CONTENTS

General chapters

Country chapters

Legal issues surrounding the use of smart contracts

Stuart Levi, Alex Lipton & Cristina Vasile
Skadden, Arps, Slate, Meagher & Flom LLP

"Smart contracts" are a critical building block in the development and evolution of many types of transactions executed on distributed ledger technologies such as blockchains.[1] By automating processes and increasing outcome certainty, smart contracts can offer important benefits in a system that effectively relies on computer networks to process transactions. This article examines whether smart contracts are enforceable legal agreements under contract law in the United States, and highlights certain legal and practical considerations that will need to be addressed before smart contracts can be widely adopted in commercial contexts.

Smart contracts: An introduction

"Smart contracts" is a term used to describe computer code that automatically executes all or parts of the transaction steps of an oral or written agreement between two parties. The code can either be the sole manifestation of the agreement between the parties ("code-only smart contracts") or complement a traditional natural language-based contract by effectuating certain provisions of that contract ("ancillary smart contracts"). The critical difference between smart contracts and natural language contracts is how they handle performance: natural language contracts generally rely on the parties to perform the contract's obligations, whereas smart contracts perform the parties' obligations automatically once triggered. By eliminating the need for human intervention, smart contracts potentially reduce the execution and enforcements costs of the contract process. As a basic example, consider an agreement between an insurer and a farmer that will pay the farmer in the event temperatures drop below a certain degree. In a natural language contract, the farmer would need to check the temperature each day, make a claim if the temperature falls below the agreed-upon degree, and then wait for the insurer to verify the claim and pay the farmer (or dispute the claim). If a smart contract component was added, the smart contract could automatically receive a feed of the official recorded temperature (using a measure agreed by the parties) and then automatically transfer funds from the insurer's account to the farmer's account if the temperature drops below the agreed-upon level.

Standards organizations and trade associations have also begun to acknowledge the impact that smart contracts could have on transactions in their areas. For example, the International Swaps and Derivatives Association ("ISDA") has signaled an openness to smart contracting in the derivatives context, though ISDA noted that any use of smart contracts must comply with existing legal requirements such as ISDA's documentation standard.[2]

The concept of smart contracts was first articulated by the computer scientist and cryptographer Nick Szabo and predates the development of blockchain technology.[3] Since then, the ability to store immutable code and data in a transparent way on a blockchain, and the interest in disintermediating human intervention, has generated widespread interest in developing smart contracts. As with other data stored on a blockchain (such as the amount of cryptocurrency held by an address), smart contract code is replicated across multiple nodes and executed according to the same consensus mechanism on a blockchain. Moreover, because smart contracts use the same asymmetric cryptography, in which users rely on private keys and public keys, as other blockchain-based transactions, smart contracts allow parties to authenticate each other, and provide a level of security not present in many other automated transactions.

Although smart contracts have great potential to reduce transaction costs and minimize outcome uncertainty, they currently can replace only the types of contractual provisions that can be represented in specific and objective terms, such as "if X occurs, then execute step Y." Subjective provisions, such as whether a party used commercially reasonable efforts, cannot be translated into smart contracts. In this respect, smart contracts are not particularly "smart." It is therefore important not to confuse smart contracts with efforts being made in the areas of artificial intelligence and machine learning.

In addition, smart contracts will often need to rely on external (i.e., "off-chain") resources before they can execute a transaction. In the crop insurance example above, the recorded temperature would be such an off-chain resource. The reliance on off-chain resources presents several problems. For example, smart contracts cannot "pull" data from off-chain resources; rather, that data must be "pushed" to the smart contract, so the parties need to agree on a single, definitive, off-chain resource willing to and capable of pushing relevant data to the smart contract. Without such clarity, there would not be a consensus as to whether the contract should trigger, and the transaction would not execute. In our example, the farmer may argue that the weather service he consulted recorded a temperature of 31 degrees, while the insurer might claim a temperature of 33 degrees.

In order to address these issues, parties to smart contracts use "oracles"—trusted third parties that retrieve mutually-agreed off-chain information and then push that information to the smart contract at predetermined times. While oracles represent an elegant, and for the time-being necessary, solution to smart contracts' functional need to access off-chain resources, they introduce a potential point of failure in what might otherwise be a fully automated and decentralized transaction system. An oracle might cease conducting business, experience a system failure, be hacked, or provide erroneous data. Indeed, a hacker looking to impact smart contracts would likely have an easier time exploiting the oracle's data feed than hacking the smart contract itself.

Are smart contracts legally enforceable under contract law in the United States?[4]

Given that the use of smart contracts is in its incipient stages, there is no case law precedent that directly addresses the enforceability of smart contracts and, as discussed below, there are only a handful of state statutes purporting to address this issue directly.[5] However, the fact that smart contracts are not drafted in natural language prose should not impact their enforceability under the principles generally applicable to contracts.

The Uniform Commercial Code and Statute of Frauds

As a preliminary matter, in order to be legally enforceable, smart contracts must comply with applicable state writing and signing requirements. The most relevant requirements in this respect flow from two sources: the Uniform Commercial Code ("U.C.C."), a comprehensive set of laws governing all commercial transactions in the United States; and state laws that identify agreements that must be in writing and signed to be enforceable (referred to as the "statute of frauds"). The U.C.C. has been adopted in whole or in part by all 50 states, the District of Columbia, Puerto Rico, and the Virgin Islands; and all states except Louisiana have adopted a statute of frauds.

