All watched over by machines of loving grace: a critical look at smart contracts

Article (Accepted Version)

All Watched Over by Machines of Loving Grace: A Critical Look at Smart Contracts

Dr Andres Guadamuz[*]

Abstract

Smart contracts are coded parameters written into an immutable distributed ledger called a blockchain. There has been increasing legal interest in the application of these self-executing programs to conduct transactions. Most of the scholarly and practical analysis so far has been taken the claims of this technology being akin to a contract at face value, with legal analysis of contract formation, performance, and enforcement at the forefront of the debate. This article discusses that while smart contracts may pose some interesting legal questions, most of these are irrelevant, and smart contracts should be understood almost strictly from a technical perspective, and that any legal response is entirely dependent on the technical capabilities of the smart contract. The article proposes that smart contracts are not contracts for all practical purposes.

1. Introduction

During the 2018 FIFA World Cup in Russia, viewers in various countries were treated to a series of adverts during the half time break describing a fictional future powered by the "blockchain solution". The commercial[1] depicts a family home were all the appliances and devices "come alive" after the inhabitants leave the house, and these devices talk amongst each other about spending an exact amount of energy to complete their functions, and this amount is paid directly to the utility service provider through automated transactions.

This future envisions a network of connected Internet of Things (IoT) devices that are constantly performing transactions through smart contracts. These are coded parameters written into an immutable distributed ledger called a blockchain.[2] The idea behind smart contracts is to have seamless machine-to-machine (M2M) transactions that will remove the need for human interaction, or could be entered by humans to automatically enforce existing obligations. A smart contract advocate explains a potential future use:

> *"I could agree to a smart contract with a local library stating that if I do not return a book by a certain date (and the library scans a book as it is returned), the cost of the book will automatically be sent from my Bitcoin wallet to the library. In contrast, under a traditional contract, the library would send me a bill, which I could choose to not pay—in violation of the*

[*] Senior Lecturer in Intellectual Property Law at the University of Sussex.

[1] https://www.youtube.com/watch?v=fBYwWiGa2Do.

[2] For more about the blockchain, see: Guadamuz A and Marsden C, 'Blockchains and Bitcoin: Regulatory responses to cryptocurrencies' (2015) 20 *First Monday* http://dx.doi.org/10.5210/fm.v20i12.6198.

contract. In the extreme case, the library would resort to legal means to force me to pay."[3]

The future is already becoming a reality. In 2016 a couple translated their pre-nuptial agreement into an executable smart contract and uploaded it to the blockchain for it to be enforced automatically.[4] Thousands of people around the globe are using an implementation of smart contracts to play an online game called Cryptokitties.[5] French insurance company AXA is testing an automated flight compensation app that uses smart contracts.[6] There are proposals to include smart contracts in everything from taxes[7] to real estate.[8]

There has already been some interest from legal scholars and practitioners about the implications of smart contracts, particularly about their validity under existing contract law. While the existing literature has been doing an excellent job of tackling a few questions about cryptographic agreements, there appears to be a gap in the way we are looking at the phenomenon. This is because the majority of existing legal literature on the subject has been dealing with smart contracts at face value, namely treating them from a strictly contractual analysis, and therefore dealing with contractual issues such as validity and enforceability. While smart contracts may pose some interesting legal questions, it will be the contention of this article that most of these are irrelevant, at least from a contract law perspective, and smart contracts should be understood strictly from a technical perspective, and that any legal response is entirely dependent on the technical capabilities of the smart contract. While there could be legal issues, these are not as evident as one could envisage.

The article is divided in two parts. The first will describe the history, definition, and uses of smart contracts. While the phenomenon is starting to become more widespread and such a descriptive exercise may appear superfluous, there is considerable disparity in the understanding and application of the underlying technical solutions, and this results in often competing and contradictory definitions used. Blockchain technology in general, and smart contracts specifically, have been the subject of grandiose claims and considerable hype in recent years, so a meticulous look at the definitions and background is warranted, and this will hopefully prove useful even for those who are very familiar with the technology.

The second part of the article will try to look at smart contracts from a critical perspective. Having explored the definitions, it will be the objective of this article to make clear that thinking of smart contracts using traditional legal analysis is not only misguided, it may not be even possible. The very unique characteristic of the blockchain technology that supports many of these types of agreements make some legal questions

---

[3] Gulker M, 'Are Smart Contracts the Future of Fraud Prevention?' (2017) *American Institute for Economic Research* https://www.aier.org/article/are-smart-contracts-future-fraud-prevention.

[4] Del Castillo M, 'Prenup Built in Ethereum Smart Contract Rethinks Marriage Obligations' (2016) *Coindesk* https://www.coindesk.com/prenup-ethereum-marriage-obligations/.

[5] https://www.cryptokitties.co/.

[6] https://fizzy.axa/en-gb/.

[7] Rikken O, 'Blockchain Real Time Tax' (2017) *LinkedIn*, https://www.linkedin.com/pulse/blockchain-real-time-tax-olivier-rikken/.

[8] https://propy.com/.

irrelevant, as the contracts are often in the hands of actors that may be immune to legal scrutiny and enforcement. In the end, smart contracts are by nature designed to disrupt the legal profession, and also intend to bypass regulatory and judicial oversight, and if they work as intended, questions such as validity, contract formation, breach, error and even fraud may be moot. The article ends with the claim that for all practical purposes smart contracts are not contracts, and should mostly be considered as self-executing programs designed to automate operations between users.

2. Introducing smart contracts

2.1 Background

Can computers contract with one another? Can legal norms be transposed into computer code? The answer to these questions has been the subject of legal scrutiny for decades. The clue of the answer to the first question lies not only on issues of agency, legitimacy and contract formation, but the key is on the second question. The issue of translating norms into code, and the analysis of legal systems and autonomous artificial agents has been a well-discussed area of legal scholarship.[9] It is not the place of this paper to discuss this area in detail, but suffice it to say that through theoretical and practical research, it has been recognised that it would be possible to codify some legal expressions into machine-readable format.[10]

It is also important to point out that exaggerated claims about the capabilities of artificial intelligence and expert systems are also not new, and at some point, it was proposed that legal practitioners and judges would be using "law machines" to give advice and make decisions in areas they were unfamiliar with.[11]

Smart contracts can therefore be seen as a subset of the discussion on artificial intelligence and the law. At the very basic level, when we talk about smart contracts, we are dealing with the concept of autonomous systems designed to conduct transactions without human intervention, and even the humble vending machine could fall under the category of smart contracts.[12] With increases in computing power, it became clear that it would be economically advantageous to program computers to interface with one another in ways that traditionally would require two people to agree with. Some of these automated transactions could be contractual in nature.

The early discussion of autonomous contracts consisted of a framework for electronic contractual interaction with autonomous agents that would negotiate with each other

---

[9] Some works include Bing J and Harvold T, *Legal Decisions and Information Systems* (Universitetsforlaget; Henley on Thames 1977); Sartor G, *Artificial Intelligence and Law: Legal Philosophy and Legal Theory* (Tano 1993); Leith P, *Formalism in Al and Computer Science* (Ellis Horwood 1990); and Bourcier D, Bochereau L and Bourgine P, "Extracting legal knowledge by means of multilayer neural network. Application to municipal jurisprudence", in *Proceedings of the 3rd ICAIL*, Oxford (ACM 1991), 288.

[10] A good survey of early artificial intelligence law research can be found here: Rissland EL, 'Artificial intelligence and law: Stepping stones to a model of legal reasoning' (1990) 99:8 *Yale Law Journal* 1957-1981. See also: Governatori G and others, 'On Legal Contracts, Imperative and Declarative Smart Contracts, and Blockchain Systems' (2018) 26 *Artificial Intelligence and Law* 377.

[11] Susskind RE, 'Expert systems in law: A jurisprudential approach to artificial intelligence and legal reasoning' (1986) 49:2 *Modern Law Review* 168-194.

[12] Szabo N, *The Idea of Smart Contracts* (1997) https://bit.ly/2vh88ji.

based on pre-established parameters.[13] Theoretically, it was recognised at some point that electronic autonomous agents could conduct themselves as legal agents acquiring, and even violating norms.[14] The idea of self-executing contracts requires not only the writing of those norms into code, but also allowed some manner of autonomy in which an agent could follow orders, but also generate new obligations independently.[15]

A sophisticated corpus of research developed,[16] and this work was eventually translated into both practical application and legal recognition. The first step towards the later was the acceptance that contracts could be conducted electronically, it may be difficult to imagine nowadays, but at some point there had to be an evolution in the law to allow such agreements.[17] The main legislative recognition came in the Electronic Commerce Directive,[18] where Article 9 prompted member states to allow for the recognition of contracts conducted electronically, and to eliminate formalities that would interfere with these formats. Furthermore, case law in various jurisdictions started to recognise a variety of formats in the conclusion of electronic contracts.[19] Then followed increasing doctrinal recognition that there was considerable legal basis for contract formation and performance by autonomous agents.[20]

The main practical uses for autonomous agents were in the shape of Electronic Data Interchange (EDI) agreements,[21] which were a set of protocols designed to allow autonomous agents to transact with one another digitally by placing orders in business supply-chains. These agreements included complex terms:

> *"The ordering, delivery, and payment for such supplies means that there are contractual terms surrounding the transaction-the time of delivery, what to do if the supplies do not arrive in time or are defective, what to do*

---

[13] Sallé M, 'Electronic contract framework for contractual agents', in *Conference of the Canadian Society for Computational Studies of Intelligence,* (Springer 2002) 349-353.

[14] Conte R, Falcone R and Sartor G, 'Introduction: Agents and Norms: How to fill the gap?' 7:*1Artificial Intelligence and Law* 1-15 (1999).

[15] Boella G, and van der Torre L, 'Contracts as legal institutions in organizations of autonomous agents' in *Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems* (IEEE Computer Society 2004) 948-955; and F Dignum, 'Autonomous agents with norms (1999) 7:1 *Artificial Intelligence and Law* 69-79.

