

What To Expect When Litigating Smart Contract Disputes

By **David Zaslow** (April 4, 2018, 5:11 PM EDT)

Interestingly, perhaps the most oft-used line about smart contracts is that they are neither "smart" nor "contracts." Be that as it may, smart contracts exist, and the future will inevitably bring more of them. It is therefore worth looking at issues — some of them unique — that will arise in litigating disputes relating to smart contracts and other developing technology.

The term "smart contract" has no settled definition. Discussions of smart contracts often refer to the soda vending machine as the first smart contract. A vending machine has hard coded rules that define what happens when certain conditions are met and then executes those actions upon the fulfillment of the conditions. In an 85-cent machine, if Alice deposits 85 cents, she will receive one can of soda; If Bob deposits one dollar, he will receive one can of soda and 15 cents change.



David Zaslow

At its most basic, that is what a smart contract does. It is computer code that defines and enforces a set of conditions ("if-then" instructions). Once a smart contract's conditions are met, its terms are enforced automatically, using a blockchain. As to the latter, again, on an overly simplistic level, blockchain is distributed ledger technology. Transactions are verified across a network of computers connected to the internet (each, a "node"). Each transaction block is transmitted to each node and must be verified. Once verified, blocks are added to the ledger and, once recorded on the blockchain in this way, a block is virtually impossible to change.

Following are the types of issues that will arise in litigating smart contract disputes.

Personal Jurisdiction and Choice of Law

Some issues that will be encountered with developing technologies will be new twists on old concepts. As an example, smart contracts run on blockchain technology and there are two characteristics of these technologies that will result in new applications of personal jurisdiction concepts. First, the transactions are often pseudonymous and there will accordingly be questions about who to sue. Second, by definition, blockchain is a "distributed" ledger. In the case of the most well-known blockchain, Bitcoin, there are thousands of full nodes spread around the world. What court will have personal jurisdiction over blockchain disputes? Likewise, what law will apply? Possible candidates are the location of the parties, the location of the servers and the location of the nodes that verified the transaction. In the world of "dumb" contracts, tricky issues like these are most commonly resolved by including choice of forum and choice of law clauses. It is not yet clear whether that will be a viable solution for smart contracts but that is certainly something every lawyer should consider.

Enforceability of Smart Contracts

Lawyers learn during the first year of law school that the basics of a contract are offer, acceptance and consideration. But in a smart contract, there is no offer and acceptance. Rather, there is agreed-on computer code that causes certain actions to be performed based on the results of "if-then" conditions. Some jurisdictions, such as Arizona and Tennessee, have legislated the legality of smart

contracts. For example, a law enacted in Tennessee on March 22, 2018, defines a smart contract as "an event-driven program, that runs on a distributed, decentralized, shared and replicated ledger and that can take custody over and instruct transfer of assets on that ledger," and states that "no contract relating to a transaction shall be denied legal effect, validity, or enforceability solely because that contract contains a smart contract term." But that is rare (so far) and courts have not yet determined whether smart contract code constitutes a binding contract, to which contract law principles will otherwise apply. Thus, if appropriate, it might help to include in the smart contract certain code that is no more than statements detailing that the agreement is for "such and such purpose."

Statute of Frauds

For those transactions to which the statute of frauds will apply, meaning that the agreement must be in writing, the question will arise as to whether code is sufficient to serve as a writing. Theoretically, because the purpose of the statute of frauds is to avoid the uncertainty of alleged oral contracts, written code certainly alleviates that problem. Indeed, as far back as 1893, the U.S. Supreme Court held that a contract assented to in a telegraphic cipher code satisfied the statute of frauds and was binding.[1] Furthermore, because smart contracts run on blockchain technology and will therefore require the use of private and public keys, those keys could satisfy the requirement that the contract be signed.

Issues Relating to Immutability

As already mentioned, a smart contract is code. It is immutable in the sense that, once the code starts to run, it cannot be changed. That fact will itself create novel issues because it is sometimes necessary for performance of a contract to stop, or to be changed. As just one example, in the case of fraud or misrepresentation, a party can be entitled to have the contract rescinded and to be returned to the status quo ante. There can also be a situation in which there is a repudiatory breach, which results in the counter-party no longer having to perform. How will that counterparty stop a smart contract?

