

## How to Interpret a Vending Machine: Smart Contracts and Contract Law

Gregory Klass

### Abstract

A smart contract is software designed to do the job of a legal contract: ensuring the performance of parties who might not otherwise trust one another to do so. By running a smart contract on blockchain, users can lock themselves into future performances without relying on a third-party enforcer or platform host, thereby realizing a “fully trustless” exchange. This new technology has wide range of potential applications, and contracts are likely to become an increasingly common part of the economy.

Some have argued that smart contracts represent a new type of legal contract, analogizing the software’s code to a contractual writing. Others have suggested that smart contracts might in some transactions replace legal contracts. But the relationship between smart contracts and legal contracts is more complex than either claim acknowledges. Although using a smart contract can figure into the formation of a legal contract, it is a mistake to analogize a smart contract’s code to a contractual writing. Unlike writings, code should rarely figure into the interpretation of a legal contract; and when it does, its interpretation is different in kind. And though some have tried to use incorporation, integration or TINALEA clauses to substitute smart contracts for traditional contractual protections, it is not clear that courts would or should always enforce such provisions. And even if they do, governance by code rather than law comes at a significant cost to the parties. Smart contracts work better when they supplement, rather than supplant, legal contracts.

This article analyzes the many ways a smart contract might interact with the law of contracts by taking seriously the comparison, common in the literature, of smart contracts to vending machines. A series of thought experiments is used to explore when and how the mechanisms inside a machine, analog or digital, can affect the terms of a legal contract between its users. The resulting detailed doctrinal analysis provides support for a broader thesis about the relationship between technology, law and society. Smart contracts, though useful tools, instantiate an anemic form of human sociability as compared to the complex, even trusting, relationships for which contract law is designed.

# How to Interpret a Vending Machine: Smart Contracts and Contract Law

Gregory Klass\*

February 10, 2022

*Readwrap: By reading this draft you agree not to cite or quote it. ☺*

## Table of Contents

Introduction .....	1
1. Preliminary Remarks .....	8
1.1 Smart Contracts.....	8
1.2 Smart Contracts and Contractual Writings.....	12
2. The Standard Vending Machine and Black-Box Smart Contracts .....	15
2.1 The Standard Vending Machine .....	16
2.2 Black-Box Smart Contracts.....	20
3. The Mutually Constructed Vending Machine and Mutually Programmed Smart Contracts.....	22
3.1 The Mutually Constructed Vending Machine .....	22
3.2 Mutually Programmed Smart Contracts .....	25
4. The Glass-box Vending Machine and Glass-Box Smart Contracts of Adhesion.....	31
4.1 The Glass-Box Vending Machine .....	31
4.2 Glass-Box Smart Contracts of Adhesion .....	36
5. Smart Contracts and Legal Contracts .....	45
5.1 The Parable of The DAO.....	47
5.2 Prepackaged Coordination vs. Shared Cooperative Activities: The Relational Aspects of Many Contractual Transactions .....	52
5.3 What Legal Contracts Do .....	58
Conclusion.....	61

## Introduction

Developments to law of contract have often been driven by new methods of contracting. Early in the twentieth century, for example, businesses began using preprinted forms for their purchase orders, confirmations, invoices, packing slips, and other documents, on which they included contract terms. That practice generated so-called battles of the

---

\* Georgetown University Law Center. I am grateful to Julie Cohen, James Grimmelmann, David Hoffmann, Paul Ohm, Jedidiah Purdy and Madhavi Sunder for comments on draft of this article or thoughts that went into it. Hannah Beiderwieden and Ceara Maria Burns provided excellent research assistance.

forms, which gave rise to section 2-207 of the Uniform Commercial Code. Around the same time, companies selling to consumers began including contract terms the backs of receipts, on parking stubs, and on cruise ship tickets, and courts had to devise rules to determine the effectiveness of these new techniques. Later, businesses began providing their customers access to complete terms only after the sale, inside a shrink-wrapped box or arriving in the mail with products ordered over the phone. Again courts had to figure out how to handle this new form of contractual writing. In the 1990s, the Uniform Law Commission and legislatures responded to the new computer technologies with laws establishing the legal equivalence of electronic records and writings.<sup>1</sup> More recently, clickwrap and browsewrap have come to dominate online consumer contracts. Courts and commentators are still working out the right rules for these technologies of contracting.

The next chapter in this history is likely to involve smart contracts. A smart contract is software, perhaps run on blockchain and designed to execute future exchanges or other coordinated actions between persons who might otherwise not trust one another to perform. Recent advancements in blockchain technology have led to an explosion of commercial interest in smart contracts. Blockchain computing platforms like Ethereum now allow users to lock themselves into smart contracts without using third-party platforms, thereby enabling what some characterize as “trustless” transactions.<sup>2</sup> Investment vehicles built on such smart contracts have attracted hundreds of millions of dollars in cryptocurrency.<sup>3</sup> And many are predicting more concrete applications in areas like insurance, financial derivatives, consumer protection, corporate governance, tax filing, voting, supply chain management, bankruptcy, property rights, and repossession.<sup>4</sup> An auto loan company, for example, might use a smart contract to automatically lock the borrower out of the car if payment is not received. An insurer might employ a smart contract to make automatic payments to farmers based on temperature and rainfall data. Or an avocado wholesaler and a trucking company might install vibration sensors in the trucks connected to a smart contract that adjusts payments up or down based on how bumpy the ride was.

Smart contracts interact in complex ways with legal contracts. Enthusiasts have suggested that smart contracts might for some applications

---

<sup>1</sup> Uniform Electronic Transactions Act; Electronic Signatures in Global and National Commerce Act.

<sup>2</sup> Blockchain technology is not central to the issues this article explores. Those seeking an accessible introduction to the technology might begin with Kevin Werbach & Nicolas Cornell, *Contracts Ex Machina*, 67 Duke L. J. 314, 319-38 (2017).

<sup>3</sup> See, e.g., the discussion of The DAO in Part Five.

<sup>4</sup> Shaanan Cohny & David Hoffman, *Transactional Scripts in Contract Stacks*, 105 Minn. L. Rev. 319, 321-22 (2020) (listing proposed uses and sources).

eventually replace legal contracts. When and how that might happen raises an interesting set of questions. More likely is that smart contracts will be used in tandem with legal contracts. Before implementing a smart contract, two businesses might enter into a legal contract governing its use. Or the use of the smart contract itself might generate a legal contract. Here the legal questions will involve whether and how the use of a smart contract might affect the terms of a legal contract.

In the past few years, scholars have begun exploring the legal implications of smart contracts.<sup>5</sup> But these are still early days. One finds in the literature disagreements on such a basic question as whether smart contracts constitute promises or agreements, whether use of a smart contract always produces a legal contract, and whether the code of a smart contract is comparable to a contractual writing.

This article is the first to systematically explore the relationship between smart contracts and legal contracts. It argues that although smart contracts are neither contractual writings nor legal contracts, they share important features with each. First, a legal contract and a smart contract can each serve as a tool to secure future performance in the absence of trust. This is why some predict that the new technology will for some applications come to replace the old. Whereas today an avocado wholesaler and shipper might enter into a contractual agreement to adjust the price based on damage to the produce, tomorrow they might lock themselves into a smart contract.

Second, both contractual writings and the code in which smart contracts are written have a lexical structure. Each can be read. And one finds a similar conditional logic in each. The if-then structure of the imagined smart contract for avocado shipments (if the ride is bumpy, then the price is reduced) is similar to the if-then structure of a contractual writing designed to do the same job (“If the fruit is bruised, then the price is reduced”). Smart contracts can *look like* contractual writings.

This second similarity has suggested to some that smart contract code might figure into the construction of legal contracts in the way contractual writings do. This article examines this idea, both in light of contract law that exists today and with an eye to how that law might evolve. The doctrinal question is threefold. First, when and how might the use of a smart contract generate a legal contract? Second, when if ever should the contents of a smart contract—especially the operational code, but also programmer comments—figure into determining the content of an associated legal contract? Third, when code is relevant to the construction

---

<sup>5</sup> See the sources cited *infra* notes 18-22. See also Zvonimir Slakoper & Ian Tot, *Digital Technologies and the Law of Obligations* (2019); Larry DiMatteo, Michael Cannarsa & Cristina Poncibò, *The Cambridge Handbook of Smart Contracts, Blockchain Technology and Digital Platforms* (2020); Jason Allen & Peter Hunn, *Smart Legal Contracts: Computable Law in Theory and Practice* (forthcoming).

of a legal contract, how should courts interpret the code when determining the parties' legal relationship?

This article answers these questions using a series of thought experiments. The term "smart contract" first appeared in Nick Szabo's 1997 article, "Smart Contracts: Building Blocks for Digital Markets." Szabo suggests that "[a] canonical real-life example, which we might consider to be the primitive ancestor of smart contracts, is the humble vending machine."<sup>6</sup> A vending machine functions like a smart contract in that it enables transactions—exchanges of money for a product—between strangers who do not trust one another. Unlike a smart contract, however, a vending machine need not employ computer code, and its internal mechanisms might not have an obvious lexical structure. Several imaginative variations on the familiar vending machine serve as thought experiments to explore when and how the internal structure of a machine might figure into the terms of a legal contract between those who use it.

Like the use of a vending machine, the use of a smart contract can express one or both users' agreement to a transaction, thereby giving rise to a legal contract. In the large majority of smart-contract transactions, that agreement will also be expressed in words—the most common medium for contractual agreements. Those transactions pose no special formation issues. It is also possible to imagine parties expressing their agreement solely through the use of a smart contract, without an accompanying verbal agreement—on the model of a vending-machine transaction. Any legal contract that emerged from such a transaction would be an implied-in-fact-contract. Although the construction of implied-in-fact contracts poses some special challenges, smart contracts do not generate puzzles that are different in kind from the construction of other legal contracts.

Scholars have recently begun to focus on how a smart contract's code might figure into the interpretation of a legal contract. But in what will probably be the vast majority of smart-contract transactions, courts should *not* look to the code when determining parties' contractual obligations. This is most obviously the case when one or both parties do not have access to the code—in a "black-box smart contract." As the use of smart contracts expands, we are likely to see more and more black-box smart contracts. But even when both users have access to the code—when they are using a "glass-box smart contract"—it is not obvious that the code should figure into the construction of any legal contracts between them. This article argues that it should in only two, relatively narrow categories of transactions.

The first is when both users take a hand in the coding, producing what I call a "mutually programmed smart contract." In these instances, code is best analogies to course of performance evidence. Such mutually

---

<sup>6</sup> Nick Szabo, *Smart Contracts: Building Blocks for Digital Markets*, Extropy (1996), available at: [http://www.fon.hum.uva.nl/rob/Courses/InformationInSpeech/CDROM/Literature/LOTwinterschool2006/szabo.best.vwh.net/smart\\_contracts\\_2.html](http://www.fon.hum.uva.nl/rob/Courses/InformationInSpeech/CDROM/Literature/LOTwinterschool2006/szabo.best.vwh.net/smart_contracts_2.html).

programmed smart contracts, while interesting to think about, are expensive to produce and will be rare.

More common will be transactions in which one user programs a smart contract invites others to interact with it, or in which two users purchase smart contract software from a third-party vendor. I term these “smart contracts of adhesion.” When adhesive smart contracts are provided to consumers or other unsophisticated parties, the code should not figure into the interpretation of the associated legal contract, even if the business makes the code available to users, and even if it stipulates that the code is part of their agreement. Several scholars have suggested that between sophisticated parties, the code of a glass-box smart contract should automatically be treated as interpretive evidence of their legal contract. This article argues against such a rule. When sophisticated parties use an adhesive smart contract, the default should be that the code is not evidence of their agreement—even if all parties have access to it. Sophisticates should have the power to incorporate code into their agreement. But courts should be wary of attempts to integrate code against natural language evidence of the parties’ agreement.

The final doctrinal question concerns how courts should interpret code. Because the operational code of a smart contract is different in kind from a contractual writing, one should not assume that interpretive methods that apply to the latter apply *pari passu* to the former. When a court interprets a contractual writing, it seeks out the writing’s communicative content—what its author or authors objectively intended to convey to a reader. The legal interpretation of code, in distinction, focuses on its design—what it was intended to do. Here it is crucial to distinguish the intended operation of a piece of software from its actual operation. Parties who have agreed to the design of a piece of software have not necessarily agreed to bugs in its operation.

This article concludes with a broader consideration of the relation between smart contracts and legal contracts. Enthusiasts have claimed that smart contracts are a means of having “private law without public authority.”<sup>7</sup> Such claims represent a misguided form of techno-utopianism,<sup>8</sup> one that misunderstands both the nature of contractual relationships and the functions of contract law.

Smart contracts, like vending machines, are designed to solve a basic mistrust problem: parties do not always trust one another to perform. The law of contracts can be used to solve that basic mistrust problem. But it does much more than that. Contract law also addresses deception at the time of formation, prevents opportunism, fills gaps in the parties’

---

<sup>7</sup> James Grimmelman, *All Smart Contracts Are Ambiguous*, 2 J.L. & Innovation 1, 2 (2019). To be clear, this is not Grimmelmann’s claim. He is describing the claims of others.

<sup>8</sup> See Julie E. Cohen, *Internet Utopianism and the Practical Inevitability of the Law*, 18 Duke L. & Tech. Rev. 85 (2019).

agreement, and gives parties the flexibility they need to address unforeseen circumstances or future disagreements. And contract law serves broader social functions, such as marking breach as a moral wrong, enforcing obligations of corrective justice, denying public support to agreements society deems unfair or otherwise problematic, and providing a form of civic participation through the courts. It is not obvious that a smart contract could be designed to serve any of those other purposes, and highly unlikely that a smart contract could ever serve all of them. Suggestions that code might someday replace law rest on oversimplified and false pictures of human sociability and of what law does.<sup>9</sup> And parties are likely to do better when they use smart contracts not to supplant, but to supplement their legal contracts.

Part One of this article lays the conceptual groundwork for the analysis that follows. It discusses various definitions of “smart contract” in the literature, fixes on a useful one, and then identifies important differences between smart contracts and contractual writings. Parts Two through Four employ a series of thought experiments to explore when and how the code of a smart contract might figure into the interpretation of an associated legal contract. Part Two make the simple but often overlooked point that when a smart contract is really like a standard vending machine, whose internal mechanisms are hidden from the user, its code does not affect the parties’ legal obligations. Part Three examines the idea of a mutually programmed smart contract. It argues *inter alia* that a smart contract of this type can serve as something like course of performance evidence of the parties’ legal contract, and that when it does, the salient interpretive question is the code’s design, as distinguished from its operation. Part Four examines smart contracts of adhesion, in which one party programs the smart contract and gives it to the other with access to the code and on a take-it-or-leave it basis. Here the analysis draws on the rules governing standard contracts of adhesion, the enforcement of browsewrap terms, the incorporation of extrinsic writings, and the integration of contractual writings to examine when, if ever, the code of an adhesive smart contract should figure into the content of a legal contract between its users. Part Five draws on this doctrinal analysis to think more broadly about the optimal

---

<sup>9</sup> This article focuses on the internal logic of the law, or on reasons cognizable within legal discourse. Legal contracts are also woven into the broader fabric of social life. They have meanings and functions that are not, in a narrow sense, legal ones. For an excellent account of the limitations of smart contracts on that front, see Karen E.C. Levy, *Book-Smart, Not Street-Smart: Blockchain-Based Smart Contracts and The Social Workings of Law*, 3 *Engaging Science, Technology, and Society* 1 (2017). This article’s thesis is the legal complement to Levy’s more sociologically oriented one: that legal contracts “work” in a multitude of ways and accomplish a multitude of aims [within the broader social and relational contexts in which they appear] that are unaccounted for by the smart contract framework.” *Id.* at 2.

relationship between smart contracts and legal contracts, using as an example the epic failure of an especially ambitious smart contract, known simply as “The DAO.” It draws on contract theory to argue that smart contracts, though useful tools, will never do the job of legal contracts or contract law, and to diagnose the mistaken picture of contracting that underlies the idea that they might.

This article employs a series of thought experiments, several of which are fairly fanciful. This approach departs from the style and method of the standard law review article. But thought experiments are hardly new in legal scholarship—think of Oliver Wendell Holmes’s “bad man,”<sup>10</sup> Lon Fuller’s “Case of the Speluncean Explorers,”<sup>11</sup> Meier Dan Cohen’s imagined an “acoustic separation” between officials and the public,<sup>12</sup> or Carol Rose’s alternative narratives about the origins of property rights.<sup>13</sup> Thought experiments of this type serve two important functions. First, imaginative variation can displace standard assumptions that lead to mistaken conclusions. Because a vending machine need not employ code, thinking through when if ever its internal mechanisms might affect a legal contract between its operator and a user helps one think more clearly about when the code of a smart contract might do the same. Second, thought experiments can serve, in Daniel Dennett’s phrase, as intuition pumps.<sup>14</sup> A well-constructed thought experiment can push us to think about familiar phenomena in new ways. I examine in considerable detail, for example, the proper legal responses to standard vending machines, to a vending machine that two engineers build and use together, and to a vending machine encased in a glass box and installed in an engineering department. These thought experiments, and the effort I expend analyzing them, might seem odd, and sometimes even goofy. But their purpose is serious: to think systematically about how a relatively new digital technology, smart contracts, can and should interact with a well-established legal technology, legal contracts. It turns out that the relationship between smart contracts and legal contracts is a complex one. The thought experiments are designed both to avoid easy answers and to identify important and often overlooked doctrinal nuances. Only by getting those nuances right will it be possible to fully understand how the law should respond to this new technology of contracting.

---

<sup>10</sup> Oliver W. Holmes, *The Path of the Law*, 10 Harv. L. Rev. 457 (1897), reprinted in 110 Harv. L. Rev. 991, 992 (1996-1997).

<sup>11</sup> Lon L. Fuller, *The Case of the Speluncean Explorers*, 62 Harv. L. Rev. 616 (1941).

<sup>12</sup> Meier Dan-Cohen, *Decision Rules and Conduct Rules: On Acoustic Separation in Criminal Law*, 97 Harv. L. Rev. 625 (1984).

<sup>13</sup> Carol M. Rose, *Property as Storytelling: Perspectives from Game Theory, Narrative Theory, Feminist Theory*, 2 Yale J. L & Human. 37 (1990).

<sup>14</sup> Daniel C. Dennett, *Intuition Pumps and Other Tools for Thinking* 1-15 (2013).

## 1. Preliminary Remarks

“Contract” is a metonym.<sup>15</sup> In some uses, the word denotes a physical object: a writing that records two parties’ legally effective agreement. In others, it denotes an abstract entity: a legal relationship between persons resulting from an agreement between them. This is why we can make sense of an infelicitous sentence such as, “Tearing up a contract does not destroy the contract.” One might tear up the physical thing without altering the associated legal relationships. To avoid ambiguity, this article uses “contractual writings” to refer to contracts in the first, more concrete sense and “legal contracts” to refer to contracts in the second, more abstract one.