[16] See: Dignum V, Meyer J, and Weigand H, 'Towards an organizational model for agent societies using contracts', in *Proceedings of the first international joint conference on Autonomous agents and multiagent systems: part 2* (ACM 2002) 694-695.

[17] For more on this subject, see: MacQueen HL, 'Software Transactions and Contract Law', in Edwards L and Waelde C (eds), *Law and the Internet: Regulating Cyberspace* (Hart Publishing 1997); Kidd DL and Daugherty WH, 'Adapting Contract Law to Electronic Contracts: Overview and Suggestions' (2000) 26 *Rutgers Computer & Technology Law Journal* 215.

[18] Directive 2000/31/EC of the European Parliament and of the Council of 8 June 2000 on certain legal aspects of information society services, in particular electronic commerce, in the Internal Market [2000] OJ L 178/1.

[19] For example: contracts could be conducted by fax in *Merrick Homes Ltd v Duff* 1996 SC 497, just to give one example.

[20] Most notable, the elegant use of Roman Law to describe autonomous agents here: Weitzenböck EM, 'Good faith and fair dealing in contracts formed and performed by electronic agents' (2004) 12 *Artificial Intelligence and Law* 83.

[21] Radin MJ, 'Humans, Computers, and Binding Commitment' (2000) 75 *Indiana Law Journal* 38.

> *if the payment is late, and all the other transactional parameters that people contract about. All of this can in principle be handled primarily by machine, using computer programs that "negotiate" with each other and enter into "agreements" with each other."*[22]

This description of the functions performed by earlier electronic contracts is interesting because in many ways it resembles some of the functions that have been hyped as new and novel by smart contracts. What is different then?

Using the most basic definition, a smart contract is "an agreement whose execution is automated" which is "effected through a computer running code that has translated legal prose into an executable program".[23] This definition makes it no different than an EDI agreement, or other early variations of machine-implemented contracts. The modern implementation has the main difference that it involves dynamic cryptographic elements. While earlier electronic agreements used cryptography in the shape of digital signatures and other authentication mechanisms, the term "smart contract" was initially coined to differentiate them from EDI agreements, which were considered static, while the 'smart' variety would be dynamic and proactive by the incorporation of more sophisticated protocols.[24]

However, those traditional definitions have been superseded by the addition of other elements, particularly dynamic cryptographic elements allowing for secure and tamper-proof transactions, what some call "smart contract code"[25] or even "strong smart contracts".[26] This article will continue to use the term smart contracts to mean the more recent variation that uses cryptography, namely these strong smart contracts.

At the most basic form, a smart contract is an "if-then" statement that runs on the blockchain where "*parties can enter into a binding commercial relationship, either entirely or partially memorialized using code, and use software to manage contractual performance*."[27]

There have not been many attempts to define smart contracts in the law,[28] but there have been a few efforts to recognise the validity of smart contracts.[29] The first US state to legislate on the subject was Arizona, which defines smart contracts as follows:

---

[22] Ibid, p.1131.

[23] Raskin M, 'Law and Legality of Smart Contracts' (2017) 1 *Georgia Law Technology Review* 305, p.309.

[24] Szabo N, 'Formalizing and Securing Relationships on Public Networks' (1997) 2 *First Monday* https://doi.org/10.5210/fm.v2i9.548.

[25] Stark J, 'Making Sense of Blockchain Smart Contracts' (2016) *Coindesk* https://www.coindesk.com/making-sense-smart-contracts/.

[26] Raskin, supra note 23 at 310.

[27] De Filippi P and Wright A, *Blockchain and the Law: The Rule of Code* (Harvard University Press 2018), p.74.

[28] The European Parliament recently made a statement about blockchains and smart contracts, but it does not define the terms. See: European Parliament resolution of 3 October 2018 on distributed ledger technologies and blockchains: building trust with disintermediation (2017/2772(RSP)).

[29] A list of US state legislation dealing with blockchains and smart contracts can be found here: https://bit.ly/2L4Oksc.

> *"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."*[30]

The intention of the law is to specifically recognise the use of smart contracts defined in this manner in the conclusion of contracts, but also allows their use in commerce. Art 5(c) reads:

> *"Smart contracts may exist in commerce. A contract relating to a transaction may not be denied legal effect, validity or enforceability solely because that contract contains a smart contract term."*[31]

It is interesting that this same language, or similar variations, can be found in a few other legislative texts.[32] For example, the New York legislature has passed a definition of smart contracts that is exactly the same as the one from Arizona.[33] It would appear that in the rush to regulate on this subject, legislators across the United States in particular have been using the same legal definition. It must be pointed out that at the time of writing no other country has attempted to legislate on smart contracts, so the existing legal definition is the one presented above.

As the Arizona law is being used as a template for others, this definition requires some unpacking, as it lacks some clarity, but also it is rather restrictive, and could be leaving out several types of smart contracts. The definition contains three main elements:

a) *A smart contract means an event-driven program, with state.*

This means that a smart contract must be a computer program that follows a series of events and that it is designed to remember preceding instructions or user interactions (stateful). This means that these documents start as lines of computer code written in a language capable of making "IF-THIS-THEN-THAT" statements that can express legal concepts, but they also can identify more complex norms. The contract has to be deterministic in the sense that the same input will always produce the same result, and the contract should also contain all eventualities arising from the contract expressed in code.[34] Any computer language can be used to code a contract,[35] but there are languages that are being designed specifically for smart contracts,[36] and a particular requirement favoured by many developers is that the language should be Turing complete, that is, that every computing problem can be solved exactly or approximately by using the same

---

[30] Arizona House Bill 2417 (2017).

[31] Ibid.

[32] See for example Nebraska LB695, and Ohio SB300**.**

[33] New York A08780.

[34] Christidis K and Devetsikiotis M, 'Blockchains and Smart Contracts for the Internet of Things' (2016) 4 *IEEE Access* 2292.

[35] There is extensive literature on ontological approaches to translate legal concept into code, for example: Sartor G and others, 'Computable Models of the Law and ICT: State of the Art and Trends in European Research' in Pompeu Casanovas and others (eds), *Computable Models of the Law* (Springer Berlin Heidelberg 2008); and Han ZZ and others, 'Interoperability from Electronic Commerce to Litigation Using XML Rules' (2007) 15 *International Journal of Law and Information Technology* 233.

[36] Kim HM and Laskowski M, 'Toward an Ontology-Driven Blockchain Design for Supply-Chain Provenance' (2018) 25 *Intelligent Systems in Accounting, Finance and Management* 18.

language.[37] At the time of writing, the most popular language is Solidity,[38] with other bespoke languages such as Michelson, Ivy, Hoon, and Rust in existence. Other projects favour mainstream languages such as C++ and Javascript.

b) *The program runs on a distributed, decentralized, shared and replicated ledger.*

This means that the smart contract is written and runs on a *blockchain*. While deceivingly simple, this part of the definition is the one that requires a more thorough explanation. Countless pages have been written on the blockchain,[39] but for something that is so hyped, there is little consensus on what a blockchain actually is, and this confusion could have important legal effects. One of the most accepted definitions is that a blockchain is a public decentralised permissionless[40] cryptographic database that operates as an open ledger of all of the transactions that have been recorded, and that this record is immutable[41] and tamper-proof,[42] so that it "can be inspected by every network participant".[43]

In more technical terms, the first definition was written in the Bitcoin white paper by Satoshi Nakamoto,[44] which describes a cryptographic coin that is comprised of a chain of digital signatures that are added to a block and time-stamped and appended at the end of the chain; so all transactions are publicly available and verified, but also the system is tamper-free because to change a transaction you would also need to change all the transactions that came before the one you are trying to verify. These verifications are performed by people running the calculations, which is called 'mining'. This system is what is known as *proof of work*. Proof of work protects the blockchain from attackers and spammers because it requires those participating to perform some work in the shape of computing time. In Bitcoin, the proof of work consists in calculating numbers until the correct solution is found and then a new block is added to the blockchain.

This definition is widely accepted as the most authoritative one, but it comes with a few limitations.[45] Proof of work means that considerable amount of computing power is needed to operate the system and allocate blocks to miners, which means that the more transactions there are, the slower the system becomes because of the need to conduct verifications. This also means that proof of work can be are extremely energy wasteful by design, as the difficulty of finding a solution increases over time.

---

[37] See: Teller A, 'Turing Completeness in the Language of Genetic Programming with Indexed Memory', *Proceedings of the First IEEE Conference on Evolutionary Computation. IEEE World Congress on Computational Intelligence* (IEEE 1994).

[38] This is mostly based on general perception, it is difficult to gather statistics.

[39] For an excellent sceptical look, see: Gerard D, *Attack of the 50 Foot Blockchain: Bitcoin, Blockchain, Ethereum & Smart Contracts* (Self 2017).

[40] This means that anyone can maintain the database, and no authentication to do so is required, see: 'Blockchain and Voting' (*Benlog*, 28 December 2017) https://benlog.com/2017/12/28/blockchain-and-voting/.

[41] However, the immutable nature of blockchains is disputed, as it will be discussed later in the article.

[42] Narayanan A and others, *Bitcoin and Cryptocurrency Technologies: A Comprehensive Introduction* (Princeton University Press 2016), p.27.

[43] Christidis and Devetsikiotis, supra note 34.

[44] Nakamoto S, *Bitcoin: A Peer-to-Peer Electronic Cash System* (2007), https://bitcoin.org/bitcoin.pdf.

[45] See Gerard, supra note 39.

So other blockchain solutions have arisen to make transactions faster and to use less power, which further complicates the definition of what is a blockchain. One such solution is to use a system called a *proof of stake*[46] blockchain, which chooses the allocation of the next block between those with a stake in the system without the need for large expenditure of resources. Similarly, there are private blockchains, where the transactions are not open to public scrutiny. Another type is a permissioned blockchain,[47] where you need permission to access and modify the data, as opposed to the standard defined above where anyone can have access and add to the blockchain; these tend to be also private as the ledger is actually not open nor transparent.

