THE EDUCATION OF A CONTRACT: SMART CONTRACTS DEFINED

Joseph J. Avery, Eric Zhao, Antti Valkonen, Vinícius Pantoja

CONTENTS

INTRODUCTION

"Blockchain" refers to an open, distributed ledger that records transactions between parties in a verifiable and permanent way.[1] "Smart contract" refers to the general purpose computation that takes place on a blockchain.[2] This seems relatively straightforward, and it also seems of little import legally. But then why did the state of Tennessee, in 2018, pass legislation establishing the legal authority of smart contracts in the marketplace?[3] Why do some argue that smart contracts will revolutionize computing, ushering in a "decentralization singularity,"[4] while others are concerned about whether smart contracts are even feasible given the linguistic flexibility inherent to traditional contracts?[5] The reason is that, when one asks what is a smart contract, one receives multiple and distinct definitions. Currently, there is no settled definition for the term.[6] In this note, we discuss how smart contracts have been defined and what this means for their legal treatment.

I. HISTORICAL ROOTS

In 1969, IBM launched its law and computers division, which was precipitated by an attempt at categorizing and standardizing typical contract terms.[7] The contracting parties, actions, and conditions were rendered into computer-readable code, with the intention that, given

---

[1] Marco Iansiti & Karim R. Lakhani, *The Truth about Blockchain*, HARVARD BUSINESS REVIEW 118 (January-February 2017).
[2] Gideon Greenspan, *Smart Contracts: The Good, the Bad and the Lazy*, Multichain (November 2, 2015),
[3] S.B. 1662, 110th Gen. Assemb., Reg. Sess. (Tenn. 2018).
[4] Gian Volpicelli, *Smart Contracts Sound Boring, But They're More Disruptive Than Bitcoin*, Motherboard (February 16, 2015), https://motherboard.vice.com/en_us/article/pgaxjz/smart-contracts-sound-boring-but-theyre-more-disruptive-than-bitcoin
[5] Jeremy M. Sklaroff, *Smart Contracts and the Cost of Inflexibility*, 166 U. PA. L. REV. 263 (2017).
[6] Josh Stark, *Making Sense of Blockchain Smart Contracts*, Coindesk (June 4, 2016), https://www.coindesk.com/making-sense-smart-contracts
[7] *See* Charles F. Goldfarb, *The Roots of SGML-A Personal Recollection,* SGML SOURCE (1996), http://www.sgmlsource.com/history/roots.htm [https://perma.cc/CX29-QEJN].

sufficient standardization, these contracts could be understood and promulgated through an electronic data interchange (EDI).[8] In short, paper would be largely eliminated, and machines would be able to parse quantities and prices and, in the case of non-performance or breach, alter terms according to if-then rules. However, the costs of building out the initial contracts (with exhaustive sets of conditionals) were large, as were the costs of establishing an EDI.[9] Moreover, EDI-based electronic contracts were not able to replace human contracting because they could not support the full range of business ratiocination and failed to fully integrate across different participants' information systems.[10] Thus, the typical result was a contract that was stored electronically, but the functionality of the contract *qua* contract was still beholden to significant human intervention. In short, not much in the world of contracting had changed.

The term "smart contract" was not used to describe these EDI-based electronic contracts. Rather, the term was introduced by Nick Szabo in the 1990s to denote a set of promises, rendered in digital form, that include protocols for party performance.[11] This definition lay dormant until the rise, in the 2010s, of distributed ledgers, which were made especially salient by Bitcoin and its use of blockchain technology.[12] Distributed ledgers opened up new possibilities for autonomous execution, and thus smart contracts came to denote something quite distinct from traditional contracts. In U.S. law, a legally binding contract requires at least five elements: an offer, an acceptance, consideration of some sort, mutuality of obligation, as well as competency and capacity by the contracting parties—not to mention, in some circumstances, a written