“Contract” is used with other meanings as well. In a game of bridge, partners can “take the contract,” and a mobster might “put out a contract” on someone. Taking a contract in bridge is unlikely to have legal consequences. And although putting out a contract for murder can have significant legal consequences, they are found in the criminal law, not in the law of contracts.

Calling something a “smart contract” does not entail that it is either a contractual writing or a legal contract.<sup>16</sup> Shaanan Cohney and David Hoffman therefore suggest relabeling smart contracts “transactional scripts.”<sup>17</sup> This article keeps with the more familiar nomenclature, but with a warning not to assume that a smart contract qualifies as a contractual writing or results in a legal contract. In fact, this article aims to unpack the complex relationship between the three.

### 1.1 Smart Contracts

A challenge in making sense of the literature on smart contracts is that scholars attach various meanings to the term.<sup>18</sup> Many describe a smart contract as a special form of promise or agreement. Thus Szabo defines “smart contract” as “set of *promises*, specified in digital form, including

---

<sup>15</sup> A metonym refers to one thing using a word for something closely associated with it, as when one refers to the theater as “the stage” or a court as “the bench.” Although this statement hardly requires authority, see Wesley Newcomb Hohfeld, *Some Fundamental Legal Conceptions as Applied in Judicial Reasoning*, 23 *Yale L.J.* 16, 25 (1913); Arthur Linton Corbin, *Offer and Acceptance, and Some of the Resulting Legal Relations*, 26 *Yale Law Journal* 169, 169 (1917); Karl Lewellyn emphasizes the same point in his classic article, *What Price Contract?—An Essay in Perspective*, 40 *Yale L.J.* 704, 707-08 (1931).

<sup>16</sup> See *infra* note 21.

<sup>17</sup> Cohney & Hoffman, *supra* note 4 at 323.

<sup>18</sup> For another systematic account of various definitions, see Alexander Savelyev, *Contract Law 2.0: ‘Smart’ Contracts As the Beginning of the End of Classic Contract Law*, 26 *Info. & Communications. Tech. L.* 116, 120-28 (2017).

protocols within which the parties perform on these promises.”<sup>19</sup> More common today are definitions that conceive of smart contracts as automated *agreements*. Jason Allen, for example, defines “smart contract” as an “automatable and enforceable agreement.”<sup>20</sup> Some, though not all, writers further assume that smart contracts are performe also *legal contracts*.<sup>21</sup>

---

<sup>19</sup> Szabo, *supra* note 6 at 1 (emphasis added). See also J.G. Allen, *Wrapped and Stacked: “Smart Contracts” and the Interaction of Natural and Formal Language*, 14 Eur. R. Contract L. 307, 319 (2018) (“[A] smart contract is a piece of text in a formal language that purports to be both (i) a written instrument embodying and recording contracting parties’ mutual promises with (ii) code that performs and/or enforces those promises on a digital computer.”); Usha R. Rodrigues, *Law and the Blockchain*, 104 Iowa L. Rev. 679, 698-99 (2019) (“[A] ‘smart contract’ generates an automatically enforceable promise, but one available without recourse to the law.”).

<sup>20</sup> Allen, *supra* note 19 at 312-13; See also Grimmelmann, *supra* note 7 at 2 (describing smart contracts as “mechanisms that enforce agreements using software rather than law”); ISDA, *Legal Guidelines for Smart Derivatives Contracts: Introduction 6* (January 2019) (“A smart contract is an automatable and enforceable agreement.”); Stuart D. Levi & Alex B. Lipton, *An Introduction to Smart Contracts and Their Potential and Inherent Limitations*, Harvard Law School Forum on Corporate Governance 1 (posted May 26, 2018), available at: <https://corpgov.law.harvard.edu/2018/05/26/an-introduction-to-smart-contracts-and-their-potential-and-inherent-limitations/> (“Smart contracts’ is a term used to describe computer code that automatically executes all or parts of an agreement and is stored on a blockchain-based platform.”); Levy, *supra* note 9 at 1-2 (“Smart contracts are agreements that utilize the blockchain . . . to automatically and securely execute obligations when certain conditions are met.”); Jeffrey J. Lipshaw, *The Persistence of “Dumb” Contracts*, 2 Stan. J. Blockchain L. & Pol’y 1, 4 (2019) (adopting a definition of “smart contract” as “computer code that automatically executed agreed-upon transactions”); Scott A. McKinney, Rachel Landy, and Rachel Wilka, *Smart Contracts, Blockchain, and the Next Frontier of Transactional Law*, 13 Wash. J.L. Tech. & Arts 313, 321 (2018) (“At their base, smart contracts are self-enforcing agreements that exchange promises or consideration between parties based on a transparent set of rules using predefined inputs.”); Max Raskin, *The Law and Legality of Smart Contracts*, 1 Geo. L. Tech. Rev. 305, 309 (2017) (“A smart contract is an agreement whose execution is automated.”); Mark Verstraete, *The Stakes of Smart Contracts*, 50 Loyola U. Chi. L.J. 743, 755 (2019) (“In the broadest terms, a smart contract is a self-enforcing digital agreement.”); Werbach & Cornell, *supra* note 2 at 328 (“[W]e define a smart contract as an agreement in digital form that is self-executing and self-enforcing.”).

<sup>21</sup> Allen, *supra* note 19 at 321-22 (arguing against distinguishing smart contracts from legal contracts); Jeff Lingwall and Ramya Mogallapu, *Should Code Be Law? Smart Contracts, Blockchain, and Boilerplate*, 88 UMKC L. Rev. 285, 298 (2019) (“At heart, a smart contract is one in which part of a contract or the entire contract is encoded and executed automatically via mechanical means or computer code.”); McKinney *et al.*, *supra* note 20 at 322-23 (“At the risk of stating the obvious, a

All the above conceptions build social facts—agreement, promise, legal contract—into the definition of “smart contract,” making it difficult to tease out the relationship between code and law. In order to get at that relationship, it is helpful to recognize that a smart contract is, first and foremost, a piece of computer code. For the purposes of my analysis, I adopt a more minimalist definition: A smart contract is a piece of software designed to execute future exchanges or other coordinated actions between persons who might otherwise not trust one another to perform.<sup>22</sup> The definition is functional. It turns on the purpose for which the software was written and is used, what it is meant to do. It does not assume, however, that the creation or use of the software involves a promise, agreement or legal contract.

Some authors have used “smart contract” more narrowly to refer only to software fitting one of above definitions that is run on a distributed

---

smart contract must actually be a contract. That is to say, it must meet the characteristics of being a legally enforceable exchange of promises.”).

*But see* Chamber of Digital Commerce, “*Smart Contracts*” *Legal Primer: Why Smart Contracts are Valid Under Existing Law and Do Not Require Additional Authorization* 1 (January 2018) (arguing that not all smart contracts are legal contracts); Grimmelmann, *supra* note 7 at 4 (“[P]arties who [employ a smart contract] at the same time may or may not enter into legal obligations . . . or they may even enter into legal obligations without knowing it or intending to.”); Jonathan Rohr, *Smart Contracts in Traditional Contract Law, Or: The Law of the Vending Machine*, 67 *Clev. St. L. Rev.* 67, 68 (2019) (“Although they can be, smart contracts are not necessarily legal contracts.”).

<sup>22</sup> For similar definitions, see Chamber of Digital Commerce, “*Smart Contracts*,” *supra* note 21 at 1 (“A smart contract is computer code programmed to execute transactions based on pre-defined conditions.”); Cohny & Hoffman, *supra* note 4 at 323 (“A *transactional script* is a persistent piece of software residing on a public blockchain. When executed as a part of an exchange, the code effectuates a consensus change to the state of a ledger.”); Joshua A.T. Fairfield, *Smart Contracts, Bitcoin Bots, and Consumer Protection*, 71 *Wash. & Lee L. Rev. Online* 35, 38 (2014) (defining “smart contracts” as “automated programs that transfer digital assets within the block-chain upon certain triggering conditions”); Rohr, *supra* note 21 at 68 (“I use the term ‘smart contract’ in a general sense to refer to a computer protocol (code) that is stored on a blockchain (or distributed ledger) and which will be automatically executed by the nodes on the blockchain’s network upon the occurrence of specified conditions.”); Savelyev, *supra* note 18 at 127 (“[I]t is possible to define a Smart contract as a piece of software code, implemented on a Blockchain platform, which ensures self-performance and the autonomous nature of its terms, triggered by conditions defined in advance and applied to Blockchain-titled assets.”); Kevin Werbach, *Trust, but Verify: Why the Blockchain Needs the Law*, 33 *Berk. Tech L.J.* 487, 505 (2018) (“Smart contracts are essentially autonomous software agents. With smart contracts, a distributed ledger becomes functionally a distributed computer.”).

ledger, or blockchain.<sup>23</sup> Although blockchain technology did not exist when Szabo introduced the term, its development has led to renewed interest in smart contracts. The ability to install a smart contract on a distributed ledger effects at least three important changes. First, the software no longer runs on a single computer or server, but effectively on thousands of computers at once with an incentive structure designed to ensure reliability. As a result, the parties no longer need to trust a single software host. Second, running a smart contract on a distributed ledger can make it difficult or impossible for a party to alter or halt its operation. Third, blockchain technology enables parties to employ smart contracts while remaining anonymous.

In my view, none of these features makes an important difference in thinking about the relationship between smart contracts and legal contracts. First, although taking a single software host out of the loop can reduce some risks, including the likelihood of certain types of breach of associated legal contracts, third-party platforms also can be secure and reliable. And as Part Five discusses, running software on a distributed ledger does not rule out other forms of technical malfunction and legal breach. Second, although transactions on distributed ledgers are more difficult to halt or unwind, contract remedies rarely aim to undo or prevent breach. More commonly they seek to compensate the nonbreaching party for her losses—a project that is the same whether the breach cannot be undone because it happens on a distributed ledger or because we cannot change the past.<sup>24</sup> Finally, although anonymity can provide a practical barrier to contract enforcement, it is not a necessary feature of smart contracts run on distributed ledgers.

Running a smart contract on a distributed ledger makes other differences as well. Code on a distributed ledger is publicly accessible, meaning that it is available for all users to see. And at present, transacting on a distributed ledger requires a fair amount of expertise, meaning that anyone using the software has some degree of technical sophistication. This article argues that both the accessibility of the code and the sophistication

---

<sup>23</sup> See, e.g., Cohney & Hoffman, *supra* note 4 at 323; Levi & Lipton, *supra* note 20; Levy, *supra* note 9 at 1; Rodriguez, *supra* note 19 at 698-99; Rohr, *supra* note 21 at 68; Savelyev, *supra* note 18 at 127; Werbach, *supra* note 22 at 505. *But see* Lingwall & Mogallapu, *supra* note 21 at 299 (suggesting that use of blockchain is not essential to smart contracts, though blockchain has given them increasing salience); Verstraete, *supra* note 20 at 757 (“[T]he bedrock principles of smart contracts do not require a specific technology. . . . Recently, blockchain technology has revived and implemented the idea of smart contracts. Yet, it is important to keep in mind that the principles and ideas behind smart contracts do not rise and fall solely on the promise of the blockchain.”).

<sup>24</sup> Werbach and Cornell emphasize this backward-looking aspect of contract law, I would argue to a fault. “Contract law is a remedial institution. Its aim is not to ensure performance *ex ante*, but to adjudicate the grievances that may arise *ex post*.” Werbach & Cornell, *supra* note 2 at 328. This article suggests that contract law is a multi-function tool.

of the parties are relevant to how the code does or should figure the terms of a legal contracts. Neither feature, however, is unique to software run on a distributed ledger. As such, they do not provide a compelling reason to limit the inquiry to the narrower category. Lastly, Cohney and Hoffman observe that running software on a distributed ledger can be enormously expensive as compared to other platforms, significantly limiting when it is cost effective to use those platforms.<sup>25</sup> This is yet another reason not to limit the inquiry to software on distributed ledgers.

## 1.2 Smart Contracts and Contractual Writings

Having described what a smart contract is, it is possible to say a few words about similarities and differences between smart contracts and contractual writings.

The similarities are notable. Both smart contracts and contractual writings have a lexical structure: they are produced by assembling semantic elements in accordance with syntactical rules to produce meaningful texts. And because smart contracts are designed to do the work of legal contracts—coordinating future behavior—they often share a logical structure as well. Both often include, for example, conditionals of the form: If *x* occurs, then do *y*.<sup>26</sup> Thus if a contractual writing contains the sentence, “If payment is not received by December 31, 2020, then Buyer will owe an additional fee of \$1,000,” an associated smart contract might include a chunk of code reading something like (in JavaScript):

```
if (Date.now() > Date.parse("2020-12-31") &&
    amtDue > 0) {
    totalAmtDue = amtDue + 1000;
}
```

The smart contract’s code mirrors the structure of a contractual writing designed to do the same job. Finally, both smart contracts and contractual writings can be *read*. One who knows the relevant language can assign meaning to each. As Jason Allen puts the point, a smart contract is “written in a language that is both human-intelligible and machine-readable.”<sup>27</sup>

But there are also important differences.<sup>28</sup> A contractual writing is, in the first instance, a communicative act, designed to be read by humans with the aim of discovering its communicative content: what its author or

---

<sup>25</sup> Cohney & Hoffman, *supra* note 4 at 335-41.

<sup>26</sup> See Raskin, *supra* note 20 at 312-13 (emphasizing the conditional structure of both contracts and programming languages).

<sup>27</sup> Allen, *supra* note 19 at 312.

<sup>28</sup> For more on differences between smart contracts and contractual writings, see Jeremy M. Sklaroff, *Smart Contracts and the Cost of Inflexibility*, 166 U. Pa. L. Rev. 263, 275-77 (2017).

authors intended to convey to a reader.<sup>29</sup> One can also “read” a smart contract. But the goal when reading computer code is different. One reads code not to determine the programmer’s communicative intent, but to determine what it was designed to do and how it will in fact operate. And of course code is written to be read first and foremost in a third, more metaphorical sense: by the platform on which it runs. Code is designed to operate as part of a digital machine. Whereas a contractual writing is a communicative act, code is in the first instance a mechanism.

This is not to deny that code can be rich in meanings. Just as an architect’s plans for a building might be innovative, elegant, pedestrian, derivative, referential, ironic, or funny, a piece of operational code might, to the expert eye, be any of those and more. And just as one might recognize the craftsman’s hand in a piece of custom-built furniture, one might see in code the distinctive style of its author or authors. But code is not primarily a means of communication. It is not, in the philosopher Paul Grice’s formulation, intended to produce an effect in its audience by the very recognition of that intent.<sup>30</sup>

This is the deeper reason why it is misguided to include in the definition of “smart contract” that the software is a form of promise or agreement.<sup>31</sup> A promise is a speech act in which the speaker communicates the intent to undertake an obligation. Because code is not, as such, a communicative act, it cannot, as such, be a promise. An agreement is an understanding between two or more persons about what each shall do, or a speech act through which persons commit themselves to such an understanding. A piece of software, standing alone, is not an agreement either of these senses.

Although code is, in itself, neither a promise nor an agreement, it can be the *subject matter* of a promise or agreement. A party can agree or promise to use a piece of software, or that the operation of the software shall satisfy one or both parties’ obligations. And in some instances, the act of using software might *express* a party’s agreement to a transaction—just as two persons’ attempts to row in unison might express a tacit agreement to continue doing so,<sup>32</sup> or a seller’s shipment of goods might express its agreement to a buyer’s offer.<sup>33</sup> But though the rowers express their

---

<sup>29</sup> The *locus classicus* for this account is Paul Grice, *Meaning*, 66 Phil. Rev. 377 (1957), reprinted in Paul Grice, *Studies in the Way of Words* 213 (1989).

<sup>30</sup> *Id.* at 219.

<sup>31</sup> For more on the nature of promises and agreements, see Gregory Klass, *Promise, Agreement, Contract*, in *Research Handbook on Private Law Theory* 39 (H. Dagan & B. Zipursky eds. 2020).

<sup>32</sup> “Two men, who pull the oars of a boat, do it by agreement or convention, tho’ they have never given promises to each other.” David Hume, *Of Morals*, In *A Treatise on Human Nature* 291, 315 (David Fate Norton & Mary J. Norton eds., Oxford Univ. Press 2000) (1739-1740).

<sup>33</sup> See UCC 2-206.

agreement using the oars and the seller by shipping the goods, neither the oars nor the goods is itself the agreement, in the sense that a verbal agreement is. Neither is an artifact with a communicative content. The same goes for code to which parties agree, or whose use expresses their agreement.

This is not to say that software cannot serve as interpretive evidence of the parties' agreement, or otherwise figure into the construction of their contract. The design or operation of a smart contract that the parties have agreed to, or that the parties have used reached an agreement, might figure into the determination of their legal obligations. But—and this is the central point of this section—one should not assume that it does, and one should not be misled by smart contracts' lexical structure into thinking that those consequences can be read from the code in the way terms can sometimes be read from a contractual writing.<sup>34</sup> Perhaps in some circumstances that is so. But we need an argument for why and when it is.

A final observation before diving into the substance. A reader might object that in fact software often does contain communicative acts: nonoperational comments that programmers have placed in the code.<sup>35</sup> The above line of code, for example, might be preceded by the comment:

```
/* automatically adds late fee  
   when calculating total due */
```

A comment like this is a communicative act in the Gricean sense. And it will be important to consider the possible legal effects of such comments—when if ever they should figure into the interpretive evidence of a legal contract, and whether they might even function as contractual writings. Comments do not, however, pose the conceptual challenges that operational code does. Although it is worth thinking about how comments in the code of a smart contract might figure into the content of an associated legal contract, comments are the easier piece of the puzzle.

---

<sup>34</sup> I consider it problematic, for example, that the UK Jurisdiction Taskforce's report on cryptoassets and smart contracts regularly speaks of the "terms" of a smart contract. See, e.g., UK Jurisdiction Taskforce, Legal Statement on Cryptoassets and Smart Contracts 31 (November 2019) (asking, "In what circumstances is a smart contract capable of giving rise to binding legal obligations, enforceable in accordance with its terms?").

<sup>35</sup> Cohney and Hoffman argue, for example, that "[c]ode—when read with its natural language comments and commit logs—has communicative meaning that courts should seek to ascertain and enforce." Cohney & Hoffman, *supra* note 4 (emphasis added).

## 2. The Standard Vending Machine and Black-Box Smart Contracts

In the years since Szabo's foundational article, the comparison of smart contracts to vending machines has become virtually *de rigueur*. The reason is that vending machines and smart contracts serve, at a high degree of generality, the same function: to enable transactions between individuals who do not trust one another.