The "written agreement" requirement

Under the U.C.C. and statute of frauds, not every contract needs to be in writing. Under the U.C.C., the following contracts generally must be in writing: (i) a contract for the sale of goods priced at or over $500;[6] (ii) lease contracts relating to personal property requiring total payments of $1,000 or more;[7] and (iii) certain agreements creating a security interest.[8] The specifics of what terms must be in writing vary by the subject matter. For example, a contract for the sale of goods must generally specify the goods at issue and the price,[9] while a lease must generally include the required payments, the term, and a reasonable description of the leased property.[10] Similarly, each state's statute of frauds generally requires a written agreement for: (i) agreements relating to executorship, suretyship, marriage; (ii) performance to be undertaken over one or more years after the execution of the agreement; and (iii) agreements for the sale of an interest in land.[11]

The question is whether a smart contract, effectively a piece of computer code, can satisfy the writing requirement under the U.C.C. and statute of frauds. Historically, courts have recognized that under the U.C.C., a written agreement does not necessarily need to be natural language prose.[12] Indeed, the U.C.C. specifies that any type of "intentional reduction to tangible form" is sufficient.[13] This is consistent with the U.C.C. policy that the "writing" requirement is meant to assure that the intention of the parties is manifest. Thus, courts have held, for example, that emails can satisfy the U.C.C. "writing" requirement.[14] Smart contracts should be treated no differently than other forms of electronic records. This is not to say that all smart contracts, by definition, will satisfy the U.C.C. requirement. Just as an email may be inconclusive as to what the parties actually intended, so too a smart contract may be too vague. That said, given the objective nature of smart contract code and the parameter certainty required to effectuate a transaction, most smart contracts for the sale of goods or for leases should satisfy the U.C.C. "writing" requirement, particularly if the parties use an ancillary smart contract where the code just executes certain terms in the natural language agreement.

A similar analysis can be applied under the statue of frauds. Under these state laws, a valid writing need not be written entirely in natural language prose nor be comprehensive.[15] As with contracts interpreted under the U.C.C., courts have taken an expansive view as to what can satisfy the "writing" requirement under the statute of frauds, focusing on the intent of the parties to create a binding agreement.[16] Thus, terms conveyed through e-mail or even types of telegraphic code can form binding contracts.[17]

In addition, the writing under the statue of frauds generally need only contain the agreement's "essential terms" which can vary depending on the type of transaction.[18] As noted above, given the nature of smart contracts, the "essential terms" (such as price and what is being delivered) will likely be captured by the code itself. And, even if the essential terms are not capable of being expressed in "if-then" terms, smart contracts can be used as ancillary tools to natural language contracts that include those terms.

The signature requirement

Both the U.C.C. and the statute of frauds require that a contract have valid signatures to be binding. This requirement can also be satisfied when using smart contracts. The U.C.C. specifies that a signature can be "any symbol executed or adopted with present intention to adopt or accept a writing."[19] Similarly, the statute of frauds generally recognizes that a signature may be any symbol made by a party with the present intent to authenticate a writing or contract.[20] Courts typically look to the intent of the parties and whether the signing parties proffered a signature with an intention to authenticate the writing.[21] Since smart contract transactions on a blockchain need to be affirmatively authenticated by each party using public-private key cryptography, a digital signature on a smart contract should constitute a "symbol executed or adopted with present intention to adopt or accept a writing"[22] and satisfy the flexible signature requirements of the U.C.C. and statute of frauds.

The E-SIGN Act and UETA

The Electronic Signatures in Global National Commerce Act ("E-SIGN Act") and state laws modeled on the Uniform Electronic Transactions Act ("UETA") also provide important support for the concept that smart contracts should be treated as legally enforceable agreements. Under each of these acts, electronic records and electronic signatures used in interstate or foreign commerce transactions generally cannot be denied legal effect solely because they are in electronic form.[23] Although E-SIGN is a federal law, and generally preempts state laws, individual states may "modify, limit, or supersede"[24] the E-SIGN Act if they adopt UETA or satisfactory "alternative procedures or requirements."[25] UETA has been adopted by 47 states, the District of Columbia, Puerto Rico and the Virgin Islands.

The key question is whether the blockchains on which smart contracts are stored are "electronic records" and therefore enjoy protection under these acts, and whether the digital signatures used with smart contracts can be deemed protectable "electronic signatures."

Both the E-SIGN Act and UETA define electronic records broadly to include any "record created, generated, sent, communicated, received, or stored by electronic means."[26] An explanatory comment to UETA indicates that this includes any "[i]nformation processing systems, computer equipment and programs . . . and similar technologies" and any "information stored on a computer hard drive."[27] There should be little dispute that a blockchain satisfies this broad definition since, at a minimum, it stores records by electronic means. Moreover, at least one court has suggested that a database is an electronic record under UETA,[28] providing important guidance given that a blockchain is an encrypted and distributed database.

The E-SIGN Act and UETA also define electronic signatures broadly. Under both acts, an "electronic signature" includes any "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."[29] Moreover, UETA expressly states that this definition encompasses a "digital signature using public key encryption technology."[30] As with the statute of frauds and the U.C.C., a digital signature based on asymmetric cryptography that is used to sign a smart contract should meet the E-SIGN Act and UETA definition of a legally valid electronic signature.