---

[8] *See* Lawrence A. Cunningham, *Language, Deals, and Standards: The Future of XML Contracts,* 84 WASH. U. L. REV. 313, 320-21 (2006).
[9] Jeremy M. Sklaroff, *Smart Contracts and the Cost of Inflexibility*, 166 U. PA. L. REV. 263, 290 (2017).
[10] *Id.* at 286.
[11] Nick Szabo, *The Idea of Smart Contracts*, NICK SZABO'S ESSAYS, PAPERS, & CONCISE TUTORIALS (1997), https://perma.cc/YED2-ACVP.
[12] *See* Jenny Cieplak & Simon Leefatt, *"Smart Contracts": A Smart Way to Automate Performance*, 1 GEO. L. TECH. REV. 414 (2017).

instrument.[13] Smart contracts, by contrast, came to mean code that is stored, verified, and executed on a blockchain.[14] It was a purely programmatic sense—and this is the definition that we proffered in the introduction to this article. This may be the definition that state legislatures, such as Tennessee, had in mind when they were drafting recent legislation. Software agents fulfill obligations and exercise rights, and autonomous enforcement (seizure of assets) within the shared ledger is deemed a valid instance of these processes.[15]

Of course, a smart contract might be a specific application of the programmatic sense of the term, such that the elements of traditional contracting are preserved even though the contract itself is rendered into code and software modifies terms and enforces obligations. This goes beyond mere electronic contracting, where there is a paperless contract that is signed electronically, but it might not go far beyond those early EDI manifestations. The parameters of this progress, and the scope of these different senses of the term "smart contract," are discussed in the succeeding sections of this article.

II. A Contract Goes to School

A contract is a written or spoken agreement. Legally, as mentioned in Section I, a contract is a (usually) written agreement that represents a meeting of the minds between competent individuals for the purpose of binding the individuals to some obligation. For such a contract, performance—the act of doing that which is required by a contract—is, of course, carried out by humans. It is carried out by the contracting parties. Similarly, if one party fails to

---

13 *See* Sanford H. Kadish et al., Criminal Law and Its Processes: Cases and Materials (2016).
14 Josh Stark, *Making Sense of Blockchain Smart Contracts*, Coindesk (2016), https://www.coindesk.com/making-sense-smart-contracts.
15 Christopher D. Clack, Vikram A. Bakshi, Lee Braine, *Smart Contract Templates: foundations, design landscape and research directions* (2016), arXiv:1608.00771.

perform, enforcement of the agreement is maintained by legal authorities. Let us take this conception of a contract as our starting point. We might say it is a contract that, in terms of its formal education, has just entered kindergarten.

A slightly smarter contract is the electronic contract, although *slightly* must be emphasized. Here, a written contract is memorialized on word processing software or scanned and saved as an electronic document. The contracting parties' signatures may be attached remotely and electronically. What was written is now computerized.

In the educational development of the contract, what comes next is encoding: more than just electronic, the contract is now rendered into computer code. It is computational. Instead of being memorialized in English or Spanish or another popular language, the contract is memorialized in a set of instructions that form a program that can be executed by a computer. Of course, this shift is not overly significant, as code is simply another language, neither English nor Spanish, but nonetheless just another language.

Now, imagine that this contract, which has been rendered into computer code, has the following additional feature. When the performance of one party is actualized, and payment by another party is owed, a computer sends a signal that triggers a red light to flash. One of the contracting parties sees this red light and, in response, decides to write and mail a check in fulfillment of the contractual terms. We can describe this as a computational contract that requires human performance. Note that, in each of the contracts that we've discussed so far in this section, performance (and enforcement) is a human matter. If, instead of a red light flashing, the trigger sent an email telling the party to write and send a check, this would still be a contract that requires human performance.

What might make such a contract "smart" is that it is the computer that does the interpreting, if such a word is appropriate, of the content of the contract. More specifically, a computer algorithm has received inputs that cause an output that signals partial performance of the contract. In a genuine sense, the computer determines that one party has performed and that performance from the other party is now owing.