Vending machines are connected to legal contracts in two ways. First, as Szabo emphasized, a vending machine is designed to do a job that a legal contract might do: enable a mutually beneficial exchange between parties who do not otherwise trust one another to perform.<sup>36</sup> David Hume long ago described the problem both seek to solve:

Your corn is ripe to-day; mine will be so tomorrow. It is profitable for us both, that I should labour with you to-day, and that you should aid me to-morrow. I have no kindness for you, and know you have as little for me. I will not, therefore, take any pains upon your account; and should I labour with you upon my own account, in expectation of a return, I know I should be disappointed, and that I should in vain depend upon your gratitude. Here then I leave you to labour alone: You treat me in the same manner. The seasons change; and both of us lose our harvests for want of mutual confidence and security.<sup>37</sup>

Call this the "basic mistrust problem." A legal contract can solve the basic mistrust problem in two ways at once. First, the legal contract gives the farmer who is to perform second a new reason to do so: the threat of a lawsuit. Second, it insures the farmer who performs first against the other's defection: if the second breaches, the first will be able to recover compensatory damages.

---

<sup>36</sup> Thus Szabo has recently written, "At its core, a vending machine is a security mechanism: the amount in the till should be less than the cost of breaching the till." Nick Szabo, *Forward*, Chamber of Digital Commerce, *Smart Contracts: 12 Use Cases for Business and Beyond* 3 (December 2016).

<sup>37</sup> Hume, *supra* note 32 at \_\_\_\_\_. Hume uses the example to explain not contract law, but the social practice of promising. Hobbes makes the same point with respect to contract law:

[In a state of nature] he that performeth first, has no assurance the other will perform after, because the bonds of words are too weak to bridle men's ambition, avarice, anger, and other passions, without the fear of some coercive power. . . . But in a civil estate, where there a power set up to constrain those that would otherwise violate their faith . . . he which by the covenant is to perform first is obliged to do so.

Thomas Hobbes, *Leviathan*, 84-85 [l.xiv.18-19] (Hackett 1994/1651).

The basic mistrust problem is not ubiquitous. An unattended roadside fruit stand with a box for depositing cash is a tool for exchanges between strangers where there *is* a degree of trust. The owner leaves their vegetables at the stand, trusting that passing drivers will pay for what they take. (The trust need not be complete. The owner might put a lock on the cash box and nail it to the table.) But such trust does not always exist. The operator of a vending machine does not trust that if they leave the candy bars unattended on a table, those who take the bars will pay for them. So the operator uses a machine that releases the bars only after payment has been made. The machine does the job—solving the basic mistrust problem—that a legal contract might otherwise perform.

Second, and this is crucial to everything that follows, use of a vending machine can at the same time create a legal contract between the user and the machine's operator. This is so even though the parties exchange few if any words. And it is so even though the parties are not, in the first instance, relying on contract law to solve the basic mistrust problem. Moreover, the legal contracts that attach to these transactions are not merely vestigial. This part argues that even if the machine solves the basic mistrust problem, the resulting legal contract still adds value to the transaction.

The first section of this Part examines in greater detail these two relationships between traditional vending machines and legal contracts: a vending machine substitutes for a legal contract and its use generates a legal contract. The second draws out the implications for what I call "black-box smart contracts."

## 2.1 The Standard Vending Machine

Imagine a standard vending machine built in the middle of the twentieth century. The machine is contained in a metal box about one and a half meters high and a meter across. In the middle of the front is a glass window, behind which is a row of candy bars, each with a price tag. Below each bar is a pull handle. Below the pull handles is an opening where candy is delivered. On the upper right corner of the box is a coin slot, a lever labeled "Refund," and a small opening where coins can be returned. Other than the prices and the "Refund" label, the only word on the machine is "CANDY," printed in large letters. Inside the machine are various analog mechanisms designed to perform the functions of a standard vending machine. Call these the machine's "guts."

Suppose Acme Vending Company ("Acme") installs such a standard vending machine in a law school faculty lounge, and that immediately after teaching his Contracts course, Professor Charles Kingsfield, Jr. puts two quarters into the machine, pulls a handle, and does not receive a candy bar. Suppose also that when Professor Kingsfield depresses the lever labeled "Refund," he does not receive his coins back. Professor Kingsfield might

successfully sue Acme for breach of contract. When two persons' nonverbal conduct evinces their agreement to an exchange, the result is an implied-in-fact contract. The machine's appearance would lead a reasonable person to believe that inserting coins in the right amount and pulling a handle will result in the delivery of a candy bar.<sup>38</sup> Acme's act of installing the vending machine is therefore an implied-in-fact offer to sell candy bars. When Professor Kingsfield inserts his two quarters, he accepts that offer, after which Acme has a legal duty to provide Professor Kingsfield a candy bar or return his money.<sup>39</sup> Acme's failure to do either breaches that legal duty, exposing it to a lawsuit by Professor Kingsfield.<sup>40</sup>

The example is, of course, fanciful. The costs of filing a law suit would dwarf any amount Professor Kingsfield could recover. And the mere threat of a lawsuit would cause Acme to simply return Professor Kingsfield his money. One might replace in the example "candy bar" with "iPhone" and make the lawsuit a class action. But the existence of a legal contract does not depend on the amounts at stake.

What are the terms of the legal contract between Acme and Professor Kingsfield? The terms of any implied-in-fact legal contract depend in the first instance to the parties' objectively reasonable understanding of their agreement. In vending machine transactions, the interpretive evidence

---

<sup>38</sup> See *Barker v. Allied Supermarket*, 596 P.2d 870, \_\_\_ (Okl. 1979) ("It has been held that in a self-service store, the 'merchant's act of stocking the self-service displays with goods thereby makes an offer to the shopper,' which the shopper accepts by taking the goods from the shelf.").

<sup>39</sup> The logic of offer and acceptance is malleable. One might instead view a vending machine as a solicitation of offers, the act of dropping coins into the machine as the making of an offer, and the machine's failure to return the coins as an acceptance of that offer. Nothing in the analysis below turns on the characterizations. See Rohr, *supra* note 21 at 74-75 (offering a similar analysis of offer and acceptance in vending machine transactions).

<sup>40</sup> The caselaw on contracts generated by the use of a vending machine is thin. But see *Haynes v. Coca Cola Bottling Co. of Chicago*, 250 N.E.2d 20 (Ill. App. Ct. 1976) (Coca Cola sold through a vending machine and containing fungus violated implied warranty of merchantability); *Alfonso v. Stavnitsky*, 8 Conn. Supp. 34 (Com. Pl. 1940) (breach of implied warranty of fitness in sale from confectionary vending machine). See also *Moore v. Compass Grp. USA, Inc.*, 2019 WL 4723077 (E.D. Mo. Sept. 26, 2019) (describing but not ruling on plaintiffs' claim that credit card surcharges violated plaintiff's contract with vending machine operator); *Jaye v. Crane Merch. Sys., Inc.*, 2020 WL 406942 (N.D. Tex. Jan. 24, 2020) (same); *Oliver v. Compass Grp. USA, Inc.*, Case No. 1:18-CV-05998 (filed in the U.S. District Court for the Northern District of Illinois on Aug. 31, 2018) (complaint with similar claim).

Curiously, the Soviet Union codified this rule for vending machines. Section two of Article 498 of the USSR Civil code provided, "The contract of retail sale with the use of vending machines shall be deemed to be concluded since the time of the commission by the buyer of the actions necessary for the receipt of goods." [Cite] This section remains a part of the Russian Civil Code.

is fairly thin. There are no contractual writings. In fact, the only communicative acts in the example are the labels, “CANDY” and “Refund,” and the price tags. But there is little doubt as to what one has the right to expect from a vending machine. Our shared cultural understanding of what vending machines do determines the objectively reasonable understanding of Acme and Professor Kingsfield’s agreement, which in turn generates Acme’s legal duty to deliver a candy bar or provide a refund after the insertion of coins.

Acme’s duty to deliver a candy bar or a refund is not all there is to the legal contract. Because this is a sale of goods, the transaction is governed by Article Two of the UCC. Suppose the machine delivers a candy bar that is two-thirds the size of the one displayed. Professor Kingsfield can then invoke UCC section 2-313: “Any sample or model which is made part of the basis of the bargain creates an express warranty that the . . . goods shall conform to that sample or model.”<sup>41</sup> Or suppose the machine delivers a stale candy bar. Professor Kingsfield might now turn to section 2-314, which provides that when a merchant sells goods, there is an implied warranty of merchantability that the goods are of a quality that would “pass without objection in the trade” and that they “are fit for the ordinary purposes for which such goods are used.”<sup>42</sup>

These terms of the legal contract between Acme and Professor Kingsfield are not immutable. Acme might, for example, put a sign on the machine reading, “All candy sold as is.” Under the UCC, Acme has contracted out of the implied warranty of merchantability, meaning that a stale candy bar no longer counts as a breach.<sup>43</sup> The express warranty created by the sample candy bars, however, remains.<sup>44</sup>

A sign reading “Use at your own risk” would have different legal consequences. A court would likely construe such a sign as shifting the risk of mechanical failures to the customer. Now if neither a candy bar nor a refund is provided due to a mechanical glitch, Professor Kingsfield does not have an action for breach. But even with such a sign, Professor Kingsfield can reasonably expect that he is using a machine designed to provide candy bars in exchange for money. It is reasonable to understand “risk” in this context as referring to mechanical glitches or other accidental failures, not to the operator’s purposeful nondelivery of candy. The sign does not negate Acme’s legal duty, for example, to regularly load the vending machine with candy.

Like all offers, Acme’s implied-in-fact offer can be stated in the form of a conditional: If a customer inserts enough money into the machine,

---

<sup>41</sup> UCC § 2-313(1)(c).

<sup>42</sup> UCC § 2-314(2)(a) & (c). See *Chaffin v. Atlanta Coca Cola Bottling Co.*, 194 S.E.2d 513, 515 (Ga. Ct. App. 1972) (holding that implied warranty of merchantability applied in a vending machine transaction).

<sup>43</sup> UCC § 2-316(3)(c).

<sup>44</sup> UCC § 2-316(1).

Acme will provide a candy bar. And the guts of a standard vending machine, in some sense, mirror that conditional structure. They are designed so that if a person inserts money, the machine will provide a candy bar. There is an isomorphism between the guts and the legal contract.

That isomorphism does not, however, make the guts part of the interpretive evidence—the collection of facts that a court would use to interpret the parties’ contractual agreement. The reason should be obvious: the interpretive evidence of an agreement includes only facts available to both sides. Acme has at least constructive knowledge of the machine’s guts. But Professor Kingsfield has no clue about what is inside the machine. The design of the machine’s interior does not, therefore, figure into the agreement between Acme and Professor Kingsfield.

A somewhat fanciful example will illustrate the point and take the thought experiment a bit further. Suppose that rather than the guts of a standard vending machine, Acme installs a mechanism designed to occasionally cause non-delivery of a candy bar and no refund—what I will call a “Random No-Candy Generator” or “RNCG.”<sup>45</sup> Because Professor Kingsfield has no reason to know that the machine contains an RNCG, its presence does not figure into the his agreement with Acme. A failure to deliver candy due to the installation of an RNCG would be a breach. This would be so even if Acme added a “Use at your own risk” sign. “Risk” in such a warning is commonly understood to refer to malfunctions, not to design features contrary to the user’s reasonable expectations.

Professor Kingsfield’s action for breach is not the only legal consequence of installing an RNCG. Its presence in the machine would also allow Professor Kingsfield to argue that his assent to the transaction was induced by a material misrepresentation, giving him the power to rescind the legal contract. Even though the guts are not part of the interpretive evidence, they might affect the legal contract between the parties.

As it happens, rescinding the legal contract does not do Professor Kingsfield much good. But the intentional installation of an RNCG also exposes Acme to civil liability for the tort of deceit or for violation of the local unfair and deceptive acts and practices statute, and perhaps even to criminal liability. This additional layer of legal protection is important. Professor Kingsfield is relying not on the law of contract to solve the mistrust problem—in practice the amounts at stake are far too small for him to sue—but on the proper operation of the vending machine. How can he know that the machine’s guts, which he cannot see, are in fact designed to

---

<sup>45</sup> The example is not entirely fanciful. Kerry Segrave finds evidence from the early days of vending machines—around the turn of the last century—of “a growing tendency on the part of some early vending inventors to combine an element of chance with their merchandise machines.” Kerry Segrave, *Vending Machines: An American Social History* 20 (2002).

deliver candy bars for coins? In part because a different design would expose Acme to fines, punitive damages even criminal prosecution. In a standard vending machine transaction the law still plays a role in solving the basic mistrust problem, though the relevant law is not traditionally considered part of the law of contract.

If a “Use at your own risk” sign is not enough to avoid legal liability for installation of an RNCG, is there something Acme could say to do so? Perhaps. Suppose Acme replaces the word “CANDY” on the outside of the box with the words “CANDY SLOT MACHINE,” and installs it in a casino rather than a faculty lounge. Those actions would both put users on notice that they should not rely on their common cultural understanding of vending machines, and inform them of the machine’s atypical design.

The results of the thought experiment can be summarized as follows. First, a standard vending machine does work that, in other transactions, legal contracts do. It solves the basic mistrust problem. Second, under existing law, use of a standard vending machine nonetheless generates a legal contract between the user and operator. Third, the terms of that legal contract depend on the cultural understanding of what vending machines do, aspects of the vending machine apparent to the user, and legal default and mandatory terms. Fourth, those terms do not depend on the machine’s internal mechanism, which are not visible to the user—though those mechanisms might give rise to one or more contract defenses or separate causes of action. Fifth, the standard vending machine cannot do the job of a legal contract—solving the basic mistrust problem—without the help of the law. Laws that punish deceptive acts assure users that the vending machine’s internal mechanisms are designed to operate in accordance with users’ reasonable expectations. The machine is not a complete substitute for the law. Finally, effectively opting-out of the standard cultural meaning of a vending machine requires clear and conspicuous notice to users.

## 2.2 Black-Box Smart Contracts

The takeaway for smart contracts is fairly straightforward. None of the above results change if the guts of the vending machine include digital elements. It makes no difference if the vending machine has a microprocessor running code, or if Acme (weirdly) chooses to run that code on a distributed ledger with which the machine interacts. Because such code would not be evidence of the parties’ agreement, it would not belong to the interpretive evidence and would not affect the terms of the legal contract.

Call a smart contract whose code one or both users cannot access a “black-box smart contract.” As technology develops, black-box smart contracts are likely to become quite common. Suppose, as suggested in the Introduction, a fruit wholesaler and a trucking company employ software

that automatically adjusts the price for transporting avocados based on the bumpiness of the ride, as measured by vibration sensors in the shipper's trucks. Perhaps the trucking company has developed the system in-house, then offered it to all its customers. Or parties to a long-term shipping contract might purchase the system from a third-party vendor. Under either scenario, one or both parties might not be given access to the code.

Like the standard vending machine, use of the vibration sensing software does some of the work a legal contract might. Rather than using the software, the parties might have signed a contractual writing that provided a payment schedule based on the condition of the fruit upon arrival. The parties have automated a process that they might have otherwise accomplished with human beings acting pursuant to a legal contract.

Also like a standard vending machine, the parties' agreement to use the vibration sensing software nonetheless creates a legal contract. But there is a difference. Because more money is at stake in this new example, the legal contract is even more likely to remain part of the solution to the basic mistrust problem. If the vibration sensing software fails to work as expected—the fruit arrives in a condition that suggests a bumpy ride, but the system does not discount the price—it would be financially viable to sue for breach of the legal contract. And a party that does not have access to the code will be especially concerned to have this legal protection. In the economy of trust, a black-box smart contract does not so much substitute for the legal contract as work in tandem with it. Finally, if the smart contract is provided by one of the parties, the other might also rely on the law of deception—the tort of deceit, UDAP statutes, criminal fraud—for protection against code designed to work other than advertised.

What are the terms of the legal contract? Unlike a standard vending machine transaction, the parties to the avocado shipping contract will have discussed many aspects of the transaction, and have perhaps even memorialized parts of it in a contractual writing. If the legal contract ends up being adjudicated, those communications will figure into the interpretive evidence. The smart contract code will not. Because one or both parties do not have access to the code, it is irrelevant to interpreting their agreement. That said, if the software was provided by one side, it might be introduced as evidence of fraud or misrepresentation.

Can the supplier of a black-box smart contract stipulate, using words outside the code, that aspects of the code shall govern? If the trucking company supplies the vibration-sensing system, it might include risk-shifting language to insulate itself from liability for operational bugs in the code, in the way that “as is” or “Use at your own risk” sign on a vending machine operates. Such language would not, however, insulate the trucking company from liability for design elements contrary to a the wholesaler's reasonable understanding of how the system functions. This is especially so if the trucking company makes express representations about

what the software does. Nor would such risk-shifting language make the code evidence of the parties' agreement. Again, to the extent that the code is not shared, it says nothing about the parties' agreement and so does not affect the terms of their legal contract.

### **3. The Mutually Constructed Vending Machine and Mutually Programmed Smart Contracts**

Placing a smart contract in a black box has both legal and practical consequences. Legally, the fact that one or both sides does not have access to the code means that the code cannot serve as evidence of their agreement. As such, it does not belong to the interpretive evidence of any legal contract between them. Practically, putting the smart contract in a black box means that there remains a mistrust problem. Parties without access to the code might not trust that the smart contract is designed to do what it is supposed to do. The law can help solve that mistrust problem. If there is a legal contract, a software malfunctions might qualify as a breach. And software that is programmed to do something other than promised can expose a party providing it to civil or criminal liability—in tort, under state UDAP statutes, or even the criminal law.

Another solution to mistrust of the code is to give both users access to it. Call a smart contract whose non-compiled code is accessible to all users a “glass-box smart contract.” There two broad categories of glass-box smart contracts. The first, which is the subject of this Part, comprises software users program together. Smart contracts of this type will probably never be common. Bespoke programming is costly and requires sophistication. But just as the idea of mutually negotiated contractual writings drives many intuitions about the law of contracts, so the idea a mutually programmed smart contract will help sharpen intuitions about smart contracts.<sup>46</sup> The second category of glass-box contracts, which is the subject of the next part, comprises smart contracts programmed by one user or a third party and provided to non-programming users together with access to the code.

#### **3.1 The Mutually Constructed Vending Machine**

Suppose Orville and Wilbur are engineers who work at the same company. Orville works the day shift, Wilbur the night shift. In Orville's spare time, he runs a chocolate truffle business. In Wilbur's spare time, he runs a coffee shop. They meet for the first time at a company picnic, where they learn of one another's outside interests. Orville suggests that Wilbur

---

<sup>46</sup> James Grimmelmann, for example, starts off his analysis of smart contract code supposing that “[t]he contracting parties write a computer program that embodies their agreement.” Grimmelmann, *supra* note 7 at 2.

buy his chocolate truffles to sell in Wilbur's coffee shop. Wilbur likes the idea, and they agree on a wholesale price of \$1 per chocolate truffle. Because both are engineers, they decide to build a machine to effect the exchanges, which they will install at their workplace. Orville will load chocolate truffles into the machine during the day. When Wilbur is there for the nightshift, he will insert cash into the machine, which will deliver him the corresponding number of chocolate truffles. Orville will have the only key to the machine, allowing him to load his chocolate truffles into it and to remove the money Wilbur has deposited. The key will not, however, give Orville access to the machine's guts. The two engineers spend the following four weekends in Orville's garage building the machine, which they then install with their supervisor's permission in an unused closet. The result is a mutually constructed vending machine.