The result of this confusion is a situation in which anything can be called a blockchain, even if it does not meet the traditional definition set out by Nakamoto.[48] In fact, the legislation defining smart contracts also comes with a legal definition of what is a blockchain, and this takes a broad approach. The Arizona law (and others) read:

> *"Blockchain technology" means 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."*[49]

This can include almost anything to be a blockchain, and does away with some of the specific requirements set out by Nakamoto. As it can be private and permissioned it is not an open ledger, and therefore it need not be distributed either.[50] This means that a legal smart contract would just be a private program running in a private network, which seems to defeat the purpose of the technology.

This legal definition of blockchain contains a couple of problematic statements. The first is that the blockchain contains data that is immutable; while this is accurate for the most part, there are various scenarios in which information could be changed provided a majority of miners decide to change the record.[51] A more accurate description is that a

---

[46] King S and Nadal S, *Ppcoin: Peer-to-peer crypto-currency with proof-of-stake* (2012), http://blockchainlab.com/pdf/peercoin-paper.pdf.

[47] Kadiyala A, 'Nuances Between Permissionless and Permissioned Blockchains' (*Medium*, 17 February 2018), https://medium.com/@akadiyala/nuances-between-permissionless-and-permissioned-blockchains-f5b566f5d483.

[48] Jeffries A, '"Blockchain" Is Meaningless' (*The Verge*, 7 March 2018) https://www.theverge.com/2018/3/7/17091766/blockchain-bitcoin-ethereum-cryptocurrency-meaning.

[49] Arizona House Bill 2417 (2017).

[50] In fact, there are so-called blockchains out there with only one user: Gerard D, 'The World Food Programme's much-publicised "blockchain" has one participant — i.e., it's a database' (2017) *Attack of the 50 Foot Blockchain Blog*, https://davidgerard.co.uk/blockchain/2017/11/26/the-world-food-programmes-much-publicised-blockchain-has-one-participant-i-e-its-a-database/.

[51] Eyal I and Sirer EG, 'Majority Is Not Enough: Bitcoin Mining Is Vulnerable' (2018) 61 Communications of the ACM 95.

blockchain is "inherently resistant to data modification".[52] Secondly, the idea that this is a ledger that is protected with cryptography is a given, but the requirement that it provides an immutable version of "uncensored truth" seems not only problematic, but highly controversial. The problem is that treating information contained on the blockchain as "immutable truth" is quite simply not correct, the distributed ledger only acts as a record of the data that was entered into it, and cannot be used as "uncensored truth". It is true that some information could be authoritative, such as time stamps, or to make a permanent record of a transaction, but having a record of a transaction does not tell us anything about the veracity of the information recorded into the system.

To complicate matters, the legal definition in legislation such as the Arizona law require that the smart contract must be run on the blockchain. While this is not specified, this very well means that we need a platform that can execute the contract, and this is usually tied to the blockchain where the contract is written. Anyone can create and run a blockchain,[53] so in theory any computer can act as a platform for smart contracts, just in the same way that you could turn your computer into a web server. However, this is inefficient, and it is better to rely on existing infrastructure. There are various platforms available for writing smart contracts, some are designed specifically for that task, such as Ethereum,[54] and others are using existing infrastructure such as the Bitcoin blockchain and cryptocurrency.[55]

A relevant consideration when looking at platforms is whether it is centralised or decentralised, as well as the governance structure that it has. A platform are said to be decentralised if there is no central authority maintaining the infrastructure that runs the contracts, this is possible with the implementation of a decentralised autonomous organisation (DAO) that executes transactions and releases funds automatically, without the need of a central controlling body.[56] The governance structure is the decision-making procedure for who has control over the platform's code, who can make changes to it, and how the platform is operated. There are various degrees of centralisation in this stage, with some platforms presenting considerable centralisation at this level.

c) *The smart contract can take custody over and instruct transfer of assets on that ledger.*

The final element present in the legislative definition of smart contracts is that they can be used to transfer assets on the blockchain. This is perhaps where the definition leaves out many other useful functions for smart contracts. It may be better to think here of the management of assets as using *tokens* for various operations needed by the contract. A token is usually a part of the contract that does not fulfil an IF-THEN function, but it is instrumental to the operation of the code.[57] In other words, a token is simply code that

---

[52] Müller L et al, *Conceptual Framework for Legal and Risk Assessment of Crypto Tokens*, MME Report (2018), https://www.mme.ch/fileadmin/files/documents/180501_BCP_Framework_for_Assessment_of_Crypto_Tokens_-_Block_2.pdf.

[53] Saurel S, 'Create Your Own Blockchain in 30 Minutes – (*Medium,* January 22 2018) https://medium.com/@ssaurel/create-your-own-blockchain-in-30-minutes-dbde3293b390.

[54] https://www.ethereum.org/.

[55] A list of platforms can be found here: https://hackernoon.com/contractpedia-an-encyclopedia-of-40-smart-contract-platforms-4867f66da1e5.

[56] The concept was introduced here: Buterin V, *A next-generation smart contract and decentralized application platform* (2014) https://cryptorating.eu/whitepapers/Ethereum/Ethereum_white_paper.pdf.

[57] More about tokens in Müller L, supra note 52.

represents fungible goods[58] for trade, and these can be "coins, shares, outcomes or tickets, or everything else which is transferable and countable".[59]

The function of tokens will be entirely dependant on the needs of the contract, but some platforms have specific uses for tokens. In the Ethereum platform, there are tokens that can be used to make a payment for each transaction (usage token), and tokens that identify participation or ownership in a contract (work tokens).[60] We can have ownership tokens, identity tokens, and even tokens representing votes for governance purposes.

There are several problems with this as a workable legal definition. The main issue is that while the language attempts to be neutral, it is evident that this refers to a specific type of contracts that depend on transactions expressed on a blockchain, and ignores some of the most salient features of smart contracts. The main problem is that this definition is designed specifically with a contract that is used to transfer assets on a distributed ledger. As we will see in the next section, some smart contracts are indeed being used for transacting goods, but there are many other uses that do not fall under this definition, particularly because of the narrow specification of assets capable of being transacted.

So the definition presents a conundrum, it is both too narrow and too broad. Because it requires smart contracts to be about the transfer of assets, it appears to be only preoccupied with a narrow band of contracts that are used to perform these transactions. But these transactions have to be recorded in a blockchain, and the definition of a blockchain is extremely broad as to allow any sort of cryptographic database to qualify. So smart contracts, from a legal perspective, are only those contracts that transfer assets stored on any cryptographic ledger.

2.2 Smart contracts in practice

The advantages of automated transactions should be self-evident, proven by the existence of these types of agents for decades. However, the usefulness of the strong version of smart contracts described above may not be as apparent. Some of the interest that has arisen from the agreements written into the blockchain is that the proponents see it as an important development that will bring efficiency, transparency, and decentralisation to electronic contracting.

A smart contract would have several capabilities that make it theoretically attractive for users and businesses: the contracts are immutable, secure, provide tamper-proof evidence, allow seamless execution, can allow immediate transfer of funds, do not require a central authority, and are open to scrutiny. Another element that is present in smart contracts is that they provide a trustless way of conducting transactions; this is because the decentralised nature of the contract does away with the need to have trusted intermediaries to the conclusion of a transaction.[61] Proponents of smart contracts see

---

[58] Fungible goods are of goods that are able to replace or be replaced by another identical item; and are mutually interchangeable.

[59] Bartoletti M and Pompianu L, 'An Empirical Analysis of Smart Contracts: Platforms, Applications, and Design Patterns' in Michael Brenner and others (eds), *Financial Cryptography and Data Security* (Springer International Publishing 2017), p10.

[60] 'What is An Ethereum Token: The Ultimate Beginner's Guide' (2018) *BlockGeeks* https://blockgeeks.com/guides/ethereum-token/.

[61] Clack CD, Bakshi VA and Braine L, 'Smart Contract Templates: Foundations, Design Landscape and Research Directions' (2016) *arXiv* 1608.00771 [cs.CY] https://arxiv.org/abs/1608.00771.

them as a way to automate large number of legal functions, eventually disrupting the legal profession,[62] and some even argue that they could bring about the end of government altogether.[63]

Anything can be automated in this manner, take this example of a potential use:

> *"Let's say that we want to organize a small conference. We need 100 people to sign up and pay/deposit money, so we can rent a hotel and such. But if not enough people sign up by a certain date, then the deposits need to be refunded. With Ethereum, we can write in a JavaScript-like language to code up this contract. It'll guarantee that everyone will get a ticket to the conference, or everyone will get their money refunded, depending on how many sign up."*[64]

Similar potential applications are countless. Because the basic tools for writing a smart contract are mostly open source[65] and free to use, anyone can learn to code a contract into software, and have it written into a blockchain. The performance of the contract is automated, and no human intervention is necessary.

However, when we talk about smart contracts, the "contract" word is not used in the strictest legal sense. We will discuss this in more detail later, but in theory anything that can be expressed in IF-THEN statements can be turned into a strong smart contract, and these documents may or may not fall under the legal definition of a contracts. So for example, you can have an executable file to express eternal love in the blockchain,[66] creating a credit association,[67] and even blatant ponzi schemes.[68] Given the decentralised nature of the development environment, anything goes.

While the flexible nature of smart contracts is one of its most attractive features, there is no recognised classification of such contracts for legal and regulatory purposes. We propose the following broad classification:

Machine-to-machine transactions: As described in the commercial mentioned in the introduction, one of the biggest potentials for smart contracts is to implement transactions between machines in "a web of tiny services that enable the creation and autonomous activity of very complex systems."[69] Anything that does not require human intervention, such as automated ordering systems, supply-chain distribution and

---

[62] Davies A (ed), *Blockchain and the Legal Profession* (Ark Group 2018).

[63] Atzori M, 'Blockchain Technology and Decentralized Governance: Is the State Still Necessary?' (2015) *SSRN Scholarly Paper* ID 2709713 https://papers.ssrn.com/abstract=2709713.

[64] Graham R, "Ethereum/TheDAO hack simplified" (June 18, 2016) *Errata Security Blog* https://blog.erratasec.com/2016/06/etheriumdao-hack-similfied.html.

