2022

Governing the Interface Between Natural and Formal Language in Smart Contracts

Joshua A.T. Fairfield
Washington and Lee University School of Law, fairfieldj@wlu.edu

Niloufer Selvadurai
Macquarie University, Australia, niloufer.selvadurai@mq.edu.au

UCLA

JOURNAL OF LAW & TECHNOLOGY

SPECIAL ISSUE: GOVERNING THE DIGITAL SPACE

GOVERNING THE INTERFACE BETWEEN NATURAL AND FORMAL LANGUAGE IN SMART CONTRACTS

Joshua Fairfield* & Niloufer Selvadurai**

ABSTRACT

Much of the confusion about the proper regulation of smart contracts stems from the fact that both code and law are expressed in language. Natural (human) and formal (computer) languages are profoundly different, however. Natural language in the form of a true legal contract expresses human meaning and expectation. Code simply acts, and when code acts contrary to the understanding of the parties to a contract, courts must have a theoretical and legal basis in order to intervene—which this Article provides.

Present scholarship on the governance of smart contracts centers on logistical problems relating to the effects of automation on operation and execution, most notably problems of inflexibility and lack of enforcement discretion. However, automatic execution is nothing new in contract law. Rather, it is the legal interface between contract law and code that must

* William D. Bain Family Professor of Law, Washington and Lee University School of Law, Lexington, Virginia.

** Professor of Technology Law, Macquarie University, Sydney, Australia.

catch and hold our attention. We focus on the point where the 'natural language' of contract law crosses over into the 'formal language' of computer code. Natural language contract terms are made accessible to a human and receive some sort of confirmation to establish the contractual magic, a set of bespoke legal rules between two parties encapsulated in some document or through behavior that makes the intention of the parties unmistakable. The formal language program portion of a smart contract executes, sometimes in accordance with these expectations, sometimes not. This Article asserts that human expectations determine the legal obligations of a contract, and that code merely executes it. It then explores the legal bases and ramifications of this human-centered law of smart contracting.

TABLE OF CONTENTS

INTRODUCTION

The brilliant glare of technological advancement sometimes blinds us to the ways it erodes human autonomy and control. This is potentially the situation with smart contracts. Smart contracting is a method of executing agreements using computer code, where such code is stored on a blockchain platform utilizing distributed ledger technology (DLT).[1] While traditional contracts are centralized—that is, designed to address the specific intentions of identified parties—smart contracts are decentralized and automatically generated using predesigned algorithms to bind pseudonymized parties.

Present scholarship on the governance of smart contracts centers on logistical problems relating to the effects of automation on operation and execution; most notably, problems of inflexibility and lack of enforcement discretion. But automatic execution is nothing new in contract law—we do it each time we purchase petrol at a pump. Quasi-automated clickwrap boilerplate contracting is commonplace. Courts deem each of us to have agreed to a myriad of contracts merely by turning on a device.[2] Established law on automatic execution can be extended to encompass smart contracts. We must therefore look elsewhere to find what is new and important with respect to the relationship between law and smart contracts.

The *interface* between contract law and DLT—the point of conversion from natural language to code, the place where human intention meets

---

1. While Blockchain technologies are used in both public and private modes, this Article will primarily address contractual issues relating to public blockchain platforms. *See generally*, Peter Yeoh, *Regulatory Issues in Blockchain Technology*, 25 J. FIN. REGUL. & COMPLIANCE 196, 196–97 (2017). Public blockchains are permission-less, in which the identities of users or even their wallet addresses are not fully traceable to the relevant real individuals. In comparison, private (permissioned) blockchain ledgers involve users who identities are known and confirmed. As Yeoh notes, private blockchain ledgers form a more circumscribed and controlled application of blockchain. *Id.*; *see also* Helena Vieira, *Blockchains May Replace the Institutions That Safeguard Commercial Activities,* LONDON SCH. OF ECON.: BUS. REV. (Mar. 31, 2016), https://blogs.lse.ac.uk/businessreview/2016/03/31/blockchains-may-replace-the-institutions-that-safeguard-commercial-activities/.

2. *See generally* JOSHUA A.T. FAIRFIELD, OWNED: PROPERTY, PRIVACY, AND THE NEW DIGITAL SERFDOM (2017), demonstrating that the confluence of the RAM-copy doctrine, the Digital Millennium Copyright Act, and contract doctrines binding online users by mere use of a site or service, originating in cases like ProCD, Inc. v. Zeidenberg, 86 F.3d 1447 (7th Cir. 1996), result in the current state of affairs whereby a user is bound by license terms of service merely by turning on and using a device.

automated systems—catches and holds our attention. This we call the smart/contract interface, the point where the natural language of contract law crosses over into the formal language of computer code, and back again. Natural language contract terms are made accessible to a human and receive some sort of confirmation to establish the contractual magic, a set of bespoke legal rules between signatory parties encapsulated in some document or through behavior that makes the intention of the parties unmistakable. The formal language program portion of a smart contract mechanically executes, but does not intend, nor produce utterances capable of being interpreted as intention. It simply acts, often in ways that no human desired, anticipated, or expected.

In such a context, a critical issue for examination is the way in which contract law should govern the interface between the natural and formal language of smart contracts. In a situation where contracts are created through predeveloped algorithms in a decentralized environment, how can contract law enable human intervention and control? When should a court be able to undo a transaction executed by smart contract that clearly contravenes the objectively determinable expectations of the parties? When should a court be able to invoke traditional contract doctrines such as unconscionability and undue influence to protect vulnerable individuals? How should doctrines of frustration of purpose and commercial impracticability, which cover events that cannot be coded, apply in the smart contract system? Imagine coding for every outcome of force majeure and commercial impracticability. The mind boggles!

This Article proceeds in three Parts. In Part I, we review the discussion around smart contracting language, illustrating how new legal terms and new legal conceptions, such as the complex of issues around contracting via DLT, come about. The process is linguistic and evolutionary, solving problems more often by asking new and better questions rather than answering old and bad ones. Part II describes some principles for mediating between natural language contracts expressing human intention and formal language computer programs, as well as some problems with translation between the formats. It also describes how courts can determine when to intervene. Finally, Part III addresses some principles for mediating between natural language intention and formally expressed smart contracts as a matter of evolving contract law across the United States, Australia, and the European Union. It shows that the process of evolving legal language to deal with the interface problems of smart contracting is already underway, but is undertheorized.

This Article thus provides a framework for helping courts communicate good ideas and develop robust legal rules for handling the interface between natural language contracts and formal language programs by: (a) exploring the evolving legal language of smart contracts; and (b) considering how to operationalize the interface between the computer program and parties' intentions.

I. THE LANGUAGE OF SMART CONTRACTS: TRANSLATING LEGAL PRINCIPLES INTO CODE

A. How Language Shapes the Creation of Legal Concepts and Terms

The interface between human—natural—language and computer programs matters, and legal constructions of human encounters with automatic systems have profound legal significance. Consider, for example, the experience of a human encountering an automated system. A human clicks "I agree" to complete a purchase, while possessing certain expectations regarding the resulting legal relationship. The result of this encounter is that the human is bound to the computer's terms. As a thought experiment, consider the problem the other way around: a human enters language expressing their understanding of the legal rights and responsibilities in a contract into a form on an automated contracting site. Courts *generally* do not hold the computer bound by any of these human representations, on the (quite incorrect) view that the computer cannot understand, or that it cannot be tasked with legal liability because the automated system cannot understand natural language (again, quite incorrect).[3] This Article asks why humans have ended up as second-class contracting citizens, and whether there are conceptions of law more consistent with legal history and reasoning that can reach a different result. The questions are of particular importance in the face of DLT, because of the rising tide of claims that the outcome of a smart-contracting process supersedes human expectations for a given contractual deal. Simply put, we argue that contract is still the law of satisfying human expectation, not validating machine outcomes. Where a deal fails because of a bug, an exploit, the occurrence of a condition the non-occurrence of which was a basic assumption of the parties, or other emergent problems between software programs, courts must not lose the plot: human expectations are

3. James Grimmelmann, *Spyware vs. Spyware: Software Conflicts and User Autonomy*, 16 OHIO ST. TECH. L.J. 25, 47–49 (2020) (describing the lopsided relationship in contracting that favors technology over users).

what matter in interpreting the contract, despite all of the jargon around replacing law with code.

The question of where the handoff occurs between natural language expressing human expectations of legal relationships and the result of its encounter with formal computer languages is of decisive import. Courts see themselves as tasked with determining the arrangement of the parties as expressed in human language, on the objective theory of contract. Yet they increasingly rely on the fait accompli of computerized transactions. Courts dangerously confuse what a computer system did with what the humans must have wanted.[4] For anyone who has ever used a computer and knows how fast and how far code can vary from human expectation, this is an absurd direction for contract to take. We set about creating a framework for diagnosing and correcting this problem.

A few examples help ground the following discussion. Consider an NFT that is sold at one one-hundredth of its well-established worth—three thousand dollars, instead of three hundred thousand dollars—due to an obviously erroneous keypress.[5] A human buyer would be held to understand this was scrivener's error. As in fact happened, an automated purchasing bot sniped the purchase in the split second after it was posted, far faster than the human could correct the typo, and the NFT was transferred indelibly to a new owner.[6]

Consider a second example. In the early days of Ethereum, a distributed autonomous organization (the DAO) sought to receive investor cryptocurrency and deploy it to a range of projects, the profits of which

4. Ed Felten, *Virus With a EULA*, FREEDOM TO TINKER (Nov. 15, 2002), https://freedom-to-tinker.com/2002/11/15/virus-eula/(discussing a virus with an ostensibly enforceable EULA).