The E-SIGN Act and UETA also include an additional concept that supports the enforceability of smart contracts. Under these acts, an agreement cannot be denied legal effect because the parties used an "electronic agent" which each act defines to include a "computer program or an electronic or other automated means used independently to initiate an action or respond to electronic records or performances in whole or in part, without review or action by an individual."[31] Smart contracts which run self-executing code agreed to by the contracting parties would seem to fit squarely within this definition. The comments to UETA also contemplate the possibility that electronic agents could conduct transactions with other electronic agents or autonomously, which could occur as smart contracts and artificial intelligence continue to develop.[32]

In order to rely on the foregoing protections of UETA, the parties must first agree in a non-electronic writing that they will conduct all or part of a transaction electronically. Thus, one party could not implement a smart contract without the express written consent of the other party. Similarly, if a written record needs to be made available to a consumer, the E-SIGN Act requires affirmative consumer consent before an electronic record can be used, which consent can be withdrawn at any time.[33] The right for consumers to withdraw their consent at any time under the E-SIGN Act may create operational complications given the self-executing nature of most smart contracts.

As noted above, only 47 states have adopted UETA. Illinois (through the state's Electronic Commerce Security Act),[34] New York (through the state's Electronic Signatures and Records Act),[35] and Washington (though a state statute that recognizes the E-SIGN Act as applying to state and local transactions)[36] have each adopted their own unique e-signature statutes in lieu of a statute modeled on UETA. While these three states adopt broad definitions of electronic records and electronic signatures, none offer the added protection of electronic agents set forth in the 47 states that have adopted UETA.

Specific state laws applicable to smart contracts

Although, as discussed above, there are strong arguments that existing state laws already provide a sound basis for the enforceability of smart contracts, to date, four states have amended their laws specifically to allow for the enforceability of blockchain-based contracts. Many believe that these states have done so in order to appear "blockchain friendly" to attract blockchain-based companies. However, in their attempts to provide greater clarity on this issue and incentivize blockchain-based development, these states may have created more uncertainty, in part because of how these laws will be interpreted and in part because of the implicit suggestion that existing laws did not cover smart contract transactions.

Arizona

In March 2017, Arizona became the first state to amend its version of UETA, the Arizona Electronic Transactions Act ("AETA") to address blockchain technology. The AETA as amended provides that a "signature that is secured through blockchain technology is . . . an electronic signature," and "a record or contract that is secured through blockchain technology is . . . an electronic record."[37] The AETA further states that "[s]mart contracts may exist in commerce" and that contracts "may not be denied legal effect, validity or enforceability solely because that contract contains a smart contract term."[38] Blockchain technology is defined to mean "distributed ledger technology that uses a distributed, decentralized, shared and replicated ledger, which may be public or private, permissioned or permissionless, or driven by tokenized crypto economics or tokenless. The data on the ledger is protected with cryptography, is immutable and auditable and provides an uncensored truth."[39] A smart contract is defined as "an event-driven program, with state, that runs on a distributed, decentralized, shared and replicated ledger that can take custody over and instruct transfer of assets on that ledger."[40] Although these definitions are broad, they employ multiple ambiguous terms whose exact meaning litigants and courts may debate.

Nevada

In June 2017, Nevada amended its version of UETA, the Nevada Electronic Transactions Acts ("NETA") to state that an "electronic record" includes, without limitation, a blockchain.[41] The statute defines blockchain to mean "an electronic record of transactions or other data which is: (i) [u]niformly ordered; (ii) processed using a decentralized method by which one or more computers or machines verify the recorded transactions or other data; (iii) [r]edundantly maintained or processed by one or more computers or machines to guarantee the consistency or nonrepudiation of the recorded transactions or other data; and (iv) [v]alidated by the use of cryptography."[42] A recent amendment, which will go into effect in October 2019, clarifies that the definition of blockchain includes, without limitation, a public blockchain.[43] Smart contracts are not directly addressed in the statute, and note that the definition of blockchain is fairly different than that adopted by Arizona.[44]

Ohio

In June 2018, Ohio amended its version of UETA to state that "a record or contract that is secured through blockchain technology is considered to be in an electronic form and to be an electronic record."[45] The law also amends the definition of electronic signatures to state that "a signature that is secured through blockchain technology is considered to be in an electronic form and to be an electronic signature"[46] and that "a record or signature may not be denied legal effect or enforceability solely because . . . the contract contains a smart contract term." The amendment mirrors Arizona's definition of blockchain technology, defining it as "distributed ledger technology that uses a distributed, decentralized, shared, and replicated ledger, which may be public or private, permissioned or permissionless, or driven by tokenized crypto economics or tokenless. The data on the ledger is protected with cryptography, is immutable and auditable, and provides an uncensored truth."[47]