At yet higher levels in its education, contracts can be become machine executable. At this stage, the contract is not just rendered into code, and performance is not just registered algorithmically, but full execution is actualized by the computer as well. To return to our previous example, instead of triggering a red light or sending an email, the computer now automatically disburses the funds once the relevant conditions are met. This is a conception of smart contracts that we take up in the succeeding section.

Yet there is still another definition of smart contracts: those that employ machine transformation. Casey and Niblett define smart contracts as agreements in which the contracting parties set only broad *ex ante* objectives, and machine learning is used to generate the specific terms or directives based on information obtained subsequent to the initial contracting.[16] In other words, the contract is smart because it fills in most of the substance of the contract on its own; only the initial broad objectives are specified by the contracting parties. A current use-case is the variable-rate electricity contract where the amount due, which is dependent on fluctuating electricity prices, is resolved *post hoc* through a mechanism agreed upon *ex ante*.[17] This structure can also extend to enforcement as well; the premise of the project Juris Protocol is that parties

[16] Anthony Casey & Anthony Niblett, "Self-Driving Contracts," 43 Journal of Corporation Law 1 (2017).
[17] *Id.*

can agree on a smart contract clause to go through dispute resolution or even crowdsource resolutions through a legally binding tribunal.[18]

Finally, it is worth noting that nowhere in this discussion did we mention distributed ledgers or blockchain technology. We give these technologies consideration in Section IV. Here, though, we must mention that a distributed ledger system may accompany contracting at any of the levels we've just described. A written contract can make use of a distributed ledger just as a computational contract with machine enforcement can make use of a distributed ledger.

III. Smart Contracts as Electronic Self-Help

Let us consider in greater depth contracts that are stored electronically and enforced automatically by a computer program. Raskin has called this arrangement a form of contractual "self-help," where self-help is defined as actions taken by contracting parties to enforce their rights without recourse to courts or tribunals.[19] Of course, Raskin is correct in that a contract being automatically enforced by a computer program obviates the need for judicial enforcement and, for this reason, should be considered a form of self-help. Dam likewise discusses self-help in the context of electronic content.[20] He describes "self-help systems" as enabling the content provider to grant access to certain electronic content contingent upon payment.[21] Similarly, in the copyright setting, Cohen describes self-enforcing contracts as technological self-help.[22] There is no doubt that in everyday life such systems are ubiquitous. Subscriptions to the digital versions of one's favorite newspaper and usage of e-books or movie streaming services are but two of

[18] Kerpelman, Adam J., Ramsay Brown, Akash Desai, Matt Mayberry, Jake Dodd, Vince Enachescu, T. Dalton Combs, "Human-Powered Dispute Resolution for Smart Contracts" *Juris Project Whitepaper* Version 2.0, 2018
[19] Max Raskin, *The Law and Legality of Smart Contracts*, 1 Geo. L. Tech. Rev. 305 (2017)
[20] Dam, Kenneth W. *Self-help in the digital jungle,* 28 THE JOURNAL OF LEGAL STUDIES 2 (1999).
[21] Id. at 397.
[22] Julie E. Cohen, *Copyright and the Jurisprudence of Self-Help*, 13 Berkeley Tech. L.J. 1089 (1998).

myriad examples. Like a lock on the door of a warehouse, systems requiring payment before permitted one to log into a streaming service provide means for a party to enforce its rights without resorting to the legal system.[23] There is nothing new with these kinds of systems and most of us are familiar and comfortable using them.

However, looking at these systems as prototypes of smart contracts is a more novel and interesting approach. Digital content self-help systems point the way towards defining smart contracts as extensions of traditional contracts in which enforcement is undertaken by a computer program. If smart contracts are defined as traditional contracts with automatic enforcement by a computer program, there is little doubt that the agreement between the vendor and the user in a service such as an e-book, e-newspaper or streaming service would be a smart contract. What is interesting in these applications is that the agreement is made for content that is completely electronic. Thus, automatic and electronic enforcement is easy to implement. If the user has not paid for a streaming service, they cannot sign in to the service. On the other hand, if the user pays, they will be able to sign in and access the content. As this simple example shows, enforcement is straightforward in the case of contracts relating to electronic content.
The use of smart contracts has been an everyday reality in digital content for a long time.