5. Ryan Browne, *Bored Ape NFT Reportedly Sells For $3,000 Instead of $300,000 Due to 'Fat-Finger' Mistake*, CNBC (Dec. 14, 2021, 8:10 AM), https://www.cnbc.com/2021/12/14/bored-ape-nft-accidentally-sells-for-3000-instead-of-300000.html (reporting on an NFT being underpriced due to a clerical mistake).

6. Edward Ongweso Jr, *All My Apes Gone: NFT Theft Victims Beg for Centralized Saviors*, VICE (Jan. 6, 2022, 9:38 AM), *https*://www.vice.com/en/article/y3v3ny/all-my-apes-gone-nft-theft-victims-beg-for-centralized-saviors.

were to be remitted to the investors.[7] Through a bug in the code, the invested cryptocurrency was stolen by a third-party actor.[8] The theft was orchestrated through bugs in the DAO's smart contracts.[9] Imagine litigation were to follow: Should courts validate this theft under the understanding that whatever a computer permits to happen is fair game, even if it results from exploitation of a software bug? Or will courts look to the intentions of the parties in creating legal arrangements, and specifically look to the objectively reasonable human understanding of such legal arrangements, rather than the vagaries of how code acts when it is actively under attack by hackers exploiting software bugs?

Finally, a third example. Imagine that person A purchases cryptocurrency or NFTs from a website, which is complete with terms of service and automated contract terms à la Amazon. Imagine that person B purchases the same cryptocurrency or NFTs directly from a smart contract located on the Ethereum blockchain. Person A will be bound by all kinds of contractual limitations and restrictions from the website (most notably, in the US context, they will lose the ability to present their claim in court pursuant to a forced arbitration clause).[10] Person B will not, because the smart contract was a computer program with which the buyer's smart wallet interacted.[11] The next step in these cases (this is of course already happening) is that sellers will attempt to embed legal contract terms of the sort found on every e-commerce website into computer code itself—to nest legal language into the executable, or at least to make a kick-out directing a human to view some sort of pop-up window—to exploit the fact that when humans and computers contract, computer terms win. What, then will be

---

7. Nathaniel Popper, *A Hacking of More Than $50 Million Dashes Hopes in the World of Virtual Currency,* N.Y. TIMES (June 17, 2016), https://www.nytimes.com/2016/06/18/business/dealbook/hacker-may-have-removed-more-than-50-million-from-experimental-cybercurrency-project.html?.

8. *Id.*

9. *Id.*

10. *See* AT&T Mobility LLC v. Concepcion, 563 U.S. 333 (2011) (addressing an arbitration clause banning class actions); *see also* Rensel v. Centra Tech, Inc., No. 17-24500-CIV, 2018 WL 4410110 at *14 (S.D. Fla. June 14, 2018) (allowing a user to avoid an arbitration agreement by interacting with a software without using the website).

11. *Rensel*, 2018 WL 4410110, at *14 (holding that purchaser is not bound by terms requiring arbitration agreement because tokens purchased via smart contract and not seller's website).

the next level response? Perhaps, as above, the way to avoid the implementation of contract terms automatically proffered and hidden deep in code is to use a dumb purchasing bot that itself asserts terms and conditions and is coded to be incapable of understanding legal language, only engaging with the formal language of the program, executing the transaction. This is the world toward which we are skidding, in which robots must contract with each other, because humans are not permitted to express their desired preferences in legal language and have them respected by the courts.

At the root of these everyday problems—examples multiply by the day—lies the problem of the legal construction of the interface between natural and formal language, between human ways of understanding what words mean and machine execution of formal logic.

1. Natural Language, Formal Language, and Machine-Human Language Hybrids

Courts are confused by the fact that both natural and formal language systems use words—or rather, appear to a human to do so. When judges look at machine code, they may see things that look like words, although they might as well be bricks laid out in a specific pattern guiding specific action. To a machine, words are either logical directions or trash. Logic is expressed mechanically, not linguistically, to a computer. Computer instructions are as easily expressed in physical NAND gates as in linguistically expressed first-order logic. To unpack why the linguistic interface between human legal understandings and computerized automatic execution of contracts presents such a legal conundrum, we must first examine what natural language is, what it does, and how it works, as well as what formal language is, what it does, and how it works. Only then can we see the profound gap in understanding that occurs at the linguistic interface in contracts.

a. Natural Language: What Law Is and How Words Gain Meaning

Natural language is a process whereby humans use context to give words meaning so that they can cooperate.[12] Human language does not

12. Cooperation at the hundred-million-plus individual scale is the human superpower, and language is its method. *See* JOSHUA A.T. FAIRFIELD, RUNAWAY TECHNOLOGY: CAN LAW KEEP UP (2021).

have ostensive meaning—there is no vast dictionary in the sky of what words mean. Rather, words come to have meaning within a given context through how humans use them.[13] Is "bad," bad? Or is it kind of good? Which of the 645 meanings of the English word "run" does one mean when one discusses a run of salmon, a run in pantyhose, or a run on the bank? Humans ground meaning in context and in a community trying to perform some task.14 This explains some features of language, how it remains sticky enough to have meaning ("up" does not mean "down," and so on); yet, words change meaning over time as entire languages evolve. Anyone who has read poetry from 600 years ago can attest to the shifts.

Law is a specific kind of task, drawn from a particular kind of human linguistic community and context. Through a specific activity—natural language—humans evolve symbols, cooperative fictions like money, corporations, nation-states, fairness, and even the rule of law itself to permit cooperation at the multi-billion-person scale.[15] Contracts are a concentrated version of this task, and are historically grounded in the context of consent: By expressing in objective language the goals, intentions, and methods of cooperation, and using the broader cooperative fiction of the state to enforce the narrower cooperative fiction of the contract, parties could replace physical coercion with a socially-backed negotiated agreement—the fabled shift from status to contract.[16]

Contracts are, relatively speaking, a recent development in the law.[17] And contracts themselves are not immune to this process of shifting meaning, either internally to the contract, or externally. Externally, contracts have shifted meaning—where once they were the law of

---

13. *See* LUDWIG WITTGENSTEIN, PHILOSOPHICAL INVESTIGATIONS 1–5 (G.E.M. Anscombe trans., 3d ed. 1974) (arguing that word meanings are based on how they are used within a community to complete a task).

14. *See id.* at 1–5.

15. *See* YUVAL NOAH HARARI, SAPIENS: A BRIEF HISTORY OF HUMANKIND (2015).

16. *See, e.g.,* R.H. Graveson, *The Movement from Status to Contract*, 4 MOD. L. REV. 261, 261–72 (1941) (discussing "Sir Henry Maine's famous generalization of the movement up to his time of progressive societies from status to contract.").

17. The first known use of "contract" as a noun occurred in the 14th century. *Contract,* MERRIAM-WEBSTER, https://www.merriam-webster.com/dictionary/contract (last visited Feb. 18, 2021).

negotiated agreement, now the practical import of most contracts is that they are not negotiated at all, but are instead embedded in mandatory technology structures that remove choice from the technologically disadvantaged party, the consumer who clicks "I Agree," the employee who signs an arbitration agreement preventing them from suing for sexual assault, and so on.[18]

Internally, contractual meaning shifts as well, although this analysis is trickier. Lawyers want to control the meaning of words inside a contract by defining them. They capitalize them or use quotation marks to show certain words in a contract are more like the logically determined signifiers of computer code, below, and less like living artifacts of meaning. But even the most tightly sewn-up contract cannot be protected against the fact that a contract is comprised of natural language words. The parties' course of dealing or course of performance give meaning to words.[19] The court construes contractual terms in light of some other community of meaning—Black's Law Dictionary, Webster's Dictionary, or industry custom and practice.[20]

The problem begins when courts and programmers misunderstand the nature of contracts and contractual language. The legacy of legal positivism—the idea that law is a closed and formal system, susceptible of logical, context-free manipulation—has left scholars and practitioners with a fundamental misunderstanding of the relationship between law and the natural language environments that give its words meaning.[21] Take core legal concepts like freedom, privacy, security, even the concept of a constitution—each is a heatmap of overlapping meanings, used by humans in different contexts to express a cluster of purposes and goals. It is not without irony that the legal profession relies on its own community and context to create legal meanings, legal fictions, the legal language that ties

18. *See* Latif v. Morgan Stanley & Co., No. 18cv11528 (DLC), 2019 U.S. Dist. LEXIS 107020 (S.D.N.Y. June 26, 2019) (upholding arbitration in face of New York law that precluded arbitration in employer-employee sexual harassment cases).

19. *See* Figgie Int'l, Inc. v. Destileria Serralles, Inc.,190 F.3d 252 (4th Cir. 1999) (stating that industry customs or typical use may supplement express contract terms under the Uniform Commercial Code).