As with the standard vending machine, the mutually constructed vending machine is designed to effectuate mutually beneficial exchanges between parties who do not otherwise trust one another to perform. Orville and Wilbur might have accomplished the same thing using the law of contract. Wilbur could, for example, simply submit a daily truffle order to Orville, which Orville might accept either by using words or by delivering chocolate truffles. Each acceptance by Orville would create a legal contract, under which Wilbur would have a legal duty to pay for chocolate truffles delivered.

Although Orville and Wilbur are relying on a mechanical device rather than the law to solve the basic mistrust problem, use of the mutually constructed vending machine again generates a legal contract. Wilbur's act of putting money into the machine creates for Orville a legal duty to provide chocolate truffles, through the machine or otherwise, or to return the money. Thus if Wilbur inserts \$30 into the machine and receives only twenty chocolate truffles, Orville has a legal duty to either provide Wilbur ten additional chocolate truffles or to return him \$10. If Orville does neither, Wilbur might sue Orville for breach.

Many of the terms of the legal contract between Orville and Wilbur will be similar to the terms of a standard vending machine legal contract. If, for example, Orville provides Wilbur some sample chocolate truffles, the law will deem Orville to have expressly warranted that the chocolate truffles he delivers will conform to that sample. And because Wilbur runs a chocolate truffle business, the sales are subject to the implied warranty of merchantability. Loading stale chocolate truffles would be a breach.

But there are also differences. For example, the fact that both Orville and Wilbur have participated in the construction of the vending machine might affect a reasonable understanding of the allocation of certain risks. Because only the owner-operator has access to a standard vending machine's guts, the owner-operator is presumed to bear the risk of glitches. Because Orville and Wilbur participated equally in the design and

construction of their vending machine, it is not so obvious who should bear the risk of its malfunction.

In addition, there is considerably more interpretive evidence of Orville and Wilbur's agreement. In the basic transaction between Acme and Professor Kingsfield, the content of their contract was largely determined by the common cultural understanding of how vending machines work. The parties barely communicated with one another. The transaction between Orville and Wilbur, in distinction, began with a conversation in which they agreed to basic terms—what sorts of goods Orville would supply and how much Wilbur would pay for them. In this respect Orville and Wilbur's agreement looks more like that between the trucking company and avocado wholesaler. It is verbose.

Finally, the design of Orville and Wilbur's mutually constructed vending machine belongs to the interpretive evidence of their agreement. Orville and Wilbur's joint participation in the machine's construction means that its design reflects their agreement. Suppose Orville and Wilbur never record that agreement in writing, that Wilbur inserts \$30 and receives only twenty chocolate truffles, that Orville neither provides the additional ten chocolate truffles to Wilbur nor returns \$10 to him, and that Wilbur sues for breach. If Orville maintains that they had in fact agreed to a price of \$1.50 per chocolate truffle, Wilbur might point to the fact that the guts are designed to deliver one chocolate truffle for each \$1 inserted. That design would be strong evidence that they agreed to the \$1 price. Alternatively, if Orville can show that nondelivery of the ten chocolate truffles was due to the proper operation of a Random No-Candy Generator that they together installed, he might argue that in fact he and Wilbur had agreed to exchange not money for chocolate truffles, but money for a chance at chocolate truffles.

Here a court would need to distinguish between the mutually constructed vending machine's *design* and its *operation*. Suppose examination of the guts reveals that nondelivery of chocolate truffles was due to a faulty bill scanner, which occasionally reads a \$20 bill as a \$10 bill. This would appear to be a glitch in the machine—a bug not a feature, part of the machine's operation not its design. Because glitches are unintended, they do not serve as evidence of the parties' agreement. The installation of an RNCG, in distinction, would belong to the machine's design. Its operation would be a feature, though it might produce results identical to the those of a faulty scanner. Because an RNCG would be part of the mutually constructed vending machine's design, it would belong to the interpretive evidence.

The example illustrates a broader point that will run through the remainder of this article. One can ask two distinct questions about any device. What is it designed to do? And: How does it in fact operate? The first, design question concerns the device's purpose or function. The second, operation question is about how the device will actually perform,

what it will do when put into use. The answers are connected. To determine what an unfamiliar device is designed to do, one looks at how it is likely to operate and asks, “Why would anyone build a machine that does such-and-such?” Contrariwise, knowing that a device is designed for a certain purpose can help one predict how it is likely to operate. But the questions are distinct. Machines do not always operate as designed. What matters when seeking to determine Orville and Wilbur agreement, and thereby the terms of their legal contract, is the mutually constructed machine’s design—how they intended it to function, in distinction from how it actually operates.

### 3.2 Mutually Programmed Smart Contracts

The results remain largely the same if we change the thought experiment to involve a mutually programmed smart contract. A somewhat fanciful example will provide materials to consider additional ways smart contracts can interact with legal contracts.<sup>47</sup>

Suppose Orville and Wilbur decide to build a fully automated system for effectuating the exchanges of money for chocolate truffles. Together they write a piece of software to do three things. First, using the feed from a digital camera, the software constantly monitors the truffle supply in Wilbur’s coffee shop. Second, when the software recognizes that the supply is running low, it instructs an autonomous drone to pick up two boxes of chocolate truffles from Orville’s store and deliver them to Wilbur’s shop. Third, upon receiving confirmation of delivery from the drone, the software effects a transfer of funds from Wilbur’s bank account to Orville’s, at a rate of \$1 per truffle delivered. Because Orville and Wilbur do not trust third party platforms, they choose to run the software on a distributed ledger.

If Orville and Wilbur were lawyers rather than engineers, they might instead have drafted and signed a contractual writing. In that writing, Orville would agree to supply and Wilbur would agree to accept all the

---

<sup>47</sup> This section assumes that the parties themselves program the smart contract. One might introduce yet another layer of complexity by assuming that one or both parties employ someone else to do the programming. Thus Alexander Savelyev observes:

Due to a separation between the person programming the code and the person intending to use it in its commercial activities, there is a risk of misunderstanding between them with regard to the terms of the future agreement. Ultimately, differences may exist between implementation and intent, and this danger may be aggravated by the huge gap of abstraction between legal language and a programming language.

Savelyev, *supra* note 18 at 126. The issue also exists with respect to the interpretation of contractual writings and other texts. See Gregory Klass, *Boilerplate and Party Intent*, 82 Law & Contemp. Probs. 105, 123-24 (2019) (distinguishing between author and authorizer meaning).

chocolate truffles Wilbur requires, for which Wilbur would agree to pay \$1 per truffle. Had they taken this path, Wilbur would need only inform Orville that he was running low on chocolate truffles to trigger Orville's legal duty to supply more, after which Wilbur would have a legal duty to pay for the chocolate truffles delivered. Orville and Wilbur's automated delivery and payment system, which includes the mutually programmed smart contract, is designed to do a job that might instead be done by such a legal contract.

At this point in the analysis it should come as no surprise that, although the automated delivery and payment system is designed to substitute for a legal contract, its construction and use nonetheless generate one. The system implements and evinces Orville and Wilbur's agreement that Orville shall keep Wilbur's coffee shop supplied with chocolate truffles in exchange for Wilbur's payment of \$1 per truffle. If Orville stops making chocolate truffles for a week, thereby preventing the system from restocking Wilbur's store, Wilbur can sue Orville for breach. If Wilbur lets his bank account run low so the system cannot transfer payment into Orville's account, Orville can sue Wilbur for breach.

We might begin with four relatively straightforward observations about this legal contract between Orville and Wilbur. First, for the same reasons that the guts of the mutually constructed vending machine might be used as evidence of Orville and Wilbur's first agreement, the code of their mutually programmed smart contract might figure into the interpretation of the legal contract between them. Suppose, for example, the code includes the following:

```
if (quantNow < 4) {  
    resupply(2)  
}
```

where the value of "quantNow" is provided by software connected to the camera counting the boxes truffles and "resupply(x)" is the subroutine that cause the drone to pick up x boxes of chocolate truffles from Orville's business and deliver them to Wilbur's. The presence of that line, together with the fact that Orville had a hand in writing it, would be evidence that Orville had agreed to keep Wilbur's shop supplied with chocolate truffles. When both parties program a smart contract, the code is evidence of their agreement, and belongs to the interpretive evidence of their contract.

Second, when using the code as interpretive evidence, a court should focus on the code's design, not its operation. The UK Jurisdiction Taskforce white paper on bitcoin and smart contracts provides that "an investigation of what the code actually does (possibly with the assistance of expert evidence) may be needed as part of the exercise of interpretation."<sup>48</sup>

---

<sup>48</sup> UK Jurisdiction Taskforce, *supra* note 34 at 35.

This formulation is misguided. What matters for the purposes of contract interpretation is not a mutually programmed smart contract's operation—"what the code actually does"—but its design—what it is intended to do. Suppose over the course of a month the system delivers 2,000 truffles to Wilbur, but that due to a bug it does not transfer \$2,000 from Wilbur's bank account to Orville's. If Wilbur refuses to pay Orville the \$2,000, Orville might successfully sue Wilbur for breach. Their agreement, as embodied in software's design if not its operation, was that Wilbur would pay for truffles received.

Third, interpreting what a smart contract is designed to do is different in kind from interpreting a contractual writing. Had Orville and Wilbur used a contractual writing to effect their exchange, it might have included a sentence like, "Orville agrees to supply Wilbur all the chocolate truffles that his business requires." That sentence is an express undertaking. Although the above fragment of code is designed to perform such an undertaking, it is not itself the undertaking. It is evidence of the agreement, not an expression of it. Reading code for what it is designed to do is not the same as reading a contractual writing for its communicative content.

The code of a mutually programmed smart contract is best understood by analogy to course of performance evidence.<sup>49</sup> Code designed by both parties is part of the performance of their agreement. It is not the agreement itself. Nor does it communicate the agreement. It is, as Corbin writes, "a further expression by the parties of the meaning that they give and have given to the terms of their contract previously made."<sup>50</sup>

Fourth, Orville and Wilbur's choice to use a smart contract might make the design question somewhat easier to answer than it was for the mutually constructed mechanical vending machine. Coding languages allow programmers to add nonoperational comments, which are commonly used to explain how sections of the code are designed to operate and can therefore help in the project of differentiating between the code's design and its operation—between features and bugs. The above fragment of code, for example, might be preceded by the following comment:

```
// resupply subroutine
// No.bboxes is number of boxes to be delivered;
// resupply(No.bboxes) causes drone to pick up
// No.bboxes from Orville and deliver to Wilbur
```

If a line in the subroutine causes the chocolate truffles to occasionally be delivered to somewhere other than Wilbur's place of business, the above

---

<sup>49</sup> See, e.g., *Brooklyn Life Ins. Co. of New York v. Dutcher*, 95 U.S. 269, 273, 24 L. Ed. 410 (1877) ("There is no surer way to find out what parties meant, than to see what they have done."); U.C.C. 1-303(a) (defining a course of performance).

<sup>50</sup> 3 Arthur Linton Corbin, *Corbin on Contracts: A Comprehensive Treatise on the Rules of Contract Law* § 558, 249 (1951).

comment would be strong evidence that this operational aspect of the code is not part of the smart contract's design.

All of the above is of a piece with the analysis of Orville and Wilbur's mutually constructed vending machine. Their code-based automated delivery and payment system, however, introduces two new wrinkles.

The first comes from the availability of code comments. Unlike operational code, comments are communicative acts. They are intended to share information with the reader, as distinguished from effecting changes to a machine's state. The above hypothetical comment concerns a subroutine's design: how it is meant to function. But programmers can say anything they want in a comment—attribution, asides, jokes, complaints, or recipes for vegan chili. One can imagine Orville and Wilbur, or the creators of any mutually programmed smart contract, embedding not only comments about how the code is designed to operate, but also comments that express their agreement. Orville and Wilbur might, for example, have inserted a comment at the beginning of the code reading:

```
// Parties hereby agree Orville shall supply
// all of the chocolate truffles Wilbur needs,
// for which Wilbur will pay $1 per truffle
```

Such a comment would qualify as a contractual writing, for it seeks to memorialize details of Orville and Wilbur's agreement. And one can imagine other code comments that directly express the parties' agreement on more particular matters—Orville's obligation to keep a supply of chocolate truffles for the drones to pick up, Wilbur's duty to keep funds in his bank account, which party bears the risk of one or another malfunction in the system, and so forth.

We might call comments that do not concern the design of the code, but are intended to record aspects of the parties' agreement "contractual comments." Courts should treat contractual comments in a mutually drafted smart contract just like other writings that express the parties' agreement. In fact, one can imagine parties embedding the entirety of a standard contractual writing as comments in the code of a mutually programmed smart contract, perhaps to record it for all eternity in a distributed ledger. But we should not confuse the way such contractual comments would figure into the parties' legal obligations with the way the code's design might. Saying something within the code is different from designing code to do something.

The second wrinkle derives from the structure of Orville and Wilbur's exchange, and illustrates yet another way that contract law can serve the interests of smart contract users.

Suppose after Orville and Wilbur set up the system, Chef's Table—a streaming documentary series that profiles professional chefs—devotes an episode to Orville and his carefully crafted chocolate truffles (slow-motion

Steadicam shots of Orville dusting the chocolate truffles with cocoa powder). After the episode is released, demand for Orville's truffles goes through the roof, and with it their price. Wilbur, hoping to cash in on Orville's luck, begins selling Orville's chocolate truffles not only at his coffee shop, but to retail stores. In order to secure the quantity he needs, as soon as the drone makes a delivery, Wilbur moves the boxes into his delivery truck, out of the camera's view, thereby causing the system to immediately make another delivery.

The legal contract between Orville and Wilbur is a requirements contract. Orville and Wilbur have agreed that Orville shall supply all the chocolate truffles Wilbur requires. As such, section 2-306 of the UCC governs:

A term which measures the quantity by the . . . the requirements of the buyer means such actual . . . requirements as may occur in good faith, except that no quantity unreasonably disproportionate to any stated estimate or in the absence of a stated estimate to any normal or otherwise comparable prior output or requirements may be tendered or demanded.<sup>51</sup>

Suppose Orville and Wilbur's smart contract software does not cap the rate at which the system ships chocolate truffles to Wilbur's coffee shop. Examining the code would provide evidence that Orville and Wilbur entered into a requirements contract. The system is designed to ensure that Orville provides all the chocolate truffles Wilbur requires. The code would not, however, reveal the limit UCC 2-306 puts on Orville's legal duty to provide chocolate truffles.

Orville and Wilbur might have built such a limit into their smart contract. They might, for example, have programmed the system to track and cap the rate of deliveries. Does their failure to do so constitute a bug in the software? Not necessarily. Suppose Orville and Wilbur discussed whether to cap the rate of deliveries, and even included in the software a variable, `rate.Deliv`, to track it. But because Orville's production capacities were elastic, they decided not to limit the delivery rate and, just to be clear, included the comment:

```
// no cap on rate.Deliv
```

The lack of a cap on the delivery rate is part of the system's design. The problem is that neither party anticipated the change in value of Orville's chocolate truffles. The bug is not in the software, which accurately reflects the parties' agreement, but in the agreement itself. Orville and Wilbur failed to anticipate that Wilbur might use the system to take advantage at Orville's expense of an unexpected rise in price.

---

<sup>51</sup> UCC 2-306(1).

Smart contracts can address a basic mistrust problem: the worry that the other side will not perform their side of the exchange. Part Two emphasized a second mistrust problem that smart contracts do not solve, but which the law can: worries about deception. But even if both those mistrust problems are solved, there might remain a third: the worry that the other side will exploit advantages that the contract gives her in ways the parties did not anticipate. In the example, the problem is not that Wilbur fails to perform his side of the bargain. He is paying for every chocolate truffle he receives. The problem is that Wilbur is using their agreement in a way the parties did not anticipate and at Orville's expense: to become a wholesaler of Orville's chocolate truffles. Section 2-306 the general duty of good faith are designed to protect against opportunism of this type.<sup>52</sup>

There are familiar reasons to think that these are the legal rules parties want when they enter into exchange agreements. Just as Hume's farmers will not help one another unless they can solve the basic mistrust problem ("The seasons change; and both of us lose our harvests for want of mutual confidence and security"), so worries about opportunism might prevent some mutually beneficial exchanges from happening. And even when parties consider the risk of opportunism to be low, they are likely to prefer legal protection against unbargained-for advantage taking. Rather than expending the resources necessary to identify and address in a contractual agreement—or in their smart contract—every contingency or possible form of opportunism, parties are happy to rely on legal defaults and the generally applicable duty of good faith that the law provides.

In short, the fact that a smart contract can be used to solve the basic mistrust problem does not make it a complete substitute for a legal contract. Contract law does more than ensure performance and insure against breach. It is designed, among other things, to provide for unanticipated contingencies and prevent opportunism.

The results of this Part can be summarized as follows. First, a mutually programmed smart contract can do better at solving the basic mistrust problem than can a black-box smart contract. Whereas users without access to a black-box smart contract need a reason to trust that the software will do what they expect, all users of a mutually programmed smart contract know the code. Second, unlike the contents of a black-box smart contract, the code behind a mutually programmed smart contract might figure into the interpretation of the parties' agreement. It belongs to the interpretive evidence of that agreement. Third, when interpreting code as evidence of the parties' agreement, the proper question is what the code is designed to do, not how it in fact operates. One must distinguish features

---

<sup>52</sup> See UCC § 1-304 (duty of good faith); Restatement (Second) of Contracts § 205 (1981) (same). For a recent account of the duty of good faith along these lines, with citations to the literature, see Daniel Markovits, *Good Faith as Contracts Core Value*, in *Philosophical Foundations of Contract Law* 272 (Gregory Klass, et al. eds., 2014).

from bugs. Fourth, although both mutually negotiated smart contracts and contractual writings can serve as interpretive evidence, they do so in very different ways. In a mutually programmed smart contract, operational code is akin to course of performance evidence. It does not state the parties' agreement, but shows it. Fifth, code can, however, contain in comments statements of agreement. When both parties had a hand in programming a smart contract, such comments should be treated like any other communications that express or evince some or all of the parties' agreement. Sixth, even if a smart contract fully solves the basic mistrust problem—even if the software locks in both sides' performance—the parties might still benefit from the law of contract. Just as a contractual writing might fail to anticipate or provide for every contingency, so might a smart contract. In addition to solving the basic mistrust problem and addressing deception, the law serves to fill gaps in the parties' agreement and to prevent opportunism. It can play the same role with respect to smart contracts.

