

Draft 102517 – Very Preliminary – Please do not cite, quote, distribute etc.

## AUTOMATED LAW LIABILITY

*Susan C. Morse*\*

### INTRODUCTION

Centralized systems make more and more legal decisions. Often this happens because machines, like computers, apply law. For instance, computer programs prepare tax returns, certify compliance with environmental regulation, keep wage and hour records, and take down material allegedly subject to copyright protection. Should automated law systems be directly liable for the errors they make?

Centralized automated law involves machines working in the service of law, including regulation. Automated law means that the machine, for instance the computer, makes a legal determination. It might produce a tax depreciation calculation, a pollution emissions report, or a determination of eligibility for welfare benefits.

A well-designed automated law system can deliver accurate and speedy legal determinations. This can help both regulated parties and the government. But no matter how good the design is, errors are inevitable. Prior work has recommended government oversight to reduce mistakes, but this oversight cannot eliminate them.

The burden of errors made by automated law systems often falls on third parties, not on the user or the maker of the automated law system. For instance, if a taxpayer understates taxable income because of an error made by tax preparation software, and the error goes undetected (which is likely), then the group that bears the burden of that error is the general taxpaying public. If a polluter underreports emissions of a harmful atmospheric chemical, and the error goes undetected (which is likely), then the group that bears the burden of the error is the general air-breathing public. Indeed there is a clear incentive for an automated law system to commit errors that benefit its users at the expense of third parties, so long as those errors will probably not be detected.

Underdetection and underenforcement are common problems for regulatory compliance. It is also common for compliance errors committed

---

\*

by private regulated parties to burden third parties, or in other words create a negative externality. What is special about errors committed by automated law systems is that they are centralized. This means that automated law systems present powerful enforcement opportunities.

Enforcing the law directly against a centralized actor is an attractive strategy from the perspective of a regulator. If a regulator can hold a centralized actor liable for the total cost of a repeated error, the regulator efficiently uses government resources to solve the persistent problems of underdetection and underenforcement. The regulator need not pursue each individual taxpayer, or each particular pollution source. Instead the regulator can require the centralized automated system to pay directly not only the cost of an error initially discovered for one regulated party but also, through a damages multiplier, the total cost of the error for all filings facilitated by the system.

As more and more legal decisions, including regulatory compliance decisions, are made by automated law systems, the law will face the question of whether, and to what extent, the systems themselves ought to be directly liable for the cost of their mistakes. This approach would fundamentally change the way in which regulatory law develops. Now, disputes about regulatory law are usually between the government and a regulated party. Under automated law liability, dialogue and controversy between the automated law systems and the government would become a central source of determining the content and progression of the law. The locus of disputes would shift from the user of an automated law system to the maker of an automated law system. That is what this Article is about.

Part I of this Article illustrates the regulator's underdetection and underenforcement problem using the example of tax law. As Part I explains, even when a taxpayer uses tax software to file a return, the taxpayer has the legal liability for errors, legal and factual, on the return. Some tax software firm sell audit insurance, and under these insurance contracts the firms assume some of the taxpayer's liability. But in either case, a central enforcement problem is that tax underpayments often go undetected. In this case, other taxpayers bear the burden of the underpayment, because other taxpayers' taxes must increase to compensate.

Part II outlines how a broader regime of direct liability for errors in automated law systems could work. What would it mean for an automated law system to be "directly liable" when the person with the legal obligation to comply with the law is the user of the system? What is the "cost" of a

mistake? And when is an error caused by automated law system, so that the error is “its mistake”?

Part III explains how the market for automated law products might change as a result of the imposition of automated law liability. One likely result is differentiation between more and less aggressive products. Another is a framework of bonding and/or reinsurance, especially since appropriate regulations might require proof of creditworthiness before accepting filings prepared by an automated system. A third possibility is that the proposed burden of liability might be so heavy that it would discourage the private development of automated law products, and leave it to the less technologically adept government to develop automated law.

Part IV considers the fact that automated law systems can be developed by government, as well as by private firms. In either case, correcting system errors is important to the quality of the resulting law. The direct liability regime outlined in Part II and Part III uses government as the external check on automated law systems developed by private firms. A government automated law system presents the converse challenge: how can private parties be empowered to identify and remedy errors made by the government system?

## I. THE REGULATOR’S PROBLEM

### A. Underenforcement

Tax law provides a good illustration of the problem of underdetection and underenforcement in regulation.

- Underpayment errors undetected
  - IRS resource constraints
  - Audit rates of perhaps 1%-2%, audit lottery
- No damages multiplier

### B. Negative Externalities

In regulation and compliance, third parties – not the user, and not the system – often bear the burden of automated law system errors.<sup>1</sup> Unless

---

<sup>1</sup> Cf. Oren Bracha & Frank Pasquale, *Federal Search Commission? Access, Fairness, and Accountability in the Law of Search*, 93 Cornell L. Rev. 1149, 1185-86 (2008) (noting the problem of possible negative externalities resulting from the filtering and organizing of search results by algorithms).

errors are detected, an automated law system that underreports taxable income for a user taxpayer results in higher taxes for other taxpayers. An automated law system with undetected errors that underreports environmental emissions may help the user (due to lower compliance costs) or the system (due to higher profits) but it produces more pollution for the general public and the environment.<sup>2</sup>

- In tax, taxpayers who do not or cannot take aggressive positions (whether current or future) bear burden of insufficient revenue.
- Aggressive positions more likely under provisions not dominated by withholding. More likely e.g. for investment income or equity compensation than for cash compensation. Perhaps a distributive justice concern.<sup>3</sup>

### *C. Reporting Positions*

- Applicable law (6662, 6694) provides that taxpayers may file returns without exposure to penalties (for either TP or advisor) for a position with “substantial authority.” “Substantial authority” is something less than “more likely than not to succeed in court.”
- Legal errors that present themselves are not always obvious legal errors. Sometimes they are interpretations of law that are reasonable assertions, but that a court later rejects. Harm to other taxpayers results not just because of tax gap due to fraud or evasion, but due to underpayments on reporting positions later rejected in audit process or by a court.
- Taking “reporting positions” is a good thing because it pushes the development of the law. Taxpayers should be allowed to take a view of the law that is favorable to them. But because of the low chance of audit, reporting position practice may push the development of the law too far in favor of taxpayers. Possible example: FF miles/loyalty programs.
- A plus of the direct automated law liability idea is that it sets up two able litigants – centralized system and government – to engage in controversy over grey area issues and figure them out.

### *D. Tax Software Preparation Programs*

---

<sup>2</sup> An underlying assumption is that the regulatory policy is in some sense wise or correct, i.e., that error-free compliance would properly measure time worked, impose tax liability, set environmental pollutants and so forth.

