InPerpetuity{challenging misperceptions of the term 'Smart Contract'}

EJLT European Journal of Law and Technology

InPerpetuity{Challenging Misperceptions of the Term 'Smart Contract'}

Monica L Vessio, Arnold Beckmann, Matt Roach, Séverine Saintier, Rhys Clements and Anton Setzer*

Abstract

In law, the term 'smart contract' has been used loosely with no one definition winning out. In an attempt to ameliorate this, the Law Commission of England and Wales has endeavoured to add the word 'legal' to 'smart contract'. No relief is found in the computer coding world, where 'smart contract' is used to indicate not a single but several forms of computer codes that do not involve (necessarily) two parties. Through a sample smart contract use case, this paper identifies more than six such coding concepts and constructs which have been corroborated by the results of trend data analysis. Turning to the public, statistics gathered show a very limited understanding of the terms 'smart contract' and 'smart legal contract' (and their implications). From these findings, this paper recognises the inappropriateness of the use of the single term 'smart contract' for the many diverse iterations as used by computer scientists; and the unsuitability of the word 'contract' as part of the term 'smart contract' by computer scientists because of the term's legal import. The redundancy of the term 'smart legal contract' is established, and in conclusion a definition of 'smart contract' which only considers universal, future-proof characteristics is proposed. In this we disagree with the definitions offered by the Law Commission. The definition we have proffered actively contemplates the legacy use of the term in both law and technology, and is broad enough to be sector- and future-adaptable, and technology-agnostic.

Keywords: smart contract, contracts, code, smart legal contracts, digital contracts, static contracts.

* Monica Vessio is Lecturer in Law, University of Reading; Arnold Beckmann is Professor of Computer Science, Swansea University; Matt Roach is Associate Professor in Computer Science, Swansea University; Séverine Saintier is Professor of Commercial Law, Cardiff University; Rhys Clements is Senior Cloud Compute and Storage Engineer, Swansea University; and Anton Setzer is Reader in Computer Science, Swansea University.

1. Introduction[1]

The way we contract (and how contracts are formed) evolves in response to external factors, such as changes in legislation, economic exigencies and technological shifts. The latter is particularly clear with the rise of smart contracts, which are gaining broader adoption[2] and have increasingly become integrated in the commercial landscape.[3]

It is accepted that the 'technological revolution'[4] of smart contracting does not imply a legal revolution,[5] and that contract law, in its current state, can accommodate smart contracts without any major modification.[6] As these dynamic technological influences shape the way contracts are structured and enforced, it is nevertheless important for the law to adapt to these changes to provide the necessary framework. This requires an assessment of the impact of the technological change to evaluate the level of regulatory response needed.[7] Both steps, however, need to be preceded by a true understanding and concord on terminology. The authors posit that the Law Commission, in its scoping document on smart contracts,[8] has not gone far enough

[1] This work emanates from an interdisciplinary collaboration on an externally funded, business-led sustainability project related to creating a circular supply chain in the supply and leasing of hydrogen vehicles (B2C) and hydrogen vehicle parts (B2B), and coding some parts of the supply chain on a blockchain with a view to eventually creating an end-to-end smart contract supply chain (for details, see <https://circularrevolution.wales/about/>).

[2] They 'are expected to revolutionise the way we do business, particularly by increasing efficiency and transparency in transactions' (Law Commission, *Smart Legal Contracts, Advice to Government* (Law Com No 401, CP 563, 2021) vii).

[3] The global smart contract market is projected to grow from $2.14 billion in 2024 to $12.55 billion by 2032, <https://www.fortunebusinessinsights.com/smart-contracts-market-108635> accessed 24 June 2024. Smart contracts also have an impact for consumers (Law Commission (n 2) ch 6). Consumers are subject to consumer law where the rationale is that, as weaker parties, they need additional protection. For this article, we do not distinguish between B2B and B2C transactions since the aim is more general in terms of definition. For issues (non-exhaustive) pertaining to the deployment of smart contracts in the B2C environment, see Samuel Becher, 'Research Shows Most online consumer contracts are incomprehensible, but still legally binding' (2019) The Conversation <https://theconversation.com/research-shows-most-online-consumer-contracts-are-incomprehensible-but-still-legally-binding-110793> and Uri Benoliel and Samuel Becher, 'The Duty to Read the Unreadable' (2019) 60 *Boston College Law Review* 2255.