[65] For more on open source software, see: Guadamuz A, 'Free and Open Source Software' in Lilian Edwards and Charlotte Waelde (eds), *Law and the Internet* (3rd edition, Hart Publishing 2008).

[66] https://www.stateofthedapps.com/dapps/eternal-love.

[67] https://www.stateofthedapps.com/dapps/wetrust.

[68] https://www.stateofthedapps.com/dapps/333-eth.

[69] Glaser F, 'Pervasive Decentralisation of Digital Infrastructures: A Framework for Blockchain Enabled System and Use Case Analysis' (2017) *Hawaii International Conference on System Sciences 2017* (HICSS-50).

tracking, and the management of IoT devices,[70] could be performed through smart contracts. While many of these transactions do not require strong smart contracts,[71] it has been postulated that the addition of open distributed ledgers enhances the potential of networked transactions as all records are public, and therefore parties in a supply chain do not need to maintain separate databases, but only one.[72]

Cryptocurrency: While this function is not exclusive of smart contracts, platforms such as Ethereum offer the possibility of users to create and deploy their own cryptocurrency. Through the use of a smart contract that contains a token,[73] this can be distributed as a "coin", a tradeable and fungible currency.

Crowdfunding: Related to the use of tokens as coins, a popular development in smart contracts has been their use to crowdfund projects and startups, this is done through what is known as an initial coin offering (ICO).[74] An ICO is usually a token offered to members of the public by the entity interested in raising funds, and these pay for the token in return for some sort of participation in the project, be it by a licence to use software, a claim to an underlying asset, or a promise of participation in future profits.[75]

Governance: It is possible to use smart contracts to create an automated governance scheme that is run, organised, and enforced through code in what is known as a decentralised autonomous organisation (DAO).[76] There is no set model for a DAO, but the understanding is that all aspects of an organisation are coded into the smart contract, from financial transactions, to voting, and governance mechanisms.[77]

Decentralised applications: Better known as ĐApps, decentralised applications are pieces of software that are stored on the blockchain and run using smart contracts and tokens.[78] The applications are stored around the world in the computers connected to the platform's nodes.

---

[70] Huckle S and others, 'Internet of Things, Blockchain and Shared Economy Applications' (2016) 98 *Procedia Computer Science* 461.

[71] For example, supply chain automation has been in place for decades, see: Johnson ME and Whang S, 'E-Business and Supply Chain Management: An Overview and Framework' (2002) 11 *Production and Operations Management* 413.

[72] Christidis and Devetsikiotis, supra note 34.

[73] https://www.ethereum.org/token.

[74] Rohr J and Wright A, 'Blockchain-Based Token Sales, Initial Coin Offerings, and the Democratization of Public Capital Markets' (2017) *SSRN Scholarly Paper* ID 3048104 https://papers.ssrn.com/abstract=3048104.

[75] Zetzsche DA et al, 'The ICO Gold Rush: It's a Scam, It's a Bubble, It's a Super Challenge for Regulators' (2017) *UNSW Law Research Paper* No. 83 https://ssrn.com/abstract=3072298.

[76] Chohan UW, 'The Decentralized Autonomous Organization and Governance Issues' (2017) *SSRN Scholarly Paper* ID 3082055 https://papers.ssrn.com/abstract=3082055.

[77] Norta A, 'Creation of Smart-Contracting Collaborations for Decentralized Autonomous Organizations' in *Proceeding of the Perspectives in Business Informatics Research: 14th International Conference* (BIR 2015), pp 3-17.

[78] See an explanation here: https://www.stateofthedapps.com/whats-a-dapp.

Rights management: An interesting proposal for smart contracts would be in the allocation of management of rights, for example, in copyright.[79] "Authors can publish works on the blockchain and then "use smart contracts to automate the control of who has access to their works and under which conditions"[80] and to obtain remuneration. There are various schemes doing this type of thing, an interesting one is SuperRare, which uses a smart contract platform to allow artist to release limited-edition artworks "tracked on the blockchain, making the art rare, verified and collectible."[81]

Registries: By using a blockchain as a method to record data in an immutable manner, there are proposals to use smart contracts to power registration systems.[82] Various systems that require registration services have been suggested as test cases for the technology, including land registries,[83] copyright registries,[84] and even the issuing of identity tokens.[85] These registries are different to other smart contracts because they rely on a registration authority, which make them more centralised than most other suggested implementation.

Dispute resolution: Finally, smart contracts could be used to intermediate disputes by providing an automated way to solve differences through code.[86] Online dispute resolution[87] already uses automated means to bring parties together, and by adding smart contracts written into a distributed ledger that allows immediate execution could be a way to mechanize many legal conflicts. The question remains of what happens when there is a conflict on the terms that are written into the contract, but that will be dealt with in the next section.

These are just some of the main categories of possible uses for smart contracts. Readers may see that many of these proposed uses are in early stage of planning and have not been fully deployed. This is a common characteristic of much of the literature dealing with blockchains in general, and smart contracts are no exception. While there are thousands of examples of smart contracts online, mainstream implementation remains minimal at the time of writing. To illustrate the point, A curated list of nearly three thousand ĐApps shows just over 75 thousand transactions per day, which is an average of 25 transactions per decentralised app. So, while the potential remains great, it is justified to question the relative low adoption levels.

---

[79] Bodó B, Gervais D and Quintais JP, 'Blockchain and Smart Contracts: The Missing Link in Copyright Licensing?' (2018) 26:4 *International Journal of Law and Information Technology*.

[80] Ibid.

[81] https://superrare.co/about.

[82] Tran AB et al, 'Regerator: A Registry Generator for Blockchain', *CAiSE-Forum-DC* (2017).

[83] Rizzo P, 'Sweden Tests Blockchain Smart Contracts for Land Registry' (2016) *Coinbase*, https://www.coindesk.com/sweden-blockchain-smart-contracts-land-registry/.

[84] Bodó, Gervais and Quintais, supra note 79.

[85] Dunphy P and Petitcolas FAP, 'A First Look at Identity Management Schemes on the Blockchain' (2018) *arXiv* 1801.03294 [cs] http://arxiv.org/abs/1801.03294.

[86] Koulu R, 'Blockchains and Online Dispute Resolution: Smart Contracts as an Alternative to Enforcement' (2016) 13 *SCRIPTed* 40.

[87] Katsh EE, Katsh ME and Rifkin J, *Online Dispute Resolution: Resolving Conflicts in Cyberspace* (John Wiley & Sons, Inc 2001), p34.

Generally speaking, smart contracts run on *proof of work* blockchains, and as it was discussed earlier these can be resource-heavy, difficult to setup, slow, and expensive.[88] Another reason could be that smart contracts are still shrouded in legal uncertainty. The next section will look at the legal issues.

3. A critical analysis of smart contracts

3.1 Smart contracts are not contracts

The title of this section is purposefully controversial for a reason. It is at this point of the conversation that most articles dealing with smart contracts will move to deal with a legal aspect of the technology, assuming that smart contracts should be treated using similar rules that we use to analyse "dumb" analogue contracts. Therefore, we have legal analysis of aspects such as contract formation,[89] interpretation,[90] and force majeure,[91] just to name a few. The growing legal literature dealing with the technology has been covering almost every aspect of the phenomenon,[92] and the main common denominator is to conclude that for the most part smart contracts are indeed binding legal agreements. While some researchers identify concerns and potential uncertainties with the application of existing law to the phenomenon,[93] others tend to be more optimistic, and herald a future where parties need not bother with monitoring execution and performance of the contract, as the machine will take care of everything.[94]

However, there is a clear trend towards increasing scepticism of the application of traditional contract law to smart contracts. While some commentators have either been enthusiastic or neutral,[95] we can identify two sceptical positions when it comes to the technology. On the one hand, some argue that many of the apparent challenges by the technology are already addressed in contract law,[96] on the other hand, there are those who see smart contracts as not being related to traditional contracts at all.[97]

Nonetheless, with increasing question marks over the technology coming from existing analysis, it seems evident that the current legislative approach has been to take smart contracts at face value. After all, the few laws dealing with distributed cryptographic

---

[88] For a wider discussion of some issues, see: Guadamuz A, 'Is the blockchain hype over?' (2017) *Technollama Blog* https://www.technollama.co.uk/is-the-blockchain-hype-over.

[89] Durovic M and Janssen A, 'The Formation of Blockchain-Based Smart Contracts in the Light of Contract Law' (2018) 26 European Review of Private Law 753.

[90] Cannarsa M, 'Interpretation of Contracts and Smart Contracts: Smart Interpretation or Interpretation of Smart Contracts?' (2018) 26 European Review of Private Law 773.

[91] Tai ETT, 'Force Majeure and Excuses in Smart Contracts' (2018) 26 European Review of Private Law 787.

[92] Caria RD, 'The Legal Meaning of Smart Contracts' (2018) 26 European Review of Private Law 731.

[93] Giancaspro M, 'Is a "Smart Contract" Really a Smart Idea? Insights from a Legal Perspective' (2017) 33 Computer Law & Security Review 825.

[94] De Filippi and Wright, supra note 27 at 81.

[95] Millard C, 'Blockchain and Law: Incompatible Codes?' (2018) 34 Computer Law & Security Review 843.

[96] Rohr JG, 'Smart Contracts and Traditional Contract Law, or: The Law of the Vending Machine' (2019) 67 Cleveland State Law Review 71.

[97] Millard C, 'Blockchain and Law: Incompatible Codes?' (2018) 34 Computer Law & Security Review 843.

agreements specifically declare that a contract should not be declared invalid just because it is a smart contract.

There are many reasons why smart contracts are not contracts, and we should start treating them as something else. The problem is that the technology, as it currently exists, has features that make it incompatible with what we understand as a contract, to the extent that it may be impractical to even consider them in this light. This is not to mean that they cannot be contracts, ostensibly there are already non-cryptographic smart contracts in existence, [98] but that for all intents and purposes they should not be.