<sup>3</sup> Linda Sugin

- There are good things about tax return preparation software etc., including the saving of taxpayer time that it supports.
- Some but not all programs give aggressive advice.
- Regardless of how an error arises, the software under current law is not necessarily involved in the resolution of the error. Say tax software prepares a tax return with a legal error. The taxpayer files the return. The government will rarely discover the legal error, because its audit coverage is so low. If it does discover the error, it will hold the taxpayer responsible.<sup>4</sup> The taxpayer will be required to pay back taxes, interest, and, only sometimes, penalties. The magnitude of penalties (setting aside criminal penalties) falls far short of the multiplied damages amount that would be necessary to offset the low likelihood of detection.
- There is an incentive for the program to give aggressive advice – perhaps then it could charge more. [Evidence that programs compete on how much refund they can provide?]
- But this is not different from the incentive of other tax preparers to give aggressive advice.
- The difference is the ease of enforcement allowed by a centralized system of liabilities. Builds on some discrete strategies e.g. tax shelter promoter regs that target a centralized source of information.
- The market provides some evidence that taxpayers will pay for insurance against risk that positions are incorrect.<sup>5</sup>

## II. THE ENFORCEMENT OPPORTUNITY OF CENTRALIZED LAW

### A. *What is an Automated or Centralized Law System?*

A machine operates as an automated or centralized law system (for purposes of the analysis here) if it produces a legal determination.<sup>6</sup> For

---

<sup>4</sup> See, e.g., Rodney P. Mock & Nancy E. Shurtz, *The TurboTax Defense*, 15 Fla. Tax Rev. 443, 490-505 (2014) (explaining case law that declines to treat software companies as possibly liable tax preparers and declines to waive penalties based on the defense that the software facilitated an error). [But see Merrill Lynch settlement case]

<sup>5</sup> See Jay A. Soled & Kathleen DeLaney Thomas, *Regulating Tax Return Preparation*, 58 B.C. L. Rev. 151, 180 (2017) (describing tax preparation software audit insurance); Rodney P. Mock & Nancy E. Shurtz, *The TurboTax Defense*, 15 Fla. Tax Rev. 443, 492-94 & nn. 272-74 (2014) (noting tax preparation software “limited guarantees” of accurate calculations and advice including carveouts for errors due to taxpayer inputs of incorrect information or incorrect classification of information).

<sup>6</sup> Securities law compliance software appears to still focus on editing and compilation. In other words, it has not crossed the line into automated law territory yet. A securities law

instance, an automated law system might produce the legal determination that a party is compliant with a regulatory requirement.<sup>7</sup> Automated law systems produce wage and hour records,<sup>8</sup> tax returns,<sup>9</sup> environmental reports.<sup>10</sup> They also produce responses to copyright-based takedown requests.<sup>11</sup> These compliance cases are the focus of the analysis. They are a subset of automated law systems. Automated law also includes private law examples<sup>12</sup> (which are beyond this paper's scope) and government-generated automated law, which is addressed in Part IV.<sup>13</sup>

---

example is the NovaWorks GoFiler suite of products. *See*

<http://www.novaworkssoftware.com/index.php?page=agents/gofilercomplete.html> (last visited August 14, 2017) (promising superior editing capabilities for XBRL, which is the format in which many EDGAR filings are made).

<sup>7</sup> Rather than the law responding to the existence of a machine or other technology, the goal is to determine how machines can help the enforcement and making of law. *Compare, e.g.* Ryan Calo, *Robotics and the Lessons of Cyberlaw*, 103 Cal. L. Rev. 513 (2015) (recommending a “Federal Robotics Commission”);

<sup>8</sup> *See* Elizabeth Tippet, Charlotte S. Alexander and Zev J. Eigen, *When Timekeeping Software Undermines Compliance*, 19 Yale J. L. & Tech. 1 (2017) (providing qualitative study of 13 different timekeeping automated law systems).

<sup>9</sup> *See* Rodney P. Mock & Nancy Shurtz, *The TurboTax Defense*, 15 Fla. Tax Rev. 443 (2014) (describing tax preparation software).

<sup>10</sup> Perillon Software Inc. provides an environmental law compliance automated law example. *See* <http://www.perillon.com/environmental-data> (last visited August 14, 2017) (stating, for instance, that “[o]ur customers use our environmental data management module for GHG MRR Reporting for Subpart A, C, D, W reporting requirements including evolving electronic submission standards (e.g. e-GGRT)”). e-GGRT is the EPA’s reporting system for greenhouse gases. *See* <https://www.epa.gov/ghgreporting/e-ggrr-news> (last visited Aug. 14, 2017).

<sup>11</sup> *See* Maayan Perel & Niva Elkin-Koren, *Accountability in Algorithmic Copyright Enforcement*, 19 Stan. Tech. L. Rev. 473, 477 (2016) (“[M]ajor online intermediaries use algorithms to filter, block, and disable access to allegedly infringing content automatically, with little or no human involvement.”). These are often responses to robot-generated Digital Millennium Copyright Act takedown requests. In contrast to most of the other automated law systems considered here, they appear to be proprietary, i.e. developed, owned and used by a firm such as Google or Facebook.

<sup>12</sup> Ethereum project; Nicholas Cornell and \_\_\_, *Contract Ex Machina*, \_\_ Duke LJ (forthcoming). Smart contracts charge computers with the responsibility of verifying the fulfillment of contract terms, such as delivery of goods, and with the responsibility of executing contract terms, such as transferring funds in payment. *See also* Kiviat, Note, Duke LJ 2015 (re digital asset transfers, confirmation of authorship, title transfers, K enforcement)

<sup>13</sup> *See, e.g.* Joseph Bankman, *Simple Filing for Average Citizens: the California Ready Return*, 107 Tax Notes 1431 (2005) (describing one government system for providing pre-filled tax returns); Danielle Keats Citron, *Technological Due Process*, 85 Wash. U. L. Rev. 1249, 1256 (2008) (describing the Colorado Benefits Management System, which generates welfare eligibility decisions). *Cf.* John Golden, *Proliferating Patents and Patent Law’s “Cost Disease”*, 51 Hous. L. Rev. 455, 499-500 (2013) (considering that machine review of patent applications could produce “radical savings” but noting danger of uncorrected errors including as a result of user gaming of system algorithm).