[4] The so-called Fourth Industrial Revolution. Florian Möslein, 'Legal Boundaries of Blockchain Technologies: Smart Contracts as Self-Help?' in De Franceschi, A and Schulze, R (eds), *Digital Revolution – New Challenges for Law* (CH Beck and Nomos 2019) 313 as cited by Mateja Durovic, 'What are Smart Contracts? An Attempt at Demystification' in Slakoper, Z and Tot, I (eds), *Digital Technologies and the Law of Obligations* (Routledge 2021) 121.

[5] Roberto de Caria, 'The Legal Meaning of Smart Contract' (2019) *European Review of Private Law* 731.

[6] Law Commission (n 2) para 1.9.

[7] Not every new development will create a 'disruption' in the law. On this, see Roger Brownsword, 'After Brexit: Regulatory-Instrumentalism, Coherentism and the English Law of Contract' (2017) 24 *Journal of Contract Law* 139 and *Law, Technology and Society* (Routledge 2019) ch 8.

[8] Law Commission (n 2).

and leaves some questions unanswered in relation to the legal complexities surrounding the understanding and usage of the term 'smart contract'. We aim to highlight those gaps and explain why they matter from a regulatory perspective.

Enquiries about the definition of smart contracts have been ongoing for some time. To have the correct definition is important normatively since 'it has consequences for the legal analysis of smart contracts'.[9] Too rigid a definition will stifle innovation. A precise but nevertheless elastic definition[10] is therefore the solution to accommodate future technological developments. Our investigations, disclosed in this article, demonstrate that the computer science community uses the term 'smart contract' to describe at least six different concepts and types in relation to tamper-proof executions of computer code: four refer to types of executions, while two are more general concepts related to executions.[11] Furthermore, through empirical research it is evident that there is no common public consensus as to the import of the term 'smart contract'. The Law Commission has attempted to bridge this divide by introducing a new term, 'smart legal contract', to differentiate a [binding] smart contract from a computer science understanding of 'smart contract'. However, we find the use of the word 'legal' in 'smart legal contract' redundant. Moreover, we contend that as this term is based on the definition of a 'smart contract', which is narrower in its understanding of the term compared to what we have uncovered through this work, the Law Commission has not synthesised the harlequin nature of the term 'smart contract' and the additional category does not really add anything to the issue of the complexities surrounding the understanding of the use of this term.

Given that the term 'smart contract' is neither purely legal nor technological, but instead straddles the two disciplines, we argue that it is important for the definition to reflect this dichotomous development. To do so, it must be understood by both

[9] de Caria (n 5) 732.

[10] ibid.

[11] See for example the comments of Christian Colombo, Joshua Ellul, and Gordon J Pace, 'Contracts over Smart Contracts: Recovering from Violations Dynamically' in Margaria, T and Steffen, B (eds), *Leveraging Applications of Formal Methods, Verification and Validation*, vol 11247 (Lecture Notes in Computer Science) 300–1. The LawTech Delivery Panel, *Legal Statement on Cryptoassets and Smart Contracts* UK Jurisdiction Taskforce (2019) 8. See also Christopher D Clack, Vikram A Bakshi and Lee Braine, 'Smart Contract Templates: Foundations, Design Landscape and Research Directions' arxiv:1608.00771 (2016) 9, where they state: 'to be a "smart contract" we require that some part of the agreement is capable of being automated (otherwise it is not "smart")'. Interestingly, Allen refers to the performance of coded actions autonomously as an 'agentive function' ('Wrapped and Stacked: "Smart Contracts" and the Interaction of Natural and Formal Language' in Allen, J and Hunn, P (eds), *Smart Legal Contracts: Computable Law in Theory and Practice* (Oxford 2022) 24). According to the current but self-declared 'working definition' provided by the International Organization for Standardization (ISO) of ISO/TC 307, WG1, a smart contract is a 'computer program stored in a distributed ledger system wherein the outcome of any execution of the program is recorded on the distributed ledger'. The ISO explains that the term 'smart contract' in its original intention (Nick Szabo in 1994) 'had a different, mainly legally oriented (precise and legitimate) meaning'. It acknowledges that this often causes confusion regarding 'legally binding intentions' <www.iso.org/obp/ui/#iso:std:iso:tr:23455:ed-1:v1:en> accessed 4 May 2023.

