REPORT

Futures & Derivatives Law

The Journal on the Law of Investment & Risk Management Products

June 2019 ▪ Volume 39 ▪ Issue 6

SEPARATING THE SMART CODE FROM THE SMART CONTRACT: LEGAL AND REGULATORY IMPLICATIONS OF SMART CONTRACTS REQUIRE THEIR CLEAR DISTINCTION FROM SMART CODE

By Kenny S. Terrero and Eric Ubias

Kenny S. Terrero is a counsel in the New York Office of Sidley Austin LLP. Eric Ubias is managing partner of Ubias Law PLLC.

This article is for informational purposes only and does not constitute legal advice. This information is not intended to create, and the receipt of it does not constitute a lawyer-client relationship. Readers should not act upon this information without seeking advice from professional advisers. The content of this article does not reflect the views of Sidley Austin LLP.

I. INTRODUCTION: THE RISE OF THE SMART CONTRACT

Much has been written about the advent of "smart contracts" and the potential disruption they herald. That they are a panacea for digital transactions. That their benefits include cost savings, speed and efficiency, security, fraud protection, and improved verification and accuracy for digital transactions. That they can provide digital transactions a connection to real-world activities and the flexibility to address complex business arrangements and innovative use cases. That they have the potential to impact industries and individual activity, unlike any other recent development.[1]

But what are they? The term "smart contract" was first coined by Nick Szabo 25 years ago when he defined smart contracts as a computerized transaction protocol that executed the terms of a contract.[2] He later described them as "a set of promises, specified in digital form, including protocols within which the parties perform on the other promises."[3] Smart contracts are now closely linked with digital transactions on blockchain networks. Still, there is no universally accepted definition of what one is, and the smart code, which is only one element of a smart contract, is often confused with the smart contract itself. Different articles on smart contracts will yield differing descriptions. Varying definitions of the term in the blockchain context are being codified in state law[4] and beginning to appear in case law.[5]

Regulators have taken notice as well. The Commodity Futures Trading Commission (the "CFTC"), a key regulator and first mover in the space, issued an educational primer on smart contracts defining

them as a set of coded computer functions that could incorporate elements of a binding contract or simply execute certain terms of a contract.[6] The race to define smart contracts is emblematic of their novelty and possibilities.

Smart contracts have the potential to revolutionize the fundamental human act of exchanging value for value. Their innovation is not simply in serving as an agent of a seller or buyer that more efficiently executes the terms of a contract, but in facilitating the entire process of a bargain to exchange value.[7] This bargain is the quintessence of a contract. Therefore, smart contracts in and of themselves need to function as legally enforceable contracts under applicable law.

In this article, we distinguish smart contracts from a self-executing *smart code* that is executed as an application on a blockchain network. We also discuss the challenges and limitations, both functional and regulatory, which must be overcome before smart contracts deliver on their promise of disruption.

II. SMART CONTRACTS

BLOCKCHAIN DEFINED: THE DECENTRALIZED COMPUTING PLATFORM THAT RUNS SMART CONTRACTS

Smart contracts can be self-executing and operate on both private, permissioned blockchain networks and publicly accessible platforms that secure performance using shared consensus-establishing mechanisms.[8] This article will focus on public, non-permissioned blockchain networks, such as the Ethereum network,[9] as private, permissioned implementations of blockchains are often intracompany enterprise solutions that do not necessarily implicate the same contractual issues.[10]

At its core, a blockchain is a distributed database that maintains identical copies of discrete structured data or transactions referred to as blocks. The term blockchain is used because this distributed database contains the complete sequence of transactions from the very first transaction (known as the genesis block) to the most recent one. Each transaction is linked to the preceding transaction that collectively form a linked chain of structured data. Transactions are verified and maintained by a network of participating computers.[11] Each computer in the network (called a "node") constructs a record and holds that record for every transaction. Transactions are validated through a distributed consensus mechanism.[12] Cryptographic tools are used for verifying and recording transactions on the blockchain ledger, such as public-private key cryptography, whereby transactions are encrypted by public keys and can only be deciphered by someone who possesses the corresponding private key and vice versa.[13]

Blockchains are said to be "trustless" because they replace a trusted intermediary with a globally shared immutable record of transactions that have been verified and managed by all of the nodes on the network. The touted benefits of using blockchain technology include increased transparency, resiliency, efficiency, cost-savings, immutability, and security.

The first blockchain implementation to gain widespread adoption was Bitcoin. The Bitcoin network is a decentralized electronic payment system with a native digital asset called Bitcoin. While the Bitcoin network uses smart code, the

Bitcoin coding language is limited[14] restricting the type of operations that occur on the platform. A standard transaction on the Bitcoin network is simply a transfer of ownership of a certain amount of Bitcoin. This transaction is a one-way transfer of value. The Bitcoin coding language that executes the transfer does not reflect a return of the value to the sender that is essential to contract formation.[15]

In 2015, the Ethereum network provided a new decentralized platform capable of processing smart code through its Ethereum Virtual Machine.[16] One of the most prominent examples of smart code is its use in the issuance of tokens that adhere to the Ethereum Request for Comment ("ERC") 20 or ERC-20 standard. Ethereum developers can write and deploy smart code that issue ERC-20 compatible tokens, provide for management of transactions and support other functions.[17] The smart code is self-executing and provides for an exchange of Ether, Ethereum's native digital asset, for the newly minted token. Because the smart code runs on the Ethereum blockchain network, the transactions also enjoy the benefits of blockchain technology. The dynamic features of the Ethereum network and the ERC-20 standard made the possibility of smart contracts a reality.