An increasing variety of technologies support automated law. These include logical algorithms,<sup>14</sup> machine learning and other artificial intelligence techniques,<sup>15</sup> and computer network approaches such as blockchain, or distributed ledger, systems.<sup>16</sup> All of these technologies are fallible. Documented errors range from the mundane, such as a miscalculation of depreciation;<sup>17</sup> to the heartbreaking, such as an erroneous denial of food stamp benefits;<sup>18</sup> to the macroeconomic, such as a failure to correctly recognize risks to bank capital on the eve of the global financial crisis.<sup>19</sup>

There are several reasons for error. One is that humans design and build automated law, and people make mistakes. Another reason is that automated

---

<sup>14</sup> E.g. Maayan Perel & Niva Elkin-Koren, *Accountability in Algorithmic Copyright Enforcement*, 19 Stan. Tech. L. Rev. 472, 477, 488-91 (2016) (explaining that “platforms[] such as Google, Facebook, and Twitter ... appl[y] various algorithms to perform qualitative determinations, including the discretion-based assessments of copyright infringement and fair use” in order to respond to robot-generated takedown requests by copyright owners and suggesting that this results in over-enforcement of copyright rights).

<sup>15</sup> See Benjamin Alarie, Anthony Niblett and Albert Yoon, , *Using Machine Learning to Predict Outcomes in Tax Law* (October 16, 2016), available at <https://ssrn.com/abstract=2855977> or <http://dx.doi.org/10.2139/ssrn.2855977> (describing AI technique applied to database consisting of the text of hundreds of cases to give answer re: whether worker is an employee or an independent contractor for tax purposes).

<sup>16</sup> The use of distributed ledger or blockchain technology, which also supports the bitcoin currency, has been proposed for use by different computers in several legal capacities. For instance, computers in different jurisdictions might agree on the status of an import/export transaction. See Richard T. Ainsworth and Musaad Alwohaibi, *Blockchain, Bitcoin, and VAT in the GCC: The Missing Trader Example* (2017 working paper) (describing blockchain-based information confirmation system proposed in for new VAT system in Middle East Gulf Cooperation Council trading bloc). Blockchain technology might confirm and effect international payments. See Marcel T. Rosner & Andrew Kang, Note, *Understanding and Regulating Twenty-First Century Payment Systems: The Ripple Case Study*, 114 Mich. L. Rev. 649, 651 (2016) (suggesting that the Federal Reserve would have an interest in this regulatory solution). A proposed system based on blockchain has been built to reduce the cost of administering so-called know-your-customer regulations relevant to anti-money laundering and anti-tax evasion laws. See Jose Parra-Moyano & Omri Ross, *KYC Optimization Using Distributed Ledger Technology* (avail on SSRN 2017 paper). See generally Carla L. Reyes, *Conceptualizing Cryptolaw*, 96 Neb. L. Rev. \_\_ (2017).

<sup>17</sup> See Mock & Shurtz at 463 (describing TurboTax errors in 1994 and 1996). See also *Choe v. Commissioner*, T.C. Summ. Op. 2008-90, 2008 WL 2852249, at \*1 (July 24, 2008); Rev. Rul. 85-187 (each involving erroneous software depreciation calculations).

<sup>18</sup> Danielle Keats Citron, *Technological Due Process*, 85 Wash. U. L. Rev. 1249, 1256 (2008) (describing a state government automated law system that incorrectly denied benefits to eligible welfare recipient).

<sup>19</sup> Kenneth A. Bamberger, *Technologies of Compliance: Risk and Regulation in a Digital Age*, 88 Tex. L. Rev. 669, \_\_ (2010) (describing private automated law systems that failed to recognize risks to bank capital reported leading into global financial crisis).

law systems only have access to existing, or past-developed, information. The idea that technology can automatically determine whether a worker is an employee or an independent contractor, for instance, depends on the technology's access to a database of worker status determinations.<sup>20</sup> Historical data cannot reliably predict answers in some new situations. The issue of worker classification in the gig economy provides an example of a novel new set of facts.<sup>21</sup>

System designers also have an incentive to favor regulated parties who purchase and use their system. For instance, an automated law system may be intentionally designed to avoid law, or support aggressive reporting positions, or find loopholes. There is an incentive to “redesi[n]g behavior for legal advantage.”<sup>22</sup> Consider the tax preparation software feature that constantly updates a taxpayer on the status of his or her payment due or refund.<sup>23</sup> A more extreme example is the possible use of machine learning applied to primary sources of law to find and design new tax shelters.<sup>24</sup>

The tension between the architecture of computer systems and the goals of law or democracy has been explored before. It is at the core of Larry Lessig's work on cyberspace.<sup>25</sup> More specific shortcomings of particular centralized, automated systems have also been explored.<sup>26</sup> Often the

---

<sup>20</sup> See Anthony J. Casey & Anthony Niblett, *The Death of Rules and Standards*, [2015] draft at 3 (giving driving example to illustrate how technology might generate “microdirectives”).

<sup>21</sup> See, e.g., *O'Connor v. Uber Techs., Inc.*, 82 F. Supp. 3d 1133 (N.D. Cal. 2015) (allowing trial to proceed on classification issue); *Cotter v. Lyft, Inc.*, 60 F. Supp. 3d 1067 (N.D. Cal. 2015) (same). See generally Shuyi Oei & Diane Ring, *Can Sharing Be Taxed?* 93 Wash. U. L. Rev. 989 (2016) (considering regulatory issues presented by the sharing economy).

<sup>22</sup> Tim Wu, *When Code Isn't Law*, 89 Va. L. Rev. 679, 707-08 (2003). Wu describes a “code designer act[ing] like a tax lawyer ... look[ing] for loopholes or ambiguities in the operation of the law. *Id.* at 708.

<sup>23</sup> See Jay A. Soled & Kathleen DeLaney Thomas, *Regulating Tax Return Preparation*, 58 B.C. L. Rev. 151, 200-01 (2017) (recommending prohibition of the “prepayment-position status bar”).

<sup>24</sup> See Marcos Pertierra, Sarah Lawsky, Erik Hemberg and Una-May O'Reilly, *Towards Formalizing Statute Law as Default Logic through Automatic Semantic Parsing* (2017 working paper).

<sup>25</sup> See generally Lawrence Lessig, *Code and Other Laws of Cyberspace* (1999) (arguing that democratic mechanisms should oversee and edit the “architecture” of cyberspace).

<sup>26</sup> See, e.g., Kenneth A. Bamberger, *Technologies of Compliance: Risk and Regulation in a Digital Age*, 88 Tex. L. Rev. 669, 729-30 (2010) (recommending “dynamic model of regulation” to improve private automated law systems); Danielle Keats Citron, *Technological Due Process*, 85 Wash. U. L. Rev. 1249, 1256 (2008) (considering “a reconceived *Mathews* test [that] might permit hearings on flaws in [government] software,



recommended remedy is oversight, for instance in the form of a technology agency that regulates the content of automated systems.<sup>27</sup> Still, errors will exist.<sup>28</sup> Who should bear the responsibility for them?