Tennessee

In March 2018, Tennessee amended its UETA to clarify that "a record or contract that is secured through distributed ledger technology is considered to be in an electronic form and to be an electronic record."[48] It further provides: "[a] cryptographic signature that is generated and stored through distributed ledger technology is considered to be . . . an electronic signature."[49] Tennessee adopted some of the blockchain technology definition used by Arizona and Ohio, but categorized it as "distributed ledger technology" and made some important modifications. Specifically, distributed ledger technology is defined as "any distributed ledger protocol and supporting infrastructure, including blockchain, that uses a distributed, decentralized, shared, and replicated ledger, whether it be public or private, permissioned or permissionless, and which may include the use of electronic currencies or electronic tokens as a medium of electronic exchange."[50] Similarly, the state's definition of "smart contracts" mirrors that of Arizona and Ohio but adds some additional language. A "smart contract" is defined to mean "an event-driven computer program, that executes on an electronic, distributed, decentralized, shared, and replicated ledger that is used to automate transactions, including, but not limited to, transactions that: (A) [t]ake custody over and instruct transfer of assets on that ledger; (B) [c]reate and distribute electronic assets; (C) [s]ynchronize information; or (D) [m]anage identity and user access to software applications."[51, 52]

Other legal considerations

In addition to the foregoing statutes generally governing the enforceability of contracts, smart contracts may be subject to a variety of legal frameworks depending on their terms and consideration. This may include state and federal commodities and securities laws and regulations; anti-money laundering laws and regulations; and state money transmission laws. Developers of, and parties to, smart contracts must discern which regulations apply and what such compliance entails, including registration and documentation requirements.

Challenges with the widespread adoption of smart contracts

Given the existing legal frameworks for recognizing electronic contracts, it is quite likely that a court today would recognize the validity of code that executes provisions of a smart contract—what we have classified as ancillary smart contracts. There is also precedent to suggest that a code-only smart contract might enjoy similar legal protection. The challenge to widespread smart contract adoption may therefore have less to do with the limits of the law than with potential clashes between how smart contract code operates and how parties transact business. We set forth below certain of these challenges:

<u>How can non-technical parties negotiate, draft and adjudicate smart contracts</u>?

A key challenge in the widespread adoption of smart contracts is that parties will need to rely on a trusted, technical expert to either capture the parties' agreement in code or confirm that code written by a third party is accurate. While some analogize this to hiring a lawyer to explain "the legalese" of a traditional text-based contract, the analogy is misplaced. Non-lawyers typically can understand simple short-form agreements as well as many provisions of longer agreements, especially those setting forth business terms. But a non-programmer would be at a total loss to understand even the most basic smart contract and is therefore significantly more beholden to an expert to explain what the contract "says."

To some extent, the inability of contracting parties to understand the smart contract code will not be a hindrance to entering into ancillary smart contracts. This is because for many basic functions, text templates can be created and used to indicate what parameters need to be entered and how those parameters will be executed. For example, assume a simple smart contract function that extracts a late fee from a counterparty's wallet if a defined payment is not received by a specified date. The text template could prompt the parties to enter the amount of the expected payment, the due date and the amount of the late fee. However, a party may want to confirm that the underlying code actually will perform the functions specified in the text, and that there are no additional conditions or parameters—especially where the template disclaims any liability arising from the accuracy of the underlying code. This review will require a trusted third party with programming expertise.

In cases where such templates do not exist, and new code must be developed, the parties will need to communicate the intent of their agreement to a programmer. Simply handing that programmer a copy of the legal agreement would be inefficient since it would require the programmer to try to decipher a legal document. Parties relying on ancillary smart contracts therefore may need to draft a separate "term sheet" of functionality that the smart contract should perform and that can be provided to the programmer.

The parties also may want written representations from the programmer that the code performs as contemplated. The net result is that for customized arrangements that do not rely on an existing template, the parties may need to enter into a written agreement with the smart contract programmer, not unlike the contract that parties may enter into with a provider of services for Electronic Data Interchange transactions today.

Insurance companies could also create policies to protect contracting parties from the risk that smart contract code does not perform the functions specified in the text of an agreement. Although the parties would also want to review (or have a third party review) the code, insurance can provide additional protection given that the parties might miss errors when reviewing the code. The parties would also take some additional comfort from the fact that the insurance company likely conducted its own code audit before agreeing to insure the code.

Code-only smart contracts used for business-to-consumer transactions could pose an additional set of issues that will need to be addressed. Courts are wary of enforcing agreements where the consumer did not receive adequate notice of the terms of the agreement,[53] and may be hesitant to enforce a smart contract where the consumer was not also provided with an underlying text agreement that included the complete terms.

Finally, as the validity or performance of smart contracts increasingly become adjudicated, courts may need a system of court-appointed experts to help them decipher the meaning and intent of the code. Today, parties routinely use their own experts when technical issues are at the center of a dispute. While both federal courts and many state courts have the authority to appoint their own experts, they rarely exercise that authority.[54] That approach may need to change if the number of standard contract disputes that center on interpreting smart contract code increases.

Liability of the smart contract developer

As noted above, in many cases, the parties to a smart contract will not have the technical capability to create a smart contract, and may therefore hire a third party to create the smart contract, or may rely on a smart contract "template" offered by a third party. In such cases, there is the possibility of programmer error or that the parties did not accurately convey what they intended to the developer. Parties will need to consider the ramifications of these situations and the appropriate allocation of risk and liability.