The first challenge is that, as a matter of fact, many so-called smart contracts are anything but, and the intention of the drafting parties is not to enter into a contract. While we have set out a few commonalities to what is considered a smart contract, the distributed nature of the developing environment means that there is little or no standardisation, and therefore there may be a very large number of self-executing code written in a blockchain that may not be intended to have a binding nature in the legal sense.

Given the technical definition that is at the centre of this article, namely that a smart contract is a set of instructions written in a programming language into a blockchain, then some of these instructions could fit under the definition of a contract if the intention fulfils contractual formation formalities, but some may not. There are various uses for the technology that are clearly not intended as such, and their functionality is to provide a non-contractual service. For example, some existing code fulfils mere functions, such as time-stamping, notarising documents, creating a token, and creating a random number generator, just to name a few.[99] On the other hand, some code is clearly designed to act as a contract by describing something akin to legal obligations, such as using code to express a software licence,[100] or an agreement to charge fees.[101] The intention of the parties, and the way the code is written, will be the key of whether the code should be read as a legal contract. One could argue that all strong smart contracts are contracts in the legal sense as they set out enforceable obligations in code,[102] but this would lead to giving validity to electronic documents that are not intended to have that effect, so we will assume that each contract has to be analysed on a case by case basis.

Even if the contract is intended by the parties to be akin to a traditional paper contract, there is a problem with the assumption that all types of agreements can be conducted in a cryptographically secure manner. This is perhaps the most salient problem with agreements that that are connected to tangible goods or represent real services. Strong smart contracts operate better in a fully digital environment, where cryptocurrency funds can be immediately released as payment in exchange for digital goods or services, made possible through the tokenisation of items. But this becomes extremely difficult when real goods and services are involved, because the self-executing characteristic of

---

[98] Sandholm T, Lesser VR, 'Issues in automated negotiation and electronic commerce: Extending the contract net framework' (1995) *Proceedings of the First International Conference on Multiagent Systems* 12.

[99] Some of these functional uses can be found here: https://ethereum.stackexchange.com/questions/2940/where-can-i-find-some-solidity-smart-contract-source-code-examples

[100] See: https://www.apriorit.com/dev-blog/557-ethereum-smart-contract-licensing.

[101] See: https://github.com/ethereum/dapp-bin/tree/master/standardized_contract_apis.

[102] While not specified, this broad view can be implied here: Geiregat S, 'Cryptocurrencies Are (Smart) Contracts' (2018) 34 *Computer Law & Security Review* 1144.

smart contracts, which is one of its most important selling points, can only work if there is a direct link between the physical goods and the digital code.

Take one of the advertised uses for smart contracts, their use in transfer of property. In principle, the use of a smart contract for such an operation seems appealing, as it would provide an immutable record of the transaction, but the problem is precisely that the contract does not know whether there has been an actual transfer of property, it just operates with the data that has been entered into the blockchain, the person making the input could lie. So, we are back to trusting an intermediary, which sort of defeats the purpose. Developer Jimmy Song describes this problem:

> *"There is an intractable problem in linking a digital to a physical asset whether it be fruit, cars or houses at least in a decentralized context. Physical assets are regulated by the jurisdiction you happen to be in and this means they are in a sense trusting something in addition to the smart contract you've created. This means that possession in a smart contract doesn't necessarily mean possession in the real world and suffers from the same trust problem as normal contracts. A smart contract that trusts a third party removes the killer feature of trustlessness."*[103]

So until we can find ways of accurately describing reality in digital form in a manner that can be read by a smart contract (the so-called Oracle Problem),[104] then most smart contracts will be mostly useful only for digital goods.

Another reason not to look at smart contracts from a legal perspective is that they are purposefully designed to disrupt the legal profession, at least in accordance to many proponents, and the intention is to have systems that will not necessitate any legal analysis, and any sort of recourse or oversight will be a thing of the past.[105] There is a considerable libertarian streak to some of the cryptocurrency circles, and the blockchain is often offered as a technology that will change governance as we know it, and it may even lead to the destruction of the state altogether.[106] In the words of Nick Szabo:

> *"Trust-minimized rules are new & in many ways will be vastly superior to what trusted but often not so trustworthy kings, judges, legislatures or regulators have concocted. Trust-minimized code will bring reality to the constitutional ideal "a government of laws & not of men."*[107]

Smart contracts become a new jurisdiction, a technical realm where enforcement becomes a question of the past, and where being in control of a good smart contract will be "99% of the law".[108] This will bring about a lawyerless utopia of code where users

---

[103] Song J, "The Truth about Smart Contracts", (2018) *Medium*, https://medium.com/@jimmysong/the-truth-about-smart-contracts-ae825271811f.

[104] For a good explanation of Oracles in smart contracts, see: https://medium.com/@DelphiSystems/the-oracle-problem-856ccbdbd14f.

[105] Orcutt M, 'Hate lawyers? Can't afford one? Blockchain smart contracts are here to help' (2018) *MIT Technology Review*, https://www.technologyreview.com/s/612748/hate-lawyers-cant-afford-one-blockchain-smart-contracts-are-here-to-help/.

[106] Atzori, supra note 63.

[107] https://twitter.com/NickSzabo4/status/1055168586434535424.

[108] https://twitter.com/NickSzabo4/status/1051606530108190720.

program obligations into computer programs without the intention of ever seeing a court of law.

But the main challenges are brought by the blockchain itself, the very same feature that makes strong smart contracts supposedly so appealing. These characteristics are the distributed and immutable nature of this technological solution. These issues are not present in non-cryptographic automated contracts, the distributed ledger technology carries an entirely different dimension to the potential pitfalls encountered by these types of documents. We will now look at the problems presented by these two features.

3.2 The problem with immutability

The main concern is that distributed ledger technology, as understood generally by the Nakamoto definition, contains a requirement of immutability, the blockchain cannot be changed, it is tamper-free, and whatever is written into it cannot be changed or censored. This means that whatever smart contract is written into this ledger will also be immutable, making it difficult (if not impossible) to change the code.

To better understand this, one must understand how the blockchain is immutable from a technical perspective. Smart contracts are considered immutable because of something called a hash. Roughly speaking, a hash function is a mathematical operation that can produce a unique output depending on the input; you can take some text, turn it into numbers, and then apply a formula (the hash function) that will produce a unique number (the hash value). If you changed the original text, then the resulting number would not match the hash value.[109] Blockchains consist of blocks of transactions that are chained together by appending the hash of the previous transaction to the next, making it impossible to change, and therefore tamper-free.

While this feature is extremely useful when it comes to verifiability of data, it has the problem that once written, it cannot be changed, hence the immutable element. It must be said that there are circumstances in which there can be changes, but we will deal with those later, for now we can assume that for all intents and purposes the blockchain is immutable. If everything works well, then this feature makes it easy to trust the information entered into the blockchain. But the opposite is true, if you enter wrong data or information into the ledger, then it cannot be amended. This means that smart contracts are practically set in stone, as the technical robustness fails to take into account the complexity of real contracts, where the social element is often as important, if not more, than the formal elements written into the document.[110]

The problems that arise from perpetuating errors in code are not just abstract possibilities, this has already happened several times. The most famous case is that of the Ethereum Decentralized Autonomous Organization (DAO). As the name indicates, the objective of this application is to use smart contracts to help in the operation of decentralized organizational governance, in their words, it "leverages smart contracts on the Ethereum blockchain so that anyone, anywhere in the world can be empowered to participate."[111] The DAO operates as a facilitator for smart contracts, and also as a

---

[109] Here is a good video explaining hash functions and values https://youtu.be/2BldESGZKB8.

[110] Levy K, 'Book-Smart, Not Street-Smart: Blockchain-Based Smart Contracts and The Social Workings of Law' (2017) 3 *Engaging Science, Technology, and Society* 1-15.

[111] https://www.crunchbase.com/organization/the-dao#section-overview.

governance structure for shared resources. Imagine that you have a startup that wants to hire help quickly and efficiently in order to scale operations. Currently, you have to negotiate with each contractor individually, and this takes time and money. With the DAO, you would pay ether (ETH) into the DAO, and at the same time you set terms and conditions of what you want the contractors to do, while the ETH is held securely. The contractors participating in the DAO define the parameters of their contract in code (business model, payment, operational parameters), if the contractor code meets the DAO code, then there is an autonomous smart contract formed, upon completion of which the DAO releases payment. Those who support the DAO receive voting rights in the shape of DAO tokens, as well as possible dividends in the shape of ether. The DAO is ruled by the voting performed by those participating in the system, making it a completely decentralized governance model.

The problem started in June 2016 with a small bug in the DAO code. The DAO operates as a public pool of funds holding several millions of dollars' worth in Ether, and only those participating in the scheme, or those completing contracts, can withdraw funds according to the pre-established terms of their participation. Using the pool analogy, you can imagine several hoses being fed by the pool, each with a tap that opens and closes using pre-defined conditions. But there was a bug in the code that allowed malicious participants to call recursive functions upon the closing of a transaction that allowed the tap to remain open, and in theory it would let them drain the entire pool. This is precisely what happened, and hackers exploited the bug and were able to drain 3.6 million ETH from the DAO common fund.[112] It is indicative of the technical nature of smart contracts that this error was not solved by legal means, the developers and maintainers decided to "fork" the code, which resulted in an entirely new version of the DAO without the bug, and therefore where the "theft" had never taken place, a technical way to turn back time.[113] Forking in the software sense means a split in development, it is to take a project's source code and begin a separate development altogether, creating a fork in the code. While the original code remains, all future instances of the software will use the new code.[114]

It may be easy to underestimate the importance of this fork. The practical result was that there were two Ethereum implementations in existence, one with the DAO funds stolen, and one where the bug was fixed, and therefore the syphoning never took place. The original code was renamed "Ethereum Classic",[115] and the forked version became plain Ethereum. This is probably the single most important decision made by any smart contract developer, and it is one that continues to have serious repercussions for the claim that smart contracts and blockchains are immutable. While this is technically true, evidenced by the fact that the original Ethereum still exists and cannot be easily changed, we are presented with an example where developers intervene to change the code. Immutability is therefore dependent entirely on the will of the platform developers, and

---

[112] Thomson C, 'The DAO of ETHEREUM: Analyzing the DAO hack, the Blockchain, Smart contracts, and the Law' (2016) *Medium*, https://medium.com/blockchain-review/the-dao-of-ethereum-e228b93afc79.