As a more specific example, one of the use cases that is talked about for smart contracts and blockchain technologies is for real estate contracts. Once the conditions of the contract are satisfied, all necessary title and ownership is transferred to the buyer and registered on a title blockchain. However, it is possible for such a contract to be void as against public policy if, for example, when the buyer seeks to later sell the property, she sees that it includes a covenant that barred subsequent transfers to persons of a particular sexual orientation.

There are also situations in which contracts are not void ab initio but changed circumstances can alter performance. For example, the U.S. government has numerous different sanctions program. A common denominator in (virtually) all of them is that it becomes illegal to contract with the sanctioned parties. Thus, while performance of a particular smart contract between Alice and Bob might, initially, be completely unproblematic, if Alice suddenly becomes a sanctioned person, Bob is going to be legally obligated (under punishment of sanctions) to stop performing that smart contract.

In all of these situations, it is going to be necessary to figure out how to undo or modify the performance of a transaction that was (or is being) automatically performed by a smart contract.

Coding Errors

Another unique issue raised by the fact that a smart contract is actually computer code concerns liability. The assumption underlying a smart contract is that the code is written in a way that it will perfectly execute the intentions of the parties. But there is an inherent contradiction between that assumption and the reality that code is rarely perfect. Indeed, software engineers are not trained to write perfect code; it is expected that bugs will be identified and fixed. Consumers experience that all the time with new patches being issued regularly for our smart phones. Thus, it should not be surprising that a 2016 study of Ethereum smart contracts revealed at least 100 errors per 1,000 lines of code. Are the programmers who wrote the faulty code going to be the ones liable for the smart contract not performing as originally intended?

A more nuanced form of the liability question concerns the situation in which the code has a

vulnerability and an attacker exploits that vulnerability so that the smart contract performs differently than was intended, yet exactly the way it was programmed. This is not a mere abstraction; it happened with respect to one of the most talked about initial coin offerings —The DAO.

DAO stands for decentralized autonomous organization. It describes a “virtual” organization embodied in computer code and executed on a blockchain. The objective of The DAO was to operate a for-profit entity that would create and hold assets through the sale of DAO tokens to investors. All funds raised were to be held at an Ethereum Blockchain “address” associated with The DAO. DAO token holders were to vote on contract proposals, including proposals to The DAO to fund projects. They would also vote to distribute The DAO’s anticipated earnings from the projects it funded.

The holders of DAO tokens stood to share in the anticipated earnings from these projects as a return on their investment in DAO tokens. In addition, DAO token holders could monetize their investments in DAO tokens by reselling DAO tokens on a number of web-based platforms that supported secondary trading in the DAO tokens. After \$150 million of DAO tokens were sold, but before The DAO was able to commence funding projects, an attacker used a flaw in The DAO’s code to reroute to himself approximately one-third of The DAO’s assets.

In trying to figure out potential liability here, it is important to recognize that this was not a hack. Rather, somebody identified a vulnerability in the code and acted to exploit it. But the code performed exactly as it was programmed to perform. Thus, there was no breach of contract. There is indeed a perfectly sound legal argument to support the position of the attacker that, because he broke no laws, there would be liability on behalf of anyone who sought to prevent him from using the Ether that he collected as a result of the rerouting.

What if someone wanted to sue The DAO? There would immediately be a question of who one would sue because there is no person who represents The DAO. In addition, the terms of the contract under which the DAO tokens were sold (and they were included in various different “documents”) identified numerous risks in investing in The DAO and included broad limitations of liability. In situations like this, are the programmers the ones who should be liable?

In short, smart contracts raise liability issues that do not exist with more traditional contracts. There should be no question that issues like these will shortly reach the courts.

Arbitration

Despite there being much bad press recently about arbitration, based on the statistics of the major arbitral institutions, there continues to be a growing number of disputes that are resolved by arbitration. Many of the reasons often given for choosing arbitration would seem to apply to the resolution of smart contract and blockchain disputes.

First, one of the primary benefits of using arbitration in international disputes is that the forum is neutral and, thus, none of the parties feels that the others have a home-court advantage. As noted, blockchain transactions are often cross-border. In addition, there can be real questions about which court, or courts, will have personal jurisdiction over a dispute. Choosing arbitration can address these issues.