worlds equally, which can only be done by establishing a common lexicon. A collected glossary provides the standardising basis for a regulatory framework that can comfortably ensconce complex advances. By adopting appropriate vocabulary, other fields can establish or align consistent terminology, making cross-disciplinary collaboration more effective as harmonised terms bridge gaps between different domains. A common term, legally sanctioned, serves as a powerful tool for precision, consistency and alignment across various disciplines especially if the putative definition reflects the distinctive nature of the socio-technical relationships that the law attempts to regulate.

This claim for a definition reflecting the duality of smart contracts comes as no surprise.[12] Cooperation between the two key elements when considering the smart contract environment, law and technology, will maximise the potential for communication and encourage the evolution of smart contracts.[13] This research was carried out in this manner – collaboratively – through an equal interdisciplinary discourse, in which the law and computer science perspectives were explored. This led to some new discoveries regarding what computer scientists intend when using the term 'smart contract'. It was even a surprise to them that the term is used for so many different coding concepts and constructs.

Section 2 of the paper sets the discussion in its wider context to consider whether the computational transformation of the contractual relationship changes our understanding of the nature of a contract. This only takes us so far since the developments do not reflect the manner in which the term is used by computer scientists, as highlighted in a thorough investigation in section 3. This finding is supported by empirical research to highlight the lack of alignment in the use of the term by the computer science community and the legal community (section 4). Section 5 turns to investigate the public's understanding of 'smart contract', emphasising a further lack of comprehension. Whilst our research was originally explorative in nature, after considering the evidence and data uncovered, it became inevitable to proffer a working definition of 'smart contract' by distinguishing certain characteristics (section 6).

[12] Collaboration between computer scientists and lawyers has been called for; see Thibault Schrepel, 'Smart Contracts and the Digital Single Market Through the Lens of a "Law + Technology" Approach' (2021) Directorate General for Communication Network, Content and Technology, European Commission. The Law Commission highlights the need for collaboration at the stage of the drafting of the contract (Law Commission (n 2) ch 2, para 2.57).

[13] Schrepel (ibid), and see Robert Herian, 'Techno-legal Supertoys – Smart Contracts and the Fetishization of Legal Certainty' in Allen, J and Hunn, P (eds) *Smart Legal Contracts: Computable Law in Theory and Practice* (Oxford 2022).

2. The Context: How Computational Transformation has Shaped our Understanding of the Nature of Contract

Although smart contracts can be regulated by contract law and the technology will not replace contract law,[14] these dynamic shifts shape the way contracts are structured and enforced, so it is important to understand whether and how the technology transforms our understanding of the nature of the contract so that the law can adapt with fluidity.

It is trite that in law when an offer is accepted, supported by an intention to create legal relations, and something of benefit is exchanged, provided there are no vitiating factors, a contract is created.[15] This may be done verbally, in writing, through conduct, by electronic communication or via a combination of these. While contracts can be oral, they are often reduced to document form, especially in commercial settings. Freedom of contract is a basic tenet of both common law and civil law systems and, subject to judicial and statutory limitations, parties are free to decide not only the terms that bind them but also how those terms are recorded and performed. Legally speaking, smart contracts can therefore be viewed as an iteration of this freedom to determine how the obligations of a contract are recorded and performed.[16]