### B. *If the System Were Directly Liable*

Recall the problem is that the errors caused by an automated law system have a tendency to fall neither on the user nor on the system, but rather on the public who is supposed to be protected by the legal scheme. This is the same as a problem raised by non-automated advisors who offer aggressive advice. But it is more susceptible of a solution, because of the centralized feature of an automated law system.<sup>29</sup> The goal of making the system directly liable for errors is to solve an underdetection problem and shift the burden of errors away from third parties and onto systems and users, since systems and users presumably both benefit from errors (e.g., sharing the benefit of a lower tax bill) and are well-positioned to avoid them.

The procedure could be as follows:

- IRS audits taxpayer who has submitted return prepared by tax software, much as under the current system.
- IRS also alerts tax software firm.
- Tax software firm and taxpayer may agree that tax software firm will assume responsibility for the controversy with the IRS, much like subrogation for an insurer who pays a claim. This is easier if tax software firm has all of the responsibility for an unpaid tax as “its” error. Causation is discussed in Part II.D, below.
- If the result of the controversy is that the taxpayer is liable for

---

and recommending that agencies test and allow public comment on automated law software”).

<sup>27</sup> See, e.g., Rory Van Loo, *Rise of the Digital Regulator*, 66 Duke L. J. 1267, 1327-28 (recommending oversight by an interdisciplinary “technology meta-agency”).

<sup>28</sup> See, e.g., Henry Smith, *Fusing the Equitable Function in Private Law* (forthcoming in *Private Law in the 21st Century* (Kit Barker, Karen Fairweather, and Ross Grantham eds.) [Harvard Public Law Working Paper No. 16-27](#) (arguing that errors are an inevitable feature of law and that equity can be understood as a mechanism to correct them).

<sup>29</sup> Other have recognized the legal design opportunity presented by centralized machine gatekeepers. Cf. Susan Klein & Crystal Flinn, *Social Media Compliance Programs and the War Against Terrorism*, 8 Harv. Nat’l Sec. L. J. 53, 57 (2017) (recommending “criminalizing the failure of social media programs to institute policies that discover [and report] terrorism-related posts”); Yesha Yadav, *The Failure of Liability in Modern Markets*, 102 Va. L. Rev. 1031, 1039-40 (considering strict liability and other regimes for harms generated by high-frequency algorithmic trading).

back taxes, interest and/or penalties as a result of an error attributable to the tax software firm (again, see Part II.D), the tax software firm pays the judgment.

- The tax software firm also pays an additional amount determined by a damages multiplier. See Part II.C below. The idea is that the tax software firm is settling not only this taxpayer's case, but also the liabilities of other taxpayers to the extent they arise from the same legal error.
- The decision would be final as to some group of filings prepared by the automated law system. Perhaps all filings with this particular issue in a particular tax year.

This approach fundamentally changes the way in which the law operates. No longer is the taxpayer-government relationship the central or key compliance relationship. Rather, the taxpayer agrees that the automated law system will have the primary relationship with the government. As a result, the system will be the decisionmaker in terms of what positions to take, which to defend when challenged, how to settle them and so forth. Controversy practice between the firms that offer software and the government becomes a primary avenue for the development of the law.

There are models for this approach in tax law.

- Tax preparer audits
- TEFRA for partnerships

Any decision that assigned liability to a taxpayer would support a damages award against an automated system that prepared the taxpayer's return. This would include an administrative decision, a decision in a trial or appellate court, or a settlement. An automated law system would have the right to directly defend against a government claim that its user committed a error. Signoff from the system would be required before the claim could be settled, and the system would have the right to appeal.<sup>30</sup> Damages paid by the system to the government (including the penalty multiplier) would be

---

<sup>30</sup> The automated law system, in other words, would have the right of subrogation. As the party solely liable for legal error, it could step into the shoes of the taxpayer to litigate the question of legal error in the taxpayer's case. One issue this raises is a possible conflict of interest between the taxpayer and the system with respect to whether liability proceeded from an error of fact or an error of law. As to this concern, see ... Another issue is whether the system may force the taxpayer to participate in a lawsuit. As to this concern, one solution is that the taxpayer might be able to block the

returned by the government if the system won an appeal.

One problem with this approach is taxpayer privacy and confidentiality. Should the tax software firm have the right to drag the taxpayer through protracted litigation at the firm's option? A procedural device might be developed to avoid a situation where the user is at the mercy of the firm in this way. Perhaps the user should only be responsible for verifying the facts he or she entered into the automated system. The individual user, at that user's option, might have the right to settle the fact questions and exit the controversy, allowing the suit to proceed on an anonymized basis and on stipulated facts.

Another issue is that the taxpayer might not be satisfied with the tax software's resolution of an issue with the government. For instance, what if an issue is in a process of transformative legal change? What if a tax software firm prepares a return that allows a registered domestic partnership in a community property state to income split,<sup>31</sup> and the IRS refuses to accept the position, and the tax software firm does not contest it? Under the proposal here, the RDP taxpayers cannot themselves contest the decision for that tax year after their tax software firm settles it. But they could decline to use this firm in other years, use another firm or self-prepare, and insist again on filing under an income-splitting position.

It is also possible that the government would not be satisfied with the resolution of an issue. Can it have a second chance to litigate an issue or pursue a similar issue through the IRS controversy process? Given the desirability of having several courts consider a problem, it would be wise to leave the door open. One way to do this is to hold the automated law firm liable not for all similar errors estimated over all returns ever filed, but rather for returns filed in the same accounting period. Another controversy might develop for another year.

A different issue raised by this proposal is that the penalty imposed on the system on account of one user's error is meant also to account for errors in other users' filings. This is the purpose of the damages multiplier. It is a significant departure from existing law to suggest that the resolution of a liability associated with Taxpayer A's filing would also resolve and finalize a liability associated with Taxpayer B's filing. Yet this is what this proposal would mean.

---

<sup>31</sup> Poe v Seaborn, Pat Cain's blog

Because of the damages multiplier, it would overcount the cost of error to again impose a penalty on a second user, like Taxpayer B, for the same error that has already been accounted for by multiplying the penalties for Taxpayer A's error. Thus neither Taxpayer B nor the system should be held liable for the same legal error. The question of what the "same" legal error is should tie back to the damages multiplier. In the case of tax software, for instance, the legal error might be present in the version of the software sold for a particular tax year. If the damages multiplier is accordingly set on an annual basis – for instance, by using the annual rate of audit for the particular year – then no further liability should be imposed for errors resulting from the same legal errors in the same tax year.