Developers of smart contracts may also need to be wary of their own liability in cases where smart contract code they developed is used for unlawful purposes. In October 2018, Brian Quintenz, Commissioner of the Commodity Futures Trading Commission ("CFTC"), suggested that smart contract code developers could be held accountable for aiding and abetting CFTC violations where they "could reasonably foresee, at the time they created the code, that it would likely be used by U.S. persons in a manner violative of CFTC regulations."[55] In November 2018, the Securities and Exchange Commission ("SEC") settled charges of operating an unregistered securities exchange against Zachary Coburn, the founder and developer of EtherDelta, a decentralized digital asset exchange. Although the SEC's order appears to be based, in part, on Coburn's control over EtherDelta's operations and his role as founder, the order also lists the fact that Coburn "wrote and deployed the EtherDelta smart contract to the Ethereum Blockchain" as a factor in finding that Coburn caused EtherDelta to violate the Securities Exchange Act of 1934.[56]

While some cases of developer liability will be clear, such as where a developer was actively part of an illegal scheme, it is likely that given the open source nature of many blockchain projects, developers will have little insight into how their smart contract code is being used, or by whom.

Outside the CFTC context, jurisprudence on contributory liability in the context of peer-to-peer technologies may provide useful precedent in balancing the need to protect developers with the need to provide redress to parties that are harmed by smart contracts put to unlawful use. For example, under *Metro-Goldwyn-Mayer Studios Inc. v. Grokster, Ltd.*,[57] peer-to-peer file-sharing sites are not liable for users' infringing uses if: (1) they are not distributing their product with the "object of promoting its use to infringe"; (2) they either (a) do not have actual knowledge of specific infringements, or (b) if they do have knowledge, they are not in a position to block the infringing conduct and have failed to do so; and (3) the product is capable of substantial noninfringing use.

While *Grokster* dealt with contributory infringement under copyright law, courts may apply its core principles in the context of developer liability for blockchain-based smart contracts. In order to minimize potential liability, smart contract developers should not only avoid developing smart contracts with the object of enabling illegal use, but should also use reasonable efforts to block unexpected unlawful use.

What is the "final" agreement between the parties?

When analyzing traditional text-based contracts, courts will examine the final, written document to which the parties have agreed in order to determine whether the parties are in compliance or breach. Courts have long emphasized that it is this final agreement that represents the mutual intent of the parties—the "meeting of the minds."

In the case of code-only smart contracts, the code that is executed—and the outcome it produces—represents the only objective evidence of the terms agreed to by the parties. In these cases, email exchanges between the parties as to what functions the smart contract "should" execute, or oral discussions to that effect, likely would yield to the definitive code lines as the determinative manifestation of the parties' intent.

With respect to ancillary smart contracts, a court likely would look at the text and code as a unified single agreement. The issue becomes complicated when the traditional text agreement and the code do not align. In the crop insurance example described above, assume the text of an agreement specifies that an insurance payout will be made if the temperature falls below 32 degrees, while the smart contract code triggers the payment if the temperature is equal to or below 32 degrees. Assuming that the text agreement does not state whether the text or code controls in the event of an inconsistency, courts will need to determine—perhaps on a case-by-case basis—whether the code should be treated as a mutually agreed amendment to the written agreement or whether the text of the agreement should prevail. In some respects, the analysis should be no different than a case where the provisions of a main agreement differ from what is reflected in an attached schedule or exhibit. The fact that here the conflict would be between text and computer code and not two text documents should not be determinative, but courts may take a different view.

One solution will be for parties to use a text-based contract where the parameters that trigger the smart contract execution are not only visible in the text but actually populate the smart contract. In our example, "less than 32 degrees" would not only be seen in the text, but also would create the parameter in the smart contract itself, thereby minimizing the chances of any inconsistency.

The automated nature of smart contracts

One of the key attributes of smart contracts is their ability to automatically and relentlessly execute transactions without the need for human intervention. However, this automation, and the fact that smart contracts cannot easily be amended or terminated unless the parties incorporate such capabilities during the creation of the smart contract, present some of the greatest challenges facing widespread adoption of smart contracts.

For example, with traditional text contracts, a party can easily excuse a breach simply by not enforcing the available penalties. If a valued customer is late with its payment one month, the vendor can make a real-time decision that preserving the long-term commercial relationship is more important than any available termination right or late fee. However, if this relationship had been reduced to a smart contract, the option not to enforce the agreement on an *ad hoc* basis likely would not exist. A late payment will result in the automatic extraction of a late fee from the customer's account or the suspension of a customer's access to a software program or an internet-connected device if that is what the smart contract was programmed to do. The automated execution provided by smart contracts might therefore not align with the manner in which many businesses operate in the real world.

Similarly, in a text-based contractual relationship, a party may be willing to accept, on an *ad hoc* basis, partial performance to be deemed full performance. This might be because of an interest in preserving a long-term relationship or because a party determines that partial performance is preferable to no performance at all. Here, again, the objectivity required for smart contract code might not reflect the realities of how contracting parties interact.

Amending and terminating smart contracts

At present, there is no simple path to amend a smart contract, creating certain challenges for contracting parties. For example, in a traditional text-based contract, if the parties have mutually agreed to change the parameters of their business deal, or if there is a change in law, the parties quickly can draft an amendment to address that change, or simply alter their course of conduct. Smart contracts currently do not offer such flexibility. Indeed, given that blockchains are immutable, modifying a smart contract is far more complicated than modifying standard software code that does not reside on a blockchain. The result is that amending a smart contract may yield higher transaction costs than amending a text-based contract, and increases the margin of error that the parties will not accurately reflect the modifications they want to make.