2.1 From Paper to Data: Electronic and Digital Contracts

Contracts have traditionally been produced and maintained in paper form. With technological developments over the last 40 years, the process of contracting has changed and there has been a clear move towards the electronic conclusion and storage of agreements. Historically, contracts in electronic form were understood as being word-processed with the potential to be converted to PDF for electronic signature purposes, to be stored in document management systems.[17] The relevant software systems for such instances are used solely to document legal, commercial data and not to structure documents to render the information (data) contained in them usable in the same way as data, to be drawn (for example) from a database or webpage.[18]

An electronic document is an electronic storage of the document, that is purely textual.[19] It is not a digitisation of the agreement since it is not machine-readable. In other words, such documents have not faced the same developments and maturation of web technologies, which enable web pages to be constructed and shared.[20] Only

[14] de Caria (n 5) 748.
[15] The process of contract formation will vary in different jurisdictions; ultimately the legal requirements must be met.
[16] See, however, the discussion on the complexities by Robert Herian, 'Smart Contracts: A Remedial Analysis' (2020) <https://ssrn.com/abstract=3672932> accessed 26 June 2024.
[17] Legal Schema, *A Structured Data Format for Digital Contracts in the UK* LawTech UK 7.
[18] ibid.
[19] An agreement despite its 'electronic costume' (Kevin Werbach and Nicolas Cornell, 'Contracts Ex Machina' (2017) 67 *Duke Law Journal* 313, 320).
[20] Legal Schema (n 17) 5.

by using structured data can a purely textual and thus static document be 'converted' to a digital document. This conversion allows the document to carry added functionality, such as having open access to and sharing 'contract data' to enable search and analysis of data within the contract itself, but also allows integration of the contract data with external systems for automated reporting and operations functionality.[21] Kim posits that 'the most significant difference between digital and paper contracts is that the nature of the terms contained in digital agreements are often much more aggressive' as these include 'terms dealing with matters ancillary to or independent of the primary transaction'. In other words, the fluidity of the digital form and the adjunct functionality of this structure of document significantly affect the rights and expectations of contracting parties.[22] Surden imagined that the next advancement after a digital contract is the agreement form that has the ability to engage in 'some sort of automated comparison between the terms of the contract and relevant information concerning compliance'.[23]

The smart contract is a descendant of the electronic contract, closer in lineage to the digital contract. Szabo referred to these contracts as 'smart' because, 'new institutions, and new ways to formalize the relationships that make up these institutions, are now made possible by the digital revolution', thus making contracts 'far more functional than their inanimate paper-based ancestors'.[24]

Our view is that the word 'contract' is the essential legally concretising factor (the wrapper).[25] As explained above, when referring to a contract, certain criteria must be met to render it legally viable. The Law Commission has opted to link the definition of a smart contract to computer code[26] and so uses the term 'smart legal contract' to define a digital automated contract. The Commission defines it as 'a legally binding contract in which some or all of the contractual terms are designed in and/or performed automatically by computer program'.[27] We contend that the word 'contract' has precise legal meaning. That is, a contract has at law a binding nature.[28] Thus, it is important to insist that the (true) legal meaning of the word 'contract' be given the weight that it has carried historically as a stand-alone.[29] The word 'smart'

[21] ibid 10. Surden refers to these as 'data-oriented' contracts (Harry Surden, 'Computable Contracts' (2012) 46 *UC DAVIS Law Review* 629.
[22] Kim was specifically referring to wrap contracts, for a critical overview see her book, Nancy Kim, *Wrap Contracts: Foundations and Ramifications* (Oxford Academic 2014) 70.
[23] He referred to this as 'computable contracts' (Surden (n 21) 660).
[24] Nick Szabo, 'Smart Contracts: Building Blocks for Digital Markets' (1996) <www.fon.hum.uva.nl/rob/Courses/InformationInSpeech/CDROM/Literature/LOTwinterschool2006/szabo.best.vwh.net/smart_contracts_2.html#:~:text=New%20institutions%2C%20and%20new%20ways,of%20artificial%20intelligence%20is%20implied> accessed 28 July 2023.
[25] See Allen (n 11); UKJT Statement (n 11) paras 143–144 and Legal Schema (n 17) 13.
[26] Law Commission (n 2). See the full definition provided in section 3 below.
[27] ibid.
[28] UKJT (n 11) 8: 'a smart contract is therefore capable of having contractual force'.
[29] Rius and Delgado posit that the word 'contract' as historically used by computer scientists, politicians and economists is different to the legal understanding of contracts and therefore

