages that are received.[214] Many websites rely on captchas as an essential tool for inspiring trust in e-commerce and gaming platforms that could otherwise be overrun by robots with lightning-quick reaction times.[215] Time-release crypto enables public figures to trust that their documents will remain safely out of reach for decades to come.[216]

III. DESIGN PATTERNS OF DESIRABLE INEFFICIENCY

Desirably inefficient systems tend to possess four distinctive attributes: filtering/separating, adversarial countermeasures, tunability, and decentralization. Although not every system contains these attributes, and some may contain none, these qualities are common design patterns found in many of the desirably inefficient systems we have studied.[217] Most of our examples operate through filtering mechanisms, segregating users or actions into "good" and "bad" categories. In order to remain effective under adversarial circumstances, a system must be able to verify that all users endure the requisite amount of inefficiency. In a select few cases, the inefficiency is built into the very hardware architecture of the system. Inefficiency itself is plastic and tunable—it can be ratcheted up or down and even customized for each user. Finally, desirable inefficiency often depends on (and on the flip side, can help create the necessary preconditions for) decentralized systems.

Designers should treat these four patterns as design templates for creating new desirably inefficient systems. Scholars and regulators need to understand how these patterns can help desirably inefficient systems advance law and policy goals.

A. *Filtering and Separating*

Several of our example applications of inefficiency achieve their goals by distinguishing between different classes of users. Dwork and Naor's approach to spam, for example, aims to separate spammers from non-

214. *See* Dwork & Naor, *supra* note 21, at 139.

215. *Cf.* Yale, *supra* note 196 (discussing how CAPTCHA technology can be used to serve a deterrence purpose).

216. *See* Rivest et al., *supra* note 94.

217. *See, e.g.*, ALEXANDER ET AL., *supra* note 30, at 3, 13, 19, 617.

spammers.[218] It does so by magnifying the disproportionate number of messages that each group sends into computational work that is debilitating for spammers but negligible for anyone else.[219] In the process of doing so, this inefficiency even goes a step further: it punishes one of the groups it identifies, making the average spammer's email patterns so expensive that sending spam at all is no longer a feasible business model.[220] In this fashion, Dwork and Naor's proposal filters spammers out from other, well-behaved email users, distinguishing and economically ruining them.

Many of the examples of desirable inefficiency from Part I seek to discriminate between flows of inputs, separating the "good" from the "bad" through their behavior. Captchas do so explicitly, posing puzzles that only humans can solve in order to distinguish them from digital automatons.[221] Similarly, Dwork and Naor's proposal to limit spam introduces a "pay to play" barrier in the form of proof of work that distinguishes legitimate users, who would barely notice the obstacle at all, from spammers, who would no longer be able to afford the cost of sending millions of emails.[222] Smartphone password lock-out mechanisms are designed to punish an adversary trying to break into a phone far more harshly than a clumsy user, who might only mistype a password once or twice.[223] No doubt, somebody inside Apple studied human behavior and concluded that legitimate users tend to require up to four guesses; unauthorized password guessers often need more than six. In the liminal space between three and six, Apple finds especially clueless users and especially savvy (or lucky) imposters, and they tune their timings accordingly.

The separating done by desirable inefficiency can never be perfect. There will be false positives and negatives. Some legitimate iPhone users will mistype their passwords seven times. Some password guessers will get lucky and guess a password on the fifth try. The best we can do is try to find the sweet spot and monitor the results. But unlike other policy levers, if we misjudge our first attempt, we can retune the policy dial, trying to tamp down the false positives and false negatives.

B. *Adversarial Countermeasures and Hardware Solutions*

The world resists desirable inefficiency. Many desirably inefficient systems operate in an adversarial setting: a user or system designer is

218. *Id.*
219. Dwork & Naor, *supra* note 21, at 4.
220. *Id.*
221. Yale, *supra* note 196.
222. Dwork & Naor, *supra* note 21, at 4.
223. APPLE, *supra* note 201, at 15.

opposed by an adversary whose goals clash with her own.[224] The spammer wants to send more messages; the phone thief or FBI agent wants to guess passwords more rapidly; and the high-frequency trader wants to front-run a trade.

Given the adversarial nature of desirable inefficiency, each of these systems must ensure that the inefficiency actually takes effect—that there are no loopholes the adversaries can exploit to operate efficiently.[225] Usually, a designer competes in this adversarial contest through software, incorporating code that anticipates and counteracts the adversaries' moves, engaging in a game of chess. Captchas, proof of work methods, and Internet tromboning all rely on software routines to impose the inefficiency and to guard against workarounds adversaries might try to deploy.[226]

Sometimes designers choose to use hardware rather than software to make desirable inefficiency more difficult for the adversary to subvert. The smartphone lockout mechanism is a product of the device's software operating system but it is made difficult to avoid thanks to the clever use of hardware.[227] For example, a key ingredient that makes an iPhone difficult to access without the passcode is a dedicated chip known as the Secure Enclave.[228]

The surprising virtue of selecting hardware over software is best seen in the use of hardware by IEX. IEX creates inefficiency by forcing all communication with the exchange to travel along thirty-eight miles of fiber-optic cable.[229] Critically, this configuration does not need a software-based "checking" step to ensure that traders have endured the appropriate time delay. The mere fact that a message reached IEX at all means that it travelled through the fiber loop and, thereby, experienced inefficiency.[230] In sum, IEX's fiber loop is a self-enforcing mechanism for inflicting inefficiency, a design choice from which we will draw inspiration for policy in Part IV.

Why did IEX choose to implement this delay using hardware rather than software? There are many benefits that flow from this decision, some quite counter-intuitive, and they deserve an extended discussion. Recall that the primary benefit the thirty-eight-mile fiber loop affords is a 350-

224. *See* Paul Ohm, *Broken Promises of Privacy: Responding to the Surprising Failure of Anonymization*, 57 UCLA L. REV. 1701, 1723–24 (2010) (discussing the role of the adversary).

225. *Cf.* ADAM SHOSTACK, THREAT MODELING: DESIGNING FOR SECURITY 40–41 (2014) (describing the related computer security notion of adversary threat modeling).

226. *See* Yale, *supra* note 196 (discussing captchas); Dwork & Naor, *supra* note 21, at 142 (proposing proof of work for spam).

227. *See* APPLE, *supra* note 201, at 12.

228. *Id.* at 7.

229. *See* LEWIS, *supra* note 11, at 177–78.

230. *Id.*

microsecond delay on all communications into and out of the trading platform.[231] IEX's engineers could have created precisely the same amount of delay with a simple software routine, and they could have done so at considerably less expense.[232]

It is a widely held maxim in computer science that any hardware solution can be solved using software instead.[233] Although solving a problem using hardware or software may be equivalent from a strictly algorithmic point of view, the two approaches vary in important ways. Most often, people focus on the advantages of software over hardware.[234] Software is intrinsically more flexible than hardware.[235] Software systems are orders of magnitude less expensive to prototype than equivalent hardware.[236] Engineers have created development environments for software that allow a continuous cycle of redesign and reinvention—many forms of software exist simply to generate other software.[237] Jonathan Zittrain celebrates the "generativity" of software, contrasting it the relatively less flexible hardware platforms.[238]

The relative inflexibility of hardware is probably what inspired IEX to choose it. A thirty-eight mile loop is a promise of persistence spun out of strands of glass. IEX cannot simply reengineer away the 350-microsecond delay from a keyboard. It has bound itself to the mast by reifying its design decision. Customers can be sure that the delay is not something that can be changed on a whim.