20. *Id.*

21. *See* H.L.A. Hart, *Positivism and the Separation of Law and Morals*, 71 HARV. L. REV. 593, 601–02 n.25 (1958) (describing a legal system as a "closed logical system" in which legal decisions are deduced from predetermined legal rules).

the entire profession together, while still too often claiming that words in legal documents are susceptible to precise definition and formal use. This professional proclivity to mistake the narrow and specific activity of legal term definition for the social processes that generated linguistic fictions like constitutions, nation-states, corporations, money, time, days of the week, and other consensual hallucinations of legal social fiction sets the stage for mistaking code for legal language, the fundamental error nestled within the present smart contracts debate. Our guild sensibilities explain why lawyers are likely to accept the project of reducing laws to computer code as plausible—and even advise clients that it is possible to do so—when it is demonstrably not.

b. Formal Language: Words are Mechanical

Formal logics, such as first-order logic, number theory, and eventually computer languages, do not use words. Humans input signs that look like words in order to express directions for a machine. These directions are best thought of as physical; indeed, computers and their code can be fully expressed as a series of physical objects. There is no requirement for meaning, only logical consistency. The fundamental rule of computation is that no system of logic may lead to a theorem that states N = !N, or, in English, N equals not-N.[22]

A result of the rigid, mechanical method of symbol manipulation is that formal languages are consistent but at the cost of completeness, as Kurt Gödel proved.[23] His result was instrumental to one of the fundamental rules of computation, the Church-Turing Thesis, which states the limits of effective computability.[24] For practical purposes, these results prevent the creation of bug-free software, and impose strong limits on the kind of problems computers can solve.

---

22. *See* Richard Zach, *Hilbert's Program Then and Now, in* 5 HANDBOOK ON THE PHIL. OF SCI. 411, 431 (Dale Jacquette ed., 2007) ("Gödel announced the second incompleteness theorem in an abstract published in October 1930: no consistency proof of systems such as Principia, Zermelo Fraenkel set theory, or the systems investigated by Ackermann and von Neumann is possible by methods which can be formulated in these systems.").

23. *Id.*

24 *See generally* DOUGLAS R. HOFSTADTER, GÖDEL, ESCHER, BACH: AN ETERNAL GOLDEN BRAID (20th ed., 2000).

First, a formal language cannot be used to guarantee bug-free code.25 There is only one way to know whether an algorithm will halt—will reach an answer—or whether it will continue grinding on for infinity, consuming infinite resources: run the code.[26] We cannot run code B to see whether code A will hang or yield an answer because the only thing code B could do is run code A.[27]

Second, the Church-Turing thesis limits the complexity that computation can effectively resolve. Certain problems cannot be resolved within a reasonable period of time, particularly those for which each additional input exponentially increases the number of potential solutions to be checked. The traveling salesman problem provides an example: imagine a salesman is traveling to 100 cities—what is the shortest route between them he can take? This requires checking a gargantuan number of routes. And if one adds an additional city to the list, the number of routes to be checked increases exponentially.[28]

In short, computer programs cannot be guaranteed to be bug-free, and they do not handle complex problems of optimization well. Here we begin to see some of the give-and-take at the natural-machine language interface. Natural language can express any statement; so it is complete at the cost of consistency.[29] And natural language can evolve to express truths within frameworks that did not exist previously:[30] The scholastics never answered how many angels could dance on the head of a pin; scientists learned to ask new questions in new frames of inquiry.

Further, natural language handles social problems of enormous complexity, the least of which would stump a supercomputer. An algorithm can "learn" the word-contexts in which humans use a given word (this is how google translate functions) as a machine-human language hybrid, but no algorithm can handle the complexities of social nuance that give meaning to #metoo or Black Lives Matter, for example.

---

25. *Id.*

26. *Id.*

27. *Id.*

28. *See generally* Merrill M. Flood, *The Traveling-Salesman Problem*, 4 OPERATIONS RSCH. 61 (1956), https://doi.org/10.1287/opre.4.1.61.

29. *See* Zach, *supra* note 22.

30. *See* WITTGENSTEIN, *supra* note 13.

What this means for contracting is simple: humans express meaning in natural language; computers parse actions in systems of symbolic direction. The overlap between these systems is more accidental than essential. When we turn to the question of legal doctrine and reading smart contracts, we see the problem: there is a temptation to state that the code expresses human preference, that the code is the contract. It is not, nor can it be. It is not written in a language susceptible to expressing meaning within a context. Computers do not express meaning; they do not read meaning. They can only create meaning if they are seen, as Bruno Latour sees them, as part of a community of meaning with humans—objects alone do not generate language—and even then, their impact on language is parasitic and derivative.[31]

Humans of course can express meaning when they program, in the sense that they may have a goal. But again, the performance of the code is not equal to the intent of the coder. Code has bugs, and even perfectly designed code may have unexpected interactions with other code. In that gap—and it is a large gap—between intention and expression lies most of the human law of contract.

Consider an increasingly common event: a blockchain that, through a revision in the code, creates a bug whereby users can exploit the database to mint millions of new tokens. The entire purpose of a cryptocurrency blockchain or NFT smart contract is to create rarity and digital scarcity. If millions of bitcoin could be created with the press of a button, bitcoin would be worth nothing. The focus of blockchain software is to prevent fraudulent double spending and to impede the creation of tokens under circumstances—like these—that ruin the scarcity or uniqueness of a token. (There are sites that attempt to do precisely this, by duplicating any extant NFT and issuing a token on an inexpensive chain, as an attempt to show some of the difficulties in the scarcity model underlying NFTs.[32]

---

31. *See* BRUNO LATOUR, REASSEMBLING THE SOCIAL: AN INTRODUCTION TO ACTOR-NETWORK THEORY 71 (2005) (describing objects as actors because they "make a difference in the course of some other agent's action").

32. *See* Lisa Gibbons, *There is a Way to Protect NFTs From Being Replicated or Lost: This Company Does Just That*, COINTELEGRAPH (Oct. 28, 2021), https://cointelegraph.com/news/there-is-a-way-to-protect-nfts-from-being-replicated-or-lost-this-company-does-just-that (describing the solution of NFT replication and solutions to the problem).

Under such circumstances, natural language conflicts irreconcilably with computation. More bugs and exploits that compromise the scarcity of cryptocurrency and NFTs are found every day, especially in proof-of-stake systems where the same software is run on each node responsible for minting new tokens, such that a single bug can easily ramify back into the entire system. Exploiting such a bug destroys the rarity of the currency.[33] These bugs are the equivalent of issuing every person in the country a photocopier that could perfectly duplicate cash. Exploitation of emergent code properties (bugs, etc.) undermines and irrevocably damages the human expectations of a community reliant on digital scarcity. Contract law protects that expectation.

2. Potential Paradigms for the Human-Machine Language Interface

What paradigm, then, should we follow when bugs in code contravene clearly delineated human expectations? Should courts continue to privilege dumb statements of language by unmeaning machines—an "I Agree" contract conveyed by a computer to a human—over evidence regarding how the human or humans meant to arrange legal arrangements between them? Can there be any room or mechanism for humans to express meaning to machines, other than machine refusal to contract on any text it itself does not offer? What happens when legal language is buried deep in code?

The job of courts is particularly complex in light of tech evangelists' push to have the effect of the technology read as the intent of humans using it.[34] Consider the above-described case of the NFT seller who listed an NFT for one one-hundredth of the fair market value—a slip of the human finger immediately pounced on by bid-sniping software. The seller adopted an almost masochistic pose—NFTs are just like that, he seemed to say; by choosing to contract in NFTs, he felt that he had acceded to a system where the operation of code supervened the human expectations of the parties.[35]

This is the first paradigm of the natural-formal language interface: formal languages win. If a smart contract operates in such a way as to

---

33. *See* Dan Goodin, *Really Stupid "Smart Contract" Bug Let Hackers Steal $31 Million in Digital Coin,* ARS TECHNICA (Dec. 1, 2021, 3:41 PM), https://arstechnica.com/information-technology/2021/12/hackers-drain-31-million-from-cryptocurrency-service-monox-finance/.

34. *See* Samer Hassan & Primavera De Filippi, *The Expansion of Algorithmic Governance: From Code is Law to Law is Code*, 17 FIELD ACTIONS SCI. REPS., SPECIAL ISSUE 88, 89 (2017).

35. *See* Browne, *supra* note 5.

contravene clearly expressed human expectations, what of it? Moreover, imagine if enterprising lawyers chose to transact business this way: "By using this service, you agree that all transactions are final and valid as executed by the Code." The parties could wrap code in contracts, to the following effect: "The Parties agree to transact as determined by the following code." The difficulty is that exploits, bugs, fraud, and outright bad faith exploitation will follow from such systems. This is precisely what happened in the case of the early DAO: other distributed autonomous organizations siphoned off the value placed by investors in the original program.[36]

The second paradigm fits more closely with historical law and practice, and falls far closer to the recommendations and experience with the law and practice of smart contracts reflected in our practical analysis below. Here, human expectation governs without giving much power to the idea that whatever the code does is what the humans intended. Because we can never have guaranteed bug-free code, the interactions within and between pieces of software guarantee emergent behavior that no human expected.[37] Human expectation must still be objectively interpreted, of course, but the grounds of objectivity lie in the context, community, and task that the contracting parties sought to accomplish.[38]

This paradigm is far more consistent with law because it tracks the discipline of contract as a legal mechanism for securing cooperation through understanding, negotiation, and consent. The drawback of such an approach is that it varies far from modern contracting practice, which has thrown any concept of human consent by the wayside in favor of sticking the human half of a human-computer transaction with all legal detriment and no legal benefit by virtue of having communicated intention to the machine. Humans no longer negotiate anything in computer-assisted contracting.

There is an additional layer of complexity in machine-to-machine contracting, creating a modern (although vastly more difficult) version of the battle-of-the-forms problems that plagued industrial contracting. If the

---

36. *See A Legal Analysis of the DAO Exploit and Possible Investor Rights,* NASDAQ (June 21, 2016, 12:52 PM), https://www.nasdaq.com/articles/a-legal-analysis-of-the-dao-exploit-and-possible-investor-rights-2016-06-21.

37. *See* HOFSTADTER, *supra* note 24.