A more recent and still ongoing development is the use of self-enforcing contracts in settings outside the purely digital realm, such as those involving physical and financial assets. Szabo defines the basic principle of smart contracts as the ability to embed physical objects with contractual clauses.[24] He clarifies this "embedding" by giving a vending machine as an example. Similarly, he discusses how an automobile containing an electronic system that allows the

---

[23] Max Raskin, *The Law and Legality of Smart Contracts*, 1 GEO. L. TECH. REV. 305 (2017)
[24] Szabo, Nick. *Formalizing and securing relationships on public networks.* 2 FIRST MONDAY 9 (1997)

creditor to gain control of the car in case of default is another example. Indeed, starter interrupters are already in use by some auto creditors.[25]

It is easy to draw parallels between the electronic content protection systems and the examples of vending machines and starter interrupters. In both cases, there exists an automated self-help system that enforces a contract. In essence, the only difference between the examples is that, in case of electronic content, self-help is aimed at protecting a party's interests in intellectual property; in the case of vending machines and automobile starter interrupters, the aim is to protect interests in material property.

We have the contours of one of the definitions discussed in Section II: a smart contract is a contract embodied in code and enforced by machine. Said again, a smart contract is an auto-enforcing encoded contract.

IV. Smart Contracts and Distributed Ledgers

The Tennessee state legislature defined a smart contract as "an event-driven computer program, that executes on an electronic, **distributed, decentralized, shared, and replicated ledger**… [emphasis added]."[26] This is a precise definition—and an exclusive one. According to the Tennessee legislature, distributed ledgers are a necessary part of the smart contract definition. As discussed above, the coupling of smart contracts and distributed ledgers has been common since the rise of Bitcoin and blockchain, and it is this overemphasis on distributed ledgers that has led to much of the definitional confusion.

A blockchain is an open, distributed ledger that records transactions between parties in a verifiable and permanent way. But a blockchain is not a distributed ledger; there are subtle

---

[25] Max Raskin, *The Law and Legality of Smart Contracts*, 1 Geo. L. Tech. Rev. 305 (2017)
[26] S.B. 1662 §47-10-201(2), 110th Gen. Assemb., Reg. Sess. (Tenn. 2018).

differences between these technologies. A distributed ledger is a collection of digital records or transactions that is accessible to all computers running the same protocol.[27] This means that every computer in the network has access to the same information. Blockchain technology, which is a type of distributed ledger, does exactly this, with the additional feature of organizing the information into blocks that are connected to each other as a chain.

For our purposes, if the definition of smart contracts includes "Blockchain" instead of "distributed ledger," it would exclude all of those (electronic, machine enforceable) contracts that are not organized in chained blocks. The definition, in this case, would approach the specificity of the definition put forth by the Tennessee legislature. This is unfortunate, since there are good reasons to believe that distributed ledger technology is essential to smart contracting. This technology has properties that are beneficial, such as transparency and verifiability,[28] immutability,[29] and also the ability to maintain the anonymity of the parties involved.[30]

The term "distributed" plays a major role in generating these properties. It means that the digital data is available in multiple nodes (computers).[31] In this case the digital data is the smart contract and the nodes are the parties involved. Everyone has one version of the smart contract, and all the versions are synchronized, therefore eliminating the need for a trusted third party. For many instances of contracting, this is revolutionary. Usually, coordinating and assessing large numbers of people is hard, and thus a system like this can be beneficial because everyone