It promotes transparency and thus external verifiability. The fiber loop is stored inside a physical box inside a machine room in New Jersey.[239] An experienced external observer could in a few hours trace the cabling in that room to verify that the box sat on the critical path between the outside world and IEX. IEX could, of course, engage in complex

231. *See* LEWIS, *supra* note 12, at 177–78.

232. Causing a computer program to delay a set amount of time is a common programming task. Many programming languages provide a primitive function—often called sleep() or delay()—to provide this capability in a single function call. *See, e.g.*, Eric S. Raymond, *Time, Clock, and Calendar Programming in C*, CATB (Sept. 9, 2017), http://www.catb.org/esr/time-programming/. Other languages provide delay as a lightweight library enhancement to the language. *E.g.*, *Time – Time Access and Conversions*, PYTHON SOFTWARE FOUND. (Feb. 14, 2009), https://docs.python.org/3.0/library/time.html (describing time.sleep() function).

233. LINDA NULL & JULIA LOBUR, THE ESSENTIALS OF COMPUTER ORGANIZATION AND ARCHITECTURE 3 (3d ed. 2006) (discussing the "principle of equivalence of hardware and software").

234. *E.g.*, ZITTRAIN, *supra* note 33, at 14.

235. Ohm & Reid, *supra* note 32, at 1675.

236. *See* Paul Ohm, *We Couldn't Kill the Internet if We Tried*, 130 HARV. L. REV. F. 79, 81 (2016).

237. *See id.*

238. ZITTRAIN, *supra* note 33, at 58.

239. *See* LEWIS, *supra* note 11, at 178.

subterfuge to hide the cable that bypassed this box in a way that might evade detection, but this would be considerably more difficult to construct and considerably more likely to be detected than the equivalent bypass of a software-only delay system, which would be almost trivial to mask with very little fear of detection.

At a time when many observers have begun to focus on the rise of the "black box society,"[240] it is an interesting spin on the metaphor to contemplate how a physical box might help us deal with the problems of black boxes. Electrical engineers and computer scientists overcome their doubts about black boxes by rigorously sending varying inputs and observing their outputs.[241] We can do the same with IEX's box. We can generate trade requests as inputs and time the latency before they emerge from the box's output, verifying the 350-microsecond delay.

Perhaps most importantly for regulators and policymakers, the IEX box isolates a critical point in the path of its architecture that can facilitate new forms of oversight and intervention.[242] There is a modularity to this design that policymakers can leverage. Imagine if the SEC obligated companies committed to rooting out the ill-effects of high-frequency trading to place their IEX fiber loop boxes within a machine room under the SEC's control. This would afford the regulator a physical place to undergo black box testing and it would make it difficult for a company to use a jumper to short-circuit the box or to swap in a new box.

In the same way, the box delineates a boundary line, a physical place where regulators can conduct surveillance and impose new obligations. Like the border around a physical territory, the borders of this box can see the rise of the digital equivalent of border agents and customs rules.

Finally, for all of these reasons—persistence, external verifiability, transparency, and regulability—the box becomes a symbol, which helps engender trust and goodwill. IEX can use this powerful symbol as part of its brand-building exercises in ways that might be less effective if tied to our proprietary software platform.[243]

C. *Tunability*

Desirably inefficient systems provide a tuning mechanism. In traditional systems for which efficiency is the goal, more is almost always better—a sorting algorithm that runs faster or a network with higher

240. PASQUALE, *supra* note 25, at 568.

241. TIM RILEY & ADAM GOUCHER, BEAUTIFUL TESTING 282 (2009).

242. *See* Kroll et al., *supra* note 137, at 636–37, 641 (discussing auditability of software).

243. *See* Victor Fleischer, *Brand New Deal: The Branding Effect of Corporate Deal Structures*, 104 MICH. L. REV. 1581, 1600 (2006) (describing terms of corporate deals selected to enhance a branding message).

bandwidth and lower latency is better.[244] The primary goal of programmers and engineers solving problems of this type is to find and stamp out lingering sources of inefficiency.[245]

The same is almost never true for desirable inefficiency. Once programmers decide to inject inefficiency into a system, the key question becomes determining the proper amount. Often, the target is a difficult to pinpoint and even debatable sweet spot: a Goldilocksian ideal between too fast and too slow. Too little inefficiency would fail to advance the desired values, while too much would render the system unusable.

The creators of IEX likely chose to delay stock transactions by 350 microseconds because they calculated or discovered through trial-and-error that shorter delays kept the door open for predatory trading while longer delays unnecessarily stemmed the volume of trading.[246] It is probably arguable whether 300 microseconds or 400 microseconds would have served as a better tradeoff between these competing goals, but any delay could have been chosen by merely shortening or lengthening the fiber loop.[247] The *tunability* of their chosen source of inefficiency granted them the freedom to find the optimal middle ground.

Consider also iPhone's password-guessing-throttling system. Here is how Apple engineers have tuned this particular dial[248]:

Delays between passcode attempts	
Attempts	Delay Enforced
1-4	None
5	1 minute
6	5 minutes
7-8	15 minutes
9	1 hour

To further discourage brute-free passcode attacks, there are escalating time delays after the entry of an invalid passcode at the Lock screen. If Settings > Touch ID & Passcode > Erase Data is turned on, the device will automatically wipe after 10 consecutive incorrect attempts to enter the passcode. This setting is also available as an administrative policy through mobile device management (MDM) and Exchange ActiveSync, and can be set to a lower threshold.

On devices with an A7 or later A-series processor, the delays are enforced by the Secure Enclave. If the device is restarted during a timed delay, the delay is still enforced with the timer starting over for the current period.

Table 2: iPhone's Password-Guessing-Throttling System

244. *Supra* Section I.A.
245. *Id.*
246. LEWIS, *supra* note 12, at 176.
247. *Id.* at 178.
248. APPLE, *supra* note 201, at 15.

No upper-level seminar in algorithmic complexity or complicated proof can decide the exact timings in this chart. Instead, this table is entirely about human psychology and imperfection: when people forget their passwords, how many tries do they usually need to get it right?[249] How frustrating would they find it to be throttled for a particular number of seconds after N guesses? After N+1? The form of inefficiency that the designers of the iPhone chose—timing delays—grant them arbitrary flexibility to tune these lock-out durations in accordance with their chosen methodology.