38. *See* WITTGENSTEIN, *supra* note 13.

present legal trend—not holding machines responsible for natural language, but holding humans responsible for technical operation—continues, more humans will resort to machine contracting to protect themselves. In practice, then, we see emerging meta-contracts at the level of whatever human or corporation can be tasked with responsibility for a machine operant, and then machines contracting at a sublayer. These contractual arrangements are often placed in apps or website terms of service, human attention traps, an attempt to leverage the rule that if a human could have seen the legal terms, they must be operative between even machine counterparties. This merges with the practical reality of crypto communities. Such communities often want some kind of nearly constitutional commitment to the blockchain and the community: promises not to exploit the code, promises to give exploited currency back, indeed, contractual recognition that exploited currencies are not the property of the exploiter, and are properly subject to seizure and deletion by the community (usually through the mechanism of a hard fork or its equivalent), and so on.

A final issue is that contract law will inevitably shift in light of these new transactional forms. Yet the means and path of a legal shift in the face of technology is not the same in every culture. Linguistic and cultural shifts translate into legal shifts in profoundly different ways. We focus on common law jurisdictions (courts at the EU level follow common-law processes) because the method of reasoning by narrative and analogy is particularly adapted to the experimental, iterative approach that permits rapid prototyping of legal rules in the face of technological shift. We recognize that civil jurisdictions often produce superior rules in the face of technological shift, largely because they rely on expertise rather than raw monetary investment in law-changing litigation.[39] Perhaps some hybrid is called for: the EU's success (its legislative branches remain closely tied to civil law systems) in the face of surveillance capitalism's unprecedented drive to exploit citizens' data might serve as a model. In light of these considerations, we consider common law doctrines from a number of

39. For example, compare the abdication of the United States on questions of personal data privacy with the now-dominant worldwide privacy regime tested in the EU data privacy directive and iterated some two decades later in the GDPR.

countries. Our purpose is not merely to chart what is, but to provide a doctrinal and theoretical basis for predicting and guiding the future.

B. The Nature and Operation of Blockchain Smart Contracts

Blockchain technologies integrate innovations from the fields of distributed computing and cybersecurity to create immutable, trusted, and decentralized data-storage systems.[40] As noted, a smart contract is an automated agreement hosted on a blockchain platform that autonomously executes transactions on the occurrence of certain predetermined conditions. An automated agreement is a method of exchanging value where some aspect of the exchange is processed by a computer without human verification or approval.[41] The code is located on the blockchain and recorded on the ledger. All parties obtain a copy of the code. But, once registered, the blockchain cannot be modified. The general objectives of smart contract design are to satisfy common contractual conditions (such as payment terms, liens, confidentiality and even enforcement), minimize exceptions (both malicious and accidental), and minimize the need for trusted intermediaries. In simplest terms, a smart contract is "a piece of code on a blockchain."[42]

The rise of DAOs facilitated DAO smart contracts, computer programs that operate on peer-to-peer networks and incorporate rules for governance and decision-making.[43] It supported the creation of self-executing smart contracts by enabling the collection, verification, validation, and enforcement of terms which have been previously agreed upon by parties.[44] Such contracts are considered trustworthy because they are stored on

---

40. *See generally* Arvind Narayanan & Jeremy Clark, *Bitcoin's Academic Pedigree*, 60 COMMC'NS ACM 36 (2017), https://doi.org/10.1145/3132259. As to the distinction between public and private blockchain ledgers, *see* Yeoh, *supra* note 1, at 196–97.

41. Jake Goldenfein & Andrea Leiter, *Legal Engineering on the Blockchain: 'Smart Contracts' as Legal Conduct*, 29 L. & CRITIQUE 141, 143 (2018); MARIA GRAZIA VIGLIOTTI & HAYDN JONES, THE EXECUTIVE GUIDE TO BLOCKCHAIN 135 (2020).

42. *See* Rajesh Gupta et al., *HaBiTs: Blockchain-based Telesurgery Framework for Healthcare 4.0*, 2019 INT'L CONF. ON COMPUT., INFO. & TELECOMM. SYS. 1 (2019).

43. Madhusudan Singh & Shiho Kim, *Blockchain Technology for Decentralized Autonomous Organizations*, 115 ADVANCES IN COMPUTS. 115, 116–17 (2019).

44. Vimal Dwivedi et al., *Legally Enforceable Smart-Contract Languages: A Systematic Literature Review*, 54 ACM COMPUTING SURVS. 1, 2 (2021).

encrypted, distributed, and immutable ledgers.[45] The decentralised system enables parties to conduct transactions without reliance on central organisational entities or external legal systems. These features cause blockchain-based smart contracts to be verifiable, observable, and enforceable, and uphold privity.

The term smart contract was coined in 1994 by Nick Szabo, who described it as "[a] computerised transaction protocol that executes the terms of a contract"[46] and later as "a set of promises, specified in digital form, including protocols within, which the parties perform on these promises."[47] The development of blockchain technologies in 2009, primarily to support cryptocurrencies, involved new protocols that also facilitated the development of blockchain-based smart contracts and helped realise Szabo's early vision. While blockchain platforms were initially deployed to establish cryptocurrencies, they progressed to support a wide variety of commercial transactions, including syndicated lending and securities transactions.[48]

Consistent with the rapid escalation in the sophistication and scale of smart contracts, there has been an increase in the complexity of smart contract language.[49] De Filippi and Hassan divide the increasing complexity

---

45. As noted above, blockchains are examples of distributed ledger technologies. They are databases, maintained on many computers, to which anyone can write, but no-one can falsify, at least under standard cryptographic and game-theoretic conditions. *See* Joshua A.T. Fairfield, *BitProperty*, 88 S. CAL. L. REV. 805, 816–819 (2015).

46. *See* Nick Szabo, *Smart Contracts*, https://www.fon.hum.uva.nl/rob/Courses/InformationInSpeech/CDROM/Literature/LOTwinterschool2006/szabo.best.vwh.net/smart_contracts_2.html (last visited Feb. 24, 2019).

47. *Id.*

48. Maher Alharby & Aad van Moorsel, *Blockchain-Based Smart Contracts: A Systematic Mapping Study*, ARXIV (Oct. 17, 2017), https://arxiv.org/abs/1710.06372.

49. Goldenfein interestingly suggests that the present challenge of creating a legal framework to govern blockchain, echoes the process of systemising writs during the Medieval times. Goldenfien & Leiter, *supra* note 41, at 144–45. He terms early writs as 'technological artefacts' as they translated physical reality into the engineered register of the writs. In this way they operated to connect human conduct to the institutional systems of the courts. *Id.* at 144. In a similar way, Goldenfein argues, blockchain systems create a library of possible engagements with the techno-legal world. *Id.* In this Article,

of smart contracts into four distinct phases.[50] The first phase involved the digitization of information, converting paper text into data and creating large databases. The second phase involved the automation of decisionmaking processes, such as governmental processes in the field of tax assessment and private sector accounting and credit assessment tools. This was followed by the third phase of incorporating legal rules into software code, leading to the concept of regulation by code. This concept, termed "lex informatica,"[51] was further developed by Lessig in his thesis that "code is law."[52]

The fourth, and currently unfolding, phase identified by De Filippi and Hassan is the codification of law involving blockchain platforms.[53] This phase is distinguished by the transition from reliance on code to enforce laws to reliance on code to also draft and elaborate upon such law. They argue that blockchain has been the driver of this fourth and most dynamic phase of the evolving relationship between law and code.

Another commentator has observed that today's smart contracts differ from previous forms of automated exchange[54] in the complexity of the arrangements, with this new class of smart contracts including transfers of real property, intellectual property rights, and licences.[55] This in turn requires more complex systems to integrate such smart contracts with real world organisational and institutional systems. And as smart contract transactions become more commercially valuable, mechanisms have also been developed to contest and reverse smart contracts. Thus, in addition to

---

we further suggest that the critical connection between the technical and legal environments is language which converts legal rights and duties into code.

50. Hassan & De Filippi, *supra* note 34, at 88–89.

51. Joel R. Reidenberg, *Lex Informatica: The Formulation of Information Policy Rules Through Technology*, 76 TEX. L. REV. 553, 554–55 (1998).

52. LAWRENCE LESSIG, CODE AND OTHER LAWS OF CYBERSPACE 5 (1999).

53. Hassan & De Filippi, *supra* note 34, at 89–90.

54. Vending machines and automated parking station are early examples of automated transactions. *See* Jonathan Rohr, *Smart Contracts in Traditional Contract Law, Or: The Law of the Vending Machine*, 67 CLEV. ST. L. REV. 67, 69–70 (2019) (drawing comparisons between vending machines and smart contracts).

55. Sinclair Davidson et al., *Blockchains and the Economic Institutions of Capitalism*, 14 J. OF INSTITUTIONAL ECON. 639, 646 (2018).

creating a relationship between software code and internal processes, it is also necessary to connect the smart contracts to off-chain real world dispute resolution processes.

Smart contracts produce a variety of well-documented economic benefits, including efficiency,[56] immutability, security, convenience, and the execution of transactions in a "trustless environment."[57] By eliminating the need for relationships of trust between contracting parties, blockchain-based smart contracts serve to coordinate individuals, including large groups of individuals, that do not know or trust each other.[58] The eradication of the need for trust between parties also extends to central authorities because blockchain vitiates the need to rely on or trust such external organizations.[59] And smart contracts enable industrial society to operate more effectively by reducing reliance on intermediaries and decreasing the transaction costs.[60] Of course, immutability does not always support trust because, while blockchain guarantees that a transaction was recorded, it does not guarantee that the maker of the transaction was the lawful proprietor of the private key.[61]

As to blockchain's claims regarding security, secure smart contracts can build a secure global contracting environment, facilitating more efficient international commerce.[62] Researchers have constructed a security

---

56. *See* Sinclair Davidson et al., *Economics of Blockchain*, SSRN (Mar. 9, 2016), https://papers.ssrn.com/sol3/papers.cfm?abstract_id=2744751; *see generally* Rainer Böhme et al., *Bitcoin: Economics, Technology and Governance*, 29 J. OF ECON. PERSPS. 213 (2015).