[27] McLean, Sue, and Simon Deane-Johns. "Demystifying Blockchain and distributed ledger technology–hype or hero." *Computer Law Review International* 17.4 (2016): 97-102.
[28] CARDEIRA, H. Smart contracts and possible applications to the construction industry. In: New Perspectives in Construction Law Conference on New Perspectives in Construction Law Conference, Bucharest. 2015.
[29] MICHAEL, JW, ALAN COHN, and JARED R. BUTCHER. "BlockChain technology." *The Journal* (2018).
[30] SKLAROFF, Jeremy M. Smart Contracts and the Cost of Inflexibility. U. Pa. L. Rev., v. 166, p. 263, 2017.
[31] *Distributed Ledger Technology: beyond block chain (PDF) (Report). UK Government, Office for Science. January 2016. Retrieved 29 August 2016.*
(https://assets.publishing.service.gov.uk/government/uploads/system/uploads/attachment_data/file/492972/gs-16-1-distributed-ledger-technology.pdf)

reaches the same verdict through a consensus system. But this might not be true for smaller numbers of people, or for contracts that must be viewed by, say, only a limited number of parties.

V. Legal Implications

How a smart contract is defined is important because there are specific legal implications that apply to these distinct ways of contracting. First, where a smart contract includes machine enforcement, a chief question is how the law will handle machine errors. For traditional contracts, there is a doctrine for scriveners' errors, such that these errors may be corrected by parol evidence if the evidence is clear, convincing, and precise.[32] It would be quite simple to extend such a rule to programmers' errors, which we can think of as a 21st Century progeny of scriveners' errors.

A more complex issue arises when we consider non-traditional enforcement, such that smart contract code can lack the semantic richness to record all intentions of the parties involved and can be susceptible to exploits, bugs, and unforeseen circumstances, which was highlighted by the DAO attack in June 2016.[33] The DAO is a decentralized autonomous organization built on the Ethereum blockchain platform and underpinned by smart contracts. A "recursive calling vulnerability" allowed an attacker to recursively withdraw funds from the DAO until it was drained of $70 million in Ether cryptocurrency.[34, 35] This attack demonstrated the risks of weak code and, while the DAO smart contract was never technically breached, the attacker's actions

[32] *See* Ian Ayres and Gregory Klass, Studies in Contract Law (2017).
[33] Falkon, Samuel. "The Story of the DAO - Its History and Consequences." Medium.com. December 24, 2017. https://medium.com/swlh/the-story-of-the-dao-its-history-and-consequences-71e6a8a551ee.
[34] *Id.*
[35] Ethereum Foundation. "CRITICAL UPDATE Re: DAO Vulnerability." Ethereum Blog. https://blog.ethereum.org/2016/06/17/critical-update-re-dao-vulnerability/.

were clearly a breach of the agreement that the smart contract was intended to execute. Non-traditional enforcement failed, but the attacker never faced any legal consequences either, so the attack underscored that, in terms of traditional enforcement, the legal enforceability of smart contract code remains a difficult, gray area despite.

Another important question is how such disputes should be resolved. One solution, which has been promoted by Vitalik Buterin, the founder of Ethereum, is that blockchains and distributed ledgers should have their own adjudicative bodies.[36] OpenBazaar, an online peer-to-peer marketplace, for example, relies on a network of moderators to adjudicate disputes arising on the site.[37] But this strikes us as less of a solution and more of a mere relocating of the problem. Moderators and specially-employed adjudicators will struggle to resolve these cases, and to develop meaningful and consistent criteria for resolution, to the same extent that regular courts would. What is clear is that, given the lack of semantic richness in programmatic language, adjudicators—whomever they are—will be forced to rebuild entire agreements in order to understand the initial understandings, and this surely is an unworkable review process.[38]

Another significant issue, and it is one that suggests a tension inherent the Tennessee State Legislature's definition of smart contracts, is consideration. If we use our definition from Section III—a smart contract is a contract embodied in code and enforced by machine—then it becomes clear that such a contract could be entered into without consideration. That is, the parties could encode a gift promise to the blockchain. If one party wanted to void the

[36] Vitalik Buterin (vbuterin), Decentralized Court, REDDIT (Apr. 16, 2016), https://www.reddit.com/r/ethereum/comments/4gigyd/decentralized_court/ [https://perma.cc/X9QC-3DRV]
[37] See Sam Patterson, How Moderators and Dispute Resolution Work in OpenBazaar, OPENBAZAAR BLOG (Feb. 24, 2016), https://blog.openbazaar.org/how-moderators-and-dispute-resolution-work-inopenbazaar/#.WPn6k1Pyuis [https://perma.cc/H5J3-WNJ8]
[38] Jeremy M. Sklaroff, *Smart Contracts and the Cost of Inflexibility*, 166 U. PA. L. REV. 263 (2017).