At its core, tunability provides the ability to implement inefficiency along a spectrum, with many (perhaps infinite) gradations between the extremes. Nearly all of the sources of examples we have discussed thus far—from the exact difficulty of proof-of-work problems in Bitcoin or spam to length of the IEX cable—provide a faculty for fine-tuning the amount of efficiency they impose.

An outgrowth of desirable inefficiency's tunability is the way it can be used to shape a complex cost or penalty function. Inefficiency need not increase linearly; the cost of sending a second email within the same minute could be higher than that of sending the first. Take the smartphone lockout feature. The seventh or eighth guess requires a user to wait fifteen minutes each time.[250] The ninth requires a full hour of thumb-twiddling.[251] The delay progresses, not along a straight line, but in increasing step sizes.

The lesson is that tunability is about far more than just setting a single, global level of inefficiency. It provides the opportunity to craft a function of increasing penalties and to customize these settings for individual users. As we will find in Part IV, this flexibility is a key selling point for desirably inefficient regulatory interventions.

D. *Decentralization*

Many of these systems are structured to avoid any need for a centralized intermediary.[252] For Bitcoin, decentralization is a design constraint.[253] Proof of work and regimented inefficiency are a

249. *See* Bruce Schneier, *The Psychology of Password Generation*, SCHNEIER ON SECURITY (Mar. 2, 2006, 11:46 AM), https://www.schneier.com/blog/archives/2006/03/the_psychology.html.

250. APPLE, *supra* note 201.

251. *Id.*

252. Matt Blaze et al., *Decentralized Trust Management*, 1996 PROC'S SYMP. ON SECURITY AND PRIVACY 164, 164 (1996).

253. In many ways, Bitcoin can be seen as a political response to the financial crisis of 2008-2009. It is designed to eliminate any central entity and money is created at a predefined rate. The first Bitcoin block to be created has a headline from *The Times*: "Chancellor on brink of second

fundamental component of making this decentralization operate smoothly. In contrast, for Dwork and Naor's approach to spam, this decentralization is a pre-existing characteristic of the way email is designed.[254] As we mentioned previously, proof of work can be verified in a peer-to-peer manner between sender and recipient without involving a central authority, providing a way to filter out spam while respecting the workflow of email as it currently exists.

Time-release cryptography must ensure that information remains locked away for extended periods of time no matter what happens over the course of the intervening years or decades.[255] Although this information could simply be given to a trustworthy authority to keep in escrow,[256] doing so could be unpredictable given the length of the timescales at play. To ensure that this data remains durably protected, time-release cryptography therefore eschews a central authority for the assurances provided by the extraordinary inefficiency of cryptographic puzzles.

The decision to decentralize a system is often closely tied to the value that desirable inefficiency is designed to advance. Many argue that Bitcoin is decentralized out of the skepticism for banks that followed the 2008 financial crisis.[257] Email is decentralized in line with the principles underlying the Internet architecture.[258] In either case, it is clear that inefficiency is a tool that can enable systems that might otherwise require a central authority to instead become decentralized.

IV. DESIRABLE INEFFICIENCY AS REGULATION

Almost all of our examples of desirable inefficiency so far have come from the private sector. Market forces have incentivized private actors to devise desirably inefficient solutions that solve the same basic problems as preexisting solutions but in addition solve related, enhanced problems by injecting or creating hard-to-quantify human values such as legitimacy, fairness, or trust.

Desirable inefficiency can serve as a tool for public-sector solutions as well. Regulators should consider incentivizing or mandating that regulated entities create desirably inefficient solutions to advance policy goals important to the public. Sometimes, regulators should consider creating their own desirably inefficient solutions, perhaps requiring

---

bailout for banks." *Genesis Block*, BITCOIN WIKI (Nov. 30 2017, 3:09 PM), https://en.bitcoin.it/wiki/Genesis_block.

254. Dwork & Naor, *supra* note 21, at 4.

255. Rivest et al., *supra* note 94, at 1.

256. *Id.* at 5.

257. BITCOIN WIKI, *supra* note 253.

258. J.H. Saltzer et al., *End-To-End Arguments in System Design*, 2 ACM TRANSACTIONS ON COMPUTER SYSTEMS 277, 278 (1984).

actors to use them or placing them into competition with preexisting solutions. The government should be willing to integrate desirable inefficiency into the design of systems that it controls.

Desirable inefficiency will sometimes solve problems that have been impossible to solve through regulation thus far. More often, desirably inefficient solutions will solve problems in a more flexible, nuanced, or supple fashion. The benefits of architecture, tunability, and filtering/separating—discussed in the prior Part—make these solutions differ in important ways from traditional regulatory frameworks, such as taxes or command-and-control edicts, as well as build on the benefits of newer approaches such as nudges. Regulatory desirable inefficiency should be added to this list of tools.

A. *Case Study: A Server on Mars*

To demonstrate how desirable inefficiency could work as a tool for regulators in practice, we propose to solve various difficult problems in information privacy using a novel solution based in desirable inefficiency—what we term "a server on Mars." To address various information privacy problems, regulators might consider requiring a mandatory delay in the propagation of a signal through a system or across a network. For privacy problems, the scale of delay will likely need to be much longer than IEX's 350 microseconds, perhaps on the order of minutes or hours. We call this "a server on Mars" as an evocative reference to the problem of communicating at solar system or galactic scale. For example, it takes at least four minutes for a radio signal to make the round trip from Earth to a rover on Mars and back.[259]

How does a four-minute mandatory way station for communications help us address information privacy problems? And how does this differ from other proposals that have come before? Consider how a server on Mars requirement would address two notable privacy flashpoints: facial recognition and the right to be forgotten. Before we turn to the details of implementing a server on Mars, we connect the idea to what many scholars have said about privacy, friction, obscurity, and structural protections.

259. This is the minimum time delay that a signal could take to get between Earth and Mars. Depending on the relative positions of the two planets around the sun, it could take up to twenty-four minutes. Thomas Ormston, *Time Delay Between Mars and Earth*, EUR. SPACE AGENCY: MARS EXPRESS BLOG (May 8, 2012), http://blogs.esa.int/mex/2012/08/05/time-delay-between-mars-and-earth/.

1. Connection to Earlier Work

The server on Mars and other desirable-inefficiency regulation encompasses and extends a variety of concepts that have appeared under a number of names throughout the privacy literature. Harry Surden observes that few of the privacy protections we enjoy in our day-to-day lives flow from legal sources.[260] Instead, they are *latent*, the product of structural barriers like walls, fences, and clothing.[261] It is a conveniently privacy-preserving accident of physics, for example, that humans are unable to see through walls. Where other legal regimes could be concocted to defend this expectation of visual privacy in enclosed spaces, walls serve this purpose so effectively that further action seems gratuitous.