#### **4. The Glass-box Vending Machine and Glass-Box Smart Contracts of Adhesion**

The idea of a mutually programmed smart contract is useful for thinking through how smart contracts can figure into legal contracts. But like bespoke contractual writings, mutually programmed smart contracts are expensive to produce. One might for that reason expect them to be relatively uncommon.

The more common type of glass-box smart contract will be software whose code is made available to non-programming users. Glass-box contracts of this type might be created by one user and then given to the other together with access to the code. Or users might obtain the software from a third-party vendor that gives both access to the code. For simplicity's sake, I focus on the first category.

The analog in contractual writings is the contract of adhesion, a contractual writing drafted by one party and given to the other, or more commonly to many others, on a take-it-or-leave-it basis. I will call a smart contract produced by one party and provided the other without an opportunity for modification but with access to the code a "smart contract of adhesion" or "adhesive smart contract." Like adhesive contractual writings, adhesive smart contracts raise distinctive issues of interpretation and enforcement.

##### **4.1 The Glass-Box Vending Machine**

Again I begin with a mechanical example. Suppose Acme takes one of its standard vending machines and replaces the metal exterior with a glass box, giving users a view of the interior. Acme installs this glass-box

vending machine in the faculty lounge of a mechanical engineering department, arranged to provide an unobstructed view of its guts. Like a standard vending machine, use of this glass-box vending machine produces a legal contract. Does the glass box and machine's location make a difference in the terms of that legal contract?

Not necessarily. Suppose that after teaching Mechanics and Materials I, Professor Emmett Brown goes to the faculty lounge and encounters the newly installed glass-box vending machine. Professor Brown is intrigued. But having just delivered a two-hour lecture on multiaxial stress-strain relationships, all he wants is a candy bar. Without examining the machine's guts, Professor Brown puts two quarters into the slot and pulls a handle. Unfortunately, the machine does not deliver a candy bar. Nor when Professor Brown depresses the "Refund" lever does he get his quarters back.

In order to think about how the machine's guts might figure into the content of the legal contract, suppose that nondelivery of the candy bar was caused by a Random No-Candy Generator installed in the machine, which Professor Brown would have recognized had he examined its interior. In other words, had Professor Brown scrutinized the machine's guts, he would have known that Acme was offering not an exchange of money for candy bars, but of money for a chance at receiving a candy bar. Should this make a difference in whether Professor Brown can sue Acme for breach? That is, does the visible presence of a RNCG affect the terms of the legal contract between Acme and Professor Brown?

We are deep in the realm of the hypothetical, but there is a good argument that so far the answer should be "No." Consider the judicial treatment of browsewrap, contractual writings on a website available via hyperlink from other pages on the same site. A browsewrap page typically opens with a pronouncement that by accessing pages that link to it, the user agrees to the terms described therein. The writing then lists terms—damage limitations, warranty provisions, data usage policies, choice of law clauses, arbitration clauses, and the like—in language that suggest that they are terms of a legal contract between the user and the site operator.

Browsewrap is a relatively new form of contractual writing. But there is an emerging judicial consensus that browsewrap can generate a legal contract with the user only if other pages on the website provide sufficient notice of it.<sup>53</sup> Thus the Ninth Circuit has held:

[T]he onus must be on website owners to put users on notice of the terms to which they wish to bind consumers. Given the breadth of the range of technological savvy of online purchasers, consumers

---

<sup>53</sup> See, e.g., *Nguyen v. Barnes & Noble Inc.*, 763 F.3d 1171, 1179 (9th Cir. 2014); *Herman v. Seaworld Parks & Entm't, Inc.*, No. 8:14-CV-3028-T-35JSS, 2016 WL 7447555, at \*5 (M.D. Fla. Aug. 26, 2016); *Hines v. Overstock.com, Inc.*, 668 F. Supp. 2d 362, 367 (E.D.N.Y. 2009).

cannot be expected to ferret out hyperlinks to terms and conditions to which they have no reason to suspect they will be bound.<sup>54</sup>

At a minimum the user must be clearly informed that by accessing the site they will be bound to the browsewrap terms. A mere link is not enough.

Acme's glass-box vending machine does not employ browsewrap. The machine's guts are not a contractual writing. But the reasoning behind the above rule might be extended to the example. It is true that Professor Brown has the ability to "read" the guts of the glass-box machine. He could figure out what the machine is designed to do. In the circumstances, however, he has no reason to expect that the machine's guts are anything other than those of a standard vending machine, and so has no reason to examine them. It is perfectly reasonable for Professor Brown to expect to receive a candy bar in exchange for his payment. The glass-box vending machine's guts should be treated as evidence of the parties' agreement only if the user has sufficient notice that they might differ from those of a standard vending machine. Absent such notice, the legal contract and its terms are largely the same as those of a standard vending machine.

What would constitute sufficient notice in the case of the glass-box vending machine? Just as reasonable minds can differ as to what counts as sufficient notice of browsewrap terms,<sup>55</sup> they can differ in the case of a glass-box vending machine installed in an engineering department. But surely it would be enough if Acme replaced the single word "CANDY" on the front of the machine with "NONSTANDARD VENDING MACHINE." Now when Professor Brown encounters the glass-box vending machine, he should know not to assume it is designed to deliver candy bars for money. And because the machine is encased in glass and placed in an engineering department faculty lounge, he should infer that if he wants to understand what he will receive for his quarters, he should examine its guts. In these circumstances, the proper functioning of an RNCG in the machine would not constitute breach. So long as the mechanism is visible and easily recognizable, someone with Professor Brown's knowledge can be expected to understand that by inserting coins into the machine he is purchasing not a candy bar, but a chance at a candy bar.

Many of the points from Part Three apply *pari passu* to use of the glass-box vending machine's guts as interpretive evidence of the legal contract between Acme and Professor Brown. For example, the machine's

---

<sup>54</sup> Nguyen, 763 F.3d at 1179.

<sup>55</sup> See, e.g., *Nicosia v. Amazon.com, Inc.*, 834 F.3d 220, 237-38 (2d Cir. 2016) (reasonable minds could disagree as to whether Amazon provided the consumer with reasonable notice of contract provisions). Compare *Cullinane v. Uber Techs., Inc.*, 893 F.3d 53, 62-64 (1st Cir. 2018) (finding notice of click wrap terms insufficient), with *Cullinane v. Uber Techs., Inc.*, No. CV 14-14750-DPW, 2016 WL 3751652, at \*7 (D. Mass. July 11, 2016) (concluding that the same app provided sufficient notice).

design would be more salient than its operation. Whereas Acme would be off the hook for the proper operation of a clearly visible RNCG, Acme would still be liable for glitches in the machine—unless it adds a “Use at your own risk” sign. Nor do the machine’s guts fully determine the terms of the legal contract. As with the standard vending machine, implied warranties and other default terms apply, as do mandatory terms such as the duty of good faith.

What if someone without Professor Brown’s expertise uses the machine? Here the application of existing doctrine is less certain. Consider the so-called duty to read: One who agrees to a contractual writing they have not written and have chosen not to read is nonetheless bound by the terms in it.<sup>56</sup> Historically the rule applied even if a party was unable to read the contractual writing. “He should read it if able; or, if illiterate, have it read to him.”<sup>57</sup> One can imagine a court taking the same attitude toward a clearly labelled glass-box nonstandard vending machine. The words “NONSTANDARD VENDING MACHINE” on a device installed in an engineering department are enough to put even the non-expert on notice that it might not be designed to provide candy bars for coins, and that they want to know what they will get for their money, they should seek an engineer’s advice.

That said, the duty to read does not entail that any term in a contract of adhesion becomes part of the legal contract. Everyone knows that contractual writings often go unread, generating the worry that drafters might include unfair, unjust or otherwise socially unacceptable terms.

Scholars and courts have suggested various answers to that worry with respect to adhesive contractual writings. Karl Llewellyn argued that even in transactions between businesspeople, courts should distinguish terms to which the parties actually agreed from those in a preprinted form. “[W]here bargain is absent in fact, the conditions and clauses to be read into a bargain are not those which happen to be printed on the unread paper, but are those which a sane man might *reasonably expect* to find on that paper.”<sup>58</sup> This approach has been applied to insurance contracts. Thus the reasonable expectations rule holds that a court should honor the insured’s “objectively reasonable expectations . . . regarding the terms of insurance contract . . . even though painstaking study of the policy

---

<sup>56</sup> See Restatement (Second) of Contracts § 23 Ill. 7 & cmt. e (1981).

<sup>57</sup> Joel Prentiss Bishop, Commentaries on the Law of Contracts 2d ed 137 (rev. and enl. by Marion C. Early, Chicago, 1907).

<sup>58</sup> K.N. Llewellyn, *Review of The Standardization of Commercial Contracts in English and Continental Law* by O. Prausnitz (London: Sweet & Maxwell. 1937), 52 Harv. L. Rev. 700, 704 (1939) (emphasis added). Llewellyn suggested that courts look to the background of trade practice and other evidence to determine whether any terms in the unread contractual writing were “repugnant to the balanced nature of the *type of transaction* which the parties have obviously entered into,” and that courts should refuse to enforce such terms. *Id.* at 705.

provisions would have negated those expectations.”<sup>59</sup> Alternatively, a court might look to the terms’ substance. California courts have held that use of an adhesive contractual writing can satisfy the procedural prong of the unconscionability doctrine, opening the door for courts to examine the terms’ substantive fairness.<sup>60</sup> The draft Restatement of the Law Consumer Contracts advocates this rule for standard terms in all consumer contracts of adhesion.<sup>61</sup>

One can imagine a court taking either of these approaches to a non-expert’s use of Acme’s glass-box vending machine. Under the first, the question would be whether it is within the non-expert’s reasonable expectations that the machine is offering not candy bars, but the chance at receiving a candy bar. Under the second, the question would concern the substantive fairness of the transaction—for example, whether the prices Acme was charging reflected the chance that a user might not receive a candy bar.

But there is a third option. A court might conclude that a glass box vending machine offered to unsophisticated users is not like a contract of adhesion, and that despite the glass box, its guts should not be included in the interpretive evidence of the parties contractual agreement. Although the words “NONSTANDARD VENDING MACHINE” put the user on notice that something about the machine is different, a user who is not a mechanical engineer does not have the capacity to figure out what that difference is. The proper analogy here is not an adhesive consumer contract written in legal English, but one written in Old Aramaic. Such a contractual writing seeks to impose unreasonably high reading costs both on consumers and on a third-party adjudicator later asked to enforce it. Rather than assigning the consumer a duty to read—“He should read it if able; or, if illiterate, have it read to him”—courts should simply refuse to give legal effect to a consumer contract of adhesion written in a dead language. By the same token, a court called upon to interpret a contract between Acme and an unsophisticated user of its glass-box vending machine might simply refuse to treat the machine’s guts as evidence of their agreement.

The analysis is admittedly speculative. If, improbably, glass-box nonstandard vending machines became common, the law would need to arrive at a rule to resolve when, if ever, their guts figure into the content of legal contracts generated by their use. In doing so, lawmakers would need to balance *inter alia* society’s interest in allowing parties to use whatever

---

<sup>59</sup> Robert E. Keeton, *Insurance Law Rights at Variance with Policy Provisions*, 83 Harv. L. Rev. 961, 967 (1970). A variation on this rule can be found in the relatively uninfluential § 211(3) of the Restatement (Second) of Contract Law. See Klass, *supra* note 47 at 131-33.

<sup>60</sup> See, e.g., *Ferguson v. Countrywide Credit Indus., Inc.*, 298 F.3d 778, 783-84 (9th Cir. 2002).

<sup>61</sup> Restatement of the Law Consumer Contracts § 5 cmt. 6(c) (Am. Law Inst., Tentative Draft April 18, 2019).

means they wish to arrive at contractual agreements—words, actions or even the design of a machine—against its interest in ensuring that people know what they are getting when they enter into a legal contract.

#### 4.2 Glass-Box Smart Contracts of Adhesion

Again many of the conclusions from the vending-machine thought experiment apply also to smart contracts, though there are important differences between the two cases and additional legal questions that merit consideration. With respect to the legal questions, this is a natural point to consider how much control the law does or should give parties over whether the code of a smart contract figures into the terms of their legal contract, and if so how.

Suppose Acme decides to diversify its operations by getting into the cryptocurrency futures market. Rather than using an existing platform, Acme creates a website on which it offers to the public futures in various cryptocurrencies (Bitcoin, Ripple, Litecoin, Ether). In order to effectuate the transactions, Acme writes a smart contract that is stored on a distributed ledger, with the ledger effectively acting as the exchange. Acme's website explains that to execute a transaction, a user must first provide the smart contract access to the user's cryptocurrency wallets. The website then instructs users how to access the distributed ledger and accept one of Acme's offers. Once a user accepts an offer, the software records the transaction, waits until the date of trade, then executes the agreed-upon exchange. Thus if a user on the first of January accepts Acme's offer to sell one Bitcoin for fifty-two Ether on the first of July, on July 1 the software will automatically execute a transfer of one Bitcoin from Acme's wallet to the user's, and a transfer of fifty-two Ether from the user's wallet to Acme's. Suppose finally that Acme's website contains a clearly labeled link to the code on the distributed ledger. That is, the site employs a glass-box smart contract of adhesion.

As in the earlier examples, use of Acme's smart contract solves the basic mistrust problem, doing some of the work of a legal contract. In this instance, the smart contract solves the problem by giving users access to code run on a tamper-proof distributed ledger. Also like the earlier examples, use of the smart contract nonetheless generates a legal contract. By using the smart contract, a user expresses their agreement to the offered exchange.

Although Acme gives users access to the code, we should not be too quick to assume that a court would or should treat the code as interpretive evidence of the legal contract between Acme and its users. The situation is comparable to the glass-box vending machine. Until the glass-box vending machine was relabeled as "NONSTANDARD," Professor Brown has no reason to suppose that its guts were anything other than those of a typical vending machine, and so no reason to examine them. Similarly,

users of Acme’s website, regardless of their sophistication, have no reason to expect that Acme’s smart contract is designed to do anything other than what they reasonably believe based on Acme’s website. In fact, users have even less reason to examine the code than did Professor Brown. Whereas the glass-box vending machine contained few if any words, Acme’s website is verbose. It includes a natural-language description of the proposed transactions. Users have access to the code, but they have no reason to examine its design or operation, or to read any comments contained in it. Consequently, the code should not affect the terms of the legal contracts formed using it.

The thought experiment suggests two general points. First, although some have suggested the possibility of “code-only” smart contracts,<sup>62</sup> it is difficult to imagine multiple parties interacting through a piece of software without some natural-language communications concerning its function and purpose.<sup>63</sup> That is, smart contracts will almost always come “wrapped” in natural language communications describing the nature of the transaction, which will serve as interpretive evidence of the parties’ agreement.

Second, where there are such communications, an adhesive smart contract’s code should not, absent special reason, figure into the interpretive evidence, even if it is made available to the user. This distinguishes adhesive smart contracts from mutually programmed smart contracts. Whereas the code of a mutually programmed smart contract always evinces the parties’ agreement, the default for adhesive smart contracts should be that the code does not belong to the interpretive evidence.

What words should suffice to bring the code into the interpretive evidence? Suppose Acme adds the following text to the page on which users enroll for trades:

By trading on this website you agree to abide by all transactions executed by our smart contract. You can find the smart contract [here](#). Please examine the code closely to ensure that you understand its design.<sup>64</sup>

---

<sup>62</sup> See, e.g., Levi & Lipton, *supra* note 20 at 2 (distinguishing between “code-only smart contracts” and “ancillary smart contracts”); UK Jurisdiction Taskforce, *supra* note 34 at 35 (considering “a smart contract consisting solely of code with no natural language element”). The UK Jurisdiction Taskforce, for example, imagines a scenario in which “Alice deploys some code to a distributed ledger platform,” and “Bob, who is not known to Alice, stumbles across the code and transacts with it.” *Id.* at 34.

<sup>63</sup> See Cohney & Hoffman, *supra* note 4 at 363 (“Any scripts that have practical relevance will have some non-code language surrounding them.”).

<sup>64</sup> A sophisticated drafter of such a clause might replace “design” in the above clause with “design and operation.” The additional words would, if effective,

The word “here” is a hyperlink to Acme’s smart contract. Such a clause would be similar to a contractual writing that seeks to incorporate the contents of another writing. An arbitration clause, for example, might not describe how arbitration will work, but stipulate that it shall proceed in accordance the rules of the American Arbitration Association. The above clause on Acme’s website similarly seeks to bring the design of the glass-box smart contract into the terms of the legal contract. Like the words “NONSTANDARD VENDING MACHINE,” it puts users who read it on notice that they should not rely on expectations they might otherwise have about what they are purchasing. But the clause does more. It attempts to stipulate that that by using Acme’s smart contract, users are agreeing not only to transactions as they are described on the website, but also to the software’s design.

The above text differs from standard incorporation clauses in that it calls not on another writing, but on a piece of technology. In order to fully understand the terms of the legal contract, the user must read the code. To mark this distinction, I shall call language that seeks to add a smart contract’s code to the interpretive evidence of the parties’ agreement a “smincorporation clause.” If a smincorporation clause is effective, the terms of the legal contract depend in part on the code of the associated smart contract.<sup>65</sup>

Would or should courts give legal effect to a smincorporation clause such as the above? As with contracts of adhesion, the answer should turn in part on user sophistication.<sup>66</sup> Suppose Acme offers no trades valued at less than a million dollars, and that its website targets only professionals who

---

operate like “as is” or “Use at your own risk,” shifting risks of software bugs to the user. The example is kept simple because courts might not, and arguably should not, apply the same rules to risk-shifting clauses as to clauses that seek to incorporate a smart contract’s design.

<sup>65</sup> Although the possibility of smincorporation has not been widely recognized, the International Swaps and Derivatives Association’s 2019 Guidelines for Smart Derivatives Contracts recommend that drafters consider whether to smincorporate code:

It may . . . be necessary for the legal documentation to expressly declare that the code forms part of the single agreement as created by the ISDA documentation. Consideration should be given to whether this would be limited to records generated by operation of the code, or whether it should actually attempt to capture all actions performed by the code.

ISDA Legal Guidelines, *supra* note 20 at 22.

<sup>66</sup> See Chamber of Digital Commerce, *12 Use Cases*, *supra* note 36 at 41 (“As with clickwrap agreements, courts will need to develop standards to determine when a smart contract or term therein will be enforced, and they will likely employ notice as a key determinant. In the case of smart contracts where the “code is the contract,” parties may experience more difficulty proving that they provided notice of terms contained in the code, especially with less sophisticated customers.”).

have the resources to understand the code. In that case, it would not be unreasonable to enforce the clause and treat the design of the code as part of the interpretive evidence. Smincorporation allows Acme to shift resources away from explaining on the website every aspect of its offer and toward improving the quality of the code or offering users a better price.

What if Acme were to offer low-value trades and target the general public? Unlike the words “NONSTANDARD VENDING MACHINE,” Acme’s smincorporation clause expressly tells users to look to the code’s design to understand what they are purchasing. As such, courts might be more open to enforcing it, and so to treating the code as part of the interpretive evidence.