SMART CONTRACTS VS. SMART CODE

We define ***smart contract*** as a combination of (1) smart code capable of self-execution and automatic performance of certain terms of an agreement, (2) other terms that are not automatically performed and are expressed in natural language, and (3) other contractual elements that collectively meet the requirements of an enforceable contract under applicable law. Smart contracts should be distinguished from ***smart code.*** Simply, smart code is computer code or programming that can automatically execute commands when a variable is satisfied.[18] It is the often spoken of "if, then" logic sequence that is capable of coding commands (e.g., if a specified condition is true, then return one output and, if false, return another).[19]

In essence, smart code can facilitate performance without the need for intermediaries through the use of decentralized and tamper-proof blockchains.[20] Smart code may be self-executing and may trigger commands to take an action to satisfy a provision of an agreement or to take an action at a future time, such as one based on the occurrence or non-occurrence of an action or event (e.g., a change in reference rate).[21] Smart code can reside and operate externally or internally within the four corners of a smart contract.

Notably, smart code can be programmed to use information from the blockchain on which the smart code operates, as well as from off-chain sources through third-party data bridges, known as oracles. Oracles offer the potential for dynamic real-time interactions between the smart code and the real world.[22] Oracles can relay information to the blockchain like temperature, current interest rate, currency exchange rate information and more. Using oracles, the smart code can adjust performance for changing conditions in near real time.[23] Once the conditions have been met, several nodes automatically execute the smart code and the smart code's execution cannot be stopped unless the smart code provided for it.

Through smart code, the terms of an agreement and the state of facts relating to the performance

of the agreement can be carried out on a decentralized blockchain network.[24] The nature of the transactions facilitated by smart code resemble bilateral agreements and thus were termed "smart contracts."[25] But smart code alone is not a contract. Smart code that simply automates a particular process but does not code, or operate in conjunction with, all of the elements necessary to form a binding contract may not satisfy the requirements of a legally enforceable contract.

III. SMART CONTRACT FORMATION

Contract formation is often overlooked in the literature on smart contracts. An agreement with or without smart code, absent the elements necessary to form a legally enforceable contract, is not a contract in the legal sense of the word. By now, these concepts are relatively straightforward. However, the complexity of the smart code, automatic execution capabilities and use of oracles to collect data from an off-chain source or make it available in a smart contract's storage,[26] raise issues of first impression for the analysis of a legal obligation triggered by a smart contract (e.g., the representations, warranties and promises of the contract).

The elements required to form a legally binding contract should be no different for a smart contract or a classical contract—offer, acceptance, and consideration. Smart contracts have not had their day in court. As such, the potential legal issues raised by smart contracts must, by necessity, be analogized to the characteristics of more traditional legally enforceable contracts. While many have asserted that existing contract law can accommodate smart contracts, it has not yet been tested in a court of law.[27] We must get closer to the technology to effectively articulate the regulations governing the creation of smart contracts.

ELECTRONIC CONTRACTS

The application of contract law principles to new technology is not a new development. The Uniform Commercial Code (the "UCC") was first published in 1952 and intended to modernize commercial transactions.[28] The UCC imposes a "writing" requirement for certain contracts for the sale of goods to be enforceable.[29] The UCC's requirement of a "writing" can be satisfied by any "printing, typewriting or any other intentional reduction to tangible form," which has been interpreted to cover electronic records.[30] Courts have also imposed a "signature" requirement that is not clearly specified by the UCC. These courts have looked at whether there is evidence that the parties to a contract "adopted and accepted" the writing.[31] Courts have found the UCC's "sale of goods" to cover software under certain circumstances[32] and have applied the UCC to address mixed contracts covering both the sale of goods and services.[33]

Electronic contract formation is subject to the same basic principles as those governed by common law or the UCC.[34] For instance, an internet user can enter into certain agreements online by simply clicking "I agree" to accept the terms and conditions on an electronic form. For these "*click-through*" *agreements* that online users often see, notice of the terms and consent are essential.[35] Forming enforceable electronic contracts using electronic signatures is also not new and has long been recognized by courts.[36] However, uniform acceptance of electronic contracts and the type of evidentiary proof needed to es-

tablish enforceability were lacking, which created barriers for the growth of e-commerce. The Uniform Law Commission developed the model Uniform Electronic Transactions Act ("UETA"), in large part, to harmonize state laws by validating and establishing rules for electronic records and signatures.[37] In relevant part, UETA defined "electronic signature" to mean an electronic sound, symbol, or process attached to or logically associated with a record and executed or adopted by a person with the intent to sign the record.[38]

The U.S. Electronic Signatures in Global and National Commerce Act ("E-Sign") was enacted in 2000, establishing a standard for legal recognition and enforceability of electronic signatures. Congress acted due to uncertainty of the pace of state adoption of and potentially divergent approaches to electronic records and signatures.[39] E-Sign set a baseline that allowed states to pass the UETA or a functional equivalent so long as it did not conflict with E-Sign.[40] E-Sign provides that signatures, contracts, and records related to transactions, not be denied legal effect solely because they are in electronic form. Additionally, under E-Sign, contracts may not be denied legal effect, validity or enforceability solely because an e-signature or electronic record is used.