Unfortunately, as Surden notes, these safeguards evaporate as arbitrarily as they were originally established, casualties of the whim of technological progress.[262] A fence that for centuries might have reliably impeded visual trespasses on private property could instantly lose relevance in the face of a camera-laden quadcopter.[263] Technology, as far as Surden's conception of latent privacy is concerned, can change the laws of physics, upending the structural reality on which the bulk of implicit privacy protections rely.[264]

Our work picks up where Surden's left off. As a regulatory response to technology's impact on structural privacy, Surden proposes that lawmakers might choose to "overlay another regulatory device . . . in parallel with existing structural constraints," a recommendation that directly invites our approach but provides little operational guidance.[265] We see desirable inefficiency as capable of implementing Surden's suggestion. When a new technology reduces the cost of a privacy violation, one can mandate desirable inefficiency in its place, increasing the difficulty of encroaching on privacy to restore the prior conditions. Since inefficiency is tunable, it can be shaped to resemble, with high fidelity, the cost function of the structural barrier that it seeks to replace.[266] We advance desirable inefficiency as a class of prescriptions to remedy the paradigm that Surden descriptively observes.[267] It is important to note that Surden does not advance a normative framework—he documents a pattern of technology-driven privacy erosion and

260. Harry Surden, *Structural Rights in Privacy*, 60 SMU L. REV. 1605, 1625–27 (2007).
261. *Id.* at 1627.
262. *Id.* at 1608.
263. *Id.* at 1608, 1622.
264. *Id.*
265. *Id.* at 1626.
266. *Id.*
267. *Id.*

discusses regulation without mandating a specific level to which privacy *should* be set.[268] In a similar manner, we submit desirable inefficiency as a technique whose flexibility renders it convenient for advancing a wide range of normative ends.

The server on Mars also implements important recurring themes in privacy law scholarship—the recognition of the value of obscurity and friction. [269] Whereas our predecessors focus primarily on the normative value of obscurity and friction,[270] our work asks a subsequent, more concrete and actionable question: how can you bring about the same effects that desirable obscurity or friction achieve when they are absent or eliminated from a system?

Woody Hartzog and Frederic Stutzman formalize a legal notion of obscurity as privacy protection,[271] listing four qualities that can make information obscure if any is absent: search visibility (whether information is accessible via a keyword-based search engine), unprotected access (whether the information is protected by passwords or other access-control), identification (whether identifying information links back to a particular individual), and clarity (whether the information itself is obfuscated or disassembled).[272] When several of these factors are missing, obscurity increases in kind.[273] In the language of desirable inefficiency, obscurity protects privacy by impeding information retrieval, increasing the amount of labor necessary to find, extract, and reassemble data.[274] In this sense, obscurity is tantamount to a proof of work puzzle whose solution is the information itself.

Many of Hartzog and Stutzman's examples focus on obscurity latent in a system.[275] Hartzog and Stutzman briefly discuss injecting obscurity into a system that doesn't provide it as a protective remedy, but do not develop it into a realistic tool.[276] Two of Hartzog and Stutzman's four qualities of obscurity, whether information is indexed by a search engine or protected by a password, are easy to manipulate but provide only a limited range of outcomes.[277] Information either is or is not password-protected, for example, and the distance between these two contingencies

268. *Id.*

269. Woodrow Hartzog & Frederic Stutzman, *The Case for Online Obscurity*, 101 CALIF. L. REV. 1, 3 (2013); William McGeveran, *The Law of Friction*, 2013 U. CHI. LEGAL F. 15, 15 (2013); Neil M. Richards, *The Perils of Social Reading*, 101 GEO. L.J. 689, 689 (2013).

270. *Id.*

271. Hartzog & Stutzman, *supra* note 269, at 1.

272. *Id.* at 4.

273. *Id.*

274. *Id.* at 5.

275. *Id.* at 40.

276. *Id.* at 43.

277. *Id.* at 42.

is enormous.[278] The two qualities of obscurity that vary along a spectrum, like clarity and the presence of information that might be identifying, are difficult to control in a systematic manner and imprecise in effect. In our terms, then, obscurity can implement a variety of possible privacy schemes by making information retrieval inefficient to varying degrees, but it is unwieldy as a prescriptive tool.

Desirable inefficiency in the privacy context connects once again to Julie Cohen's concept of semantic discontinuity.[279] Desirable inefficiency, like semantic discontinuity, "tend[s] to be conceptualized" by traditional, rationalist points of view "as imperfections that detract from the realization of legal, market, and technical ideals."[280] To implement semantic discontinuity, Cohen prescribes that "privacy law and policy should reinforce and widen gaps within the semantic web so that situated subjects can thrive."[281] That seems an apt description for some of the examples we have presented in this paper, perhaps most literally the IEX exchange, which imposes a 350-microsecond "gap" in the system, in pursuit of fairness.[282] We think the applications of desirable inefficiency we present next help to "separate contexts from one another," fulfilling the role for regulation that Cohen recommends.[283]

Finally, in this vein, Bill McGeveran and Neil Richards independently examine *friction* in the context of sharing information over social media.[284] They each dissect the privacy implications of a recent trend toward frictionless sharing, in which content consumption habits on websites like Netflix and Spotify are automatically broadcasted to friends on Facebook without any action on the part of the user.[285] The friction to which these companies and the authors refer is the notice and affirmative button clicks that would typically accompany this information flow.[286] This friction is a specific form of inefficiency—the work necessary for a user to share content. In a frictionless setting, this work declines to zero, a phenomenon about which the authors both compile a lengthy list of privacy concerns.[287]

McGeveran proposes a solution in the form of a simple rule of thumb for future frictionless sharing platforms: "it should not be easier to 'share'

278. *Id.*
279. COHEN, *supra* note 4, at 239.
280. *Id.*
281. *Id.* at 248.
282. Levine, *supra* note 70.
283. COHEN, *supra* note 4, at 252.
284. McGeveran, *supra* note 269.
285. *Id.* at 62; Richards, *supra* note 269, at 691.
286. McGeveran, *supra* note 269, at 63.
287. *Id.*; Richards, *supra* note 269, at 689.

an action online than to do it."[288] That is to say, it should be at least as difficult to share the fact that an action was performed as to perform the action itself.[289] McGeveran offers as an example that Netflix could simply add a "Play and Share" button to each video next to the existing "Play" button, a change that makes it exactly as difficult to watch the movie as to share that it was watched.[290]

McGeveran's proposal arrives at a conclusion that forms the basis for our argument: "zero friction would be intolerable to most observers."[291] He explains this point by invoking Lawrence Lessig: privacy is "protected by the high cost of gathering or using" information, meaning that "friction is . . . privacy's best friend."[292] McGeveran and Lessig embrace a middle ground between cumbersome inconvenience and the efficiency-fueled social media race to zero (not unlike the race to zero that IEX resisted).[293] Some carefully measured degree of inefficiency, they reason, is a desirable tradeoff between convenience and privacy.[294] Friction is particularly conducive to achieving this balance because "[w]e can choose to calibrate the amount of friction at an infinite number of levels, responsive to the costs and benefits in each situation."[295] This tunability applies to the entire category of techniques that we bundle into desirable inefficiency, permitting policymakers to regulate privacy with fine precision. Although McGeveran's analysis focuses on one specific variety of inefficiency (friction in the form of additional user actions) in one specific domain (online sharing) he arrives at a number of broader insights that inform our more expansive framework.[296]

How exactly would a server on Mars work? What information privacy problems would it help mitigate? Consider two examples: facial recognition and the right to be forgotten.