[113] The fork announcement can be found here: https://blog.ethereum.org/2016/06/17/critical-update-re-dao-vulnerability/.

[114] See: Robles G, González-Barahona JM, 'A Comprehensive Study of Software Forks: Dates, Reasons and Outcomes', *The Eighth International Conference on Open Source Systems* (OSS 2012), https://flosshub.org/sites/flosshub.org/files/paper_0.pdf.

[115] https://en.wikipedia.org/wiki/Ethereum_Classic.

everything that is written into a blockchain, including smart contracts, are entirely dependent on the decisions made by those in control of the code. Smart contract therefore become a matter of governance, and not of contract law.

To reinforce this, there have been other high-profile errors in which developers did not intervene, displaying once more the power that the platforms have. On November 2017, a bug in a multi-signature smart contract for an Ethereum-based wallet resulted in the freezing of $280 million USD in ether at the time. As the code is immutable and written in the blockchain, there was no other way of unlocking the funds other to change the blockchain and re-write history.[116] In stark contrast with the DAO theft, where the community decided to fork the code to fix the bug that allowed funds to be taken out, in the case of the Parity wallet there has not been a similar technical solution, and at the time of writing the funds remain frozen and no legal action has been forthcoming. In fact, the Parity wallet team have released a statement apologizing to the community for the bug and claiming that they are still committed to developing in the Ethereum environment.[117] It is a testament of the reliance of the community on developers that there has not been any legal action taken so far, even though this error involved considerable amounts of money. The community seems intent on carrying on despite the problems. The immutable code prevailed, and future implementations of the Parity Wallet code had to be re-written without the bug, but the existing error was not or forked, and therefore still remains.

Bugs and backdoors[118] in contracts are exceedingly common, with one report placing the figure at about 25% of all smart contracts having some sort of error,[119] while another study analysed 19,366 Ethereum-based smart contracts, and found that 44% of them contained some sort of bug.[120] As a lot of code is shared between contracts, the same error can often be replicated to multiple documents, for example, a bug on transaction orders was found in 15% of all Ethereum contracts.[121] As David Gerard said: "there are endless guides to writing a secure smart contract for Ethereum, but most Ethereum contracts ignore them, with the obvious consequences."[122]

And these figures are only about unintentional bugs, it is possible for coders to purposefully build a contract with flaws so that all funds are removed from a wallet,[123] or

---

[116] Thomson I, 'Parity calamity! Wallet code bug destroys $280m in Ethereum' (2017) *The Register*, https://www.theregister.co.uk/2017/11/07/parity_wallet_destroys_280m_ethereum/.

[117] Statement here: https://www.parity.io/our-commitment-to-ethereum-and-a-decentralised-future/.

[118] A backdoor is a bug, often intentional, that allows Unauthorised access. For example of a smart contract backdoor, see: https://medium.com/unchained-reports/bancor-unchained-all-your-token-are-belong-to-us-d6bb00871e86.

[119] Sedgwick K, '25% of All Smart Contracts Contain Critical Bugs' (2018) *Bitcoin News*, https://news.bitcoin.com/25-of-all-smart-contracts-contain-critical-bugs/.

[120] Olickel H, 'Why Smart Contracts Fail: Undiscovered bugs and what we can do about them' (2016) *Medium*, https://medium.com/@hrishiolickel/why-smart-contracts-fail-undiscovered-bugs-and-what-we-can-do-about-them-119aa2843007.

[121] Ibid.

[122] Gerard, supra note 39.

[123] Demolino et al, supra note 130.

to create a contract that operates intentionally as a pyramid scheme, with a bug that allows funds to be taken outside of the original parameters.[124]

Having so many errors and mistakes in immutable code has been one of the most salient and discussed aspects of smart contracts. This is where it is tempting to talk about the will of the parties, error, mistake, frustration, and the enforcement of contractual obligations set out in code.[125] Experience with non-cryptographic automated contracts in the past lead us to believe that the contract law is in general quite well prepared to handle many of the issues that could arise from the performance and enforcement of a contract that contains an error, and that there may not be liability arising from an honest mistake that makes it impossible for the parties to perform the contract as originally drafted.[126] There is extensive case law dealing with such cases in various jurisdictions[127] and one could think that there is no reason why such rules would not apply to strong smart contracts either.

But the immutable element makes such strict legal considerations almost moot. When presented with a situation such as the Parity Wallet, we are witnessing some bugs that could frustrate the performance of a contract, and this may arise from an honest technical error, but there is nothing that could remedy such a situation from a legal perspective. There may not be a party to sue, and even if a court examined the circumstances and agreed that there had been an obvious error, the court may be powerless to make any changes, or to redress the situation. Immutable self-executing code cannot be taken to court.

We could then be in the presence of an alternative legal space where "parties can transact outside of the legal system".[128] While a party may want to obtain redress, it would be technically impossible to do so. The contract itself is code written in a distributed ledger, with a set of instructions that are meant to be executed by a machine without human intervention. If such an interaction cannot be modified by external factors, then enforcement would not be a viable solution, and may not even be possible.

While smart contracts could make a number of transactions cheaper and more efficient, parties will have to consider that redress in the face of bugs and errors could render the contract inoperable. In a recent speech on the subject of financial technology, Lord Hodge of the UK Supreme Court made the following comments accepting the challenges of these new technical realities:

> *"But the law has to address how to provide a remedy if contractual consent has been vitiated, for example, by misrepresentation or fraud. Smart contracts are self-executing as the terms of the agreement between a buyer and a seller are written into lines of code which exist in a blockchain. When the coded conditions are met, a product is released or a payment*

---

[124] Szilágyi P, 'How to PWN FoMo3D, a beginners guide' (2018) *Reddit r/ethereum*, https://www.reddit.com/r/ethereum/comments/916xni/how_to_pwn_fomo3d_a_beginners_guide/.

[125] See for example: Raskin, supra note 23 at 328.

[126] Giancaspro, supra note 93.

[127] See Davis Contractors Ltd v Fareham Urban District Council [1956] UKHL 3 regarding frustration, and Raffles v Wichelhaus [1864] EWHC Exch J19 with regards to error.

[128] Mik E, 'Smart Contracts: Terminology, Technical Limitations and Real World Complexity' (2017) 9 Law, Innovation and Technology 269.

> *made. No-one, including a court, can stop the performance of a smart contract. The courts will not be able to cancel the performance of the contract. But a remedy may lie in the law of unjust enrichment in both common law and civil law jurisdictions to compel the parties to re-transfer the property or money which was the subject of the transaction."*[129]

It is interesting that the legal establishment already recognises that self-executing contracts may fall outside of the legal judicial powers to enforce, and Lord Hodge may be unto something with the proposed solution of looking at redress directly to the parties through unjust enrichment, and not through contract law as such. But there is one more feature of smart contracts that could even frustrate such a strategy, and it is the potential of anonymity, particularly with some contracts that could be part of a larger pool of resources, such as the DAO. The party benefiting from the mistake could not be easily identifiable, and then we would be back to having an unenforceable document that cannot be the subject of legal action.

It must be said that the above is a feature, not a bug. As we have seen, some of the most vocal proponents for strong smart contracts envisage an arrangement that sits outside of the traditional legal system of courts and lawyers. The immutable nature of distributed ledgers is precisely designed to frustrate legal intervention. The existence of smart contracts as a techno-anarchy without laws and judges, only code, may not sit well with many, and in fact it could even dissuade some parties from attempting to adopt the technology. Given the likelihood that something will go wrong due to an error, the absence of legal enforcement would become a risk-factor to take into account when looking at the viability of contracts written into immutable public ledgers.

3.3. The problems of decentralisation

This brings us the second relevant characteristic of smart contracts, and that is the fact that they are distributed. What this means is that in a public and trustless blockchain, a smart contract is not run exclusively by the parties in a contract, but would be executed by the network of nodes that maintains the blockchain. In other words, the contract would be executed by "the network of miners who reach consensus on the outcome of the execution and update the blockchain accordingly."[130] While this may seem like a technical detail on the surface, sort of in the same way that the Internet itself is a "network of networks", this aspect can be one of the most relevant features of this type of smart contracts from a formal perspective, as the contract does not exist if there is no network to verify the transactions.

Most smart contracts nowadays rely on platforms such as Ethereum. While it is possible to have a private blockchain, the main appeal of decentralised databases using proof of work is the use of an open source, distributed ledger that is maintained by thousands of nodes, and where transactions are verified by miners. The miners are the most important part of this scheme, as they perform the mathematical calculations required for the contract to be performed. The incentive of these actors to perform such verifications is

---

[129] Lord Hodge, *Financial Technology: Opportunities and Challenges to Law and Regulation*. Speech at the East China University of Political Science and Law, Shanghai, China (October 2018), https://www.supremecourt.uk/docs/speech-181026.pdf.

[130] Delmolino et al. *Step by step towards creating a safe smart contract: lessons and insights from a cryptocurrency lab* (2015) https://eprint.iacr.org/2015/460.pdf.

that they are rewarded with a block after some time and resources have been spent, these miners are what make possible to run contracts in a distributed manner. In some sense, these miners are the ones that are executing and performing the contract. A smart contract platform consists of these miners, as well as the developer that released the code they are using to mine.[131]

This is one of the characteristics of blockchains that is touted as giving it reliance and security,[132] the network is distributed and therefore all participants verify the transactions through the mining for coins, and the permanent record of the transaction is shared publicly for all to see. In smart contracts, the ledger is both a record and verification, sort of like having thousands of notaries verifying a signature.