Similar challenges exist with respect to terminating a smart contract. Assume a party discovers an error in an agreement that gives the counterparty more rights than intended, or concludes that fulfilling its stated obligations will be far more costly than it had expected. In a text-based contract, a party can engage in, or threaten, so-called "efficient breach," *i.e.*, knowingly breaching a contract and paying the resulting damages if it determines that the cost to perform is greater than the damages it would owe. Moreover, by ceasing performance, or threatening to take that step, a party may bring the counterparty back to the table to negotiate an amicable resolution. Smart contracts do not yet offer analogous self-help remedies.

Projects are currently under way to create smart contracts that are terminable at any time and more easily amended. While in some ways this is antithetical to the immutable and automated nature of smart contracts, it reflects the fact that smart contracts only will gain commercial acceptance if they reflect the business reality of how contracting parties act.

Objectivity and the limits of incorporating desired ambiguity into smart contracts

The objectivity and automation required of smart contracts can run contrary to how business parties actually negotiate agreements. During the course of negotiations, parties implicitly engage in a cost-benefit analysis, knowing that at some point there are diminishing returns in trying to think of, and address, every conceivable eventuality. These parties no longer may want to expend management time or legal fees on the negotiations, or may conclude that commencing revenue-generating activity under an executed contract outweighs addressing unresolved issues. Instead, they may determine that if an unanticipated event actually occurs, they will figure out a resolution at that time. Similarly, parties may purposefully opt to leave a provision somewhat ambiguous in an agreement in order to give themselves the flexibility to argue that the provision should be interpreted in their favor. This approach to contracting is rendered more difficult with smart contracts where computer code demands an exactitude not found in the negotiation of text-based contracts. A smart contract cannot include ambiguous terms nor can certain potential scenarios be left unaddressed. As a result, parties to smart contracts may find that the transaction costs of negotiating complex smart contracts exceed that of a traditional text-based contracts.

It will take some time for those adopting smart contracts in a particular industry to determine which provisions are sufficiently objective to lend themselves to smart contract execution. As noted, to date, most smart contracts perform relatively simple tasks where the parameters of the "if/then" statements are clear. As smart contracts increase in complexity, parties may disagree on whether a particular contractual provision can be captured through the objectivity that a smart contract demands.

Do smart contracts really guarantee payment?

One benefit often touted of smart contracts is that they can automate payment without the need for dunning notices or other collection expenses and without the need to go to court to obtain a judgment mandating payment. While this is indeed true for simpler use cases, it may be less accurate in complex commercial relationships. The reality is that parties are constantly moving funds throughout their organization and do not "park" total amounts that are due on a long-term contract in anticipation of future payment requirements. Similarly, a person obtaining a loan is unlikely to keep the full loan amount in a specified wallet linked to the smart contract. Rather, the borrower will put those funds to use, funding the necessary repayments on an *ad hoc* basis.

If the party owing amounts under the smart contract fails to fund the wallet on a timely basis, a smart contract looking to transfer money from that wallet upon a trigger event may find that the requisite funds are not available. Implementing another layer into the process, such as having the smart contract seek to pull funds from other wallets or having that wallet "fund itself" from other sources, would not solve the problem if those wallets or sources of funds also lack the requisite payment amounts. The parties might seek to address this issue through a text-based requirement that a wallet linked to the smart contract always have a minimum amount, but that solution simply would give the party a stronger legal argument if the dispute was adjudicated. It would not render the payment operation of the smart contract wholly automatic. Thus, although smart contracts will render payments far more efficient, they may not eliminate the need to adjudicate payment disputes.

Risk allocation for attacks and failures

Smart contracts introduce an additional risk that does not exist in most text-based contractual relationships—the possibility that the contract will be hacked or that the code or protocol simply contains an unintended programming error. Given the relative security of blockchains, these concepts are closely aligned; namely, most "hacks" associated with blockchain technology are really exploitations of an unintended coding error. As with many bugs in computer code, these errors are not glaring, but rather become obvious only once they have been exploited. For example, in 2017 an attacker was able to drain several multi-signature wallets offered by Parity of $31 million in ether.[58] Multi-signature wallets add a layer of security because they require more than one private key to access the wallet. However, in the Parity attack, the attacker was able to exploit a flaw in the Parity code by reinitializing the smart contract and making himself or herself the sole owner of the multi-signature wallets. Parties to a smart contract will need to consider how risk and liability for unintended coding errors and resulting exploitations are allocated between the parties, and possibly with any third party developers or insurers of the smart contract.

Governing law and venue

One of the key promises of blockchain technology, and by extension smart contracts, is the development of robust, decentralized and global platforms. However, global adoption means that parties may be using a smart contract across far more jurisdictions than might exist in the case of text-based contracts. The party offering terms under a smart contract would therefore be best-served by specifying the governing law and venue for that smart contract. A governing law provision specifies what substantive law will apply to the interpretation of the smart contract, whereas a venue clause specifies which jurisdiction's courts will adjudicate the dispute. In cases where governing law or venue is not specified, a plaintiff may be relatively unconstrained in choosing where to file a claim or in arguing which substantive law should apply given the wide range of jurisdictions in which a smart contract might be used. Given that many early disputes concerning smart contracts will be ones of first-impression, contracting parties will want some certainty surrounding where such disputes will be adjudicated.