2. Facial Recognition

There are many concerns about the rise of technology that can be used to identify people based only on photos of their faces,[297] only some of which can be addressed through a state-mandated server on Mars.

---

288. McGeveran, *supra* note 269, at 63.
289. *Id.* at 42.
290. *Id.* at 64.
291. *Id.* at 53.
292. *Id.* at 60; *see* LESSIG, *supra* note 3, at 202.
293. McGeveran, *supra* note 269 at 58; *see* LESSIG, *supra* note 3, at 202.
294. McGeveran, *supra* note 269 at 53; *see* LESSIG, *supra* note 3, at 200–01.
295. McGeveran, *supra* note 269, at 53.
296. *Id.*
297. *See generally* Yana Welinder, *A Face Tells More than a Thousand Posts: Developing Facial Recognition Privacy in Social Networks*, 26 HARV. J.L. & TECH. 165 (2012) (discussing the "risks of face recognition technology when combined with the vast amount of personal

One such concern is that facial recognition might empower people bent on committing fraud and assault.[298] People might use a smart phone's camera and Internet connectivity to identify strangers in their close proximity.[299] Armed with knowledge of the stranger's identity, augmented with additional facts gleaned from an Internet search, a fraudster or stalker could engender false trust in their victim or use the Internet to identify vulnerabilities.[300] Imagine a child predator using this technology to identify a child walking home from school, lying that the child's parent is in the hospital, and referring to the child and parent by name.[301]

Desirable inefficiency expands the toolkit we can use to respond to this particular threat model from facial recognition. A regulator sufficiently worried about this threat could enact a new rule: anybody who provides a facial recognition capability directly to end users must implement a mandated minimum latency for any facial identification. They must place their server "on" Mars.

The server on Mars implicates all of the properties of desirable inefficiency we discussed in Part II.[302] The four-minute latency is tunable. Four minutes might be fine if it prevents a stalker from identifying a child passing them on the street. But it might seem insufficiently short if the adversarial threat model is a child being watched at play in a park (or a person being watched in a bar). To address different threats, we might mandate a server on the Sun (approximately seventeen minutes round-trip) or maybe even a server on Neptune (eight hours).

---

information aggregated in social networks"); Note, *In the Face of Danger: Facial Recognition and the Limits of Privacy Law*, 120 HARV. L. REV. 1870, 1873, 1881–82 (2007) (noting that "many scholars have been preoccupied with the collection of personal information in large, easily abused databases that threaten to create detailed portraits of individuals").

298. In a letter from Senator Al Franken to Kevin Alan Tussy of FacialNetwork, Senator Franken described the concern about the NameTag app for Google Glass, which could match photos with social media profiles. *See* Conan Milner, *Latest Google Glass Controversy: Face-Recognition Search Engine App*, EPOCH TIMES (Feb. 11, 2014), https://m.theepochtimes.com/latest-google-glass-controversy-face-recognition-search-engine-app_500481.html; *see* Bridget A. Sarpu, Note, *Google: The Endemic Threat to Privacy*, 15 J. HIGH TECH. L. 97, 126–27 (2014).

299. Sarpu, *supra* note 298, at 126–27.

300. Senator Franken further described his concern to Kevin Alan Tussy of FacialNetwork. Milner, *supra* note 298 ("'I am especially concerned that NameTag plans to scan dating websites such as Match and OKCupid. . . . It is easy to envision how this technology could facilitate harassment, stalking and other threats to personal security. Your company has an obligation to protect users from these threats.'").

301. Kate Webb, *New Facial Recognition App 'Creepy', Says Kids Entertainer Raffi*, TORONTO STAR, (Jan. 10, 2014), http://www.metronews.ca/news/canada/2014/01/10/new-facial-recognition-app-creepy-says-kids-entertainer-raffi.html (citing threat of facial recognition app to children specifically).

302. *Infra* Section II.C.

When tuned correctly, a server on Mars requirement will separate users of facial recognition into favored and disfavored groups. With four-minute latency, regulators can discourage street-side stalking while not interfering with apps that might help people better navigate cocktail party conversations. With a server on Neptune you won't be able to use facial recognition to trick a stranger in a bar, but you can still use it to organize your digital photos.[303]

Finally, the regulator might literally borrow IEX's solution, mandating a hardware implementation of the server on Mars. Granted, it might be difficult to deploy the 33.9 million miles of fiber-optic cable that would be necessary to directly mimic IEX's approach, but other materials through which light propagates more slowly or other forms of specially-designed hardware could satisfy this purpose.[304] A hardware solution will ensure the same persistence, external verifiability, transparency, and regulability benefits discussed earlier.

3. The Right to Be Forgotten

The server on Mars solution could provide a nuanced approach for protecting what is known as the "Right to be Forgotten."[305] This right protects the ability of individuals to wipe the digital slate clean of truthful but outdated information by giving them the power to demand the removal of records from databases.[306]

The right has been given its most prominent expression in law in the European Union, which has recognized the right to be forgotten in case law and legislation. The decision by the Court of Justice of the European Union in *Google Spain v. Costeja*, recognizing the right to be forgotten in the Data Protection Directive, may at first glance be interpreted as a decision rooted in desirable inefficiency.[307] By purportedly drawing a line between search engines, which were obligated under the decision to unlink "inadequate, irrelevant or no longer relevant, or excessive" search results from their indices, and publishers, which were not, some might credit the court for finding the middle ground.[308] Other commentators have resisted this conclusion, noting (we think correctly) that being

303. Andreas Girgensohn, et al., *Leveraging Face Recognition Technology to Find and Organize Photos*, *in* PROCEEDINGS OF THE 6TH ACM SIGMM INT'L WORKSHOP ON MULTIMEDIA INFO. RETRIEVAL 99, 99–101 (2004).

304. Nola Taylor Redd, *How Long Does it Take to Get to Mars?*, SPACE.COM (Nov. 14, 2017, 10:22 AM), http://www.space.com/24701-how-long-does-it-take-to-get-to-mars.html.

305. JONES, *supra* note 34 (discussing the right to be forgotten).

306. *Id.*

307. Case C-131/12, Google Spain SL v. Agencia Española de Protección de Datos, 2014 E.C.R. 317, http://curia.europa.eu/juris/document/document.jsf?text=&docid=152065&doclang=EN.

308. *Id.* at ¶ 93.

removed from the search engines is to be effectively severed from public accessibility.[309]

Imagine if, in the alternative, the European Court of Justice imposed a server on Mars requirement. Google could be obligated to withhold the offending search query results not entirely but only for a prespecified delay.[310] The Court would probably leave the implementation details to Google, which could, for example, take advantage of the dynamism of browsers that speak Javascript and HTML5 to implement the delay in code.[311] The browser would display the first page of search query hits without the "forgotten" link for a specified length of time—say four minutes—after which it would appear. The idea is that offending hits would not appear for most users, but for the dedicated researcher, it would appear with patience.