57. Eghbal Ghazizadeh & Tong Sun, *A Systematic Literature Review of Smart Contract Applications*, 2020 PROC. OF THE FUTURE TECHS. CONF. 877, 881 (2020).

58. *See* Primavera De Filippi & Samer Hassan, *Blockchain Technology as a Regulatory Technology: From Code is Law to Law is Code*, FIRST MONDAY (Nov. 14, 2016), https://firstmonday.org/ojs/index.php/fm/article/view/7113.

59. *See* Yeoh, *supra* note 1, at 196.

60. *See* Max Raskin, *The Law and Legality of Smart Contracts,* 1 GEO. L. TECH. L. REV. 305, 309 (2017).

61. *See* Kevin Werbach, *Trust, But Verify: Why the Blockchain Needs Law*, 33 BERKELEY TECH. L.J. 489, 494–96 (20162018) (discussing the theft of Ether and how the immutability of blockchain records prevented stopping or reversing the theft).

62. *See generally* RICHARD MA ET AL., FUNDAMENTALS OF SMART CONTRACT SECURITY (Lisa MacLean ed., 2019), for a discussion of the security vulnerabilities associated with smart

framework to connect people, systems, and processes, creating an institutional governance model for investigating technical errors and vulnerabilities.[63] Others have further examined the problems created by the leaking of private data and its potential criminal exploitation.[64] One commentator, in outlining the benefits of smart contracts, notes their potential to lower losses generated by fraud.[65] Another notes that smart contracts enable the creation of "pools of resources and their allocation according to agreed criteria," facilitating innovative financial activities such as crowdfunding.[66] Finally, in relation to enforcement, the economic benefits of smart contracts have been said to include avoiding the cost of arbitration proceedings and court enforcement costs.[67]

However, as has been widely noted, smart contracts are, in many circumstances, accompanied by certain legal uncertainties and vulnerabilities. Due to a lack of common understanding between lawyers and computer programmers, many legal loopholes exist in present smart contracts. It is therefore critical to align the formal language of computing with the natural language of contract law. This littoral zone between the virtual and the real, between the automated and the non-automated, and between formal and natural language, will be the focus of the next Subpart of the Article.

C. The Evolving Techno-Legal Smart Contract Language

The legality of smart contracts depends on properly connecting the formal language of computational transactions to the natural language of

---

contracts. The authors argue that while blockchains are secure, smart contracts are not, and presents a variety of smart contract principles and practices that can help strengthen security.

63. *See id.*

64. *See* Yilei Wang et al., *Randomness Invalidates Criminal Smart Contracts*, 477, INFO. SCI. 291 (2019).

65. Nick Szabo, *Smart Contracts*, PHONETIC SCIS. (1994), https://www.fon.hum.uva.nl/rob/Courses/InformationInSpeech/CDROM/Literature/LOTwinterschool2006/szabo.best.vwh.net/smart.contracts.html.

66. Alexander Savelyev, *Contract Law 2.0: 'Smart' Contracts as the Beginning of the End of Classic Contract Law*, 26 INFO. & COMMC'NS. Tech. L. 116, 120-122 (2017).

67. *See* Szabo, *supra* note 65.

contracts. In this techno-legal blockchain-based smart contract environment, technical standards, guidelines, and protocols are the language that mediates between code and law. This language is created by hybrid techno-legal instruments and operates to model the potential actions and behaviors of parties to smart contracts and create packages to govern both technical and legal transactions. This language is critical in supporting compatibility and interoperability between the smart contract software and the law of contract. Thus, it is important to get it right.

A variety of smart contract languages (SCLs) have been developed to translate contractual concepts and principles into code, and some valuable research has been conducted by computer engineers on the operation and merits of the various SCLs. DAO smart contracts are characterised by their incorporation of sophisticated rules as to governance and decision-making.[68] SCLs implementing self-executing smart contracts include Solidity, Michelson, and Rholang.[69]

Each SCL has semantic peculiarities and spectres of use.[70] For example, domain-specific SCLs are commonly used to support online voting and crowdfunding.[71] In contrast, formally verifiable SCLs provide sophisticated semantics for framing contractual obligations and rights. So-called "easy-to-use" SCLs translate high-level Solidity Code into low-level Bytecode, making it easier to apply the language.[72]

But how successful is a particular SCL language at capturing the intentions of parties, including ensuring that the smart contracts retain their desired level of flexibility and discretion? How successful are such technical standards and guidelines in ensuring that transactions comply with the principles of contract law? In the field of computing, substantial

---

68. *See* Singh & Kim, *supra* note 43, at 116–17.

69. *See* further Dwivedi et al., *supra* note 44, at 2 (discussing Solidity, Michelson, and Rholand); *cf.* KEVIN SOLORIO ET AL., HANDS-ON SMART CONTRACT DEVELOPMENT WITH SOLIDITY AND ETHEREUM: FROM FUNDAMENTALS TO DEPLOYMENT (O'Reilly 2020) (discussing Solidity).

70. *See* Dwivedi et al., *supra* note 44, at 2.

71. *See* Ilya Sergey et al., *Safer Smart Contract Programming with Scilla*, 3 PROC. ACM ON PROGRAMMING LANGUAGES 1, 2 (2019), https://doi.org/10.1145/3360611.

72. *See* Dwivedi et al., *supra* note 44, at 19.

work is being done to develop mechanized means of validating formal syntax and semantics, such as Solidity and Lolisa.[73]

Moving from the technical to the legal, lawyers' unfamiliarity with the technology and the programmers' unfamiliarity with law have made smart contracts an uncertain and little-researched area of law. As a variety of computing scholars have noted,[74] although blockchain supports the drafting of legal contracts, "the underlying contractual concepts and properties necessary to render said smart contracts legally binding (which [computer scholars] refer to as "suitability"), are still less researched."[75] The relationship between software code and the natural language is often unclear. And in the case of immutable blockchain smart contracts, it is also inflexible and opaque. Within this context, the next two Subparts consider how the language of smart contracts differs from traditional contracts, and analyze how such differences lead to challenges in seamlessly applying established contractual principles and protections. Rather than seek to consider all concepts and principles of contract law, the Subparts will focus on two specific concepts that are particularly difficult to apply and uphold in the environment of blockchain smart contracts: discerning intent and determining when a contract has been properly performed.

D. The Challenge of Articulating Intent in Smart Contracts

A defining feature of a smart contract is its establishment of consensus between parties who may not know or trust each other. This feature, however, gives rise to certain legal challenges related to discerning intent and establishing consent. While the smart contract establishes technological consensus, to be valid under contract law, it must also embody the genuine contractual consent of all individuals to the

---

73. *See id.* at 16 (discussing Solidity and Lolisa); *cf.* Karthikeyan Bhargavan et al., *Formal Verification of Smart Contracts: Short Paper*, *in* PLAS '16: PROCEEDINGS OF THE 2016 ACM WORKSHOP ON PROGRAMMING LANGUAGES AND ANALYSIS FOR SECURITY 91 (2016), https://doi.org/10.1145/2993600.2993611 (discussing Solidity).

74. *See, e.g.*, Usman W. Chohan, *The Decentralized Autonomous Organization and Governance Issues*, *in* NOTES ON THE 21ST CENTURY 1 (2017), https://doi.org/10.2139/ssrn.3082055; Mark Giancaspro, *Is a 'Smart Contract' Really a Smart Idea? Insights from a Legal Perspective*, 33 COMPUT. L. & SEC. REV. 825 (Dec. 2017), https://doi.org/10.1016/j.clsr.2017.05.007; Goldenfein & Leiter, *supra* note 41.

75. *See* Dwivedi et al., *supra* note 44, at 2.

transaction. One commentator frames this problem as the tension created between the establishment of "decentralised consensus" and information distribution.[76] This idea could also be defined as the challenge of reconciling "decentralised consensus" with individual intent. These two are not easy to reconcile—while the validity of contracts is dependent on the proper application of the contractual principles of offer, acceptance, consideration, and capacity, smart contracts are drafted within the limitations of smart contract language. Moreover, while blockchain systems do maintain detailed records of individual transactions that can be used to establish the required intent and consent to terms, many algorithms supporting such decentralised systems come down to some form of majority vote.[77] This problem is exacerbated if a hacking incident causes the blockchain platform to be modified to maintain security, causing forking and a change to the underlying system.[78] This then calls into question the continuing relevance of the initial consent of parties.

Moreover, even if a contract evinces an intention by all parties to enter the contract and accept its terms, the contractual doctrines of mistake, unconscionability, duress, and undue influence allow for contract to be rescinded if it can be shown that such consent was illegally obtained, or that the contract as executed contravenes objectively discernible intentions of the parties or valuable social norms (e.g. unconscionability). In the case of smart contracts, there is a high likelihood that the parties will misunderstand or abuse the terms of the blockchain agreement. This is accentuated when such agreements involve members of the general public (such as in crowdfunding situations) who may have limited understanding of the operation of smart contracts and how they differ from online consumer contracts (such as those relating to online shopping). In determining the capacity of a party to a contract, the court in *Saunders v.*

---

76. Lin W. Cong & Zhiguo He, *Blockchain Disruption and Smart Contracts*, 32 REV. FIN. STUD. 1754, 1755 (2019), https://academic.oup.com/rfs/article/32/5/1754/5427778.