Conclusion

As smart contracts are in their nascent stages, so is the law surrounding their enforceability and use. While there are strong arguments that properly constructed smart contracts are enforceable under existing statutes governing electronic contracts, certain issues must be resolved before they can enjoy widespread adoption in complex commercial transactions. While smart contracts have potential to change the way markets operate, their impact will invariably be shaped by how such applications fit within the contours of the law.

* * *

Acknowledgment

The authors acknowledge the assistance of Daniel Chase, a law student at Berkeley Law.

* * *

Endnotes

[1] Blockchains are one type of "distributed ledger technology" in which data is organized in blocks and new data can only be appended to the chain. For purposes of this article, we refer to blockchains, but most of the legal issues presented here apply to other forms of distributed ledger technology as well.

[2] *See* International Swaps and Derivatives Association, *ISDA Legal Guidelines For Smart Derivatives Contracts: Introduction* (Jan. 2019), https://www.isda.org/a/MhgME/Legal-Guidelines-for-Smart-Derivatives-Contracts-Introduction.pdf.

[3] *Compare* Nick Szabo, *Smart Contracts: Building Blocks for Digital Market* (1996) *with* Satoshi Nakamoto, *Bitcoin: A Peer-to-Peer Electronic Cash System* (2008).

[4] There is no federal contract law in the United States; rather, the enforceability and interpretation of contracts is determined at the state level. Thus, while certain core principles apply consistently across state lines, and there has been a drive to harmonize state laws by the National Conference of Commissioners on Uniform State Laws, any conclusions regarding the enforceability of smart contracts must be tempered by the reality that states may adopt different views.

[5] For a comprehensive overview of the enforceability of smart contracts, see *"Smart Contracts" & Legal Enforceability* (Cardozo Blockchain Project Research Report No. 2, Oct. 16, 2018), https://cardozo.yu.edu/sites/default/files/Smart%20Contracts%20Report%20%232_0.pdf; *see also* Uniform Law Commission, *Guidance Note Regarding the Relation Between the Uniform Electronic Transactions Act and Federal ESIGN Act, Blockchain Technology and 'Smart Contracts'* (Feb. 11, 2019) (opining that state UETA provisions do not require amendment to enable use of blockchain technology and smart contracts in electronic transactions).

6. U.C.C. § 2-201.
7. *Id.* § 2A-201.
8. *Id.* § 9-203(b)(3)(A).
9. *Id.* § 2-201.
10. *Id.* § 2A-201.
11. *See, e.g.*, Restatement (Second) Contracts § 110. Contracts that fail to comply with the statute of frauds remain enforceable in some cases, such as cases wherein promissory estoppel applies. *See* Restatement (Second) Contracts § 90.
12. *See, e.g.*, Apex Oil Co. v. Vanguard Oil Serv. Co., 760 F.2d 417, 420, 423 (2d Cir. 1985).
13. U.C.C. § 1-201(43).
14. *See, e.g.*, Bazak Int'l Corp. v. Tarrant Apparel Grp., 378 F. Supp. 2d 377, 392 (S.D.N.Y. 2005) ("Although e-mails are intangible messages during their transmission, this fact alone does not prove fatal to their qualifying as writings under the UCC[.] [F]orms of communication regularly recognized by the courts as fulfilling the UCC "writing" requirement, such as fax, telex and telegraph, are all intangible forms of communication during portions of their transmission. Just as messages sent using these accepted methods can be rendered tangible, thereby falling within the UCC definition, so too can e-mails.")
15. *See*, *e.g.*, Bibb v. Allen, 149 U.S. 481, 497–98 (1893) (holding that a contract written in telegraphic cipher code was binding); Cloud Corp. v. Hasbro, Inc., 314 F.3d 289, 295–96 (7th Cir. 2002).
16. *See, e.g.*, Leeds v. First Allied Connecticut Corp., 521 A.2d 1095, 1097 (Del. Ch. 1986) (explaining an agreement is binding when "a reasonable negotiator . . . would have concluded, in that setting, that the agreement reached constituted agreement on all of the terms that the parties themselves regarded as essential[.]").
17. *See*, *e.g.*, Bibb, 149 U.S. at 497–98; Naldi v. Grunberg, 908 N.Y.S.2d 639, 645 (App. Div. 2010).
18. *See*, *e.g.*, *Ross v. Ross*, 172 A.3d 1069, 1075 (N.H. 2017); Simmonds v. Marshall, 292 A.D.2d 592, 592 (2d Dep't 2002); Leeds v. First Allied Connecticut Corp., 521 A.2d at 1097.
19. U.C.C. § 1-201(37).
20. Restatement (Second) Contracts § 134; U.C.C. § 1-201(37).
21. *See*, *e.g.*, SD Protection, Inc. v. Del Rio, 498 F. Supp. 2d 576, 584 (E.D.N.Y 2007); U.C.C. § 1-201 cmt 37.
22. *See* U.C.C. § 1-201(37); *see also* Restatement (Second) Contracts § 134.
23. 15 U.S.C. § 7001(a)(1); UETA § 7(a), (c)-(d). There are certain exceptions to these acts (such as wills) that will not impact the majority of smart contract usage.
24. 15 U.S.C. § 7002(a).
25. 15 U.S.C. § 7002(a)(2)(A).
26. UETA § 2(7); *see also* 15 U.S.C. § 7006(4).
27. *Id.* § 2 cmt. 6.
28. *See* Godfrey v. Fred Meyer Stores, 124 P.3d 621, 631 (2005) (Armstrong, J., concurring).