arrangement, the Tennessee definition would seemingly prohibit such voidance, even though, according to traditional contract doctrine, no contract would ever have existed.[39]

When it comes machine transformations, so-called "self-driving" contracts, there are no legal rulings, to our knowledge, that establish the validity of such contracts. However, a strong case can be made that such contracts do fall within the definition provided by the Tennessee State Legislature, so long as they are executed on a distributed ledger. After all, a contract that uses machine learning to produce specific terms that are guided by broad *ex ante* objectives certainly qualifies as "an event-driven computer program." Such contracts raise a few specific legal concerns, although each one cuts in favor of enforcing such contracts. It turns out that self-driving contracts may be legally robust contracts.

The first is lack of mutual assent. Also called the "meeting of the mind" requirement, courts will not enforce contracts in which all parties did not know what they were agreeing to. When it comes to self-driving smart contracts, the argument can be made that the parties *never* know what they are agreeing to. They know the broad objectives, but the specifics are left undecided at the time of contracting (and are filled in, over time, by means of machine learning). But this may not be as significant hurdle as it first appears. In *Hill v. Gateway 2000*,[40] for example, the Seventh Circuit held that an arbitration clause could be enforced even though the customer purchased the product over the phone and never even read the clause. Judge Easterbrook wrote, "[A] contract need not be read to be effective."[41] If reading of the contract is not required, then assent is a nebulous requirement indeed. It seems likely that, so long as the

---

[39] Kevin Werbach and Nicolas Cornell, *Contracts* Ex Machina, 67 DUKE L. REV. 313, 370 (2017).
[40] *Hill v. Gateway 2000, Inc.*, 105 F.3d 1147 (7th Cir. 1997).
[41] *Id.* at 1148.

parties clearly indicate their assent to the algorithms at issue, the meeting of the minds requirement should not fell such smart contracts.

A second issue is what to make of indefiniteness when it comes to machine transformations. Courts routinely refuse to enforce contracts when the essential elements are missing. In the classic case of *Varney v. Ditmars*,[42] "a fair and reasonable percentage of the net profits" was deemed to indefinite to withstand a legal challenge.[43] As Casey and Niblett point out, though, self-driving contracts actually decrease the possibility of indefiniteness challenges.[44] The machine learning processes would have calculated a fair and reasonable percentage of the net profits pursuant to the *ex ante* objectives of the parties. In a sense, a contract that allows for extensive machine transformations is unlikely to ever fail a test of indefiniteness. This same argument also applies to claims of mutual mistake. Here, if the factual situation suggests that the parties were mistaken at the time of the contract (to use a classic example, the parties believe that a cow is barren when it actually is not), the self-driving contract should adjust to remain in accordance with the parties' initial broad objectives.[45]

CONCLUSION

In this article, we have attempted to cast greater light on smart contracts, outlining how they are commonly defined, how they might be defined, and how these definitions reflect the confusion evinced in common usage. These definitional concerns are not of mere academic importance. In the Tennessee example we discussed, for instance, one can see how even a state legislature might over-define the term and thereby eliminate from protection contractual

[42] *Varney v. Ditmars*, 111 N.E. 822 (N.Y. 1916).
[43] *Id.* at 823.
[44] Anthony Casey & Anthony Niblett, "Self-Driving Contracts," 43 Journal of Corporation Law 1, 19 (2017).
[45] Anthony Casey & Anthony Niblett, "Self-Driving Contracts," 43 Journal of Corporation Law 1, 18 (2017).

instruments that it did not intend to eliminate. We do not pretend that smart contracts are simple or easy to understand. But we do aver that smart contracts warrant clearer and more consistent treatment, especially as they and their variants begin to hold increasing importance in the commercial, financial, and legal realms.