77. *See id.* at 1761.

78. *See* Stephen Penzo & Niloufer Selvadurai, *A Hard Fork in the Road: Developing an Effective Regulatory Framework for Public Blockchains,* INFO. & COMMC'NS TECH. L. (July 27, 2021), https://www.tandfonline.com/doi/full/10.1080/13600834.2021.195972; *see also* Angela Walch, *The Bitcoin Blockchain as Financial Market Infrastructure: A Consideration of Operational Risk,* 18 N.Y.U. J. LEGIS. & PUB. POL'Y 837, 866–67 (2015).

Anglia Building Society held that an inability to properly understand computer code does not constitute an "innate capacity."[79] Nonetheless, the inability to understand code could, under the right circumstances, constitute an incapacity that vitiates consent. This would be consistent with cases such as *Petelin v. Cullen*[80] which have held that an inability to understand English amounted to an incapacity.[81]

Additionally, the law relating to third-party beneficiaries in smart contracts is also under-theorized. Many blockchains are managed by foundations, or at least have interests that are promoted by foundations. Those foundations pre-mine currency, sell some of it, and try to set terms and conditions of use. For example, a proof-of-stake chain might have a transactional wallet with terms and conditions built into the End User License Agreement (EULA) which arguably benefit other members of the community. If someone dupes the currency, it is unclear whether community members whose holdings are diluted can sue on that contract. Lawyers are trained to eliminate third-party beneficiaries from online contracts, but in smart contracts, the insertion of express third-party beneficiary clauses may be necessary.

E. The Language of Smart Contract Performance and Breach

While blockchain platforms offer the benefits of immutability, the corollary is that smart contracts only encompass perfect performance. The extensive scholarly literature on the inflexibility of smart contracts,[82] while focused on the logistics of execution, also demonstrates this problem. One commentator points out that while smart contracts create rights, they do

---

79. Saunders v. Anglia Bldg. Soc'y [1971] AC 1004 (HL) 1016 (appeal taken from Eng.).

80. *Petelin v Cullen* (1975) 132 CLR 355, 359–360 (Austl.).

81. *See also* Gabriel Olivier Benjamin Jaccard, *Smart Contracts and the Role of Law*, JUSLETTER IT (Nov. 23, 2017), https://jusletter-it.weblaw.ch/en/issues/2017/23-November-2017/smart-contracts-and-_42155d7e26.html__ONCE&login=false (subscription required).

82. *See* José Carlos Pereira, *The Genesis of the Revolution in Contract Law: Smart Legal Contracts*, *in* ICEGOV2019: PROCEEDINGS OF THE 12TH INTERNATIONAL CONFERENCE ON THEORY AND PRACTICE OF ELECTRONIC GOVERNANCE 374 (2019), https://dl.acm.org/doi/10.1145/3326365.3326414; *see also* Judah A. Druck, *"Smart Contracts" Are Neither Smart Nor Contracts,* 37 NO. 10 BANKING & FIN. SERVS. POL'Y REP., Oct. 2018, at 5, 7; *see also* KRISTIAN LAUSLAHTI ET AL., SMART CONTRACTS – HOW WILL BLOCKCHAIN TECHNOLOGY AFFECT CONTRACTUAL PRACTICES? 9 (Rsch. Inst. Finnish Econ., ETLA Reports No. 68, 2017).

not create legal obligations as understood by traditional contract law. The concept of obligation is not limited to specific rights or duties, but encompasses the whole relationship between the parties and serves to regulate human interactions. In comparison, a smart contract involves the self-limiting of rights through technical means. Hence, once the smart contract is entered into, it is not feasible to change the terms, such as those governing place and time of performance, even in response to unforeseen events. This creates a direct conflict with doctrines like commercial impracticability or force majeure.

Further, it is not possible for the terms governing breach of contract to be interpreted in response to events that have not specifically been addressed in the smart contract terms. The embedding of obligations in immutable code means that traditional doctrines of frustration and impossibility will not apply. For example, if an event took place that rendered the smart contract unperformable, but that had not been foreseen and embedded in the code, a party would be in breach regardless of the impossibility of performance. Hence, a cyber-attack that results in the inability of one or more parties to perform the contract will not change the terms of the smart contract. Given the variations in possible cyber-attacks and the many ways in which such a cyber-attack could affect the terms of the contract, it will often be impossible to plan for all such contingencies when setting up the smart contract.

Even where both parties wish to modify the smart contract, such as in response to a changed commercial circumstance, the blockchain system will not typically permit such amendments to rights and duties. This raises critical issues relating to the application of established contractual principles, such as those governing force majeure events. This is unsatisfactory as both consumers and companies want some flexibility in commercial dealings, and may wish to avoid a breach situation by renegotiating certain terms.[83] In such circumstances, while it would of course be possible to create a new smart contract, the initial smart contract would continue its self-execution on the basis of the predetermined terms.

This limitation of the capacity to modify is exacerbated because the language of smart contracts also does not typically allow ambiguity. In a

---

83. *See* Danielle D'Onfro, *Smarts Contracts and the Illusion of Automated Enforcement,* 61 WASH. U.J. L. & POL'Y 173, 182–83 (2020) (discussing why consumers and corporations value such flexibility.).

traditional contract, parties may use widely defined terms and deliberately vague language to encompass a myriad of unforeseen circumstances. Such a use of language, while reducing certainty, has the benefit of providing flexibility regarding operation of the contract. But as one commentator has observed, while "[a]mbiguity is celebrated in human language . . . [a]mbiguity is anathema to computer language."[84]

Finally, remedies for breach of contract, including damages, penalties, and equitable remedies such as injunctions and specific performance, would be unavailable unless expressly provided for in the terms of the blockchain. This has the potential for a smart contract to limit contractual remedies available under the general law. Again, this raises potential for the misleading of consumers, especially those who do not have a firm understanding of how smart contracts operate.[85]

II. GOVERNING THE INTERFACE BETWEEN NATURAL AND FORMAL LANGUAGE IN SMART CONTRACTS

Building on the above discussion of the challenges of translating concepts and principles into code, and the corresponding tension between the formal language of computers and the natural language of contract, it is useful to consider whether, and to what extent, these challenges have been addressed through legal reform. In the last two decades, nations around the world have enacted legislation to address the challenges created by electronic contracts. While no nation has enacted legislation to regulate smart contracts, it is useful to examine the extent to which electronic contract laws address the legal challenges relating to smart contracts. The

84. *See* Raskin, *supra* note 60, at 325.

85. In addition to the issues discussed, there is the further challenge of enforcing smart contracts in the international sphere and the challenges of governing the seamless internet with jurisdictional rules that align to the geographical sovereign borders of nations. This issue of internet jurisdiction is outside the ambit of the present discussion. *See generally* Niloufer Selvadurai, *The Proper Basis for Exercising Jurisdiction in Internet Disputes: Strengthening State Boundaries or Moving Towards Unification?*, 13 PITT. J. TECH. LAW L. & POL'Y 124 (2013); *see also* Bedrettin Gürcan, *Jurisdiction on the Blockchain*, in CONFERENCE PROCEEDINGS OF THE OXFORD CONFERENCE SERIES: MARCH 2020 14 (2020), https://www.researchgate.net/publication/345176938_JURISDICTION_ON_THE_BLOCKCHAIN (specifically discussing jurisdiction on blockchain-based smart contracts); Karen Yeung, *Blockchain, Transactional Security and the Promise of Automated Law Enforcement: The Withering of Freedom Under Law?*, in 3TH1CS: A REINVENTION OF ETHICS IN THE DIGITAL AGE 1, 13 (Philipp Otto & Eike Gräf eds., 2017).

jurisdictions of the United States and Australia have been selected for analysis because they have relatively sophisticated electronic contracts laws and have a mature law reform discourse considering further enhancements. The European Union's e-Commerce Directive has also been selected for analysis because it provides useful insights to overcoming the contractual challenges generated by automation.[86] It may be useful to highlight our conclusions at the outset: as countries have enabled electronic contracting, including smart contracting, they have adopted the paradigm that such contracts express human expectation, and they have rejected both expressly and by implication the emerging trend of holding human expectations hostage to machine outcomes. Or, simply put, these frameworks support our conclusion that where human expectations denoted in natural language differ from machine outcomes expressed in formal computing code, expectations prevail.

As both the United States and Australia have adopted many of the United Nations Commission on International Trade Law (UNCITRAL) enactments relating to electronic commerce, there is a degree of harmony between these two nations.[87] The United States, Australia, and certain European Union nations have enacted domestic legislation to give effect to the UNICTRAL provisions. And several of these jurisdictions have enacted further specific laws to support electronic commerce.

The most widely enacted UNCITRAL model law, the UNCITRAL Model Law on Electronic Commerce,[88] is based on the principles of

---

86. Directive 2000/31/EC, of the 0001 – 0016.European Parliament and of the Council of 8 June 2000 on Certain Legal Aspects of Information Society Services, in Particular Electronic Commerce, in the Internal Market, 2000 O.J. (L 178).