29. UETA § 2(8); *see also* 15 U.S.C. § 7006(5).
30. *Id*. § 2 cmt. 7.
31. *Id*. § 2(6); 15 U.S.C.§ 7006(3).
32. *Id*. § 2 cmt. 5
33. 15 U.S.C. § 7001(c)(1).
34. 5 Ill. Comp. Stat. 175/5-110.
35. N.Y. State Tech. § 304.
36. Wash. Rev. Code § 19-360.010–360.040.
37. Ariz. Rev. Stat. Ann. § 44-7061.
38. *Id.*
39. *Id.*
40. *Id.*
41. Nev. Rev. Stat. Ann. § 719.090.
42. Nev. Rev. Stat. Ann. § 719.045, as amended by 2019 Nev. S.B. 162. Note that the amended version of this statute will become effective on October 1, 2019.
43. 2019 Nev. S.B. 162.
44. *See also* 2019 Nev. S.B. 163.
45. Ohio Rev. Code Ann. § 1306.01(G).
46. *Id.* § 1306.01(H).
47. *Id.* § 1306.06(A).
48. Tenn. Code Ann. § 47-10-202(b).
49. *Id.* § 47-10-202(a).
50. *Id.* § 47-10-201(1).
51. *Id.* § 47-10-201(2).
52. Note that other states, including California, Colorado, Connecticut, Delaware and Vermont, have enacted blockchain-related laws as well, though these laws do not specifically address the issue of blockchain-based contracts.
53. *See*, *e.g.*, Nicosia v. Amazon.com, Inc., 834 F.3d 220, 237–38 (2d. Cir. 2016) (reversing the district court's dismissal for failure to state a claim and holding that reasonable minds could disagree as to whether Amazon provided the consumer with reasonable notice of the mandatory arbitration provision at issue).
54. *See* Charles Alan Wright & Arthur R. Miller, *Federal Practice and Procedure*, Section 6304 (3d ed. supp. 2011) ("In fact, the exercise of Rule 706 powers is rare under virtually any circumstances. This is, at least in part, owing to the fact that appointing an expert witness increases the burdens of the judge, increases the costs to the parties, and interferes with the adversarial control over the presentation of evidence."); Stephanie Domitrovich, Mara L. Merino & James T. Richardson, *State Trial Judge Use of Court Appointed Experts: Survey Results and Comparisons*, 50 Jurimetrics J. 371, 373–74 (2010).
55. Brian Quintenz, Commissioner, U.S. Commodity Futures Trading Commission, Remarks at the 38th Annual GITEX Technology Week Conference (Oct. 16, 2018), https://www.cftc.gov/pressroom/speechestestimony/opaquintenze16.

56. In re Zachary Coburn, Securities Act Release No. 84553 (Nov. 8, 2018), https://www.sec.gov/litigation/admin/2018/34-84553.pdf.
57. 545 U.S. 913, 918–19; 936–37 (2005) (holding that one who "distributes a device with the object of promoting its use to infringe copyright, as shown by clear expression or other affirmative steps taken to foster infringement, is liable for the resulting acts of infringement by third parties").
58. *See* Haseeb Qureshi, "*A Hacker Stole $31M of Ether—How it Happened, and What it Means for Ethereum*," *FreeCodeCamp* (July 20, 2017), https://medium.freecodecamp.org/a-hacker-stole-31m-of-ether-how-it-happened-and-what-it-means-for-ethereum-9e5dc29e33ce.

Stuart Levi
Tel: +1 212 735 2750 / Email: stuart.levi@skadden.com

Stuart D. Levi is co-head of Skadden's Intellectual Property and Technology Group, and he coordinates the firm's blockchain, outsourcing and privacy practices. Mr. Levi has a broad and diverse practice that includes outsourcing transactions, technology and intellectual property licensing, fintech and blockchain matters, privacy and cybersecurity advice, branding and distribution agreements, cloud computing agreements, technology transfers, strategic alliances and joint ventures. Mr. Levi also counsels clients on website and technology policies, intellectual property strategy and regulatory compliance. His background in computer science and the information technology industry allows Mr. Levi to understand the technology and business drivers underlying transactions and agreements in these areas.

Alex Lipton
Tel: +1 212 735 3006 / Email: alex.lipton@skadden.com

Alex is an Intellectual Property and Technology associate in Skadden's New York office. He earned his A.B. from Harvard University (2011) and J.D. from NYU School of Law (2016).

Cristina Vasile
Tel: +1 212 735 2247 / Email: cristina.vasile@skadden.com

Cristina is an Intellectual Property and Technology associate in Skadden's New York office. She earned her B.A. and M.A. from NYU (2008, 2009) and her J.D. from NYU School of Law (2016).

Skadden, Arps, Slate, Meagher & Flom LLP

4 Times Square, New York, New York 10036, USA
Tel: +1 212 735 3000 / URL: www.skadden.com