UETA and E-Sign both recognize the creation of electronic contracts and set the foundation for the creation of smart contracts. They also presage potential limitations of smart contracts as UETA and E-Sign do not cover certain types of agreements such as wills, codicils, testamentary trusts, and certain UCC transactions.[41] These statutes underscore a pillar of contract formation in the United States, that no matter the format, parties must mutually agree to the contract terms.

Several distributed ledger technology and blockchain technology platforms incorporate the requirements established by existing rules and regulations like UETA and E-Sign. The Corda Private permissioned blockchain, for example, has smart contracts (called "CorDAPPs") that include smart code and a function that Corda calls "legal prose."[42]

There are similar efforts underway for public blockchains. Ethereum specific browsers like Mist or plug-ins for traditional internet browsers, such as Metamask, allow users access to the Ethereum applications (called decentralized applications or "DApps") with a user-friendly interface. While data on the Ethereum blockchain is publicly visible, it is unintelligible to most lay users. DApps allow developers of blockchain based smart contracts to present customers with a requirement to accept the terms and conditions for use of the DApp in a readable and user-friendly way. As with traditional click-through agreements, such terms and conditions agreements in DApps should also be enforceable.[43]

The contours of the regulatory regime for developing smart contracts are already in place. Developers must take into account the objective and purpose of the smart contract, which will dictate how to analyze it from a risk management perspective and what legal and regulatory rules may apply (e.g., sale of goods, currency swap, interest in real estate or security interest in digital assets).

IV. SMART CONTRACT ENFORCEABILITY

All contracts are subject to performance issues and smart contracts are no different. Performance is the fulfillment or accomplishment of a promise, contract, or other obligation according to its

terms.[44] The typical issues with performance cannot be eliminated with automation because they arise when there is a question as to whether the performance meets or falls short of the contracting parties' expectations.

A smart contract is enforceable either by automated execution of smart code or via legal enforcement of rights and obligations.[45] Performance in the context of a smart contract is mediated by technology.[46] For example, blockchain networks that have total control over the platform have been able to incorporate typical contractual mechanism that greatly mitigate any enforceability concerns. To date, there is no reported case of a court adjudicating issues of first impression around enforceability of smart contracts. As such, any analysis of the legal enforceability of a smart contract must necessarily rely on untested hypotheticals and analogies.

Consider a smart code that governs the performance of a loan agreement. In this example, the smart code would automate the release of an amount of Ether to satisfy a loan repayment schedule. One can write code that establishes if and when a payment is due and the steps required for performance (e.g., charge the payment amount), but it cannot guarantee performance. In order for the contract to be satisfied, the payment must be readily available at the time that the Ether is due. Yet, what guarantees that the digital wallet will have the Ether at the time the smart code is executed and attempting to collect the Ether?

The more challenging aspects of smart contract performance deal with provisions that require value judgments. For instance, whether someone has used commercially reasonable efforts is a concept that we are currently unable to effectively code into a smart contract.[47] The proposition that the *code is the contract* means that smart contracts can be written entirely in code (including every provision of the contract, its representation, warranties, promises, indemnity, etc.). This proposition is one which smart contracts likely will not be able to fulfill either through native blockchain information or, off-blockchain information (e.g., via an oracle) until smart contracts can leverage technological developments, such as those in the area of artificial intelligence, to assess the satisfactory compliance of subjective provisions like those requiring a "reasonable" effort.

We caution that the use of and reliance upon smart code to automate performance can lead to unwanted results. Coding is not error proof. Converting a written or spoken agreement into executable smart code can have unintended consequences. For contracts with a long shelf life, there is a greater chance that a divergence between the agreed upon terms and the performance of the smart code will develop over time. What happens when the performance triggered by the code does not meet the contracting parties' expectations? Will one of the parties be held liable for breach of contract or will the programmer bear some responsibility?

RECOURSE AND CHOICE OF LAW

Typically, the parties to a contract are the ones that enforce it. More often than not parties to a contract adhere to the agreed upon terms. For simple agreements, legal requirements around contract formation and enforcement may be viewed as burdensome. The rubber meets the road, however, when there is a dispute in a contractual arrangement and the aggrieved party resorts to the courts to seek redress.

In the United States, contract law is generally the province of state law. Smart contracts present jurisdictional issues that are not typically present in other forms of contractual arrangements. Choice of law provisions, typically used in standard contracts, may not provide an answer in the context of smart contracts. Smart contracts on public blockchains may have connections to a large number of jurisdictions. As in the United States, countries around the world are grappling with how to deal with blockchain technology and issues like the ones raised here. As such, there are questions about whether and how a smart contract governing a cross-border transaction upsets established principles.[48] In the case of choice of law, some possible factors for making this determination are the location of the parties, the location of the servers and the location of the nodes that verify the transaction.[49]

V. LIABILITY RISKS CREATED BY SMART CONTRACTS

Vulnerabilities in smart contract code have resulted in significant monetary losses as a result of hacking and coding errors.[50] These hacks are commonly performed by exploiting an error in critical functions of smart contracts, which can lead to indefinitely locked funds, leaked funds, backdoor transfer/minting functions, and the ability to arbitrarily kill a contract.[51]

These issues and others discussed below have raised the question of whether liability in the context of a smart contract should include parties that were previously omitted from liability for breach of contract. There is a well-worn path for determining which parties have potential liability for the breach of a classical contract. The complexities of smart contracts and the new contributing parties (i.e., the programmers and the developers of platforms where the smart contracts reside and operate) raise the question of which parties have potential liability not only for breach of contract but also for any violations of law perpetrated through the use of a smart contract.