We are not taking a stand on whether the *Costeja* decision represents a triumph for privacy, unwise censorship, or something in between. Our point instead is to imagine ways in which desirable inefficiency and proof of work might have provided the court with different, perhaps more palatable middle grounds.

B. *Compared to Other Regulatory Approaches*

Desirable inefficiency will provide benefits and capabilities that other regulatory approaches lack. It threads a needle between traditional command-and-control measures and newer proposals for nudges. Better than command-and-control approaches, desirably inefficient regulation (such as a mandated server on Mars) acts more like a "policy dial" than a "policy lever," expanding the regulatory art-of-the-possible from a binary off/on choice to a range of possibilities.[312] At the same time, desirably inefficient solutions can be far more specific and prescriptive than most nudges, which have been described as "libertarian paternalism" by Sunstein and Thaler.[313] When a nudge is too libertarian to address a problem, desirable inefficiency can often do more and go further.

---

309. Jonathan Zittrain, *Don't Force Google to 'Forget,'* N.Y. TIMES (May 14, 2014), https://www.nytimes.com/2014/05/15/opinion/dont-force-google-to-forget.html.

310. This is similar to a proposal by Evan Selinger and Woody Hartzog for a middle ground result rooted in obscurity, one that would have pushed the objectionable material off of Google's first page of results. *See* Evan Selinger & Woodrow Hartzog, *Google Can't Forget You, but it Should Make You Hard to Find*, WIRED (May 20, 2014, 3:33 PM), http://www.wired.com/2014/05/google-cant-forget-you-but-it-should-make-you-hard-to-find/.

311. For an online tutorial on how to accomplish a delayed reveal, see *CSS3 Transitions Using Visibility and Delay*, GREYWYVERN.COM (Apr. 21, 2011), http://www.greywyvern.com/?post=337.

312. *See* Dan L. Burk & Mark A. Lemley, *Policy Levers in Patent Law*, 89 VA. L. REV. 1575, 1612 (2003).

313. Cass R. Sunstein & Richard H. Thaler, *Libertarian Paternalism is Not an Oxymoron*, 70 U. CHI. L. REV. 1159, 1160–62 (2003).

Desirable inefficiency will often be the best choice for a regulator faced with a difficult problem, particularly one involving information flow. That said, it is not a panacea, and there will be many situations in which a command-and-control approach or a nudge continues to be the best solution.

Finally, we note both similarities and important differences between desirably inefficient regulation and the imposition of a tax. Taxes exhibit many of the benefits of desirable inefficiency, most notably the ability to filter and separate based on different behaviors. But unlike a mere tax, desirable inefficiency can impact quantities beyond just money (e.g., time or energy) and lends itself to decentralized and architectural solutions, which we argue will provide implementation benefits over a tax in many situations.

1. Command-and-Control

A familiar metaphor in policy circles is that of a policy "lever."[314] This evokes a switch, an on/off, binary condition. A policy lever can be "flipped" or left alone. Because desirable inefficiency is inherently a tunable feature, policymakers will find that it allows new choices between "on" and "off"—between "permitted" and "forbidden." For this reason, we propose selecting a new metaphor to capture the implications of desirable inefficiency: a policy dial. Policy dials are intrinsically tunable mechanisms. Deciding to deploy a policy dial is but a first step; policymakers need also to decide where to set or tune the dial to effectuate a particular goal. This tuning process doesn't always entail setting some magic global threshold. A dial need not be calibrated to the same position in every instance; instead, one can alter its setting depending on the circumstances of individual cases, weighing various factors to calculate an overall remedy.

For these reasons, policy dials often present appealing "middle ground" or "third way" possibilities between unappealing or nonviable polar choices.[315] Rather than choosing between an unending deluge of spam and an overprotective ban on unsolicited emails, for example, an appropriate proof of work solution could set a dial somewhere between these two extremes.

2. Nudges

Like nudges, mandated desirable inefficiency is less direct than classic command-and-control regulation.[316] These three options sit along a

314. Burk & Lemley, *supra* note 312, at 1630.

315. THALER & SUNSTEIN, *supra* note 36, at 252.

316. *Id.* at 253.

continuum, so it is important not to overstate the distance between them. Nudges, like mandated desirable inefficiency, also tend to separate those being regulated based on the strength of their preferences.[317] For example, switching from an opt-in to an opt-out for retirement benefits changes rates of participation because default choices tend to be sticky.[318] Nevertheless, those with a strong preference not to participate will still have the freedom to opt out, suggesting that the nudge will separate based on this preference. In other words, the impact of a nudge is most heavily felt by those who do not have strong preferences.[319]

On this continuum, mandated desirable inefficiency usually sits much closer to command-and-control than to nudges. The most prominent proponents of nudges have aptly described them as methods of libertarian paternalism.[320] Although some critics have debated whether nudges tend to be more paternalist than the proponents have suggested, it is hard to deny that placing healthy food at eye level is far less paternalist or interventionist than stalling digital communications for four minutes or longer.[321] As we have noted previously, desirable inefficiency often appears in adversarial settings where participants have significant incentive to overcome inefficiency if possible. This target audience of highly-motivated high-frequency traders, Bitcoin miners, or spammers is a far cry from the easily-confused but well-meaning "humans" that Sunstein and Thaler address.

Nudges are therefore able to influence behavior gently. They are intended to be seamless, shaping behaviors without creating disruption or adding friction.[322] Desirable inefficiency, in contrast, must respond to its adversaries heavy-handedly, embracing cost as a mechanism for altering outcomes.

Finally, nudges tend to arise in situations that can easily be reduced to a quantifiable performance metric, such as amount saved, calories consumed, or plastic grocery bags used (a proxy for sea animals endangered). Desirable inefficiency may be much more suitable for advancing human values that are difficult to reduce to a measurement, such as fairness or trust, values we might otherwise implement through traditional command-and-control.

---

317. *Id.* at 247.

318. Brigitte C. Madrian & Dennis F. Shea, *The Power of Suggestion: Inertia in 401(k) Participation and Savings Behavior*, 116 Q.J. ECON. 1149, 1151–52 (2001).

319. THALER & SUNSTEIN, *supra* note 36.

320. *Cf.* Sunstein & Thaler, *supra* note 313, at 5 (using the term "libertarian paternalism" to describe a policy that preserves freedom of choice but steers people toward a desired behavior).

321. THALER & SUNSTEIN, *supra* note 36.

322. *Id.* at 5–6.

3. Taxes

At first glance, many of our examples of desirable inefficiency might seem indistinguishable from taxes. Dwork and Naor's proposal for spam is a tax on email exacted in computational effort.[323] Captchas assess a tax via mental exertion in the form of pattern matching, a currency that (for now) happens to be disproportionately plentiful for humans and scarce for machines.[324] The smartphone lockout mechanism is a tax on time that penalizes anyone who needs to make a large number of attempts—such as an intruder who is guessing the password brute-force—while imposing minimal inconvenience on a user who is simply clumsy.[325]

In many instances, desirable inefficiency filters "good" users from "bad" in largely the same manner as a tax.[326] Although some taxes exist simply to create government revenue streams, taxes can also serve to alter prices in ways that change incentives and, ultimately, behaviors.[327] For example, a nation might decide to assess a gasoline tax to encourage people to use more fuel-efficient alternatives by making them more cost-competitive.