Is it acceptable for Taxpayer A's case to set the system's liability for errors in the returns of all Taxpayers B? Does this deprive Taxpayer B of an important opportunity to argue that her tax return was correct as submitted?

One response is that Taxpayer B has entered into a contract with the automated and/or centralized law provider. Under the contract, Taxpayer B allows the system to adjudicate legal errors using other taxpayers' facts. For instance, Taxpayer B has implicitly agreed that the cost of the software may increase because of a determination in Taxpayer A's case.

A concrete illustration is provided by the question of whether a taxpayer may deduct interest on a purchase money home mortgage up to a \$1.1 million principal balance, or whether the maximum principal amount is \$1 million. The statutory language is not clear.<sup>32</sup> An early case called *Pau* held that \$1 million was the limit.<sup>33</sup>

Under the proposal here, if a tax preparation system had prepared the *Pau* return, and had allowed deductions on a maximum principal amount of \$1.1 million, the system would be liable for the total estimated cost of those errors, not only in the taxpayer's case, but also in all cases. As a result, the cost of the software might increase to cover the penalties, and also future versions of the software would presumably feature the lower purchase money mortgage principal limit.

Faced with this response by the tax software firm, a taxpayer with a \$1.1 million purchase money mortgage has several choices. She can continue to use the software and accept the mortgage interest deduction haircut. But she need not do so. That is, she need not permanently sacrifice the right to

---

<sup>32</sup> Lawsby [others]

<sup>33</sup> *Pau*

directly defend her legal claim. She can switch to another method of tax preparation that takes the opposite view on the legal question. Or, she can self-prepare.<sup>34</sup> In this example, the taxpayer who deducted interest on a \$1.1 million purchase money mortgage would eventually have prevailed, as the government changed its view several years after *Pau*.<sup>35</sup>

### C. What is the “Cost” of a Mistake?

The full cost of the negative externality will not be assigned to the maker of an automated law system unless the approach solves the problem of underdetection. The appropriate tool is a damages multiplier. Penalties should increase according to a damages multiplier designed to account for the error costs incurred across the system, not just for the user whose specific case is discovered.

A damages multiplier is a well-known tool suggested by literature including Gary Becker’s foundational economic model of crime.<sup>36</sup> Say a person decides whether to comply by comparing the cost of compliance (“ $c$ ”) with the cost of noncompliance (“ $nc$ ”) multiplied by the probability of detection and liability (“ $p$ ”). The person considers whether  $c < nc * p$ . Compliance will be the attractive answer only if  $nc$  (in other words, the penalty in the case of noncompliance) is greater than  $c$  by a factor of more than  $1/p$ . In other words, the damages multiplier should be more than  $1/p$ .

There are a number of issues with damages multipliers. But these issues

---

<sup>34</sup> The possibility of a user overriding the system could also be added here. This raises the question of whether software would attempt to shift liability to users by providing them with choices that required them to reach legal decisions. If the software in fact requires users to make such decisions, the system might avoid liability. But it would also sacrifice a central selling point, which is removing the burden of understanding the law from the shoulders of its users. Efforts on the part of an automated law system to have the best of both worlds, for instance through fine print that purports to leave the user with the legal decision but in fact does not engage the user in the decision, should be treated as legal decisions made by the system. There is a parallel with some courts’ refusal to enforce arbitration provisions buried too deeply in the fine print of an online contract. To determine whether the software or the user makes a relevant legal decision, a court might ask “whether the circumstances support the assumption” that the purchaser made an independent legal decision rather than accepting the system’s default answer. *Cf. Sgouros v. TransUnion Corp.*, 817 F.3d 1029, 1034-35 (7<sup>th</sup> Cir. 2016) (asking in the context of online contract terms “whether the circumstances support the assumption that the purchaser receives reasonable notice of [the terms and conditions of the agreement]”).

<sup>35</sup> See Rev. Rul. 2010-25, 2010-44 I.R.B. 571.

<sup>36</sup> See Gary Becker, *Crime and Punishment: An Economic Approach*, 76 J. Pol. & Econ 169 (1968); see, e.g., Michael G. Allingham & Agnar Sandmo, *Income Tax Evasion: A Theoretical Analysis*, 1 J. Pub. Econ. 323 (1972).

are less problematic for automated law systems. One challenge is that political and rule of law proportionality constraints limit the ability to vastly increase penalties imposed on a single person based on the idea that her transgression was difficult to detect.<sup>37</sup> A second consideration is that a fixed damages multiplier across different offenses fails to account for the variation in probability of detection and in particular for the likelihood that more serious offenses are more likely to be detected.<sup>38</sup> A third issue is that factors other than the cost of compliance influence the magnitude of penalties. These include aggressiveness, culpability and intent.<sup>39</sup> They also include whether the defendant has deep enough pockets to pay the larger penalty.

A damages multiplier for automated law system can sidestep each of these issues.

First, the imposition of the penalty on the centralized system, not the individual violation, reframes the issue of ensuring that the punishment fits the crime. The idea is that the centralized system itself has the responsibility to correctly state the law or pay appropriate damages. The individual user's penalty is only the starting point for measuring the system's total error.

The idea of a damages multiplier based on other users of the system means a fundamentally different method for adjudicating claims of legal error in compliance systems. Now the litigants, effectively, are the firm that makes the automated system and the U.S. government. In addition, this case is no longer just about the audited taxpayer. Instead it is a general test case that automatically will determine liability for dozens or hundreds or thousands of tax returns, not just the taxpayer's.

Second, the damages multiplier can be customized. As a starting point, the damages multiplier, calculated as  $1/p$ , might be based on  $p$  equal to a typical audit rate. But an automated law system could prove out of the high damages by presenting its own data to rebut the calculation of the penalty

---

<sup>37</sup> See, e.g., Michael J. Graetz & Louis L. Wilde, *The Economics of Tax Compliance: Fact and Fantasy*, 38 Nat'l Tax J. 355, 358 (1985) ("That an economic model analyzing the expected utility calculation of a would-be tax evader recommends large increases in the applicable sanction in light of the very low probability of its application quickly becomes irrelevant as a policy matter. In this country, at least, legal, moral and political constraints make this necessarily so. Coherence in our criminal law generally demands that 'punishment fit the crime'....").

<sup>38</sup> See Richard Craswell, *Deterrence and Damages: The Multiplier Principle and its Alternatives*, 97 Mich. L. Rev. 2185, 2192 (1999).