brings to the word 'contract' an implication of digitalised automaticity,[30] but beyond that, we submit that 'contract' should not be diluted. This ties in with Szabo's original intention (he coined the term 'smart contract').[31] Furthermore, we find that the use of the word 'legal' in 'smart legal contract' is redundant, and that the use of the word 'contract' as part of the term 'smart contract' by computer scientists is for several purposes (static, applies to more than one coding concept) unsuitable. We argue therefore that computer scientists should not retain dominance of the use of this term. To bandy the word 'contract' about indiscriminately, sometimes intending it to have legal meaning and sometimes not, simply muddies and confuses the legal and societal waters.

2.2 The Origins of the Term 'Smart Contract'[32]

The term 'smart contract' was introduced in 1994 by Szabo,[33] when he defined it as a 'computerized transaction protocol that execute[s] the terms of a contract. The general objectives of smart contract design are to satisfy common contractual conditions (such as payment terms, liens, confidentiality, and even enforcement), minimize exceptions both malicious and accidental, and minimize the need for trusted intermediaries'.

---

elect, for the purposes of their paper, to use the term 'legal contract' when referring to a binding agreement. Oddly they make this decision despite giving the Cambridge and Merriam-Webster online dictionaries definition, which confirms the standalone word 'contract' is an agreement (i.e., a collection of promises between two or more parties) that gives rise to a corresponding set of legally binding obligations and confirming that from a legal practitioner's perspective quoting (Hugh Beale (ed), *Chitty on Contracts* (31st edn, Sweet & Maxwell 2012) para 1-016: it is 'a promise or set of promises which the law will enforce'. It is submitted that this is an inconsistent conclusion (Alfonso Delgado Rius, 'Smart Contracts: Taxonomy, Transaction Costs, and Design Trade-Offs' in Allen, J and Hunn, P (eds), *Smart Legal Contracts: Computable Law in Theory and Practice* (Oxford 2022)).

[30] UKJT (n 11) 31; see also Max Raskin, 'The Law and Legality of Smart Contracts' (2017) *Georgetown Law Technology Review* 305, 309; Philip Paech, 'Law and Autonomous Systems Series: What is a Smart Contract?' (Oxford Business Law Blog, 9 July 2018) <https://blogs.law.ox.ac.uk/business-law-blog/blog/2018/07/law-and-autonomous-systems-series-what-smart-contract> accessed 26 June 2024. Some contracts may not be suitable for automated contracting as some obligations may not be adept to conditional logic and may require 'the exercise of discretion, reasonableness, best endeavours or some element of human judgement' and will thus be difficult to translate into code (Law Commission (n 2) 14).

[31] Nick Szabo, 'Smart Contracts' (1994) <www.fon.hum.uva.nl/rob/Courses/InformationInSpeech/CDROM/Literature/LOTwinterschool2006/szabo.best.vwh.net/smart.contracts.html> accessed 26 June 2024. This observation is also posited by the ISO (n 11).

[32] The writers share Schrepel's observation that '[l]ooking at smart contracts through the lens of an evolutionary perspective helps understand and capture the dynamism and complexity of the ecosystem' (Schrepel (n 12)).

[33] See Szabo (n 24); see also Maria Vigliotti, 'What do we mean by Smart Contracts? Open Challenges in Smart Contracts' (2021) Frontiers in Blockchain <www.frontiersin.org/articles/10.3389/fbloc.2020.553671/fullVW%20doi:10.3389/fbloc.2020.553671> para 3.1).