A court taking this approach might additionally adopt either or both of the protective rules mentioned above: reasonable expectations and substantive unconscionability. Under the first, code that violates an unsophisticated user’s reasonable expectations would not be included in the interpretive evidence. Unexpected design elements might then breach the legal contract, despite Acme’s attempt to smincorporate the code. Under the second, substantively unfair or unreasonably one-sided aspects of the code’s design would not figure into the interpretive evidence of the legal terms, and their operation might again constitute breach.

It is also possible, however, that courts would—and I believe should—take the third path described above. When given to unsophisticated users, courts should decline to give a smincorporation clause legal effect and refuse to treat the smart contract code as interpretive evidence of the legal contract. Acme’s clause asks unsophisticated users to agree to code they would be unable to understand even if they took the time to examine it. To treat the code as part of their agreement would involve not only a legal fiction—that users read standard terms—but science fiction—an assumption that users have capacities that we know they do not possess. Rather than imposing on nonsophisticated parties the costs of noncomprehension, it is fairer and more efficient to require that Acme explain terms in language its target users can understand.

From this point forward, assume that only sophisticated traders use Acme’s glass-box smart contract of adhesion, that Acme’s website includes the above smincorporation clause, and that, because the users are sophisticated, the clause is legally effective. We have now arrived at a second example in which a smart contract’s code figures into the interpretive evidence of an associated legal contract—in which code partly determines legal terms. In the case of a mutually programmed smart contract, the code belonged to the interpretive evidence because it was part of the course of performance, and as such was highly probative of the parties’ actual agreement. In a smincorporated glass-box contract of adhesion, the code belongs to the interpretive evidence because the parties have so stipulated. They have agreed to bind themselves to the code’s design.

I have argued that the code of a mutually programmed smart contract constitutes course of performance evidence. Cohney and Hoffman suggest that this is true of smart contract code generally.<sup>67</sup> This is incorrect when parties have smincorporated an adhesive smart contract. Smincorporating code is more like choosing to memorialize part of the agreement in a technical argot or usage of trade.<sup>68</sup>

Smincorporation of the smart contract does not entail that the code is the *only* evidence of the parties' contractual agreement and the terms of their contract. In the example, Acme's website contains many representations about how the system operates and what users can expect. These various forms of evidence might in some instances point in different directions. Suppose, for example, Acme's website states that certain transactions shall be executed at 5:00 pm Eastern on the date of sale, but the code is designed to execute at midnight. How such a conflict in the interpretive evidence be resolved?

Contract law gives courts several tools to handle such situations. One is to engage in an all-things-considered analysis of how a reasonable user would likely understand the interpretive evidence.<sup>69</sup> A court might conclude, for example, that users are more likely to treat the words on Acme's website as dispositive than they are the code. Alternatively, a court might apply the rule of *contra proferentem*, which preferences interpretations against the drafter's interests.<sup>70</sup> Because Acme created both its website and smart contract, that rule would suggest whichever reading is preferred by the nondrafting user.

Acme might seek to avoid these results by adding additional language to its smincorporation clause. A standard integration clause stipulates that the contractual writing in which it appears is the final, and therefore legally dispositive, statement of some or all of the parties' agreement. When effective, an integration clause limits parties' ability to rely on evidence from outside the writing—"parol evidence"—to identify the terms of their legal contract. Thus Cohney and Hoffman suggest that the drafter of a contractual writing might integrate it against contrary elements in an associated smart contract's code.<sup>71</sup> And the International Swaps and Derivatives Associate recommends "include[ing] some provision within the smart derivatives contract stating that the natural language version of the contract will prevail in the event of any inconsistencies."<sup>72</sup> (Such clauses

---

<sup>67</sup> Cohney & Hoffman, *supra* note 4 at 368.

<sup>68</sup> "Usage of trade" is sometimes used to refer to common practices within an industry, and at other times used to refer to special meanings words have in an industry.

<sup>69</sup> See Restatement (Second) of Contracts § 202 (1981).

<sup>70</sup> See Restatement (Second) of Contracts § 206 (1981).

<sup>71</sup> Cohney & Hoffman, *supra* note 4 at 390.

<sup>72</sup> ISDA, *Legal Guidelines for Smart Derivatives Contracts: The ISDA Master Agreement* 33 (February 2019). The ISDA suggests that if drafters do not take this

would not be necessary if there were a clear legal default that smart contracts of adhesion do not serve as interpretive evidence of the parties' agreement.)

What if Acme wants the code to control? It might then add the following notice to its enrollment page:

By enrolling on this website you agree to abide by all transactions executed by our smart contract, *even if those transactions are contrary to Acme's statements or representations on this website or elsewhere*. You can find the smart contract [here](#). Please examine the code closely to ensure that you understand its design.

The italicized language is akin to a integration clause, as it aims to limit evidence of the parties agreement. But rather than identifying a contractual writing as dispositive, it designates the smart contract's code as such. In order to mark the difference, I call this a "smintegration clause."<sup>73</sup>

Although integration clauses are widely used, they are not always effective. The Draft Restatement of the Law of Consumer Contracts, for example, suggests that a business's attempt to integrate its standard terms—the small print few if any consumers read—will not exclude evidence of the business's contrary consumer-directed representations.<sup>74</sup>

Since the standard contract terms do not result from a combined effort by both parties to draft a negotiated agreement, there is less justification to view them as a joint affirmative memorialization of a mutually designed agreement, and thus less reason to allow them to override affirmations of fact or promises made to the consumer.<sup>75</sup>

---

route, they should include a provision "ensuring there is some mechanism in place to confirm, to the extent necessary, that the legal effect of any coded part of the smart derivatives contract has been appropriately validated by lawyers." *Id.*

<sup>73</sup> The possibility of integrating code against other interpretive evidence is not widely recognized in the literature, but see Lingwall & Mogallapu, *supra* note 21 at 299 (suggesting that "the parties may use code to express the *entire* agreement, so that the code becomes the integrated contract").

<sup>74</sup> "A standard contract term that contradicts, unreasonably limits, or fails to give the reasonably intended effect to a prior affirmation of fact or promise by the business does not constitute a final expression of the agreement regarding the subject matter of that term and does not have the effect under the parol evidence rule of discharging obligations that would otherwise arise as a result of the prior affirmation of fact or promise." Restatement of the Law of Consumer Contracts § 8 (Am. Law Inst., Tentative Draft, April 18, 2019).

<sup>75</sup> *Id.* cmt. 1. See also *id.* cmt. 3 ("Because consumers are not likely to notice, read, or understand the effect of . . . merger clauses, they do not control the conclusion of whether the standard contract terms constitute a partially or completely integrated agreement, and thus do not preclude a finding that the standard contract terms do not constitute the parties' final expression of a particular matter.").

I have stipulated that Acme’s glass-box smart contract targets not consumers, but sophisticated investors. Nonetheless, a court might apply similar reasoning to Acme’s smintigration clause. Whereas an integration clause stipulates which communicative acts figure into the interpretive evidence, a smintigration clause attempts to stipulate that software design decisions can exclude evidence of contrary express commitments. A court might question whether the design of a smart contract of adhesion should be permitted to override express natural language affirmations or commitments made to users of it. Nor is smintigration comparable to an “as is” or “Use at your own risk.”<sup>76</sup> Whereas those contractual mechanisms allocate the risk of unknown or unintended departures from the parties’ express agreement—bugs—a smintigration clause seeks to shift the risk of departures that one side built into the design of its smart contract.

That said, once can also imagine courts giving legal effect to a smintigration clause between sophisticated parties. Such parties might have good reasons for wanting to smintigrate a glass-box contract of adhesion. Judicial interpretation of natural language is neither predictable nor perfectly accurate. If both parties know the code, a smintigration clause might reduce the risk that a court will later misinterpret their communications. And a party who has not written the code might have other reasons, such as reputation and repeat play, to trust that the code is designed to do what that party expects. If these grounds provide the nonprogramming party all the assurances it needs, smintigration can serve to prevent courts from later clouding the waters by misinterpreting one or both parties’ other statements or actions.<sup>77</sup> Lastly, smintigration does not extinguish all legal protections against code that does not conform to communications. A party who intentionally programs a smart contract in ways that are contrary to other representations commits promissory fraud, against which the parol evidence rule is no defense.<sup>78</sup> Anglo-American contract law gives parties a remarkable degree of control over the framework rules that govern the construction and enforcement of their legal agreements. Courts might well extend that attitude to sophisticated parties’ attempts to smintigrate smart contracts.

---

<sup>76</sup> For thoughts on how a smintigration clause might be drafted to also allocate the risk of errors in the code, see *supra* note 64.

<sup>77</sup> The point here is parallel to observations about how non-contextual approaches to contractual writings can, by rendering legal contracts more predictable and less malleable, enable parties’ reliance on extra-legal forms of trust. See Lisa Bernstein, *Merchant Law in a Merchant Court: Rethinking the Code’s Search for Immanent Business Norms*, 144 U. Pa. L. Rev. 1765, 1787-95 (1996); Robert E. Scott, *Conflict and Cooperation in Long-Term Contracts*, 75 Cal. L. Rev. 2005, 2050-53 (1987).

<sup>78</sup> See Ian Ayres & Gregory Klass, *Promissory Fraud without Breach*, 2004 Wis. L. Rev. 507, 520-22 (discussing actions for promissory fraud where an action for breach is precluded by the parol evidence rule).

But even if they do, the rules governing attempts to smintigrate smart contracts should not track those for the integration of contractual writings. Many U.S. courts have held that a contractual writing that does not contain an integration clause might nonetheless be found to be integrated if it appears to be a final statement of some or all of the parties' agreement.<sup>79</sup> The best argument for that rule is that when parties sign a contractual writing that appears to be legally binding, they should understand that it has a special legal effect—that it is likely to exclude parol evidence of contrary or additional terms. It is difficult to imagine a court finding the implied smintigration of a glass-box smart contract of adhesion. The reason should by now be familiar. The mere fact that a user is given access to a smart contract's code does not put them on notice that its design or operation is likely to displace other evidence of the parties' agreement. Code should displace other interpretive evidence only when the parties have expressly agreed that it shall.

One should also keep in mind that, for all its power, the parol evidence rule concerns only the evidence that goes into interpreting the parties' contractual agreement. As I have emphasized, the terms of a legal contract are rarely coextensive with the parties' agreement. Default terms, such as implied warranties, apply absent evidence of the parties' contrary agreement. And mandatory terms, such as the duty of good faith, apply even in the face of the parties' contrary agreement. And other aspects of the transaction might give rise to a defense such as mistake, duress, or misrepresentation. All this mean that no matter how completely parties are able to smintigrate their glass-box smart contract, the code will not fully determine the resulting legal state of affairs. Contract law is designed to give parties the terms they choose. But that is not all it is designed to do.

I have so far been discussing attempts to smincorporate operational sections of the code into the parties' legal contract, or to smintigrate operational code against contrary interpretive evidence. Software can also contain nonoperational comments. Should one party be able to smincorporate or smintigrate programmer comments contained in a glass-box smart contract of adhesion?

Unlike operational sections of the code, programmer comments are communicative acts, typically formulated in natural language. Anglo-

---

<sup>79</sup> See, e.g., *Vanhook Enterprises, Inc. v. Kay & Kay Contracting, LLC*, 543 S.W.3d 569, 572 (Ky. 2018) (“As a matter of law, a document which on its face appears to be a complete integration is a complete integration.”); *Coll v. PB Diagnostic Systems, Inc.*, 50 F.3d 1115, 1123 (1st Cir. 1995) (“Where the parties reduce an agreement to a writing which in view of its completeness and specificity reasonably appears to be a complete agreement, it is taken to be an integrated agreement unless it is established by other evidence that the writing did not constitute a final expression.”); *Bellman v. Am. Int’l. Grp.*, 865 N.E.2d 853, 857 (Ohio 2007) (“A contract that appears to be a complete and unambiguous statement of the parties’ contractual intent is presumed to be an integrated writing.”).

American contract law imposes few requirements on how parties express their contractual agreement. (Contractual writings aimed at consumers are more likely to be subject to typeface or other requirements.<sup>80</sup>) There is no principled reason why the creator of a glass-box smart contract given to sophisticated users could not, if it provided sufficient notice, include legally effective comments in the code. Sufficient notice is, however, key. Because parties are unlikely to expect aspects of their contractual agreement to appear in comments buried in the code, such comments should be given legal effect only where there has been clear and presumptively effective notice of them—when they have been expressly incorporated in communications outside the code. Similarly, an attempt to integrate programmer comments should be effective only if it appears outside the code.

These notice rules need not be written in stone. Contemporary judicial treatments of browsewrap, for example, reflect cultural practices and understandings, which might change over time. Suppose that as browsewrap becomes increasingly common, users come to expect (correctly or not) that by accessing a website they will be legally bound to unread browsewrap terms, no matter the quality of the notice.<sup>81</sup> If common cultural understandings shift in this direction, courts will have a new reason to hold that using a webpage that links to browsewrap expresses the user's assent to the browsewrap terms, no matter the quality of notice. If that happens, browsewrap will have become a self-fulfilling prophecy.

One can imagine a similar shift in the common understanding of using a glass-box smart contract of adherence. To return to Cohney and Hoffman's suggestion, perhaps someday the mere use of a glass-box smart contract, together with a link to the code, might be widely understood to express the user's agreement to the content of that smart contract—to the operations the code is designed to perform and to any contractual writings embedded as comments in it. In that case, courts might not require express incorporation of the code.

Perhaps. But such a change would involve a much more radical shift in common understanding than a new rule for browsewrap. Browsewrap is a type of contractual writing. It is not merely evidence of the parties' agreement, but says what that agreement is. To repeat a point I have been emphasizing throughout, a glass-box smart contract is first and foremost a mechanism for doing something. Although one can imagine a

---

<sup>80</sup> See, e.g., N.Y. C.P.L.R. 4544 (McKinney 2020) (providing that the drafter of a printed consumer contract or residential lease may put the document into evidence only if the print is clear and legible and in 8-point type or larger).

<sup>81</sup> This change in common understanding might happen whether or not courts treat browsewrap as legally enforceable. Some studies suggest that consumers often believe themselves to be bound by terms that are legally unenforceable. See, e.g., Tess Wilkinson-Ryan, *A Psychological Account of Consent to Fine Print*, 99 U. Iowa L. Rev. 1745 (2014).

world in which code is impliedly smincorporated into the parties legal contract, or even impliedly smintegrated against other interpretive evidence of the parties actual agreement, it would be radically different from the one we live in.

The results of this section can be summarized as follows. First, the fact that users have access to the code is not sufficient reason to treat the contents of a glass-box smart contract of adhesion as interpretive evidence of the parties' contractual agreement. Mere access to the code does not put users, regardless of their sophistication, on notice that its design or comments contained in it might affect their legal rights and responsibilities. This distinguishes glass-box smart contracts of adhesion from mutually programmed smart contracts, as the latter always qualify as evidence of the parties' agreement. Second, in transactions between sophisticated parties we might expect courts to enforce parties' express agreement to incorporate, or to smincorporate, the contents of a glass-box smart contract of adhesion. Smincorporation might apply to both the code's design and to contractual comments imbedded in it. There are, however, good reasons to reject attempts at smincorporation in transactions with nonsophisticated parties. Third, by the same token courts might enforce sophisticated parties' attempts to integrate, or smintigrate, the contents of their smart contract against other interpretive evidence of their contractual agreement. That is, a court might enforce an agreement between sophisticated parties that their legal obligations shall be determined in the first instance solely by the code's design and comments, despite other contrary evidence. Again, agreements with nonsophisticates should be treated differently. Fourth, unlike the integration of a contractual writing, both the smincorporation and the smintigration of a smart contract should require the parties' express agreement. Fifth and finally, even when a smart contract is fully smintegrated, the parties' legal contract is likely to include terms that are not expressed in its design or comments. The terms of a legal contract are rarely coextensive with the parties' agreement, including when they agree to be bound by a piece of software.

## **5. Smart Contracts and Legal Contracts**

Parts Two, Three and Four have gone fairly deep into the doctrinal weeds. Having explored those details, it is now possible to step back to examine the relationship between smart contracts and legal contracts generally.

Many scholars have emphasized the technical limits of smart contracts. It is difficult to imagine how a smart contract might be programmed to implement familiar standards such as reasonableness, good

faith or due diligence, which contracting parties often wish to employ.<sup>82</sup> Systems that implement code do not have the resources to recognize or fill gaps in an agreement.<sup>83</sup> The practical inevitability of programming errors means that parties should never rely solely on the operation of code to secure intended outcomes.<sup>84</sup> Conditioning software on events in the world requires the use of oracles—software or hardware that provides inputs from outside the smart contract—creating points of possible failure and exploitation.<sup>85</sup> When a smart contract is run on blockchain, it may be difficult to modify or terminate.<sup>86</sup> And the out-of-pocket costs of running code on a distributed ledger could mean that only the simplest of smart contracts can practicably employ the technology.<sup>87</sup>

These technical limitations are important. But there is a broader point to be made. Smart contracts are designed to do only one of the many jobs of a legal contract: solving the basic mistrust problem. An exclusive focus on this function reflects an anemic understanding of what contract law does and suggests a false picture of the forms of sociability to which contract law applies.

This Part argues that point by first reexamining the story of the 2016 DAO project (hereinafter “The DAO”) and its subsequent hack. The story is often told to illustrate the technical limitations of smart contracts. Its plot elements include software bugs, blockchain inflexibility, and the costs of anonymity. But The DAO also embodied a simplistic understanding of what legal contracts do, and reflected the misguided ambitions of some smart contract enthusiasts. The second section distinguishes the form of prepackaged coordination that characterized The DAO, and that appears in smart contracts generally, from forms of sociability common in contractual transactions. The third section discusses the multiple functions of contract law—beyond solving the basic mistrust problem—and argues that they suggest a cautious approach to the idea of using smart contracts to supplant rather than supplement legal contracts.

---

<sup>82</sup> Chamber of Digital Commerce, *12 Use Cases*, *supra* note 36 at 43; Werbach & Cornell, *supra* note 2 at 365; Lipshaw, *supra* note 20 at 30-51; Raskin, *supra* note 20 at 326; Sklaroff, *supra* note 28 at 279-84.

<sup>83</sup> Rodriguez, *supra* note 19 at 682.

<sup>84</sup> Cohney & Hoffman, *supra* note 4 at 337, 358; Adam Kolber, *Not-So-Smart Blockchain Contracts and Artificial Responsibility*, 21 *Stan Tech. L. Rev.* 198, 204 (2018); Werbach, *supra* note 22 at 515-16; Werbach & Cornell, *supra* note 2 at 365.

<sup>85</sup> Levi & Lipton, *supra* note 20 at 4-5.

<sup>86</sup> Chamber of Digital Commerce, *12 Use Cases*, *supra* note 36 at 43; Kolber, *supra* at 205-14; Levi & Lipton, *supra* note 20 at 6; Werbach & Cornell, *supra* note 2 at 367; Sklaroff, *supra* note 28 at 284-86.