<sup>39</sup> See, e.g., Alex Raskolnikov, *Six Degrees of Graduation: Law and Economics of Variable Sanctions*, 43 Fla. State L. Rev. 1015 (2016).

multiplier.

Customization of the damages multiplier is available in this case as for relatively few others,<sup>40</sup> because the information about all similar cases should be within the reach of the automated law provider. Different customized factors might be taken into account in particular cases. As an example, if an issue splits circuits, thus raising the *Golsen*<sup>41</sup> rule, a tax software provider should be allowed to argue for damages to be based only on the tax returns filed in such-and-such circuit.

Third, the penalty itself (aside from the multiplier) imposed on automated law systems could be set without reference to aggressiveness or culpability.<sup>42</sup> The automated law liability idea does not mean to use damages as a message that the system wronged or hurt someone. It is not meant to act as a corrective justice tool. It is more like a “public mechanism of accident regulation.”<sup>43</sup> Since the goal of the liability regime is to force automated law systems to internalize the costs of legal error, it should be sufficient to set the penalties equal to the cost of legal error without, for instance, an upward adjustment for culpability. Admittedly, this is easier to figure for some automated law systems as opposed to others. The cost of underpaid taxes equals the tax shortfall.<sup>44</sup> In contrast, the cost of environmental noncompliance may be more difficult to calculate.

The idea of a damages multiplier that a system can rebut with adequate evidence of errors on other filings raises another problem of confidentiality

---

<sup>40</sup> One example of a torts case in which a damages multiplier may have been customized is a case in which the Seventh Circuit upheld an award of punitive damages against a defendant who operated a bedbug-infested 191-room hotel. Two hotel guests sued, and the total damages award was \$10,000 in compensatory damages plus \$372,000 in punitive damages -- \$2000 for every room of the hotel. *Mathias v. Accor Economy Lodging, Inc.*, 347 F.3d 672, 678 (7<sup>th</sup> Cir. 2003).

<sup>41</sup> The *Golsen* rule provides that the Tax Court follows the law in a taxpayer’s circuit of residence. *See Golsen v. Commissioner*, 54 T.C. 713 (1957). The proper damages multiplier in a circuit split situation might be designed to calculate the total cost of the legal error for all tax returns filed for residents in the circuit that gave the pro-government answer. The automated law system could bear the burden of supplying the information necessary to determine its users’ residence.

<sup>42</sup> violations corrective justice

<sup>43</sup> Alex Stein, *The Domain of Torts* 117 Colum. L. Rev. 535, 594 (2017). A similar idea is of “licensing-based liability,” distinct from “liability imposed on the basis of wrongdoing.” *See* John C.P. Goldberg & Benjamin C. Zipursky, *The Strict Liability in Fault and the Fault in Strict Liability*, 85 Fordham L. Rev. 743, 745 (2016). The proposal of automated law liability here stretches beyond the domains of inherently dangerous activities and the like in which common law tort imposes licensing-based liability. *See id.* at 784.

<sup>44</sup> Although the appropriate discount rate might be controversial. [Check: Remedies.]

and privacy. This issue relates to the confidentiality and privacy of the other taxpayers – not those that have been directly audited, but whose information is relevant to the calculation of total damages. At least initially, the solution proposed here allocates that problem to the automated law system. It is well-positioned to determine the extent to which it should require its users to participate in a test case damages calculation (in which case the users would presumably have to agree to this involvement in a contract with the company) and the extent to which it might, for instance, attempt to prove lower damages using anonymized data or statistical techniques.

#### *D. When Does an Automated Law System Cause an Error?*

Automated law systems are involved in at least three different kinds of errors: clear errors of law, unclear errors of law, and mixed errors of fact and law. The systems can be said in some sense to cause each of these kinds of errors. Bringing all of these kinds of errors into the direct liability system proposed here would maximize the ability of automated law liability to price, surface and debate questions of law.

To illustrate the error types, consider a new example of automated law -- timekeeping software products.<sup>45</sup> These are built to comply with the Fair Labor Standards Act (FLSA), which requires employers to “make, keep and preserve” wage and hour records.<sup>46</sup> They are used to keep records for [a large fraction of] the [50 million] hourly workers in the United States.

Both errors of law and errors of fact may occur in timekeeping software systems. For instance, the software may prompt employers to enter scheduled break and/or meal times for employees, and then automatically deduct that time from paid time. This connects with a legal error if some state laws do not allow break and/or meal times to be deducted from paid time. It connects with a factual error if an employer enters the wrong information. An employer’s incorrect data entry usually would seem to be the employer’s fault, not the system’s fault. But even in this case one can find mixed questions of fact and law. What if the system makes it very hard to change entered time if it turns out that an employee works through a break? It is conceivable that this could be cast as an error of law, perhaps as a legal error

---

<sup>45</sup> The descriptions of errors and features of electronic timekeeping systems is based on a qualitative empirical examination of thirteen such systems. See Elizabeth Tippet, Charlotte S. Alexander and Zev J. Eigen, *When Timekeeping Software Undermines Compliance*, 19 Yale J. L. & Tech. 1 (2017).

<sup>46</sup> 29 USC § 211(c).



because the design of the system so strongly suggests that scheduled time worked, not actual time worked, is the relevant input.

When errors are errors of law, the mistakes may or may not be clear. Let us assume that state law unambiguously states that break and/or meal times count toward paid time. The software's mistake on this front would be a clear legal error. But timekeeping software also interacts with grey areas of law. Consider the software's interpretation of a time rounding rule. FLSA regulations accept the practice of rounding "starting and stopping time ... to the nearest quarter of an hour" so long as it does not cause "a failure to compensate the employees properly for all the time they have actually worked."<sup>47</sup> Timekeeping software apparently implements this guidance with a default setting that rounds time to the hour if a punch-in or punch-out time is within seven minutes of an hour.<sup>48</sup> But if employer rules effectively prevent tardiness, so that employees are sometimes early, but never late, then the software's rounding default may systematically reduce the time recorded for an employee. In this case, the software's default rounding rule encourages an employer to take an aggressive, but not clearly illegal, filing position.

The errors most obviously attributable to the system are clear errors of law. If the system gets state law wrong, that is on the system.

Somewhat more difficult are errors of law that are not clear. In the parlance of the tax law, these might be positions as to which there is a substantial authority "reporting position," so that a filer would not face "substantial understatement" penalties as a result of the filing. The position may be an incorrect statement of the law, though this is not obvious before it is tested in an audit and perhaps a litigation process.