87. U.N. Comm'n on Int'l Trade L. [UNCITRAL], United Nations Convention on the Use of Electronic Communications in International Contracts, U.N. Sales No. E.07.V.2 (2005), 2007; together with the UNCITRAL, UNCITRAL Model Law on Electronic Transferable Records (2017), U.N. Sales No. E.17.V.5 (2018) [hereinafter UNCITRAL Electronic Transferable Records]; UNCITRAL, UNCITRAL Model Law on Electronic Signatures (with Guide to Enactment 2001), U.N. Sales No. E.02.V.8 (2002) [hereinafter UNCITRAL Electronic Signatures]; and the UNCITRAL, UNCITRAL Model Law on Electronic Commerce with Guide to Enactment (1996) with Additional Article 5 Bis, as Adopted in 1998, U.N. Sales No. E.99.V.4 (1999) [hereinafter UNCITRAL Electronic Commerce]. These all facilitate electronic commerce and have been adopted by over 100 nations.

88. UNCITRAL Electronic Commerce, *supra* note 87.

technology neutrality and nondiscrimination against the use of electronic means. It recognises the functional equivalence of contracts made by electronic and paper-based means. Additional rules on the use of electronic signatures are provided by the UNCITRAL Model Law on Electronic Signatures.[89] More recently, the UNCITRAL Model Law on Electronic Transferable Records[90] recognises the use of electronic transferable documents, including bills of exchange, bills of lading, cheques, and promissory notes. Most recently, the UNCITRAL Notes on the Main Issues of Cloud Computing Contracts[91] considers the development of an instrument on the use and cross border recognition of electronic identity management services (IdM services) and authentication services (trust services).

A thread runs through these statutes and the ones that follow: electronic contract, including smart contract, is subject to the regular rules of contracting practice. The rule is one of *equivalence*, not *exceptionalism*. Across the board, enabling statutes have determined that the principles of contract doctrine extend to online and electronic practices. Although this principle may seem almost pedestrian (and the statutes that enshrine it are certainly not flashy), the implications of this point are far-reaching. We must enable true contracting doctrine in smart contracts. Neither the statutory or case law have a legitimate basis for smart contract exceptionalism, for the exclusion of the practice of discerning human intent from the language, facts, and circumstances surrounding a smart contract.

In the United States, state and federal common law, together with specific statutory laws governing certain types of contracts, govern the formation and enforcement of contracts. The Uniform Commercial Code (UCC) proposed Article 12 may develop an equivalence between smart-contract-transferred assets and intangible assets subject to control (as used in Article 8 governing securities, for example). The state Uniform Electronic Transactions Act (UETA) applies to contracts that have been enacted in a digital format for which the contracting parties have "agreed to conduct

89. UNCITRAL Electronic Signatures, *supra* note 87

90. UNCITRAL Electronic Transferable Records, *supra* note 87.

91. UNCITRAL, Notes on the Main Issues of Cloud Computing Contracts, 2019 U.N. Doc. A/CN.9/974, U.N. Sales No. Not Printed (2019).

transactions by electronic means."[92] The federal Electronic Signatures in Global and National Commerce Act (ESIGN)[93] contains further provisions as to the recognition and authentication of electronic signatures and electronic records.

While these statutes do not expressly refer to smart contracts, the broad references to digital format encompass blockchain-based smart contracts. They operate to prevent a contract from being denied legal effect solely because it is in an electronic form. Hence, the use of a blockchain platform will not affect validity. The UETA and ESIGN operate to further ensure that an electronic signature will satisfy legal requirements as to signature, and that an electronic record will satisfy the requirement for a record to be in writing. Cryptographic signatures used in blockchain are thus electronic signatures. Further information stored on the distributed ledger will be legally recognised as an electronic record.

But, while these provisions operate to uphold the contractual validity of blockchain transactions, they do not offer a solution to the problems of discerning intent and ensuring performance discussed above. The statutes thus entrench some of the problems caused by immutability discussed above.[94] Further it is relevant to note that the UETA and ESIGN framework does not grant legal effect to all electronic contracts: A variety of exemptions apply to the use of electronic signatures with respect to, for example, the formation of wills, codicils, testamentary trusts, official court documents, and documents related to family law matters. Accordingly, blockchain-based contracts in these areas may not be recognised, despite the fact that using a smart contract to transfer assets upon the occurrence of a easily discernible event (e.g., death) is a core application of the technology.

In Australia, the federal Electronic Transactions Act 1999 (ETA),[95] mirrored by the various State and Territory Acts, also operate to give formal recognition to contracts made using electronic means. The stated aim of the ETA is to provide a regulatory framework to facilitate and support e-

92. Unif. Elec. Transactions Act § 5(b) (Unif. L. Comm'n 1999).

93. 15 U.S.C. § 7001.

94. *See id.*

95. Electronic Transactions Act 1999 (Cth) (Austl.).

commerce, inspire public confidence in electronic trading, and enable "businesses and the community to use electronic communications in their dealings with government."[96] More specifically, it aims to eliminate barriers to the conduct of electronic transactions and ensure that a transaction would not be unenforceable because it was created in an electronic environment.[97] As the Act governs "information systems," defined as "systems for generating, sending, receiving, storing or otherwise processing electronic communications,"[98] it would encompass blockchain platforms. While its definition of "electronic communication" is technology specific, being "a communication of information in the form of data, text or images by means of guided and/or unguided electromagnetic energy . . . ," it would encompass blockchain transactions. And while the definition of "data storage device" is likewise technology specific, defined to be "any article or material (for example, a disk) from which information is capable of being reproduced, with or without the aid of any other article or device," it would encompass distributed ledgers.

As with its equivalent in the United States, the Australian legislation mandates the functional equivalence of electronic and paper contracts, stipulating that a transaction cannot be denied legal effectiveness because it is wholly or partly created by the use of electronic communications.[99] Electronic communications are stipulated to satisfy the writing requirement and requirements as to electronic signatures.[100] Special provisions as to presumptions applying to time and place of dispatch and receipt[101] help clarify the application of traditional contract law principles to electronic transactions. Of course, this general rule can be displaced by a more specific provision in the Act.[102]

---

96. *Id.* § 3.

97. *Id.*

98. *Id.* § 5.

99. *Id.* § 8(1).

100. *Id.* § 10.

101. *Id.* §§ 14–14B.

102. *Id.* § 8(2). As with the United States equivalent, the Australian Act does not recognize electronic signatures for certain prescribed transactions, including wills.

In the European Union, the e-Commerce Directive[103] forms the foundational governance framework for online services, creating harmonised rules to support transparency in the provision of online services and facilitate commercial communications and electronic contracts. In Section 3 on Contracts concluded by electronic means, Article 9.1 provides that "Member States shall ensure that their legal system allows contracts to be concluded by electronic means." Member States shall, in particular, ensure that the legal requirements applicable to the contractual process neither create obstacles to the use of electronic contracts nor result in such contracts being deprived of legal effectiveness and validity on account of their having been made by electronic means. Article 9.2 further provides:

Member States may lay down that paragraph 1 shall not apply to all or certain contracts falling into one of the following categories: (a) contracts that create or transfer rights in real estate, except for rental rights; (b) contracts requiring by law the involvement of courts, public authorities or professions exercising public authority; (c) contracts of suretyship granted and on collateral securities furnished by persons acting for purposes outside their trade, business or profession; (d) contracts governed by family law or by the law of succession.[104]

"Electronic contract" is not defined in Article 2, but the breadth of the term seems to encompass smart contracts which can be run automatically.

Therefore, while the United States' UETA and ESIGN, the Australian ETA, and the European e-Commercial Directive provide a measure of legal and commercial certainty to the use of blockchain-based contracts, they do not solve the fundamental problems raised above.[105] A more developed governance framework is required. This will be the focus of the next Part of this Article.

III. TOWARDS A NEW LEGAL FRAMEWORK TO GOVERN THE INTERFACE BETWEEN NATURAL AND FORMAL LANGUAGE IN SMART CONTRACTS

A. Indicia for Judging the Effectiveness of Smart Contract Law

103. Directive 2000/31/EC, *supra* note 86.

104. *Id.* art. 9.2.

105. *See supra* Part I.

When natural language human expectations for legal arrangements conflict with the inevitably buggy emergent nature of code, especially where bugs are actively exploited by humans, what happens? When a hacker exploits vulnerabilities in the DAO to extract investors' money, should the community have the right to fork the currency to set the clock back?[106] Moreover, should community members have causes of action in unjust enrichment, money had and received, conversion, or replevin against counterparties who exploit software vulnerabilities? Blockchain is a community coordination technology, and blockchain agreements often operate to benefit communities whose members come and go. Contract is a mode intended to capture an agreement at a specific time and place between specified parties. How should courts respond to legal language buried in code, where no human sees it, or code buried in legal language, where few humans can understand it (and none can be sure of its operation)?

Our approach is marked by a few signposts. First, we believe contracting must remain a human-centered activity. Courts should reject the rising trend of treating exploits in blockchain contractual arrangements as expressions of the deal human counterparties wanted. Second, we believe that progress is best made by building on and extending common law and historically established legislative approaches for blockchain-based negotiated agreement rather than attempting to build a new law from scratch. Third, we recognize wrinkles created by the technology (immutability and the community nature of these arrangements foremost among them) but believe these can be addressed best by looking to what humans wished to do—the expressions of their agreement in natural language—rather than looking to software outcomes.