<sup>87</sup> Cohney & Hoffman, *supra* note 4 at 335-41.

## 5.1 The Parable of The DAO

The DAO (decentralized autonomous organization) was an organizational form embedded in a smart contract run on the Ethereum blockchain and was designed to create a crowd-funded venture capital fund that would operate without fund managers or a centralized governing body.<sup>88</sup> Funds in the form of a virtual currency, Ether, were to be raised through the sale of tokens to the public, permitting token holders to participate in the fund’s management. Once funding was complete, token holders—whom the technology allowed to remain anonymous—were to have voting rights in decisions regarding which projects The DAO would fund. Because the code was run on a distributed ledger, it could be read by those with the technical skill to do so and was nearly impossible to alter once put into motion. Over the course of a 29-day token sale in May 2016, The DAO raised from thousands of investors approximately 12 million Ether, which at the time had a value of \$150 million. The project collapsed the next month when one or more users exploited a vulnerability in the code to siphon off Ether worth in the neighborhood of \$55 million.

The DAO was a glass-box smart contract of adhesion. Many or most of the investors did not participate in writing the code, but had access to it through the distributed ledger. The DAO is especially interesting in this context not only because of its epic hack, but also for the several ways its creators attempted to determine the smart contract’s legal effects. Part Four identified two mechanisms the provider of a glass-box smart contract of adhesion might use to attempt to alter codes’ effect on a legal contract: smincorporation and smintegration. The documentation surrounding The DAO attempted both, plus a third.

The DAO came with two sets of salient contractual writings: a website describing the project and a readme file bundled with the code. The DAO website included a browserwrap terms page. At the top of the landing and other pages was a link labelled “Terms,” which directed users to a webpage titled “Terms: Explainer of Terms and Disclaimers.” The first paragraph of that Terms page reads as follows:

The terms of The DAO Creation are set forth in the smart contract code existing on the Ethereum blockchain at [address]. Nothing in this explanation of terms or in any other document or communication may modify or add any additional obligations or guarantees beyond those set forth in The DAO’s code. Any and all explanatory terms or descriptions are merely offered for educational

---

<sup>88</sup> This article’s account of The DAO draws primarily from the Security and Exchange Commission’s Report of Investigation Pursuant to Section 21(a) of the Securities Exchange Act of 1934: The DAO, Release No. 81207 (July 25, 2017), available at <https://www.sec.gov/litigation/investreport/34-81207.pdf> (last visited May 12, 2020).

purposes and do not supercede or modify the express terms of The DAO's code set forth on the blockchain; to the extent you believe there to be any conflict or discrepancy between the descriptions offered here and the functionality of The DAO's code at [address], The DAO's code controls and sets forth all terms of The DAO Creation.<sup>89</sup>

Several paragraphs later, the page addresses the same topic more expansively.

The DAO's smart contract code governs the Creation of DAO tokens and supercede any public statements about The DAO's Creation made by third parties or individuals associated with The DAO, past, present and future.<sup>90</sup>

These are combined smincorporation and smintigration clauses. The first is the more limited one. It seeks to smintegrate the code against contrary statements on The DAO website. The second seeks to smintegrate the code against all other communications. In other words, the second clause seeks to stipulate that the smart contract shall be the only interpretive evidence of the parties' contractual agreement.

The Terms page is not a model of legal precision. Although it states that the website does not create any contractual relations, it also purports to effect a number of legal changes that are contractual in nature.<sup>91</sup> More interesting for the present purposes, however, is the readme file bundled

---

<sup>89</sup> *Explanation of Terms and Disclaimer*, DAOHUB, <https://daohub.org/explainer.html>, archived at INTERNET ARCHIVE, <https://web.archive.org/web/20160704190119/https://daohub.org/explainer.html> (captured on Jul. 4, 2016) (last visited May 10, 2020).

<sup>90</sup> *Id.*

<sup>91</sup> The Explainer of Terms and Disclaimers page states, for example, that by clicking "I Accept," the user agreed to "tak[e] part in The DAO's Creation under the terms set forth in The DAO's smart contract code at [the user's] own risk," and that if the user is acting on behalf of an entity, the user warrants that the user has "legal authority to bind that entity to these terms as set forth in The DAO's smart contract code." The page then lists a number of representations and warranties that the user makes "[b]y interacting with The DAO's smart contract code." And in two allcaps paragraphs, it provides that the user expressly agrees to use The DAO at the user's sole risk, without any express or implied warranties, and "to the fullest extent permitted by any applicable law" to "not hold third parties or individuals associated with the DAO creation liable for any and all damages or injury whatsoever caused by or related to use of, or inability to use, DAO tokens or the DAO platform."

with the smart contract on the distributed ledger.<sup>92</sup> This file contained several provisions that sought to go beyond smincorporation and smintegration.

Like the Terms page, the readme file purports to effect several contractual changes. For example, it grants users a GNU license to the code and disclaims any warranties. At the same time, several clauses appear to stipulate that use of The DAO will not produce a legal contract. Buried in a section disclaiming any legal advice one finds two especially interesting sentences: “Your use of the Software does not, in and of itself, create a legally binding contract in any jurisdiction.” And: “You agree that in order for you to form a legally binding contract that you shall seek legal advice from an appropriately qualified and experienced lawyer within your jurisdiction.”<sup>93</sup> Along the same lines, the readme file elsewhere provides:

Although the word “contract” is used in the DAO code, the term is a programming convention and is not being used as a legal term of art. The DAO does not practice law and the authors of this code do not practice law are not your lawyers. The term is a programming convention, not a representation that the code is in and of itself a legally binding and enforceable contract.

Taken together, these provisions read like a TINALEA (“This is not an enforceable agreement”) clause: a written provision that seeks to opt-out of legal enforcement altogether.

One often finds TINALEA clauses in letters of intent or preliminary agreements—documents generated in the course of negotiations and used to clarify points of agreement and outstanding issues. Adding a TINALEA clause to such documents permits the parties to generate the written clarification while maintaining the ability to back out of the deal. Another common site of TINALEA clauses is employee handbooks. Here the employer seeks to communicate its policies without effecting a change to its employment contracts.

The TINALEA provisions in The DAO readme file try to do something very different. Rather than excluding a writing from the interpretive evidence of an anticipated or an existing legal contract, these clauses attempt to opt-out of legal enforcement altogether. The Terms page states that the only contractual commitments are those in the smart contract, and the smart contract states that its “use . . . does not, in and of itself, create a legally binding contract.” Whereas smincorporation and smintegration clause seek to affect how the code will figure into the interpretive evidence of legal contract, these provisions seek to replace the

---

<sup>92</sup> Stephan Tual, *Updated Readme*, GITHUB: SLOCKIT/ DAO (Apr. 11, 2016), <https://github.com/slockit/DAO/commit/aceec3efcc8afd4277396ebc42628f2e5ca8dff2#diff-04c6e90faac2675aa89e2176d2eec7d8> (last visited May 10, 2020).

<sup>93</sup> *Id.*

legal contract with a smart contract. I suggest calling contractual writings that state that the use of a smart contract shall not generate a legal contract “SMINALEA clauses” (“[This] smart contract is not a legally enforceable agreement”).

Section 21 of the Second Restatement provides that a TINALEA clause “*may prevent the formation of a contract.*”<sup>94</sup> The comments explain that an attempt to opt-out of legal enforcement might not be effective “where a bargain has been fully or partly performed on one side, a failure to perform on the other side may result in unjust enrichment,” or where the clause is determined to be “against public policy because it unreasonably limits recourse to the courts or as unconscionably limiting the remedies for breach of contract.”<sup>95</sup> There is, however, relatively little caselaw in this area. And there are as of yet no judicial decisions on the effects of SMINALEA clauses.

It would be interesting to think through whether courts would or should give effect to an unambiguous SMINALEA clause. More salient to my purposes, however, is a different question: What do participants in a smart contract stand to gain and to lose from opting out of legal enforcement?

To repeat a point I have been emphasizing, both legal contracts and smart contracts can be used to solve the basic mistrust problem. The DAO smart contract sought to solve it by locking users into a piece of software that they could read but not alter. The thought experiments in Parts Two through Four have identified two other salient functions the law serves that relate to potential breaches of trust. Laws of deception provide assurances of honesty, especially at the time of formation. And some default and mandatory obligations, such as the duty of good faith, provide assurances that parties will not engage in opportunism. A smart contract designed to solve the basic mistrust problem—say by running accessible code on a distributed ledger—might not be capable of serving these other functions.

The DAO hack illustrates the point. Although the readme file explained that “[t]his code been reviewed by hundreds of pairs of eyes from our community and by one of the most respected auditing company in the world,”<sup>96</sup> the code in fact contained a bug that permitted one or more anonymous users to transfer \$55 million from investment pool to their own accounts. This is a classic form of opportunism. The hacker or hackers took advantage of a flaw in the code to extract from the transaction a benefit that was not part of the bargain at the expense of other participants.

---

<sup>94</sup> Restatement (Second) of Contracts § 21 (1981).

<sup>95</sup> *Id.* cmt. b. A TINALEA clauses might also be ineffective when the agreement qualifies as a partnership, such as in The DAO. See Rodriguez, *supra* note 19 at 706-07.

<sup>96</sup> *Updated Readme, supra* note 92.

The flaw in The DAO code is comparable to an error in a contractual writing.<sup>97</sup> In fact, when, prior to the hack, The DAO programmers unsuccessfully attempted to fix the bug, they included in-line commentary explaining that the change was intended to prevent just the sort of hack that later occurred.<sup>98</sup>

Contract law provides multiple tools for dealing with such errors. If the mistake originated in the transcription of the parties' agreement into a contractual writing, a so-called "scrivener's error," a court might reform the writing.<sup>99</sup> Standard rules of interpretation also provide protections. The words in a contractual writing are always "read as a whole to determine its purpose and intent."<sup>100</sup> Thus Samuel Williston—as ardent a textualist as any—explained in the first edition of his treatise that

in giving effect to the general meaning of a writing particular words are sometimes wholly disregarded, or supplied. Thus "or" may be given the meaning of "and," or vice versa, if the remainder of the agreement shows that a reasonable person in the position of the parties would so understand it.<sup>101</sup>

Another tool for dealing with errors is the mistake defense. A material mistake of fact by one or both parties can give the disadvantaged side the power to rescind the contract and bring an action for unjust enrichment.<sup>102</sup> The unconscionability doctrine also can be used to prevent the enforcement of terms that involve unfair surprise and are unreasonably one-sided.<sup>103</sup> And as explained above, the mandatory duty of good faith can

---

<sup>97</sup> Consider the casebook staple, *Market Street Associates Ltd. Partnership v. Frey*, 941 F.2d 588 (7th Cir. 1991), in which a lessee attempted to use a paragraph in a 20-year-old lease to purchase a property at far below market value, perhaps based on the lessor's failure to attend to the terms of the lease. Todd Rakoff has argued that the situation resulted from the failure to require notice of the relevant lease provision. Todd D. Rakoff, *Good Faith in Contract Performance: Market Street Associates Ltd. Partnership v. Frey*, 120 Harv. L. Rev. 1187, 1192–94 (2007).

<sup>98</sup> Cohney & Hoffman, *supra* note 4 at 383. The comment read:

```
// we are setting this here before the CALL() value transfer to
// assure that in the case of a malicious recipient contract trying
// to call executeProposal() recursively money can't be transferred
// multiple times out of the DAO
```

<sup>99</sup> See Restatement (Second) of Contracts § 155 (1981).

<sup>100</sup> *W.W.W. Assocs., Inc. v. Giancontieri*, 77 N.Y.2d 157, 162 (1990). See also Restatement (Second) of Contracts § 202(2) (1981) ("A writing is interpreted as a whole and all writings that are part of the same transaction are interpreted together.")

<sup>101</sup> 2 Samuel Williston, *A Treatise on the Law of Contracts* § 619, 1199 (1920).

<sup>102</sup> See Restatement (Second) of Contracts § 152, 153 & 158 (1981).

<sup>103</sup> See Restatement (Second) of Contracts § 205 (1981); U.C.C. § 1-304.

prevent one party from taking advantage of unintended loopholes in a contractual writing.<sup>104</sup>

Had The DAO been organized as a multilateral legal contract between identified investors, a court could have used one or more of these tools to sort out the mess created by the flaw in its code—even though the code was both smincorporated into the legal contract and smintegrated against extrinsic communications. A court would likely treat the flaw in the code as just that: not an element of the parties’ agreement, but an error calling for reformation and whose exploitation would give rise to an action for breach. In other words, the users who were victims of the hack would have the ability to recover money damages from those who hacked The DAO. Jonathan Rohr notes the irony: “the consequences of the DAO organizers’ attempt to opt-in to an extra-legal framework for the relationship between and among holders of DAO tokens and other participants in the enterprise are arguably best addressed by the legal framework that they attempted to disclaim.”<sup>105</sup>

## 5.2 Prepackaged Coordination vs. Shared Cooperative Activities: The Relational Aspects of Many Contractual Transactions

The DAO hack illustrates how an exclusive focus on contract law’s role in solving one form of mistrust—worries about performance—can lead to ignoring its role in addressing other forms of mistrust—worries about deceptive and opportunistic behavior. But the hack was not the end of The DAO story. This section discusses its aftermath and the fixed commitment of some project participants to the idea a “trustless” social architecture. That commitment suggests false pictures of contractual relationships and of the form of sociability they embody. Most contractual relationships involve ongoing adjustments and mutual responsiveness that presuppose some degree of trust. Although contract law is designed to solve multiple mistrust problems, it is designed also to accommodate trust and mutual responsiveness. The architecture of a smart contract cannot.

Victims of The DAO hack had no legal recourse not because of the SMINALEA cluses in the DAO’s readme file, but because the software enabled users to remain anonymous. To date, no one knows who siphoned off the \$55 million dollars from the fund, making it impossible to find a defendant.

With legal recourse off the table, The DAO’s creators addressed the hack by altering the blockchain protocol going forward in a way that returned the siphoned monies to the fund. In July 2016, a supermajority of DAO token holders adopted the software updates needed to activate that

---

<sup>104</sup> This was the law’s answer in *Market Street*, *supra* note 97.

<sup>105</sup> Rohr, *supra* note 21 at 87.

new protocol.<sup>106</sup> The effect was to transfer all funds raised, including those that had been siphoned off, to a new chain, permitting token holders to exchange their tokens for cryptocurrency. A vocal minority of participants declined to accept the fix, arguing that it violated The DAO's original vision: governance by code. The result was a hard fork in the code—a permanent divergence from the previous blockchain. Holders of a majority of tokens shifted to the new, forked Ethereum blockchain; a minority continue to use what is now known as “Ethereum Classic.”

At least a few of The DAO token holders who rejected the hard fork did so for principled reasons. In July 2016, a group calling itself “The Ethereum Classic Community” released a four-page unsigned document titled, “The Ethereum Classic Declaration of Independence.”<sup>107</sup> The Declaration raises numerous objections to the hard fork, many founded on a specific vision of governance by code, according to which “code is law.” The authors emphasize the value of the immutability and fungibility of distributed-ledger smart contracts, which immunize users against “the subjective morality judgments” of those who might consider those individual's actions wrongful—including the judgments of a majority of other users, or “the ‘tyranny of the majority.’” The argument assumes moral relativism and a strong commitment to individual choice. “In a global community, where each individual has their own laws, customs, and beliefs, who is to say what is right and wrong?” The blockchain protects individual choice and judgment by enabling a form of governance that is “censorship-resistant, trustless and immutable.” A person's choice to opt into the code binds them to its rules, and because those rules are fixed and apply to the community as a whole, fully protects them against interference by others.

The Declaration's picture of governance by code—or *lex cryptographia*<sup>108</sup>—recalls a familiar reading of Thomas Hobbes.<sup>109</sup> On this reading, the social contract is an arrangement between fearful rational egoists, each of whom agrees to the sovereign's command only in order to

---

<sup>106</sup> “The DAO Heist Undone: 97% of ETH Holders Vote for the Hard Fork,” *Futurism* (July 19, 2016), available at: <https://futurism.com/the-dao-heist-undone-97-of-eth-holders-vote-for-the-hard-fork>. This option was available because of a 27-day waiting period in which funds could not be transferred from the address where they were originally located.

<sup>107</sup> The Ethereum Classic Declaration of Independence, Ethereum Classic (Aug. 11, 2016, grammatical and design update July 2019), available at: [https://ethereumclassic.org/ETC\\_Declaration\\_of\\_Independence.pdf](https://ethereumclassic.org/ETC_Declaration_of_Independence.pdf) (last visited May 13, 2020).

<sup>108</sup> See Aaron Wright and Primavera De Filippi, *Decentralized Blockchain Technology and the Rise of Lex Cryptographia* (March 10, 2015), available at: <https://ssrn.com/abstract=2580664>.

<sup>109</sup> Familiar but not necessarily correct. See Richard Tuck, *Introduction, in Hobbes's Leviathan: Revised Student Edition* \_\_\_\_, \_\_-\_\_ (1991).

protect himself from others.<sup>110</sup> So long as the sovereign is itself composed of people, however, there remains a risk of infringement or exploitation. Subjecting oneself to a human sovereign means not yet fully escaping the state of nature.<sup>111</sup> If code can be made impersonal, discretionless and immutable, a governance system where “code is law” can be entirely trustless. In such a system, individual users are free to pursue their own aims without fear of interference by or the judgment of anyone, be it other citizens or the sovereign.

The Declaration’s picture of a “truly free and trustless computer” is a radical one. That said, it is consistent with a familiar picture of how law in general, and contract law in particular, work. The picture of law is the that of the sovereign’s command backed by force.<sup>112</sup> It is law as compulsion. The relevant picture of contract is what Ian Macneil labels “presentation”:

[Presentation] is a recognition that the course of the future is so unalterably bound by present conditions that the future has been brought effectively into the present so that it may be dealt with just as if it were in fact the present. Thus, the presentation of a transaction involves restricting its expected future effects to those defined in the present, *i.e.*, at the inception of the transaction.<sup>113</sup>

---

<sup>110</sup> Thomas Hobbes, *Leviathan* (Richard Tuck ed., Cambridge Univ. Press 1996) (1651). I have lifted the phrase “fearful rational egoists” from Kwame Anthony Appiah. *As If: Idealization and Ideals* 154 (2017).

Primavera De Filippi suggests that this picture of human nature is built into the structure of blockchain:

[T]he introduction of market-based incentives into the system is such as to distort the underlying motivations of people, who are ultimately brought to compete with each other in order to be the first to find the solution to the mathematical puzzle associated with each block and acquire the corresponding reward.

Primavera De Filippi, *Blockchain Technology and Decentralized Governance: The Pitfalls of a Trustless Dream*, in *Decentralized Thriving: Governance and Community on the Web 3.0* (2019), available at: <https://hal.archives-ouvertes.fr/hal-02445179>.