Second, smart contract technology is relatively new, evolving and, in many ways, highly technical. Few judges, if any, have had to deal with the technical issues presented by smart contracts. A known advantage of arbitration is that it gives parties the ability to choose the person who will resolve their dispute, thereby giving them the opportunity to select a person who is knowledgeable about the field and the underlying technology. This is, accordingly, another reason why arbitrating smart contract disputes would be preferable to having them decided in court. Moreover, it would be wise for parties to include in their arbitration clause specific language concerning the qualification of the arbitrator(s).

Confidentiality is another feature of arbitration that is often cited by in-house lawyers as one of its chief advantages over litigation. That is becoming even more important as courts, at least in the U.S., are making it more and more difficult for filings to be made under seal. It should be expected that there will be situations where parties to blockchain and smart contract disputes will want to maintain the confidentiality of their source code. Arbitration provides the possibility to do so. In such

situations, parties should make certain to include robust confidentiality provisions in their arbitration clause because it is incorrect to assume that all arbitral institutions, and all jurisdictions, provide the needed level of confidentiality based only on the rules of the arbitral institution.

Liability in Artificial Intelligence

Artificial intelligence is another of the developing technologies to which much digital ink is devoted every day. This article will limit itself to a discussion of just one issue and, more specifically, to one issue in what will soon be one of the most disruptive applications of AI — self-driving cars.

One of the most interesting discussions around autonomous vehicles was provoked by an article in Science magazine that looked at the ethical dilemmas posed by such vehicles.[2] Namely, how should such vehicles be programmed if forced to choose between killing passengers or pedestrians? These questions are, of course, simply a variation of the well-known "trolley" thought experiment in which a trolley is heading straight toward five people and will kill them, but a person has the opportunity to move the trolley to a side track, where there is one person on the track who will be killed. How many people would actually pull the switch to change tracks and thereby be the active cause of the death of a person? Paradoxically, in the study done with respect to the self-driving cars, although a majority of respondents acknowledged that there were circumstances under which the vehicle should be programmed in a way that would result in the death of the passenger, those respondents also said that they would prefer not to ride in such vehicles.

For our purposes, let us assume a situation in which there is one passenger in the car and the car finds itself in a circumstance in which three children are suddenly in front of the car. The car can continue straight, killing the three children or veer off to the side, where it would hit a wall and kill the passenger. Let us further assume that the car is programmed so that it results in the death of the passenger. One should expect the estate of the passenger to sue the car company. Should there, however, be any liability on the part of the company?

From the perspective of the car company, it was not negligent in any way. The vehicle performed exactly as it was programmed to do. Nor, under these circumstances, can one say that there was any product liability. Be reminded again that the study respondents said they would prefer not to ride a car that it is programmed in a way that will result in the death of the passenger. Might the law develop in such a way that there is an assumption of risk on the part of a passenger who rides in an autonomous vehicle which performs precisely as it was programmed to perform? That is not an easy question to answer and, until now, it was a question that did not have to be answered. This issue is, however, an example of liability issues that courts and legislatures will now have to consider in the context of products that operate with artificial intelligence.

Conclusion

It is often said that the law is always playing catch-up. In a sense, that it is inevitable. Law cannot possibly get ahead of technology. It must react to how new technology affects individuals and business. Indeed, one of the reasons that the SEC has been reluctant to regulate the initial coin offering (ICO) market is because so many people are arguing that the blockchain and ICO wave is similar to the internet wave and that government should not kill the industry in its infancy. That constituency is urging the government to follow the same, general "do no harm" attitude that government showed with respect to the internet.

Government regulation, however, will be just one piece of the puzzle. As described above, litigating smart contract disputes will raise different twists on standard issues. But it will also raise new issues that will be unique to these new and developing technologies. It will likely be evolutionary, rather than revolutionary. Relatedly, a follow-up to this article a year from now will probably answer some of the questions raised herein, yet also include a host of new ones.

David Zaslowsky is the chairman of the litigation department in the New York office of Baker McKenzie. He often litigates technology disputes.

The opinions expressed are those of the author(s) and do not necessarily reflect the views of the

firm, its clients, or Portfolio Media Inc., or any of its or their respective affiliates. This article is for general information purposes and is not intended to be and should not be taken as legal advice.

[1] See *Bibb v. Allen*, 149 U.S. 481 (1893)

[2] Jean-François Bonnefon, Azim Shariff, Iyad Rahwan, "The Social Dilemma of Autonomous Vehicles," *Science* (June 24, 2016).