CODING ERRORS

The hack of The Decentralized Autonomous Organization ("DAO") in 2016 served as an early high-profile example of smart contract coding gone awry.[52] After raising hundreds of millions of dollars in its initial coin offering ("ICO"), but before the DAO project commenced activity, computer programmers identified a bug in the DAO's smart contract code.[53] The DAO's smart contract was coded to be self-executing and entirely reliant on the code that resided on the Ethereum blockchain. Even though the potential threat was known, the smart contract had already been deployed and an attacker was able to exploit the vulnerability by draining approximately one-third of the Ether raised in the ICO before the bug could be addressed. It could be said that the smart contract performed differently than intended, yet the smart code, as written, performed exactly the way it was coded. The essential roles played by the parties involved in the blockchain networks, the core developers, and the programmers of the smart code, raise potentially novel liability issues that are not present in classical contracts and raise more traditional ones in novel ways.[54]

RISKS FOR CORE DEVELOPERS

"Core developers" are a group of developers who release periodic updates and new versions of blockchain software. They enjoy certain control over the operations of the network that are

not enjoyed by other nodes on the network. The actions or inactions of the core developers can have significant implications for the network and its users. Take, for example, the audit of a recently proposed upgrade to the Ethereum blockchain, Constantinople, that revealed a dangerous unintended consequence of one of Constantinople's changes: certain smart contracts would now become vulnerable to a type of malicious attack called "reentrancy." Attackers could have exploited the bugs to rig a smart contract to, for example, redirect currency to the hacker's wallet.[55] Shortly after the audit company brought this issue to the attention of the Ethereum core developers, they decided to postpone the Constantinople upgrade.[56] This underscores that the decisions made by the core developers of a given blockchain network, such as upgrades, can have a significant impact on blockchain network participants both from a financial and functional perspective. The potential liability for core developers is an issue that has become the subject of legal debate among blockchain scholars.[57]

RISKS FOR SMART CODE DEVELOPERS

Smart code developers also face a number of unanswered questions around the potential for civil and criminal liability created by their involvement in programming the smart codes. Some have questioned whether creating a smart contract and advising on its use and operations can be deemed to be practicing law without license.[58] Additionally, some regulators have signaled the potential for personal liability for computer programmers and smart code developers. This later approach deviates from the historical treatment of developers as insulated third parties that do not create violations because the end user typically controls the program's operations. This is not the case for smart contract code that can carry out every step of a transaction once initiated. As such, developers have entered into a new realm with several potential legal and regulatory pitfalls.

VI. REGULATORY OVERSIGHT OVER SMART CONTRACTS AND THEIR PARTICIPANTS

Smart contracts are, by definition, subject to state and federal rules regarding contractual arrangements. However, depending on their design, application, and use, smart contracts may also be subject to other regulatory and legal regimes. These regimes include obvious ones like the previously discussed UETA and E-SIGN, and some that are less obvious like the U.S. securities laws or commodity laws. When the operations of smart contract transactions encroach on their jurisdiction, U.S. regulators have signaled their intent to regulate smart contracts, computer programmers and participants in these transactions.

The U.S. Securities and Exchange Commission's (the "SEC") enforcement action against Zachary Coburn, more commonly known as the EtherDelta case, is an example of the potential regulatory consequences of smart contracts.[59] EtherDelta is a decentralized trading platform that provides a secondary market for trading of ERC-20 tokens, such as Ether and other Ethereum-based tokens. EtherDelta has a smart contract deployed on Ethereum that allows users to execute trades pseudonymously through the smart contract with only an Ethereum wallet and token (ERC-20 or Ether).[60] EtherDelta's smart code defined and executed its business

operations.[61] The smart code was programmed to validate order messages, confirm the terms and conditions of orders, execute paired orders, and direct the distributed ledger to be updated to reflect a trade. Partly due to the functionality of EtherDelta's smart code, the SEC concluded that it had the underlying functionality of an online national securities exchange.[62]

Section 5 of the Securities Exchange Act of 1934, as amended (the "Exchange Act"), makes it unlawful to operate an unregistered securities exchange.[63] Section 3(a)(1) of the Exchange Act defines an "exchange" to include any organization that provides a marketplace or facilities for bringing together purchasers and sellers of securities.[64] Rule 3b-16(a) uses a "functional test" that looks at whether the system "brings together the orders for securities of multiple buyers and sellers" and "uses established, nondiscretionary methods" whereby "such orders interact with each other, and the buyers and sellers entering such orders agree to the terms of a trade."[65] Because, in relevant part, the EtherDelta smart code trading protocols had the sophistication to perform these functions, the SEC held that EtherDelta's functionality satisfied the definition of an "exchange" within the meaning of the securities laws.[66] Accordingly, the SEC concluded that Zachary Coburn should have registered EtherDelta as an exchange or operated pursuant to an exemption.[67]

Identifying Zachery Coburn was relatively straightforward, as he was a visible figure for the platform and he personally developed the smart code that powered the decentralized exchange, which he also operated.[68] However, this case raises interesting questions of what happens when a third party is responsible for writing the smart code for the smart contracts. The SEC has not made it clear that it will make a distinction between the developer of smart code and the platform operator.[69] The SEC indicated that "[a]n entity that provides an algorithm, run on a computer program or on a smart contract using blockchain technology, as a means to bring together or execute orders, could be providing a trading facility."[70] As such, programmers of such smart code under which these platforms operate, could conceivably become liable for securities violations.