For the most part the system works as intended. Parties write a contract in code and publish it in a blockchain which is maintained by developers and miners. There is an ecosystem that does not rely on trust, all the parties are self-interested in maintaining the status quo, miners execute the calculations needed to verify the transactions because they get rewarded. Some systems may also offer transaction fees. The contracting parties can rely on those involved doing their job.

But the fault in the system is the very reliance on these third parties, the permanent record cannot exist without the blockchain, and the blockchain cannot exist without the infrastructure of developers and miners, and these could become very important to the whole endeavour. As with a paper contract, there are usually at least two parties directly involved in a cryptographic smart contract, and under normal circumstances these would be the only parties to any conflict should one arise. But in automated contracts on a distributed ledger, there could be many other indirect participants, as blockchains are composed of a multiplicity of actors that have a relevant participation on its operation; this includes participants, developers, administrators, and gateways.[133]

So smart contracts are not only dependent on the two or more parties that have entered into an agreement, they also involve the software developer that produces the program, and it also involves thousands of miners around the world that perform the verifications required for the contract to run. The idea behind distributed applications and contracts is that all of these participants do not care about any one specific transaction, they run thousands of contracts at the same time. Miners operate independently, so they all get to verify the "history" as written in the blockchain together. These are not participant in any specific transaction, they just make the system work for everyone.

However, this works as long as the integrity of the blockchain remains intact. Proof of work is built in a way that makes it very difficult to change, but it is not completely impossible, the system relies on a majority of miners operating rationally to protect the integrity of the ledger. But what happens if a single entity managed to get enough computing power to re-write the blockchain? Then we get what is called a "51%

---

[131] See Nakamoto, supra note 44.

[132] De Filippi and Wright, supra note 27 at 79.

[133] Rauchs M and others, *Distributed Ledger Technology Systems: A Conceptual Framework* (2018), Cambridge University Judge Business School Report, https://www.jbs.cam.ac.uk/faculty-research/centres/alternative-finance/publications/distributed-ledger-technology-systems/#.W3HmKtJKiUk.

attack",[134] one mining entity (or a multitude working in unison) gain control of more than half of the computing power dedicated to make calculations required to maintain the blockchain, they can re-write the blockchain, could change transactions, or even re-write contract code. This would mean that in theory, contracts could be changed at will with a sufficient level of technical resources spent to change the ledger.

This opens up several new of legal issues, most of which have gone unexplored for now. It is technically possible for miners to get together at some point and perform an attack on a specific contract. This is not fully hypothetical, it became a possibility in the past when a popular smart contract called FOMO32 started accumulating large amounts of money, and several developers discussed that it could be profitable for miners to get together and empty the contract's funds.[135] However, proof of work systems do not make it easy for such an attack to take place, for starters, the resources needed to mount a successful majority attack may be prohibitive. But while miners are competing with others for rewards, most cryptocurrencies are nowadays so difficult to mine that there is an incentive for collaboration in what is known as a mining pool, a collection of miners working together to maximise profits.[136] While the mining pools can consist of individuals, the cost of mining has meant that some sort of centralisation has been taking place, particularly involving Chinese entities, and nowadays most of the cryptocurrency mining is taking place in China.[137] Such centralisation makes it more likely that some sort of majority attack could take place on an entire infrastructure.

It is possible to calculate the amount of computing power needed to undertake an attack, and there are lists that cite the needed to successfully re-write the blockchain for specific coins.[138] At the time of writing, the cost in computing power of taking over the Bitcoin network for an hour is $263,416 USD, and while expensive, it is by no means excessively onerous for a dedicated party. For comparison, attacking the Ethereum network would cost $66,763 USD, and some coins such as Litecoin Cash can be taken over with just $59 USD worth of computing power. One can make the argument is that one should pick only mature and popular cryptocurrencies to write a smart contract, future success is not assured, so one could write a smart contract using a blockchain that is supported by a popular network, only to see that support collapsing in the future. After all, mining is only sustainable if the miners are ensured to stay profitable, and sudden drops in prices[139]

---

[134] For more about this, see: Bae J and Lim H, 'Random Mining Group Selection to Prevent 51% Attacks on Bitcoin', *2018 48th Annual IEEE/IFIP International Conference on Dependable Systems and Networks Workshops (DSN-W)* (2018); and Harwick C, 'Cryptocurrency and the Problem of Intermediation' (2016) 20 The Independent Review 569.

[135] For a discussion of this eventuality, see: 'Millions of Dollars Are Being Sent to a Ponzi Like Ethereum Smart Contract Game That Might Never End (2018) *TrustNodes*, https://www.trustnodes.com/2018/07/22/millions-dollars-sent-ponzi-like-ethereum-smart-contract-game-might-never-end.

[136] Lewenberg Y and others, 'Bitcoin Mining Pools: A Cooperative Game Theoretic Analysis', *Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems* (International Foundation for Autonomous Agents and Multiagent Systems 2015).

[137] Kaiser B, Jurado M and Ledger A, 'The Looming Threat of China: An Analysis of Chinese Influence on Bitcoin' (2018) *arXiv* 1810.02466 [cs.CR] https://arxiv.org/abs/1810.02466.

[138] See for example: https://www.crypto51.app/.

[139] For example, at the time of writing Bitcoin has experienced a drop of 30% in value in just one week.

could be translated into a fall in participating miners in the network, which also makes a majority attack more likely.

These concerns stopped being an eventuality in November 2018, when cryptocurrency Vertcoin was subjected to a successful 51% attack, with over 300 blocks being changed from the blockchain, resulting in changes that exceeded $100,000 USD in value.[140] This specific attack is worrying because it was sustained, and it also involved multiple blocks, this means that attackers were able to wrestle control of the system for a sustained period of time, and also means that not only one block was attacked, but several.

But the most worrying development took place in early January 2019 when the aforementioned Ethereum Classic was subject to a massive 51% attack.[141] After the DAO fork, most of the Ethereum community migrated to the new version, but the original code and blockchain still remained usable, and the coin was still being traded in exchanges. However, when most people migrated, so did the miners, so the coin remained a prime target to a concerted attack, which successfully took place. The attackers conducted what is known as a "double-spend" attack, which is in fact a reversal of a transaction in which coins spent are given back to the assailant. This is normally not possible because the transactions cannot be changed, but if one has gained control of the network, it is possible to transmit a version of history in which the money is still in my wallet. It is calculated that in this occasion the perpetrators managed to conduct 15 double-spend transactions worth over $1 million USD.[142]

There is evidence of other successful attacks against other smaller coins.[143] The problem is that once a platform becomes disused, or the price of the cryptocurrency used to support transactions decreases, then there is no incentive for miners to put resources into maintaining the integrity of the system. Contracts written into such a ledger would be liable to be changed. The result is that the much-vaunted immutability of smart contracts can only be assured as long as it is not worthwhile for a dedicated party to mount an attack. In the case of contracts involving large sums of money, this could be both viable and profitable for the attackers.

As stated above, there is growing concern that mining has become increasingly centralised, with large mining pools operating in coordinated fashion.[144] Moreover, a study of mining practices in some cryptocurrencies has found that some networks and developers have a more direct hand in mining practices, when it was discovered that popular currency Monero had been secretly mining its own coins in a coordinated manner, and they were spending about 50% of all of the computing power in the network

---

[140] Nesbitt M, 'Vertcoin (VTC) is currently being 51% attacked' (2018), *Coinmonks*, https://medium.com/coinmonks/vertcoin-vtc-is-currently-being-51-attacked-53ab633c08a4.

[141] De Silva M, 'Ethereum Classic is under attack' (2019) *Quartz*, https://qz.com/1516994/ethereum-classic-got-hit-by-a-51-attack/.

[142] Yap R, 'Ethereum Classic's 51 Percent Attack Highlights the Challenges of Proof-of-Work Coins' *CryptoSlate*, https://cryptoslate.com/ethereum-classics-51-percent-attack-highlights-challenges-proof-work-coins/.

[143] BTG, XVG, and MONA.

[144] Sui D, Ricci S, and Pfeffer J, 'Are Miners Centralized? A Look into Mining Pools' (2018) *Consensys*, https://media.consensys.net/are-miners-centralized-a-look-into-mining-pools-b594425411dc.

at that time.[145] Obviously, this would mean that a smart contract written in these types of blockchains and using those coins as tokens could be vulnerable, and not immutable at all.

But miners are not the only parties involved, platforms rely on shared code, and a set of developers that allow the system to exist. To understand the relevance of the developers to smart contracts, we need to understand the governance structure of the code that sustains the contracts.

Take Ethereum, still the most popular smart contract platform. The Ethereum ecosystem consists of a number of developer tools, which include a programming language, an open source operating system to run ĐApps, the cryptocurrency Ether, and a blockchain. The platform was proposed in 2013 by developer Vitalik Buterin, and it came into existence in 2015 after using an ICO to crowdfund its development. The organisation as such started life as a company, and it is now a non-profit Foundation based in Switzerland. The Foundation's role is to "is to promote and support Ethereum platform",[146] and as such it has little or no technical role. Currently, the technical decision-making structure is based on the Ethereum Improvement Proposal (EIP-1), which is a governance system that makes decisions on standards and technical implementation, and it tries to gain consensus from the developer community.

The practical effect of this structure is that it is the community that makes decisions that are relevant to the platform used to run smart contracts. This is vital, because it is the resilience of the technology what makes many of the claims of the security and immutability of smart contracts possible. If changing the code that supported an agreement was easy, then contracting parties would be subject to the whims of a nameless and faceless developing community of coders.

This is precisely what happened with the DAO hack and its subsequent fork, a decision that could have wide-ranging consequences of how we think about smart contract platforms from a legal perspective. When the bug and subsequent Ether theft was uncovered, most developers seemed to agree that the best solution was to fork the platform code and retroactively re-write the blockchain to a stage where the vulnerability would not be possible. Buterin did not have the power to do this on his own, and needed the rough consensus of other developers to do it, but he advocated strongly for a fork to fix to the issue so that developers would be able to trust the system again.[147] But fixing a bug in this way, and re-writing the blockchain, is a highly controversial decision, one that was opposed by some other participants. For example, a developer wrote:

> *"One solution is to roll-back the blockchain before the theft. Of course, that means screwing over everybody who made a transaction since then. You'd be screwing people out of $1 million in order to compensate the theft of $100 million. This is, of course, the type of corrupt thinking that gets us*

[145] Vorick D, 'The State of Cryptocurrency Mining' (2018), *Sia Blog*, https://blog.sia.tech/the-state-of-cryptocurrency-mining-538004a37f9b.