We can see a similar effect in several of our examples of desirable inefficiency. At the moment, the price of sending an email is negligible, requiring so little computation time that a concerted spammer can send a massive number of messages each day. Dwork and Naor propose raising the price of each email dramatically by increasing the amount of computation necessary to send a message.[328] Although the average user would still be willing to make this small but appreciable outlay to send a few dozen emails each day, the price hike would increase the cost of a single spam message well beyond the profit that a spammer will make from sending it.

Taxes provide one key benefit that desirable inefficiency lacks—they allow the government to recapture the costs of tax-driven behavioral changes in the form of tax revenue.[329] Every penny paid in gasoline tax, although a financial burden for a consumer, can at least be redirected somewhere else in the economy to soften or reshape the impact of the tax. Doing so ensures that the increased costs of price changes do not entirely go to waste. In contrast, waste is a hallmark of desirable inefficiency. The computational work of sending an email or validating Bitcoin

323. Dwork & Naor, *supra* note 21.

324. Yale, *supra* note 196.

325. APPLE, *supra* note 201, at 15.

326. Brian Galle, *Tax, Command . . . or Nudge?: Evaluating the New Regulation*, 92 TEX. L. REV. 837, 844 (2014).

327. *Id.* at 839.

328. Dwork & Naor, *supra* note 21.

329. Galle, *supra* note 326, at 851.

transactions is lost as waste heat. Likewise, captchas waste 150,000 hours of productivity every day.[330]

Not only is this waste entirely justifiable, but, in many circumstances, it is actually preferable. Suppose that Dwork and Naor had replaced proof of work with a financial tax instead. Every email sent would incur a fee, the Internet equivalent of postage.[331] This system would achieve the same spam-prevention goals as Dwork and Naor's proposal, but would substitute waste heat for tax revenue, seemingly an upgrade.

However, implementing this solution would make it necessary to entirely rethink the way email functions. Unlike paper mail, email is fundamentally decentralized.[332] Paper mail is processed and delivered by a single entity, the United States Postal Service, which makes it easy to collect stamp revenue and reject letters that lack enough postage. In contrast, email messages travel from the sender's computer to the recipient's without passing through any central intermediary. To enact an email tax, every email account would have to have an associated payment method. To ensure that the tax was paid, all email messages would have to pass through a central third party. In effect, we would have to redesign email to build out the infrastructure for an email tax—it would no longer be email as we know it today. It would arguably lack the dynamism, fault-tolerance, and generativity of decentralization that has enabled its success.

In general, tax-like forms of desirable inefficiency make a tradeoff: in order to remain decentralized, they forgo the opportunity to collect tax revenue. Tax collection requires a centralized system of detection, collection, and enforcement. When it comes to the goal of preventing spam, however, collecting a tax is far less important than making sure it was paid. Proof of work comes with a built-in, decentralized mechanism for doing so—a recipient can easily check whether a sender completed a proof of work problem, thereby ensuring the tax was paid.[333] In this light, Dwork and Naor's approach is what we might call a *wasteful tax*—a tax that must be paid but not collected. This sort of tax maintains its potency as a pricing instrument but sacrifices revenue in order to preserve decentralization.[334]

---

330. David Pogue, *Time to Kill Off Captchas*, SCI. AM. (Mar. 1, 2012), http://www.scientificamerican.com/article/time-to-kill-off-captchas/.

331. A decade ago, entrepreneurs were proposing precisely this type of system. *See* Saul Hansell, *Postage is Due for Companies Sending E-mail*, N.Y. TIMES (Feb. 5, 2006), http://www.nytimes.com/2006/02/05/technology/postage-is-due-for-companies-sending-email.html (describing service called Goodmail).

332. NARAYANAN ET AL., *supra* note 102, at 6.

333. NAKAMOTO, *supra* note 20.

334. It is worth noting that there are still limited ways of recapturing some of desirable inefficiency's waste. Computational work, for example, could be repurposed for good causes. The

4. The Political Economy of Desirable Inefficiency

Regulators may prefer mandated desirable inefficiency over traditional approaches because of the political economy of the regulation of code. Because desirable inefficiency approaches will often reveal alternatives beyond a permit-or-ban binary, they can be used to strike meaningful compromises.

The intrinsic tunability of desirable inefficiency allows the regulator to wield it as a form of responsive regulation, a regulatory approach that advocates starting with softer, less punitive approaches to regulation, turning to more onerous approaches only when these fail.[335] An information privacy law or rule mandating a server on Mars, for example, could set an initial value for the mandated delay while building in a mechanism for assessing whether the delay continues to make sense every six months, say. Especially as technology continues to evolve, regulators may realize that the delay should be increased or decreased to take into account new threat models.

5. Regulating Drones

Consider one final example, which highlights the potential for desirable inefficiency to serve as a tool for regulators. The rise of inexpensive, capable drones sold directly to consumers has raised concerns about privacy and airspace safety.[336] In the United States, both the FAA and state regulators have begun to tentatively propose and promulgate rules limiting, for example, where drones are permitted to fly.[337]

These regulators might turn to desirable inefficiency to address concerns about drones, leveraging the beneficial design patterns described in Part II and the advantages they afford over traditional approaches to regulation.

Folding@Home project seeks to use spare computational cycles on personal computers to solve protein folding problems that could help cure diseases like Alzheimer's or Parkinson's. This collective computational power is able to match that of a large supercomputer. One could imagine developing proof-of-work problems that contribute to causes like Folding@Home, mitigating some of the costs of existing approaches that merely waste electricity. FOLDING@HOME, http://folding.stanford.edu (last visited Aug. 22, 2016). Similarly, Google's reCaptcha effort exploits humans reading garbled text to interpret the difficult-to-decipher words in scanned books. *reCAPTCHA: Tough on Bots Easy on Humans*, GOOGLE, https://www.google.com/recaptcha/intro/android.html (last visited Apr. 14, 2018). Even so, repurposed work is a far less flexible currency than money or electricity.

335. *See generally* IAN AYRES & JOHN BRAITHWAITE, RESPONSIVE REGULATION: TRANSCENDING THE DEREGULATION DEBATE 4–5 (1992) (describing the responsive regulation approach).

336. Troy A. Rule, *Drone Zoning*, 95 N.C. L. REV. 133, 135–36 (2016).

337. *Id.* at 142–44.

Currently, the FAA has designated sensitive airspaces as no-fly zones.[338] You cannot fly a drone over the White House, in Yosemite (or any national park), or within five miles of an airport.[339] Nor can you fly a drone at an altitude of greater than 400 feet.[340] This is a binary, policy-lever approach to the problem.