The CFTC's unofficial view of potential enforcement action similarly puts smart code programmers in the crosshairs of potential civil and criminal liability. At a conference in Dubai, CFTC commissioner Brian Quintez expressed the view that if a contract is a product within the CFTC's jurisdiction (e.g., it has the characteristics of a swap, futures or option), then it is subject to CFTC regulation.[71] Commissioner Quintez suggested that smart code programmers who could reasonably foresee, at the time that they created the smart code, that it would likely be used by U.S. persons in a manner that could violate CFTC regulation, could potentially be charged as aiding and abetting violations of CFTC regulations.[72] Commissioner Quintez also stated that "[i]t is certainly possible that the software code does not represent the entirety of the participants' agreement and must be interpreted in connection with traditional contract law concepts like good faith and fair dealing."

While this may seem incredible, smart code programmers need look no further than the CFTC's recent case against software developer Jitesh Thakkar and his firm, Edge Financial Technologies Inc.[73] In that case, the CFTC

charged Thakkar, a software developer, with aiding and abetting a trader's spoofing scheme (bidding or offering with the intent to cancel before execution) by developing customized trading software for a trader.[74] The CFTC alleged that Thakkar knew, or should have known, that the program was being used to violate CFTC rules and regulations. Sound familiar? The CFTC has signaled it will assert jurisdiction over smart contracts and participants in transactions dealing with smart contracts that encroach on the CFTC's jurisdiction.

The CFTC is responsible for promoting the market integrity of derivatives markets. CFTC-registered entities like Designated Contract Markets (DCMs) and Swap Execution Facilities (SEFs) may implement smart contracts using smart code programmed into options, futures, and swaps. Depending on its structure, operation, and relevant facts and circumstances, a smart contract could be considered a commodity, forward contract, futures contract, swap agreement or an option on futures contracts.[75] All of these are subject to regulation by the CFTC.

Any potential enforcement action by the SEC or the CFTC would present unique challenges. Yet, these developments reflect a clear effort by these agencies to scrutinize all aspects of the cryptocurrency and blockchain ecosystem. U.S. regulators will continue to police digital transactions involving smart contracts. As such, network core developers and programmers should be mindful of these risks.

VII. CONCLUSION

Technology has disrupted everyday activities—how we hail a cab, purchase products and even how we manage the temperature in our homes. As technologists turn their attention to other long established ways of doing things, the way that we have historically exchanged value for value is in the crosshairs. It is not a question of if, but when contracts and contract formation, as we know it, will be truly disrupted.

Smart contracts will force a reevaluation of how the existing legal regimes apply to contractual relationships. Although existing sources of law can govern even the most complex transactions, smart contracts present novel issues that have been raised in the abstract but have not been tested. Companies, individuals and the gatekeepers they enlist have long since learned that the SEC and CFTC have targeted, and may continue to target, digital transactions and activities that use or occur on blockchain networks they view as unlawful. As the use of smart contracts proliferates, parties seeking to develop, deploy and/or use them (including programmers and coders) should be mindful of the potential risk and legal landmines that abound.

ENDNOTES:

[1]Smart contract programming languages, such as Solidity or Vyper, are said to be Turing complete. That is, given enough resources, the programming language is able to compute anything computable. *See e.g.*, Kyle Wang, *Ethereum: Turing Completeness and Rich Statefulness Explained*, Medium Corporation (July 9, 2017), https://hackernoon.com/ethereum-turingcompleteness-and-rich-statefulness-explained-e650db7fc1fb. These coding languages should be distinguished from the Ethereum blockchain itself, which is not. *See* Dr. Gavin Wood, *Ethereum: A Secure Decentralized Generalized* (2018), https://ethereum.github.io/yellowpaper/paper.pdf (noting that the Ethereum Virtual Machine is a *quasi*-Turing complete machine because it is "intrinsically bounded through a

parameter, *gas*, which limits the total amount of computation done."). Setting aside the theoretical foundations for these technical differences, the important point is that the Ethereum blockchain provides for the executing of sophisticated business logic and programs allowing it to function as a computer. *See* Gavin Wood & Andreas M. Antonopoulos, Mastering Ethereum: Building Smart Contracts And Dapps, 2 (explaining that "Ethereum can straightforwardly function as a general-purpose computer.")[hereinafter Mastering Ethereum].

[2]Nick Szabo, *Smart Contracts* (1994), https://web.archive.org/web/20011102030833/http://szabo.best.vwh.net:80/smart.contracts.html.

[3]Nick Szabo, *Smart Contracts: Building Blocks for Digital Free Markets* (1996), https://web.archive.org/web/20040622133315/http://szabo.best.vwh.net/smart_contracts_2.html [hereinafter *Building Blocks*] (partial rewrite of article first appearing in *Extropy:* Journal Of Transhumanist Thought, no. 16, 1996.

[4]*See e.g.*, Ariz. Rev. Stat. Ann. § 44-7061.E.2 (" 'Smart contract' means an event-driven program, with state, that runs on a distributed, decentralized, shared and replicated ledger and that can take custody over and instruct transfer of assets on that ledger.").

[5]*See e.g.*, *CFTC v. McDonnell*, 287 F.Supp.3d 213, 247 (E.D.N.Y. 2018).