In order to develop a new governance framework for smart contracts, it is first necessary to establish appropriate criteria for judging effective laws in this area. Analyzing the issue through a technical lens, some have suggested that the ontological suitability of a SCL can be judged using two

106. Because a blockchain is a ledger with an immutable history—what was written cannot be unwritten—the only way to correct gross errors on the chain itself is to "fork" the chain; that is, the community accepts that a new record, a new chain, is the one true accepted chain, and the old record is disregarded by the software. This has been used in both the case of the DAO and the Icon exploit to revert or freeze assets that a hacker used an exploit to obtain. *See* Penzo & Selvadurai, *supra* note 78.

indicia. First, the appropriateness of "the choreography or workflow of processes" in the SCL in relation to concepts. Second, the appropriateness of the semantics that define processes in relation to properties.[107] But what legal lens should be used to identify such criteria? As discussed earlier, Lessig famously postulated that code is law:

> The code regulates. It implements values, or not. It enables freedoms, or disables them. It protects privacy, or promotes monitoring. . . . Thus the choice is not whether people will decide how cyberspace regulates. People—coders—will. The only choice is whether we collectively will have a role in their choice.[108]

Of course, a number of scholars have however contested this notion. One such scholar proposes there are two distinct aspects of code's relationship with law.[109] The first—Lessig's concept "that computer code can substitute for law or other forms of regulation"—must be accompanied by the second: an understanding of how code can operate as an "anti-regulatory mechanism," a tool that certain groups will use to their advantage to minimise the costs of legal compliance.[110] This scholar suggests that the design of code should be studied as just one aspect of interest group behavior.

Extending this notion, another scholar proposes a sharp analytical distinction between the realms of technology and of law, arguing that while "[t]he question to what extent the law 'can' be digitalized relates to technology, whereas the question to what extent it 'may' be digitalized falls

---

107. *E.g.*, Dwivedi et al., *supra* note 44; *see also* Alex Norta et al., *eContractual Choreography-Language Properties Towards Cross-Organizational Business Collaboration*, 6 J. INTERNET SERVS. & APPLICATIONS, no. 8, 2015, https://doi.org/10.1186/s13174-015-0023-7.

108. Lawrence Lessig, *Code Is Law: On Liberty in Cyberspace*, HARV. MAG. (Jan. 1 2020), https://www.harvardmagazine.com/2000/01/code-is-law-html; *see also* Lessig, *supra* note 52.

109. Tim Wu, *When Code Isn't Law*, 89 VA. L. REV. 679, 682 (2003).

110. *Id.*

within the realm of the law."[111] While Lessig's first intuition that law is being replaced by automation is largely true, his second implication—that this makes machine code equivalent to legal understandings—is wholly inaccurate, and forms the foundation of what has gone wrong in the law and theory of smart contracting.

Still, other scholarly works have analysed the intersection between conventional law produced and enforced by national legal systems—the "code of law"—and the internal rules of blockchain systems which form executable software code and cryptographic algorithms which operate across distributed computing networks, or "code as law."[112] But, while this scholar concludes that the success of blockchain will depend on "effective and legitimate governance structures," encompassing "both the code that controls the operation of digital technologies (code as law) as well as the conventional rules provided by national legal systems (code of law),"[113] precise guidance is not provided on the potential nature and operation of such "effective and legitimate governance structures."

B. A New Legal Framework to Govern the Interface between Formal and Natural Language of Smart Contracts

To effectively proceed, we stress the similarity of contractual arrangements through blockchain to regular everyday contracts (since it is only through this law of contract that an arrangement can obtain legal validity), with specific reference to satisfying human preferences through the division of consumer surplus as a result of negotiated agreement. Humans want something out of smart contracts, and the law should tend to ensure that they get it. The law of unconscionability should minimize oppression and surprise, as it always has. The law of good faith and fair dealing should continue to operate to penalize contracting parties who exploit software bugs to subvert the expectations of the parties. Courts will need to interpret contract terms for their objective meaning between the parties rather than rely on the operation of code. Courts will need to admit parol evidence regarding the actual understanding of the parties to agreements expressed in code—formal code, particularly buggy formal

---

111. Jan Oster, *Code is Code and Law is Law—The Law of Digitalization and the Digitalization of Law*, 29 INT'L J. LAW L. & INFO. TECH., 101, 101 (2021).

112. Karen Yeung, *Regulation by Blockchain: The Emerging Battle for Supremacy Between the Code of Law and Code as Law*, 82 MOD. L. REV. 207, 207 (2019).

113. *See id.* at 239 (emphasis omitted).

code, has no expression of meaning for oral or extrinsic evidence to contradict.

We believe an iterative, experimental, and humble analysis based on analogy to regular contracting practice will yield better results than tech evangelism. Blockchain is a database in which human preferences can be recorded in natural language and executed in code. Where the two diverge, courts can take steps to align the execution of the contract with the expectation of the parties.

In particular, this requires specific legal moves with respect to the interface between natural language and formal language. At a minimum, courts should focus on and privilege natural language expressions of the intent of the parties, particularly those that show indicia of dickered consent. Courts should disfavor boilerplate buried in the code of a smart contract where no human can see it, and should question the value of assuming that where a human and computer meet and exchange text, that the machine-proffered version should prevail. Courts should take seriously the fact that human intent *cannot* be perfectly expressed in code, since both internal bugs and external emergent interactions with other software can always create variance between intent and execution. And the concept of a contract that cannot be breached is both legal nonsense and likely violates rules against exculpation—why would one have the legal form at all if the mechanical operation of a software program were synonymous with legal performance?

Once a background rule of similarity through analogy and a human focus are established, courts can more easily deal with the few and relatively unimportant differences between contracts expressed in natural language and automatically executed via blockchain, and any other database technology. The first key technological difference is immutability. Imagine Anne forces Bill, at gunpoint (or by fraud, or by software exploit), to authenticate a smart contract transfer of the digital Mona Lisa. The form of the transaction will look like any other transaction executed by smart contract. Bill will want the Mona Lisa back. Of course, courts should protect Bill's interests despite the facially valid contractual transfer. It is true that the transfer will be immutably recorded on the blockchain, but this is no bar to sane adjudication. If a court can reach Anne or her assets, it can impose sanctions requiring her to produce and return the NFT. If it cannot, it cannot force the blockchain to recognize the transfer. But neither can a court force the return of a physical asset if it cannot find the asset or the perpetrator. And in the meantime, the perfect provenance of blockchain

technology will render the digital Mona Lisa dead in the commercial water: everyone will know it is stolen, and any transfer of the asset would be a public transaction recorded on the blockchain. In short, immutability creates no more problem for NFTs than it does in any other circumstance of force, fraud, or the like.

The most significant challenge to evolving common law and historical legislative contract doctrine to cover smart contracts will be problems of community. Contracts are a form intended to capture the intentions of two or more parties at a given moment in time for a set expectation in the future. That is not at all to say that contracts cannot be long-term fluid arrangements between larger groups, but merely that some innovation will be required to standardize such practice. The natural language expression of the intentions of parties who engage in smart contracting will likely be EULA-style framework contracts that set the rules of the road, similar to a master trading agreement, a constitution for the online community, or the master agreements for electronic data interchange. The question of whether natural language or automatic execution should be deemed the expression of the parties' intent is less problematic in these scenarios, with the law of boilerplate fairly well established. The harder question will be enabling contractual rights between people who enter the community at different times, with promises that are not directly reciprocal to one another (that is, both owe contractual duties to the community to refrain from exploitation or token duplication). There are problems of third-party beneficiaries and of virality. Imagine Alan gives Meg cryptocurrency governed by a community constitution. Alan is likely bound by the constitution because he purchased the tokens through a website, or because he downloaded the community's wallet app. But the degree to which Meg is bound is unclear.

Yet even this hardest problem is hardly insurmountable if courts follow the approach espoused here. It may simply require a foray into the law of rights in things—common law property. Contracts can set rights that travel with property, and if those rights are properly recorded (there is no better technology for doing so than blockchain), then successors in interest can be bound despite not signing on to the original community contractual agreement.

There are more examples than this relatively brief Article can contain. The heart of our approach is to reject technological exceptionalism for blockchain-based contracting, and to reject the idea that code expresses human meaning in the way that natural language does. It demonstrably does not, and indeed cannot. Natural language contracts cannot be reduced

to code—not even a simple, commercial impracticability clause can be so expressed. And smart contracts cannot obtain legal validity other than through the law of contract, which is powered by human expectations expressed in natural language. Once courts understand these key facts, they unlock the ability to draw on the bank of analogous common law doctrines and historical examples of legislation to iterate contracting practice conducted in this new medium.

Conclusion

This Article suggests that the modern trend of assuming that humans intend whatever computers in fact do when they transact via decentralized ledger technology does not fit well with the law of contract. Smart contracts may act, but they do not intend. Nor do their bugs and emergent behaviors suggest intention that should be protected by law. We thus need a law of contracts that understands the distinction between natural and formal languages, between human expression and machine code, between contractual intent and contract execution.

Through the adoption of such a governance framework, the computer mechanism that generates the terms of the contract and forms the formal language of computer code can be better aligned with the natural language of the contract. If a human makes a mistake in contracting by smart contract, the law of mistake should be applied. If the automated system encounters an un-codeable event of force majeure, impossibility, impracticability, or frustration of purpose, then those doctrines ought to apply to contracts automatically executed by computers through decentralized ledgers just as they apply to contracts automatically executed by any other means.

Moreover, we must recognize that the blockchain enthusiast's dream of displacing law through automated execution cannot reduce the role of human understanding and community expectations, processes, and norms. Nor should it. The community—not code—will return NFTs to rightful owners even though they are conveyed to fraudsters via code exploits. The community—not code—will hard-fork blockchains to stop disasters like the DAO. Courts will order rescission of contracts paid for in cryptocurrency, and the return of NFTs obtained through technologically valid transactions that are nevertheless caused by force or fraud. The examples multiply, but the solution is in each case the same. Computer systems cannot reach outside of themselves to square actions with intention. For that function—and it is the only function of contract law, after all—we will need courts

ready to interpret the intent of the parties and correctly manage the interface between the language of law and the language of code.