[146] https://www.ethereum.org/foundation.

[147] Buterin V, 'CRITICAL UPDATE Re: DAO Vulnerability' (2016) Ethereum Blog, https://blog.ethereum.org/2016/06/17/critical-update-re-dao-vulnerability/.

into banking failures in the real world, as we screw over everyone else in order to protect those banks who are too big to fail."[148]

As it is evident from the above, and many other writings at the time,[149] the Ethereum development environment is filled with libertarians and crypto-anarchists that see such centralized decision-making as anathema to the very ethos of the blockchain. But more importantly, by getting together and making a decision to change the blockchain, the Ethereum developers opened the door for future legal recourse against platforms such as the Ethereum Foundation. If there is a bug in your code you can try to ask the developers to have it changed, as has been the case several times when smart contract bugs have been uncovered.[150] We need not think about smart contracts using the terms of contract law, but should rather think of them as a service where the legal remedies sought would be not against the party, but against the developers asking for a court-mandated fork to fix the blockchain.

While this notion started as a joke[151] in some developer circles, it has started to gain traction. Earlier this year, a class action suit was filed in New York[152] against the developers of the cryptocurrency NANO (XRB), where the plaintiffs requested a fork to recover $170 million USD that had been stolen due to a hack to a cryptocurrency exchange in February 2018. In the complaint,[153] the plaintiffs argued that the developers of NANO had the ability to return the stolen funds by forking its blockchain in a manner that would re-allocate the stolen tokens to their original and lawful owners. The case was dismissed because the plaintiff abandoned it, so we never got to read a decision on the legal merits of asking for a court-mandated change to the blockchain. While the case does not involve smart contracts directly, it could give an indication of where parties involved in large contracts could try to tackle the potential unenforceable nature of smart contracts.

But even here most legal action could be futile. The developers working in cryptocurrency and smart contract development are often operating outside of a centralised structure using open source licensing and repositories. While there may be a central organisation such as the Ethereum Foundation, these tend to rely on an army of independent developers who contribute to the code. Even if we were to imagine a court issuing a mandated fork, or some change to try to fix an error in a smart contract, the developers could ignore it. Enforcement of such an order would be difficult, if not impossible.

The above has led some researchers to argue that software developers could be considerably more liable than it has been previously assumed. Walch claims that in the

---

[148] Graham, supra note 64.

[149] See for example a statement by the very hackers claiming that the exploit was not theft, as it was a bug in a smart contract: https://pastebin.com/CcGUBgDG.

[150] See for example this unsuccessful pull request on the Ethereum code to remove a bug from a smart contract: https://github.com/ethereum/EIPs/issues/1030.

[151] Developer Pierre Rochard commented that perhaps we should have a formal system involving lawyers and judges to adjudicate property rights on the blockchain through forks: https://twitter.com/pierre_rochard/status/996322728973041665.

[152] Brola v. Nano et al (1:18-cv-02049), New York Eastern District Court.

[153] https://www.morrisoncohen.com/siteFiles/files/2018_04_06%20-%20Brola%20v_%20Nano%20et%20al.pdf.

context of cryptocurrencies, coders operate more like fiduciaries than anything else, and as such the trust that is given to them should open the organisation to being legally accountable as fiduciaries if they "fail to act with competence on behalf of those who rely on the blockchain".[154]

While the proposal of treating developers as fiduciaries is interesting, what is more likely to happen is that plaintiffs and courts will use a range of legal options that could be deployed against platforms.

Once we recognise that there are intermediaries that could be capable of making changes to a blockchain to change the terms of a smart contract, then anything that could be used to compel those intermediaries can and will be used. It could be a question of the law of equity, trusts, negligence, unjust enrichment, and maybe even intellectual property. What matters is that a party that feels they are bound by an erroneous and/or fraudulent smart contract, they will not seek redress against the other party, but rather will try to compel the platforms as active intermediaries. Already regulators are looking[155] at intermediaries as the true actors in this space, so perhaps this will offer us an indication of where most enforcement will take place. However, the decentralised nature of the development environment would probably continue to frustrate users.

This could not only stop with the software developers, but could be extended to those with real control, namely the miners. A future where a court orders a mining pool to make changes to a contract would also be a possibility, particularly in situations where it has been demonstrated that the developers have exercised majority control over the mining resources, such as those discussed earlier.

Interestingly, we could be entering an era in which the limitation of liability of intermediaries that was experienced during the early years of the Internet[156] could be tested again in court, but this time the defending intermediaries would be software developers, cryptocurrency miners, and blockchain platform maintainers.

The general idea is that the intermediaries are not liable because they just facilitate the infrastructure that can be used to run the system, much like telecommunications companies are not liable for fraud that is committed using a telephone, this is responding to what is called Szabo's Law,[157] which postulates that developers should not "implement changes to the blockchain protocol unless the changes are required for the purpose of technical maintenance." This is because making any proactive decisions could open platforms to legal liability. For the most part, this theory has held, even with a few

---

[154] Walch A, 'In Code(rs) We Trust: Software Developers as Fiduciaries in Public Blockchains', In Dimitropoulos G and others, *The Blockchain Revolution: Legal & Policy Challenges* (Oxford University Press 2018).

[155] Del Castillo M, 'SEC Cyber Chief Puts A New Type Of Cryptocurrency Exchange On Notice' (2018) *Forbes* https://bit.ly/2Du2p2l.

[156] This topic falls outside of the remit of this article. For more on this subject, see: Mac Síthigh D, 'The Fragmentation of Intermediary Liability in the UK' (2013) 8 *Journal of Intellectual Property Law & Practice* 521.

[157] Zamfir V, ' Against Szabo's Law, For A New Crypto Legal System', (2019) *Crypto Law Review*, https://medium.com/cryptolawreview/against-szabos-law-for-a-new-crypto-legal-system-d00d0f3d3827.

regulators claiming that they view platforms as mere conduits.[158] In theory developers are not involved in the contracts themselves, and the system runs like the self-executing code as intended. However, the more we learn about the practicalities of the implementation, the more it becomes evident that the system relies entirely on these intermediaries.[159]

Both developer-led forks, and 51% attacks give us enough ammunition to think that the law governing smart contracts is not contract law, but rather anything that covers liability by intermediaries.

Therefore, smart contracts are not contracts, but they are to be treated just as any other piece of software, there may be some contract law involved in the shape of licences and such, but they should be seen as a service, with all of the legal implications that it entails.

4. Conclusion: Code is not law

In his seminal (and often misunderstood) work *Code: The Laws of Cyberspace*, Lawrence Lessig wrote:

> *"[Code] will present the greatest threat to both liberal and libertarian ideals, as well as their greatest promise. We can build, or architect, or code cyberspace to protect values that we believe are fundamental. Or we can build, or architect, or code cyberspace to allow those values to disappear. There is no middle ground. There is no choice that does not include some kind of building. Code is never found; it is only ever made, and only ever made by us."*[160]

Lessig was talking specifically about the difficulties of regulating cyberspace, and he argued that the "invisible hand of cyberspace" was shaping a new regulatory solution in the architecture of the Internet that could be used to regulate the space. This is where the famous maxim of "Code is Law" comes from.

Interestingly, this has recently been taken over as the rallying cry of smart contract enthusiasts, completely subverting the true regulatory meaning of the phrase, and it has morphed into a techno-libertarian article of faith. They tend to consider the words to mean that code can be used to replace law, and the smart contract will be the foundational tool of a revolution that will end in the replacement of traditional contracts and private law with a "digital private law" enforced by code,[161] and even going as far as changing the famous "pathetic dot" chart to change the dot with a smart contract in the middle.[162]

While some see smart contracts as the end of lawyers, the practical implementation of smart contracts proves that perhaps such statements are still premature. If anything, the

---

[158] Nathan D and Pesok J, 'A Foreboding View of Smart Contract Developer Liability' (2017*) Lexology Blog*, https://www.lexology.com/library/detail.aspx.

[159] Zetzsche DA, Buckley RP and Arner DW, 'The Distributed Liability of Distributed Ledgers: Legal Risks of Blockchain' (2018) 2018 *University of Illinois Law Review* 1361.

[160] Lessig L, *Code Version 2.0* (2nd ed, Basic Books 2006), p.6.

[161] Szabo N, 'Towards a digital and private common law' (2007) *Unenumerated Blog*, https://unenumerated.blogspot.com/2007/05/towards-digital-and-private-common-law.html.

[162] Glatz F, 'What are Smart Contracts?' (2014) *Medium*, https://medium.com/@heckerhut/whats-a-smart-contract-in-search-of-a-consensus-c268c830a8ad.

lack of mainstream implementation of smart contracts could act as a sobering reminder that we have a legal system for a reason, and that perhaps placing all our trust in faceless machines that conduct transactions without possibility of legal redress may still be a bridge too far for most developers. Smart contracts remind us that laws are still useful. This is why smart contracts may not be legally a contract, but they could still end up having considerable legal effects in other areas of law, such as criminal law, data protection, intermediary liability, etc.

While there is certainly scope for the use and implementation of strong smart contracts, there are still enough questions about the power of platforms and other intermediaries to change the blockchain.

And this is precisely where there is a serious issue with the way in which contracts are being discussed at the moment. The prevalent idea is to take smart contracts as cryptographic versions of "real" traditional contracts, and to conduct all of the analysis in that way. The argument presented here is that we cannot think of smart contracts as perfect self-executing code expressed in an immutable distributed ledger. It is more useful to think of them as computer programs running on the cloud. There is no need for a new legal analysis, just look precisely at the infrastructure that supports the platforms for legal recourse in case something goes wrong.