[6]*See* LabCFTC, *A Primer on Smart Contracts* (Nov. 27, 2018), https://www.cftc.gov/sites/default/files/2018-11/LabCFTC_PrimerSmartContracts112718.pdf [hereinafter *Primer*].

[7]Coded computer functions or computerized transactions protocols that automate the performance of contractual obligations are not new. These descriptions could encompass established technology like Electronic Data Interchange ("EDI") systems that companies have used for years. An EDI system allows buyers and sellers to facilitate business transactions by exchanging electronically communicating predefined structured data in standardized EDI forms. *See e.g.*, Judith E. Payne & Robert H. Anderson, *Electronic Data Interchange (EDI) Using Electronic Commerce to Enhance Defense Logistics*, at v (RAND Nat'l Def. Research Inst. 1991), https://apps.dtic.mil/dtic/tr/fulltext/u2/a238559.pdf ("[EDI] has been advocated as one of the most important applications of computer technology, and one that holds the greatest potential for improving the nation's productivity—for with EDI will come 'electronic commerce[.]' ").

[8]It is important to distinguish smart contracts on private, permissioned platforms from those on public, permissionless platforms such as Ethereum. Under private platforms, participants can be required to execute a commercial license and other related agreements to use the enterprise platform. Public blockchains, on the other hand, do not require any such agreement to access the network. Differences like these play an important role in how the smart contract is enforced and regulated.

[9]Blockchain technology continues to evolve. There are a number of potential changes for the Ethereum network on the horizon and potential for other distributed ledger technology innovations.

[10]*See e.g.,* Primavera De Filippi & Aaron Wright, Blockchain And The Law: The Rule Of Code, 32 (2018).

[11]Blockchain is a specific implementation of distributed ledger technology with the centrality of a single, global ledger as the single source of trust as its distinguishing characteristic.

[12]Nolan Bauerle, *What is a Distributed Ledger?*, Coindesk, https://www.coindesk.com/information/what-is-a-distributed-ledger/.

[13]*Id.*

[14]Mastering Ethereum, *supra* note 1, at 2.

[15]Bitcoin's source code supports multi-signature schemes that require more than one private key signature to redeem a transaction which can be used to backup a Bitcoin wallet and/or for escrow transactions. Adding hurdles to the release of payment does not change the one-way transfer into an exchange of value reflected in the terms of the multi-signature smart code *See P2SH Multisign, PsSH Multisig Output*, Bitcoin Developer Glossary, https://bitcoin.org/en/glossary/p2sh-multisig.

[16]Cardozo Blockchain Project, *"Smart Contracts" & Legal Enforceability, Research Report No. 2* (2018), https://cardozo.yu.edu/sites/default/files/Smart%20Contracts%20Report%20%232_0.pdf [hereinafter Cardozo].

[17]*EIP 20: ERC-20 Token Standard*, Ethereum Improvement Proposals, https://eips.ethereum.org/EIPS/eip-20.

[18]Dickson Chin, *Smart Code and Smart Contracts, in* Blockchain For Business Lawyers, 87 (Mark W. Rasmussen & James A. Cox eds., 2018).

[19]A stop loss order is a good example of "if, then" statement in use in the markets. A typical stop loss order would be something like "if" the price of [A] falls below [X], then sell [Y] number of shares at market.

[20]The blend of technologies underlying blockchain technology fostered a decentralized, highly resilient, and tamper-resistant database allowing parties to transact without knowing or trusting their counterparty and with limited fear of third-party intervention. *See* De Filippi, *supra* note 10, at 2.

[21]Scott A. McKinney, Rachel Landy, & Rachel Wilka, *Smart Contracts, Blockchain, and the Next Frontier of Transactional Law*, 13 Wash. J. L., Tech, & Arts 313, 323 (2018); *see also* Richard Holden & Anup Malani, *Can Blockchain Solve the Holdup Problem in Contracts?* (Univ. Chi. Coase-Sandor Inst. for. Law & Econ., 2017), https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3093879.

[22]Mastering Ethereum, *supra* note 1, at 253.

[23]Ethan M. Katsh, Law In A Digital World (1995).

[24]Max Raskin, *The Law and Legality of Smart Contracts*, 1 Geo. L. Tech. Rev. 305, 308 (2017).

[25]*Building Blocks*, *supra* note 3.

[26]Mastering Ethereum, *supra* note 1, at 255.

[27]*See e.g.*, Cardozo, *supra* note 16, at 26 (concluding that "recent state amendments to the UETA are largely unnecessary"); *but see*, Shearman & Sterling LLP, R3 & BAFT, *Code is Not Law: The Legal Background for Trade Finance Using Blockchain* (2018), https://www.jdsupra.com/legalnews/code-is-not-law-the-legal-background-70089, 1-2 (identifying legal obstacles to the use of blockchain technology for certain digital trade finance transactions and proposing and exploring amending the UETA as a possible solution); Smart Contracts Alliance, *Smart Contracts: Is the Law Ready?* (Chamber of Digital Commerce, 2018), https://digitalchamber.org/smart-contracts-whitepaper, 54-59 (identifying practical and legal challenges to the perfection of security interests under the UCC in blockchain-based assets).

[28]*See* UCC § 1-103 (identifying the UCC's underlying purposes and policies, which are: "(1) to simplify, clarify, and modernize the law governing commercial transactions; (2) to permit the continued expansion of commercial practices through custom, usage, and agreement of the parties; and (3) to make uniform the law among the various jurisdictions.").