The liability of systems for legal errors should be strict. That is, liability should not be limited to a negligent clear error of law, like the failure to research wage and hour law in a particular state. Instead, it should include liability for the close case that happens to come out in favor of the government and to the detriment of all the taxpayers who took the position. If a court invalidates the practice of rounding time according to the 7-minute rule, the automated law system should bear that liability even though it was not clear when the return was filed that the 7-minute rounding rule was illegal.

---

<sup>47</sup> 29 C.F.R. § 785.48.

<sup>48</sup> See Tippet, Alexander & Eigen at 37 ("A common unit of rounding appears to be seven minutes.")

The reason for strict liability goes to the heart of the proposed direct liability idea for automated law systems. The idea is that legal questions will be priced, and then debated and decided. The interesting questions, those in need of development, are the close ones. This centralized mechanism of discovering and discussing these questions will be of little use unless it covers the matters that will fuel the development of the law. Also, the system controls and makes these legal decisions as much as it makes the decisions that involve clearer legal error. It is still the least cost avoider.

A final question is whether systems should be liable for mixed questions of law and fact. This is the issue that demarcates when a user has to be involved in the controversy over an allegedly incorrect filing. It raises the question of whether the system is liable for design choices that nudge a user to present facts in a certain way. An example from wage and hour software involves scheduled breaks. Assume that a software system encourages an employer to enter breaks that are scheduled, and breaks are sometimes not taken in full. Assume further that the software system fails to prompt to confirm breaks were taken as scheduled, and/or the system does not permit for easy correction of the data if an employee works through a break.

The system should also have direct liability for errors arising from mixed questions of law and fact. The capacity of software programs to manipulate or influence human users' responses through their design is well established.<sup>49</sup> If a system's nudge influences a data input, the centralized system's design is at least partly responsible for the data input. Including the mixed question of fact and law in the direct liability space empowers the law might encourage the system to stop nudging users toward noncompliance by suggesting an inappropriate legal framework for the relevant facts.

In the presence of a competitive market for different automated law products, automated law presents a classic case for strict liability.<sup>50</sup> The automated law systems considered here cover matters of public regulation. Moreover, the users of the system are likely to accept the answers given by the system and move on with the filing. There may be aggressive position-taking in tax filing, but it is not the kind of harm done one person by another that motivates a wrongs-based system of liability. Automated law liability does not mean to achieve corrective justice. It means to properly regulate.

---

<sup>49</sup> DeLaney Thomas / Soled.

<sup>50</sup> See, e.g., Steven Shavell, *Strict Liability Versus Negligence*, [2] J. Leg. Stud. 1, 3 (198\_) (explaining that strict liability is appropriate for cases of "accidents between sellers and strangers" because if sellers are forced to pay for harm to strangers, market forces will adjust the prices charged to customers until the outcome is efficient").

Success means a bureaucratic exercise that shifts costs of error until they fall on the right party. This is consistent with strict liability.<sup>51</sup>

Even if strict liability is theoretically correct, it conflicts with existing law. In particular, the idea of strictly attributing errors that show up on user filings to the system that prepared the filings contradicts existing law applicable to lawyers' responsibility for positions taken on behalf of clients. The starting point for the liability of lawyers for mistakes is malpractice and ethics laws. In a litigation setting, these allow lawyers to ethically take any position [that is not frivolous]. A lawyer who takes an aggressive position in court and loses is not liable to anyone because she turns out to have been wrong.

In the compliance space, lawyers often have less leeway. Under applicable tax rules, for instance, lawyers and other tax preparers may be liable for penalties and under the ethical rules that govern tax practice before the IRS if they advance a position that lacks "substantial authority."<sup>52</sup> "Substantial authority" means weaker than "more likely than not" to succeed, but certainly means a stronger position than "not frivolous." Existing law recognizes that the preparation of a tax return will operate in an underenforcement, underdetection regulatory environment. It is appropriate to charge a tax preparer with a higher standard of legal certainty than a litigator because the chance of a tax return audit is so small. The tax preparer is required to shift away from the position of advocate and toward the position of officer of the court, or in this case officer of the agency.

The analogy to tax preparer standards is a parallel in existing law that may help explain the liability of automated law systems for negligent errors.<sup>53</sup> But strict liability is proposed here, both for errors of law and for mixed questions of law and fact. What else might justify strict liability for such errors made by an automated law system?

Another key is that even though a user and a system are in a contractual

---

<sup>51</sup> See, e.g., John C.P. Goldberg & Benjamin C. Zipursky, *The Strict Liability in Fault and the Fault in Strict Liability*, 85 *Fordham L. Rev.* 743, 745 (2016) (contrasting licensing-based liability and wrongs-based liability); Alex Stein, *The Domain of Torts* 117 *Colum. L. Rev.* 535, 594 (2017) (contrasting "public mechanism of accident regulation" and wrongs-based torts laws).

<sup>52</sup> 6694

<sup>53</sup> Need to work out: If tax preparer law remains as is for TPs that do not use automated law systems like TurboTax, is the effect of this to drive taxpayers back into private preparers, so no effect of automated law liability but once again way too expensive to prepare? Is a workaround a different idea for tax preparer law, e.g. if an automated law system, not a preparer? Should Big 4 software count as automated law?

relationship with each other, they are not in a lawyer-client relationship. Just as a legal self-help book does not create an attorney-client relationship, so too the use of a software program does not create such a relationship. The relationship is more similar to the relationship between a withholding agent and a payee. Automated law systems are more like the banks that determine the character and amount of income and basis, and like the employers who identify and quantify compensation. These banks and employees have strict liability for errors for certain failures to withhold. Likewise, the idea of this automated law liability proposal here is to cause automated law systems to balance, on one hand, the importance of compliance and, on the other hand, the benefit of different reporting positions to their clients.

The core of the idea presented here is to engage automated law systems and the government in a dialogue about what the law should be, and to ensure that if the law is not as first reported, the costs of the error fall on the right shoulders, which is to say on the users and the system rather than on the public in general. This analysis applies equally for clear errors and close calls. Another way to look at this is that someone needs to bear the burden for close-call errors of law. As between the system and user on one hand and the public on the other, the right party is the contracting unit of the system and the user. As between the system, which makes the decision, and the user, who accepts the system's decision, the right party is the system.

### III. HOW AUTOMATED LAW LIABILITY MIGHT PLAY OUT

#### *A. Market Differentiation*

A competitive market for different automated law products provides the best-case scenario for the imposition of automated law liability. Under current law, as described above in Part I, problems of underdetection and underenforcement mean that the government rarely audits taxpayers. Many close-call issues are never raised. If automated law liability appeared in a competitive marketplace for automated law products, close-call issues would be in effect debated in the market, through the pricing of different products.