<sup>111</sup> As Werbach and Cornell put the point with respect to contracts, “An inherent constraint on traditional contracting is that the parties must trust the state, and a variety of private intermediaries that facilitate efficient operation of the system.” Werbach & Cornell, *supra* note 2 at 330.

<sup>112</sup> See H.L.A. Hart, *The Concept of Law* \_\_\_ - \_\_\_ (2d ed. 1994) (discussing the command theory of law).

<sup>113</sup> Ian R. Macneil, *Contract: Adjustment of Long-Term Economic Relations Under Classical, Neoclassical, and Relational Contract Law*, 72 *Nw. U. L. Rev.* 854, 856 (1978) [hereinafter “Macneil, *Adjustment*”]. See also Ian R. Macneil, *Restatement (Second) of Contracts and Presentation*, 60 *Va. L. Rev.* 589, 589 (1974) [hereinafter “Macneil, *Restatement (Second)*”] (“Presentation is thus a recognition that the course of the future is bound by present events, and that by those events the future has for many purposes been brought effectively into the present.”); Ian R.

Macneil argues that “traditional contract theory created a legal structure which in theory attempted to presentiate not just part of the relation between contracting parties, but virtually all of it.”<sup>114</sup> Or to put the point in contemporary terms, traditional contract theory pictured the ideal legal contract as a present act of choice that secures the parties’ desired outcomes in all possible future states of the world. “The aim was to establish, insofar as the law could, the *entire* relation at the time of the expression of mutual assent.”<sup>115</sup> The Declaration envisions a similar act of presentation in computer code. By binding themselves to an immutable, impersonal code, users can act today to determine their collective futures going forward.

There are several reasons to doubt both the viability and the wisdom of such projects. One, illustrated by The DAO, is the persistence of coding errors.<sup>116</sup> Equally important is our lack of foresight. Code that has been designed for possible futures  $F_1$  through  $F_n$  might not serve the parties’ needs in future  $F_{n+1}$ .<sup>117</sup> There is, however a deeper problem. The idea of complete presentiation relies on a false picture of contractual relationships.

The error can be put in terms of Michael Bratman’s analysis of acting together. Bratman distinguishes “prepackaged coordination” from “shared cooperative activities.” You and I engage in prepackaged coordination when we intend to engage together in some joint activity  $J$ —singing a duet, painting a house, taking a walk—and “we work out, in advance, just what roles we each will play in our  $J$ -ing.”<sup>118</sup> In a prepackaged duet, for example, each singer sings her part on tempo and in key, but does not modulate her own performance in response to the other’s. “[W]e each go off and play our role with no further interaction with the other: there is no responsiveness in action.”<sup>119</sup>

---

Macneil, *The Many Futures of Contracts*, 47 S. Cal. L. Rev. 691, 800-804 (1974) (discussing the idea of presentiation in traditional contract theory).

<sup>114</sup> Macneil, *Restatement (Second)*, *supra*, at 592.

<sup>115</sup> *Id.* at 593.

<sup>116</sup> See Cohney & Hoffman, *supra* note 4, at 337, 358.

<sup>117</sup> This is another way of putting the point that smart contracts entail higher drafting costs, as it must anticipate and answer all future contingencies. See, e.g., Sklaroff, *supra* note 28 at 291.

<sup>118</sup> Michael E. Bratman, *Shared Cooperative Activity*, in *Faces of Intention: Selected Essays on Intention and Agency* 93, 106 (1999), *originally in* 101 *Phil. Rev.* 327 (1992).

<sup>119</sup> *Id.* Danah Boyd suggests that programmers often attempt to parse social relations in terms something like prepackaged coordination.

As geeks, we were trained to separate policy and mechanism through systems courses. We rely on people to figure out the policies, not realizing that we've framed what is possible through our technology.

From an autistic perspective, social life can and must be programmatically and algorithmically processed and understood on

Although prepackaged coordination is possible,<sup>120</sup> it is not characteristic of agreements to engage in projects together, or what Bratman calls “shared cooperative activities.” Agreements commonly anticipate a degree of flexibility and mutual responsiveness in their performance, as well as various forms of trust along the way. At the outset, participants understand their shared plan to be incomplete and even defeasible, that the details are to be filled in together depending on how things go. They do not commit themselves to fixed courses of action, but to working together to advance the interests of each and of both. Joint projects of this type does not require that each participant fully commit herself to the other’s ends. “A joint activity can be cooperative down to a certain level and yet competitive beyond that.”<sup>121</sup> But they do involve an expectation—or trust—that each will exercise her discretion in accord with the other and with the plan’s broader purposes.

This is, of course, Macneil’s point about contractual transactions.<sup>122</sup> Although complete presentation might be possible in a truly discrete exchange transaction<sup>123</sup>—“the cash purchase of gasoline at a station on the New Jersey Turnpike by someone rarely travelling the road”<sup>124</sup>— the vast majority of contractual exchanges both require and anticipate flexibility. Macneil identifies a host of reasons why this is so. Because the future is not entirely known, complete presentation is not possible in transactions that

---

simplistic categorical levels. The nuanced relationships that people regularly manage in everyday life are boiled down to segmented possibilities.

Autistic Social Software, (2004), available at: <http://www.danah.org/papers/Supernova2004.html> (reprinted in Best Software Writing I 35 (Joel Spolsky ed. 2005)).

<sup>120</sup> The example of the two singers might have seemed fanciful in 1992, when Bratman published his essay. During a pandemic that requires social distancing and with a much more advanced internet, it is a familiar musical genre.

<sup>121</sup> *Id.* at 107. Bratman provides a somewhat technical description of the degree of cooperation as follows:

So even if the participants have differing preferences about how they are to *J*, neither participant will be in a position to pursue such preferences in ways that by pass the other’s intentions/subplans. This makes it likely that in such cases the demand on each agent that her plans be means-end coherent will lead to rational pressure in the direction of bargaining that is framed by the shared intention.

Michael E. Bratman, *Shared Intention*, in *Faces of Intention: Selected Essays on Intention and Agency* 109, 125 (1999), *originally in* 104 *Ethics* 97 (1993).

<sup>122</sup> For more on the relationship between Bratman’s theory of shared intentions and Macneil’s relational contract theory, see Gregory Klass, *A Conditional Intent to Perform*, 15 *Legal Theory* 107, 141-42 (2009).

<sup>123</sup> Macneil, *Adjustment*, *supra* note 113, at 856 (“A truly discrete exchange transaction would be entirely separate not only from all other present relations but from all past and future relations as well.”)

<sup>124</sup> *Id.* at 857.

extend over time. Because it is difficult to anticipate how people will choose, complete presentation is not possible where the transaction's success depends on acts within one or both parties' control. Because judgment is situation specific, complete presentation is not possible where performance is not easily measured. Nor is complete presentation possible where the parties themselves anticipate future cooperative behavior; where benefits and burdens are to be shared rather than allocated; where one or both parties might choose to exit the relationship; where the "entangling strings or friendship, reputation, interdependence, morality, and altruistic desires are integral"; where "trouble is expected as a matter of course"; or finally and more generally where "the participants never intend or expect to see the whole future of the relation as presentiated at any single time, but view the relation as an ongoing integration of behavior which will grow and vary with events in a largely unforeseeable future."<sup>125</sup>

Contracts scholars are not of one mind as to how the law should take account of the relational aspects of contractual transactions. Macneil argues that contract law and the adjudicators who apply it should become more attuned to relational obligations. The legal reference point should not be the original agreement, but "the entire relation as it has developed to the time of" the dispute, and legal relief should be "in great (or even entire) measure forward-looking, fashioned ad hoc on flexible and broadly remedial lines."<sup>126</sup> Robert Scott, in distinction, argues that parties to relational contracts often prefer contractual obligations that are off-the-rack, binary and fixed, even if those obligations do not track their actual agreement over time. Legal duties fixed at the beginning of the transaction make room for extralegal forms of trust that can achieve flexibility at a lower cost.

The parties, in essence, [learn] to behave under two sets of rules: a strict set of rules for legal enforcement and a more flexible set of rules for social enforcement. It may be that the great lesson for the courts is that any effort to judicialize these social rules will destroy the very informality that makes them so effective in the first instance. . . . Understood in their broader context, binary legal rules serve as an effective compliment to the more flexible means of social adjustment.<sup>127</sup>

Where there is no disagreement, however, is that contractual transactions typically involve a thicker web of expectations and obligations than is captured in the contractual writing, and that the parties themselves benefit

---

<sup>125</sup> Macneil, *Restatement (Second)*, *supra* note 113, at 595.

<sup>126</sup> Macneil, *Adjustment*, *supra* note 113, at 890 & 892.

<sup>127</sup> Robert E. Scott, *A Relational Theory of Default Rules for Commercial Contracts*, 19 J. Legal Stud. 597, 615 (1990). See also Robert E. Scott, *The Case for Formalism in Relational Contracts*, 94 Nw. U. L. Rev. 847 (2000).

from flexibility in the performance of those obligations.<sup>128</sup> Legal contracts are not and should not be straight-jackets. They are more like a good parka: designed to protect against the elements, but also to allow for freedom of movement, to be unzipped when it gets too warm, and to be shed when no longer necessary.

The Declaration's commitment to governance by code utterly fails to account for these aspects of human sociability and the corresponding flexibility of the law of contracts. The claim that smart contracts might eventually supplant even a limited range of legal contracts rests on a false picture of contractual relationships.

### 5.3 What Legal Contracts Do

When Szabo coined the term, "smart contract" was a useful metaphor. It leveraged a familiar fact about legal contracts—that they can be used to solve the basic mistrust problem—to explain something one might do with software. That heuristic enabled Szabo to explain how code could be designed to do something similar to what many legal contracts do. But like all metaphors, the similarity is only partial.

Contract law, and by extension legal contracts, do much more than solve the basic mistrust problem.<sup>129</sup> This is not the place for a complete taxonomy of those functions. But a partial list will be useful in thinking about the proper relationship between smart contracts and legal contracts.

Because breach makes for a good story, Parts Two through Four focused on the law's solutions to various mistrust problems. A mistrustful party might worry that the other side will breach their agreement—a problem a smart contract might solve. But they might also worry that the other side is lying about or failing to disclose material information, or that the other side might adhere to the letter of the agreement but not its spirit.

---

<sup>128</sup> As Scott memorably put the point, "We are all relationalists now. . . . [T]he debate, rather, is over the proper nature of *contract law*. All contracts are relational, complex and subjective." Scott, *The Case for Formalism*, *supra* at 852.

<sup>129</sup> This was one of Lon Fuller's basic insights, too often ignored in recounting of his famous distinction between the restitution, reliance and expectation interests. "[I]t is impossible to assume that when a court enforces a promise it necessarily pursues only one purpose and protects only one 'interest'." L.L. Fuller and William R. Perdue, Jr., *The Reliance Interest in Contract Damages: 1*, 46 *Yale L.J.* 52, 66 (1936).

Others have argued that smart contracts do not fulfill one or another core functions of contract law. See Levy, *supra* note 9 at 4-10 (discussing uses of unenforceable terms, purposively vague terms and strategic nonenforcement); Werbach & Cornell, *supra* note 2 at 318 ("While smart contracts can meet the doctrinal requirements of contract law, they serve a fundamentally different purpose. Contract law is a remedial institution. Its aim is not to ensure performance *ex ante*, but to adjudicate the grievances that may arise *ex post*." (footnotes omitted)).

The law provides multiple tools that address both deception and opportunism, ranging from the tort of deceit to the mandatory duty of good faith. If smart contracts are like legal contracts in that they can be designed to solve basic mistrust problems, it is not obvious that they can address these other forms of mistrust.

But contract law does more than enable exchanges between fearful rational egoists. As H.L.A. Hart observed, the law does not only deter the Holmesian “bad man” from committing wrongs. It also provides answers to “the ‘puzzled man’ or ‘ignorant man’ who is willing to do what is required, if only he can be told what it is.”<sup>130</sup> Many rules of contract law are designed to clarify what parties owe one another when something unexpected happens. The rules for impracticability and frustration, anticipatory repudiation, constructive conditions of exchange all serve this function. Even the default remedies for breach can be viewed in this light. Again, smart contracts provide only compulsion, not guidance.

As the discussion of the relational aspects of contractual transactions suggests, legal contracts are also often structured to give parties flexibility and discretion in the performance of their agreements. Both contractual writings and default rules employ standards—good faith, best efforts, reasonableness, and the like—that leave room for party choice in their satisfaction. The rules for modifications, conditions, and waivers allow the parties to alter their obligations as they go along. And the remedies for breach do not compel performance—parties do not have the power to penalize breach<sup>131</sup>—but merely require compensation for any unavoidable harms that foreseeably resulted from nonperformance. Whether or not one accepts the theory of efficient breach, it is difficult to deny that the expectation measure gives parties the ability to choose breach when performance becomes too expensive.<sup>132</sup> Because a smart contract locks users into prepackaged performances, this is not something it can provide.<sup>133</sup>

Finally, the law of contracts is designed to do more than help parties realize their own ends. It serves social interests whose realization does not run through the satisfaction of party preferences. By marking defections from exchange agreement as legal wrongs, contract law arguably supports

---

<sup>130</sup> Hart, *supra* note 112 at 40.

<sup>131</sup> See, e.g., Restatement (Second) of Contract § 356 (1981); UCC § 2-718.

<sup>132</sup> Even specific performance provides some degree of party choice. The availability of specific performance does not punish the breach or compel the promised performance, but orders the breaching party to provide the next best thing to their willing performance. See John Gardner, *Torts and Other Wrongs* 339-40 (2020).

<sup>133</sup> See Werbach & Cornell, *supra* note 2 at 366 (“One need not accept the theory that the law sanctions efficient breach to appreciate that the law does not lock parties into performance.”).

the social moral practice of making and keeping agreements.<sup>134</sup> By attaching a remedy to breach, contract law can express a collective judgment that such wrongs generate new obligations to make the other side whole.<sup>135</sup> By refusing to enforce some terms and mandating others—the unconscionability rule, nonenforcement on the basis of public policy, minimum wage laws, the mandatory warranty of habitability, and so forth—contract law reflects social assessments of the substantial fairness or public harms of certain agreements. And by providing parties off-the-rack of legal contracts with a mix of mandatory and default rules—employment, partnership, franchise, consumer sales, and others—contract law can help parties find their way into socially valued and personally valuable relationships.<sup>136</sup> These are not functions that a smart contract, programmed or purchased by the parties, can perform.

What does all this say about whether courts would or should give legal effect to attempts to replace legal contracts with smart contracts? It is doubtful that a court would have given legal effect to The DAO's SMINALEA clauses, which were buried in the fine print of an adhesive readme file, in section of the writing that was primarily a disclaimer of legal advice, and which were not a model of clarity. The case would be stronger if there were clear evidence that all participants had actively chosen to opt-out of legal liability, that no one was deceived in doing so, and that doing so was unlikely to harm others. In such a case, one might well imagine a court giving effect to a SMINALEA clause. Legal contractual obligations are, first and foremost, chosen obligations. In addition to all of the above purposes, contract law serves to expand our range of choices by allowing us to develop and engage in new forms of acting together. If people want to experiment with purely technological solutions to the basic mistrust problem, why not let them try?

But the above analysis casts doubt on the wisdom of such experiments. The techno-utopian dream of governance by code rests on an anemic view of human sociability—prepackaged coordination as distinguished from

---

<sup>134</sup> See, e.g., Joseph Raz, *Promises in Morality and Law*, 95 HARV. L. REV. 916, 937 (reviewing P.S. Atiyah, *Promises, Morals, and Law* (1981)) (arguing that Mill's harm principle "does not preclude the law from encouraging moral, cultural or other valid goals"); Seana Valentine Shiffrin, *The Divergence of Contract and Promise*, 120 HARV. L. REV. 708, 752 (2007) (arguing that contract law "is not an effort to legalize as much as possible the interpersonal moral regime of promising, but rather to provide support for the political and public values associated with promising").

<sup>135</sup> See, e.g., Gardner, *supra* note 132, at 139-41 (describing contract law as serving a corrective justice function).

<sup>136</sup> See, e.g., Hanoch Dagan & Michael Heller, *The Choice Theory of Contracts* 76 (2017) ("[W]hen contract law offers us multiple contract types, it participates in the ongoing social production of stable categories of human interaction by consolidating people's expectations of themselves and others. In this, law enlarges the range of valuable options available to us.").

shared cooperative activities—and the law’s role in it. Julie Cohen observes that “hacker culture speaks the intertwined languages of liberal individualism and libertarianism and posits enlightened self-reliance and, by necessary implication, technical meritocracy as cardinal virtues.”<sup>137</sup> Many contracting parties are attuned to a broader range of values and virtues. So too is contract law. Smart contracts are single-use tools. Legal contracts serve both the parties and society in multiple ways at once. Parties are likely to achieve better results by using a smart contract to supplement their legal contract, rather than supplant it.

## Conclusion

Smart contracts are the latest in a long line of new techniques parties have invented to structure future-oriented exchanges. As has happened before, lawmakers will need to adapt the existing law of contract to these new forms of contracting.

This article has identified several limitations of smart contracts. Smart contracts are not contractual writings, and should not be treated as such. And code should figure into the interpretive evidence of a legal contract only in a narrow band of transactions: when the parties have worked together to write the code and when sophisticated parties have expressly incorporated it into the agreement. Even less often should code displace other evidence of the parties’ actual agreement. More broadly, in the vast majority of transactions, a smart contract will not do everything for the parties that a legal contract can do. Parties are likely to achieve more value when they use smart contract to supplement rather than supplant their legal contracts. So too society as a whole.

None of this argues against the expanded use of smart contracts. A smart contract is comparable to a surety bond in which S guarantees that A will perform A’s legal contract with B. Like a smart contract, the bond can help to solve the basic mistrust problem. Parties use surety bonds when legal-contractual assurance of performance are, for one reason or another, insufficient for their purposes. But no one would think that the surety bond S provides does or should displace the legal contract between A and B. The two work in tandem to maximize the value of the transaction to the parties. The same should go for smart contracts.

Of course a smart contract can do much more than a surety bond. By automating large parts of their transaction, perhaps using a distributed ledger, parties might be able to achieve greater security and predictability at a lower cost. But this new tool brings its own risks. A circular saw can accomplish some jobs with greater speed and efficiency than can a hand

---

<sup>137</sup> Cohen, *supra* note 8 at 92 (crediting the idea to Gabriella Coleman, Hacker, Hoaxer, Whistleblower, Spy: The Many Faces of Anonymous (2014); Gabriella Coleman, Coding Freedom: The Ethics and Aesthetics of Hacking 183–205 (2012).)

saw. But no one accidently loses a finger to a hand saw. Contract law provides a host of tools not only to do jobs that smart contracts cannot, but that can protect parties against new risks smart contracts might generate. The key is to understand how contract law can productively interface with smart contracts, which has been the subject of this article.