[29]*See* UCC § 2-201.

[30]*See Bazak Intl. Corp. v. Tarrant Apparel Group*, 378 F. Supp. 2d 377, 383-386 (S.D.N.Y. 2005) (holding that in New York an e-mail satisfied the UCC § 2-201(2) writing requirement).

[31]*See* Cardozo, *supra* note 16, at 11.

[32]*See Pain Ctr. of SE Ind. LLC v. Origin Healthcare Solutions LLC*, 893 F.3d 454, 459 (7th Cir. 2018) (noting that under Indiana law contracts involving the purchase of "preexisting, standardized software" are subject to the UCC); *but see*, *e.g.*, *Geneva Intern. Corp. v. Petrof, Spol, S.R.O.*, 608 F. Supp.2d 993, 999 (N.D. Ill. 2009) (finding that under Illinois law non-exclusive license for use of specific typeface in software was not a sale of goods because no transfer of title occurred.).

[33]*See e.g., Rottner v. AVG Technologies, USA, Inc.*, 943 F. Supp.2d 222, 231 (D. Mass. 2013) (noting that "software is not clearly a good or a service in the abstract, and may qualify as either depending on the particular circumstances of the case."). There were prior efforts to bring more clarity to the application of the UCC to software agreements. However, those efforts were never widely adopted. The American Law Institute

("ALI") originally proposed Article 2B, an amendment to the UCC to apply specifically for transactions involving software. However, ALI subsequently withdrew its support for the draft model provision, which was renamed the Uniform Computer Information Transaction Act (UCITA). To date only Virginia and Maryland have passed UCITA, while other states such as West Virginia, North Carolina, Vermont and Iowa have proactively passed anti-UCITA statutes (referred to as "bomb-shelter" legislation) prohibiting the application of UCITA for residents of those states such as through choice of law provisions in software contracts.

[34]*See Specht v. Netscape Commc'n Corp.*, 306 F.3d 17, 28 (2d Cir. 2002).

[35]*Id. at 29* (noting that "a consumer's clicking on a download button does not communicate assent to contractual terms if the offer did not make clear to the consumer that clicking on the download button would signify assent.").

[36]The New Hampshire Supreme Court recognized electronic signatures as far back as 1869. *See Howley v. Whipple*, 48 N.H. 487, 488 (1869) (noting "it makes no difference whether [the telegraph] operator writes with a steel pen an inch long attached to an ordinary penholder, or whether his pen be a copper wire a thousand miles long. Nor does it make any difference that in one case common record ink is used, while in the other case a more subtle fluid, known as electricity, performs the same office.").

[37]UETA has been adopted in 47 states (Illinois, New York and Washington have their own laws covering electronic signatures).

[38]Notably, the comments to the UETA state that even a "digital signature using public key encryption technology would qualify as an electronic signature, as would the mere inclusion of one's name as a part of an e-mail message—so long as in each case the signer executed or adopted the symbol with the intent to sign." UETA § 2 cmt. 7 (1999).

[39]Adam R. Smart, *E-Sign versus State Electronic Signature Laws: The Electronic Statutory Battleground*, 5 N.C. Banking Inst. 485, 485 (2001).

[40]15 U.S.C.A. § 7002 (Exemption to Preemption).

[41]*See* UETA § 3(b) (Scope) and 15 U.S.C.A. § 7003 (Specific exceptions).

[42]A cryptographic hash function creates a digital fingerprint for the bytecode of the contract that is available and relied upon if the smart code is insufficient. *See* Corda Data Model Overview, https://docs.corda.net/releases/release-M9.2/key-concepts-data-model.html?highlight=legal%20prose (last visited May 3, 2019).

[43]David W. Tollen, The Tech Contracts Handbook: Cloud Computing Agreements, Software License, And Other It Contracts For Lawyers And Businesspeople, 255-256 (2d ed. 2015).

[44]*See e.g., Payoutone v. Coral Mortgage Bankers*, 602 F.Supp.2d 1219, 1225 (D. Colo. 2009) (noting that to create an enforceable contract "there must be an exchange of one party's promise or performance for the other party's promise or performance.").

[45]Christopher D. Clack, Vikram A. Bakshi, & Lee Braine, *Smart Contract Templates: foundation, design landscape and research direction*, 4 (2016), https://arxiv.org/pdf/1608.00771.pdf.

[46]Nicholas Berry, *Norton Rose Fulbright, The future of smart contracts in insurance*, Insurance Day (Sept. 19, 2016), https://www.nortonrosefulbright.com/en/knowledge/publications/88244592/the-future-of-smart-contracts-in-insurance.

[47]McKinney, *supra* note 21, at 329.

[48]Giesela Rühl, *The Law Applicable to Smart Contracts, or Much Ado about Nothing?*, (Jan. 23, 2019), https://www.law.ox.ac.uk/business-law-blog/blog/2019/01/law-applicable-smart-contracts-or-much-ado-about-nothing.

[49]David Zaslowsky, *What to Expect when Litigating Smart Contracts*, LAW360, (Apr. 4, 2018), https://www.law360.com/articles/1028009/what-to-expect-when-litigating-smart-contract-disputes.