Assume two tax software products. One, TaxDragon, takes very aggressive positions. Another, CleanTax, takes conservative positions. TaxDragon will face a higher likelihood of liability for the errors made on returns it prepares. It will cost more to insure those errors. A TaxDragon customer will receive the benefit of a lower tax bill because of the aggressive positions, but will have to pay more for the product because of the high cost of insuring against the prospect of liability for the automated law system. In

contrast, a CleanTax customer will have a higher tax bill, but the product will cost less because of the lower cost of insuring against automated law liability.<sup>54</sup>

An additional twist is that CleanTax might be likely to check with the government about the validity of its product. This may mean that it would be easier for the government to discover errors in the CleanTax software, because of its familiarity with CleanTax. If so, then the damages multiplier for CleanTax ought to be lower than the damages multiplier for TaxDragon, because the damages multiplier is supposed to be inversely related to the probability of detection.

### *B. Insurance and Reinsurance*

- Competitive market requires assumption that neither firm is judgment proof.
- Deep pockets to absorb the burden of penalty increased by a damages multiplier are another reason why a damages multiplier can be imposed on a larger actor but not as effectively on an individual.<sup>55</sup>
- Regulating agency should require evidence of the system's creditworthiness before allowing the automated law system to prepare compliance submissions or other legal determinations or filings.
- Reinsurance schemes might emerge to support the good credit of automated law systems.<sup>56</sup>
- A TBTF<sup>57</sup> problem may arise.

### *C. Push Automated Law Into Government Hands*

The idea that a competitive market among automated law products will allow the pricing of error risk relies on the idea that automated law providers will respond to the incentive to provide regulated parties with aggressive,

---

<sup>54</sup> There could also be differentiation within a software product if an automated law provider charged different amounts of insurance based on different positions. An "audit insurance cost" bar, like the "refund due" bar might show a taxpayer how a decrease in tax liability related to an increase in audit insurance cost.

<sup>55</sup> *Cf.* Guido Calabresi, *The Costs of Accidents: A Legal and Economic Analysis* 50-54 (1970) (identifying deep pockets as a possible reason supporting enterprise liability).

<sup>56</sup> This centralization of liability and insurance is the opposite of the prediction of peer-to-peer insurance and "radical financial disintermediation" suggested elsewhere. *See* Michael Abramowicz, *Cryptoinsurance*, 50 *Wake Forest L. Rev.* 671, 673 (2015).

<sup>57</sup> Too big to fail.

cost-saving answers. The best result under this system is one in which there are different choices available and some offer more aggressive positions than others. Then, the market would participate in a helpful dialogue about the content of the law. The risk and total cost of error would be diversified across many returns under the umbrella of an automated law system, and then priced in. This is the picture of market differentiation presented in Part III.A.

There are reasons why such a competitive market might not exist. One possibility is that the market might be occupied by one dominant player. Or, there might be collusion among players. Or, regulated parties might not be able to see and process information about the offsetting costs and benefits of, for instance, a lower tax bill but a higher price due to a higher cost of insurance.

One possible outcome is that automated law systems might lean too strongly in favor of the government, because this approach reduces their risk. If a system can charge the same amount to a taxpayer whether or not it produces pro-taxpayer positions in grey areas, then under an automated law liability regime it will avoid these pro-taxpayer positions, because they cost more to insure. Alternatively, the automated law system might not be created to begin with, because of concerns about the cost of automated law liability.<sup>58</sup>

Yet it is not the goal of this proposal to stamp out pro-taxpayer reporting positions. The goal is to encourage the risk of these positions to be priced and their costs to be properly allocated. The goal is to debate and resolve controversies, not to eliminate them by resolving every close question in favor of the government.

If automated law liability encourages some automated law providers to tilt in favor of the government, another problem will arise which is the converse of the problem that this paper has so far focused on. So far, this paper has focused on the problem that errors in favor of regulated parties should be corrected so that they do not fall on the general public. If an automated law system is too pro-government, then the problem is the reverse. The system will cause undetected errors that should be corrected at the cost of the general public, but instead remain on the shoulders of, and burden, the taxpayers or other regulated parties who use the system. How can pro-government errors made by automated law systems be detected and corrected? Government enforcement is a poor answer in this case. Instead, the solutions may be drawn from the ideas outlined in Part IV, regarding

---

<sup>58</sup> Note argument that movement from traditional strict-liability trespass to negligence was motivated in part by goal of fostering economic development. [Horwitz (1977)]

liability for government automated law.

#### IV. GOVERNMENT AUTOMATED LAW SYSTEMS

- Example: CO welfare determination system. Some errors blocked welfare recipients from receiving benefits to which they were entitled.<sup>59</sup>
- Whether government or private, errors arise
- When a government system provides automated law (e.g.: welfare determinations) the likely problem is the inverse of the one investigated in this paper. It involves a decision that is too pro-government, that denies a regulated party some right or benefit.<sup>60</sup>
- But a similar solution: External mechanism to find and correct will improve quality of law.
- For instance, an improper denial of welfare benefits hurts the user of the regulatory scheme – the applicant – and helps the general public – because less is spent on welfare. The right answer would transfer value from the general public to the welfare applicant.
- Third party AG, collective action, third party standing, damages multiplier might help solve.<sup>61</sup>
- Perhaps some programs are combo government and taxpayer. For these, both error correction mechanisms might deserve consideration.
- But transaction costs will quickly climb. Is it worth it?
- Thought experiment reveals: Private automated law, government enforcement may be the best combination. The idea of automated law liability solves an age-old dilemma for regulators, brings market forces to bear on the question of pricing risk, and transforms controversy practice in a way likely to raise and resolve more issues. But its soft spot is the possibility that the competitive market will not be robust enough. In this case, the government is left to run a system, or the private automated law system may become quasi-governmental. This raises the inverse set of issues.

---

<sup>59</sup> Citron

<sup>60</sup> *Cf. e.g.*, Joshua D. Blank & Leigh Osofsky, *Simplicity: Plain Language and the Tax Law*, 66 *Emory L.J.* 189 (2017) (describing IRS guidance that outlines conservative or safe harbor guidance).

<sup>61</sup> Much here to cite, e.g. Ventry, others. The availability of private enforcement of public rights varies by subject matter and over time. One view is that it has been systematically cut back over half-century between 1964 and 2014. *See* Stephen B. Burbank & Sean Farhang, *Rights and Retrenchment* (2017).

## CONCLUSION

Automated law liability will become more and more important as machines make more and more legal decisions. Implementation should proceed with caution. But some version of liability for centralized legal decisionmakers seems a necessary and correct development from a regulators' perspective if automated law continues to grow and occupy the different fields of compliance.