In contrast, desirable inefficiency could offer a more nuanced balancing of costs and benefits. Assume the primary concern in a given city is the nuisance of a sky abuzz with dozens of noisy devices. With this goal in mind, rather than prohibiting drone flight entirely, one could require drones to drain their batteries at higher than their ordinary rate when flying within the city limits or over particular classes of property (residential, commercial, schools, etc.).[341] For example, these rules could be structured to channel (but not compel) package delivery drones to travel long distances over low-drain streets rather than high-drain private property.[342] The higher the rate of mandated battery drain, the fewer number of drones would be aflight in that airspace at any given time. This creates a tunable dial that policymakers can set to encourage drone owners to limit flight in some areas.

This approach acknowledges and permits the positive uses of drones while altering incentive structures to balance these benefits with undesirable side-effects.[343] It operates somewhat like a tax, altering the incentive structure, limiting flight over sensitive areas to drone operators with the most strongly expressed preferences, namely those willing to accept the performance hit.

To draw on the relative strengths of desirable inefficiency, imagine regulators model their battery drain rules on proof of work. Rather than issuing an open-ended edict instructing operators to consume energy in any manner they choose, regulators could require a proof-of-work-style method for mandated battery drain.

---

338. 14 C.F.R. § 107.51(b) (2016); *see generally Unmanned Aircraft Systems*, FED. AVIATION ADMIN. (Mar. 12, 2018, 12:00:29 PM), https://www.faa.gov/uas/ (giving an overview of information for people looking to fly drones in the United States).

339. *See Airspace Restrictions*, FED. AVIATION ADMIN. (Oct. 4, 2017, 1:18:01 PM), https://www.faa.gov/uas/where_to_fly/airspace_restrictions/; *Unmanned Aircraft in the National Parks*, NAT'L PARK SERV., https://www.nps.gov/articles/unmanned-aircraft-in-the-national-parks.htm (last visited Mar. 16, 2018).

340. 14 C.F.R. § 107.51(b).

341. Troy Rule proposes a "drone zoning" approach to addressing the diversity of concerns about drones by channeling different permissible uses into the various parts of a city. Rule, *supra* note 336, at 184–200. This proposal would be a natural fit for implementation through a desirable inefficiency mandate.

342. *See id.*

343. *Id.* at 186–87 (advocating for a utilitarian costs-benefits approach to channeling various drone uses).

For example, regulators could literally require drones to solve pre-specified, difficult math puzzles while flying over restricted spaces, the way Bitcoin does.[344] They could select puzzles known to require heavy computation, and concomitantly high energy usage, resulting in predictable, accelerated reductions of battery life.[345]

Drone proof of work might have an enforcement benefit as well. Currently, the FAA uses a registration system for drones.[346] Even hobbyists with inexpensive drones must submit personal information on a government website including the serial number of their drone.[347] Many owners cannot be bothered to take these steps, meaning the rate of compliance is quite low.[348]

If drones are required not only to solve but to broadcast the solution to proof of work problems, law enforcement officials could monitor those broadcasts to verify compliance. The police officer with the scanner on the ground would not know who owned an unregistered drone overhead, but she would at least have mathematically verifiable proof that the operator was following the restricted zone rule.

This proof of work solution faces a problem not faced by Bitcoin or spam: processor offloading. The drone operator might solve the puzzles on a computer outside the device—say a laptop on the ground or a computer in the cloud—and transmit the answer wirelessly to the drone. Bitcoin and spam assume that users will solve proof of work using the most powerful computer they can spare.[349] Drone proof of work depends on the use of the drone's CPU itself, in order to effect battery drain.

One solution might be to effect battery drain not by computation, but by ordeal. For example, regulators might require drones to fly a pattern known to require greater work from rotors and motors, perhaps a sinusoidal pattern through the air. Or regulators might require the installation of baffles on the body of the drone in configurations known

344. NAKAMOTO, *supra* note 20, at 3.

345. Researchers have calculated the energy consumption consequences of Bitcoin mining. KARL J. O'DWYER & DAVID MALONE, NAT'L UNIV. OF IR. MAYNOOTH HAMILTON INST., BITCOIN MINING AND ITS ENERGY FOOTPRINT 1–3 (2014), http://karlodwyer.com/publications/pdf/bitcoin_KJOD_2014.pdf.

346. *FAADroneZone*, FED. AVIATION ADMIN., https://faadronezone.faa.gov/#/ (last visited Mar. 16, 2018).

347. *Id.*

348. John Goglia, *Over 450,000 Hobby Flyers Have Registered. So Why Does Fighting FAA's Drone Registry Still Matter?*, FORBES (June 15, 2016, 7:31 PM), https://www.forbes.com/sites/johngoglia/2016/06/15/over-450000-hobby-flyers-have-registered-so-why-does-fighting-faas-drone-registry-still-matter/.

349. Eric Limer, *The World's Most Powerful Computer Network is Being Wasted on Bitcoin*, GIZMODO (May 13, 2013, 10:33 AM), http://gizmodo.com/the-worlds-most-powerful-computer-network-is-being-was-504503726.

to change the aerodynamics of the device, like flaps on an airplane's wings. Finally, they might require drones to emit high-powered radio signals at a battery-draining wattage that can be measured. Any of these three methods would be observable and very difficult to fake to the observer on the ground.

Regardless of the precise mechanism, a proof of work for drones would have advantages over any other approach. Unlike a no-flight zone, proof of work makes flight through a particular airspace possible but unlikely.[350] Like a tax, the burden of the proof of work can be tuned to balance the competing interests. It allows for some on-the-ground enforcement and verification, without being as burdensome or arguably invasive as a central registry.

It might seem odd to imagine drones careening across the sky following some government-mandated, intentionally inefficient flight pattern. This, of course, is the common response to proof of work, perhaps no more bizarre than thirty-eight miles of fiber-optic cable or computers solving math puzzles to no direct end.

CONCLUSION

Computer scientists and engineers turn away from efficient solutions when faced with the need to inject complex human values into systems. Desirable inefficiency is an important and emerging area of computation that deserves further investigation and development. Not only should computer experts focus on desirable inefficiency, but regulators, policymakers, and legal scholars should also take up this promising new tool. We need to develop strategies for ensuring the peaceful coexistence of human life and technological progress. Desirable inefficiency can serve as a critical part of such efforts.

The story we have just told, which begins with a peculiar and unappreciated trend in computer science and ends with an entirely new (and, at times, bizarre) approach to regulating digital systems, is not unique. Although this Article grapples with the minutiae that certify particular techniques as "desirably inefficient" to regulators, it should be seen as just one source of inspiration among many. The moral of this Article is that technical design decisions can shape the values that digital systems advance in predictable ways. The emergence of desirable inefficiency highlights what computer scientists have known at some level for years, if not decades. As regulating design becomes ever more popular, regulators can look to desirable inefficiency—and the other unwritten and yet-to-be-written rules of design—for inspiration and opportunity.

350. Rule, *supra* note 336, at 184–200.