[50]*See* CipherTrace, *Crypto Currency Anti-Money Laundering Report 2018 Q3* (2018), https://ciphertrace.com/wp-content/uploads/

2018/10/crypto_aml_report_2018q3.pdf [hereinafter CipherTrace Report]. According to the CipherTrace Report, more than $927 million was stolen in cryptocurrencies through the third quarter of 2018—a substantial amount of which came from smart contract vulnerabilities—and that number was expected to surpass $1 billion for full-year 2018.

[51]Ivica Nikolic, Aashish Kolluri, IIya Sergey, Prateek Saxena, Aquinas Hobor, *Finding the Greedy, Prodigal, and Suicidal Contracts at Scale* (2018), https://arxiv.org/abs/1802.06038; *see also* Amrit Kumar, *The Importance of Formal Verification in Smart Contracts*, Networks Asia (Jan. 21, 2019), https://www.networksasia.net/article/importance-formal-verification-smart-contracts.1548047083.

[52]*See generally*, Michael S. Sackheim et al., *United States* in The Virtual Currency Regulation Review (Michael S. Sackheim & Nathan A. Howell eds., 2018), https://thelawreviews.co.uk/edition/the-virtual-currency-regulation-review-edition-1/1176673/united-states.

[53]Report of Investigation Pursuant to Section 21(a) of the Securities Exchanger Act of 1934: The DAO, Exchange Act Release No. 81,207, at 9 (July 25, 2017), https://www.sec.gov/litigation/investreport/34-81207.pdf.

[54]Complex smart contracts can raise intractable issues that go well beyond coding errors.

[55]*See* Mike Orcutt, *Ethereum's got a hard forking problem thanks to another delayed upgrade*, MIT Tech. Review (Jan. 17, 2019), https://www.technologyreview.com/s/612769/ethereums-got-a-hard-forking-problem-thanks-to-another-delayed-upgrade.

[56]*Id.*

[57]Andrea Tinianow, *When Blockchains Crash, Who Can You Sure?* Forbes (Feb. 7, 2019), https://www.forbes.com/sites/andreatinianow/2019/02/07/when-blockchains-crash-whom-can-you-sue/#578e9eaa7775.

[58]John Storino, Justin C. Steffen, & Matthew T. Gorden, *Decrypting the Ethical Implications of Blockchain Technology*, LAW.COM (Nov. 13, 2017), https://www.law.com/legaltechnews/sites/legaltechnews/2017/11/13/decrypting-the-ethical-implications-of-blockchain-technology.

[59]In re Zachary Coburn, Exchange Act Rel. No. 84,553 (November. 8, 2018), https://www.sec.gov/litigation/admin/2018/34-84553.pdf.

[60]*Id.* at 4-5. The EtherDelta smart contract smart code can be found at https://etherscan.io/address/0x4aea7cf559f67cedcad07e12ae6bc00f07e8cf65#code.

[61]Press Release, Sec. Exch. Comm'n, SEC Charge EtherDelta Founder with Operating an Unregistered Exchange (Nov. 8, 2018), https://www.sec.gov/news/press-release/2018-258.

[62]*Id.*

[63]15 U.S.C.A. § 78e.

[64]15 U.S.C.A. § 78c(a)(1).

[65]17 C.F.R. § 240.3b-16(a) (Definitions of Terms Used in Section 3(a)(1) of the Act).

[66]Coburn, *supra* note 60, at 9.

[67]*Id.*

[68]Kai Sedgwick, *Smart Contract Developers May Be Held Liable by the SEC*, BITCOIN-.COM, (Nov. 18, 2018), https://news.bitcoin.com/smart-contract-developers-may-be-held-liable-by-the-sec.

[69]*See e.g.*, Letter from Rainey Reitman, Chief Program Officer, & Aaron Mackey, Staff Attorney, Electronic Frontier Foundation, to Brent Fields, Secretary, SEC (Feb. 12, 2019), https://www.eff.org/files/2019/02/12/correspondence_from_eff_re_in_the_matter_of_zachary_coburn_file_no._3-18888-2.pdf (noting that the SEC's Order involving the EtherDelta smart contract and related contemporaneous public statement by the SEC "could be read to imply that anyone merely writing and publishing code that becomes part of a decentralized exchange could be subject to licensing requirements or liability under U.S. securities laws.").

[70]*See* Public Statement, Sec. Exch. Comm'n, Statement on Digital Asset Securities Issuance and Trading, (Nov. 16, 2018), https://www.sec.gov/news/public-statement/digital-asset-securites-issuuance-and-trading.

[71]Brian Quintenz, Commissioner, Commodity Futures Trading Comm'n, Remarks at the 38th Annual GITEX Technology Week Conference (Oct. 16, 2018), https://www.cftc.gov/PressRoom/SpeechesTestimony/opa quintenz16.

[72]The Commodity Exchange Act (the "CEA") provides, in relevant part, that "any person who commits, or who willfully aids, abets, counsels, commands, induces, or procures the commission of, a violation of [the CEA or CFTC rules]. . . may be held responsible for such violation as a principal." 7 U.S.C.A. § 13c.

[73]*See* Press Release, Commodity Futures Trading Comm'n, CFTC Charges Jitesh Thakkar and Edge Financial Technologies Inc. with Aiding and Abetting Spoofing and Manipulative and Deceptive Scheme (Jan. 29, 2018), https://www.cftc.gov/PressRoom/PressReleases/pr7689-18.

[74]*Id.*

[75]*Primer*, *supra* note *6.*