Another iteration of the smart contract, although not labelled as such, is the 'Ricardian contract' introduced and described in the early 2000s by Ian Griggs[34] as 'the design pattern to capture: as a human-readable, contractually significant document, digitally signed and including sufficient but simple markup tokens such that a computer program could extract out the handful of important values: face, rates, issuer, etc. The document could then be hashed cryptographically, providing a secure, unique and cost-free identifier'.

When it arrived in November 2008, it was widely considered that Bitcoin[35] was the first implementation of Szabo's vision.[36] After rejected attempts to contribute and extend the computational capabilities of the Bitcoin related platform Mastercoin[37] – where the Mastercoin team considered the proposed changes too radical for their development roadmap, despite being impressed with the ideas – Buterin introduced Ethereum,[38] a more extended, Turing complete[39] version of 'smart contracts',[40] as contemplated by the computer science community.[41]

---

[34] Ian Griggs, 'Why the Ricardian Contract Came About: A Retrospective Dialogue with Lawyers' (2012–2013) 22 *Journal of Law, Information and Science*. Griggs' work is primarily focused on the financial sector.

[35] Satoshi Nakamoto, 'Bitcoin: A Peer-to-peer Electronic Cash System' (2008) <https://nakamotoinstitute.org/library/bitcoin/> accessed 26 June 2024. See also Allen (n 11) 26-27.

[36] It seems that the underlying ideas behind blockchain technology in effect emerged in the late 1980s and early 1990s, with Lamport developing the Paxos protocol in 1989, and submitting a paper in relation to this in 1990, which was eventually published in 1998 (Leslie Lamport, 'The Part-Time Parliament' (1998) 16 ACM Transactions on Computer Systems <https://dl.acm.org/doi/10.1145/279227.279229> accessed 20 July 2022). These concepts were combined and applied to electronic cash in 2008 and described in a pseudonymous paper, 'Bitcoin: A Peer-to-Peer Electronic Cash System' (2008) <https://nakamotoinstitute.org/bitcoin/> accessed on 20 July 2022. Nakamoto's paper is said to contain the blueprint for most modern cryptocurrency schemes (applied with variations and modifications) (Dilan Yaga, Peter Mell, Nik Roby and Karen Scarfone, 'Blockchain Technology Overview', National Institute of Standards and Technology Internal/Interagency Report 8202 (2018). <https://nvlpubs.nist.gov/nistpubs/ir/2018/nist.ir.8202.pdf> accessed 19 April 2023.

[37] Mastercoin is an altcoin, i.e., a cryptocurrency alternative to Bitcoin with a substantially different implementation.

[38] Vitalik Buterin, Ethereum White Paper (2014) <https://ethereum.org/en/whitepaper/> accessed 26 June 2022.

[39] In computer science, 'Turing complete' is a theoretical notion indicating that a system can compute anything that is computable in principle.

[40] Ethereum said to have moved blockchain technology to the second generation (Andrea Pinna, Simona Ibba, Gavina Baralla and Roberto Tonelli, 'A Massive Analysis of Ethereum Smart Contracts. Empirical Study and Code Metrics' (2019) IEEE Access <www.researchgate.net/publication/333682492_A_Massive_Analysis_of_Ethereum_Smart_Contracts_Empirical_Study_and_Code_Metrics> accessed 7 March 2023.

[41] We acknowledge that there have since been developments of many other Turing complete smart contracts languages using various approaches. A list of 32 different smart contract languages appears in Mark Daniels, 'All Smart Contract Languages' (2022) <https://medium.com/coinmonks/all-smart-contract-languages-2022-701afc8c0289>.

It is clear that the term 'smart contract' has been used indiscriminately by legal scholars and practitioners as well as computer scientists. To obfuscate the matter even further, in 2022, LawTech introduced yet another label – the 'smarter contract' – and commenced a Smarter Contracts Project.[42] 'Smart contract' as a term seems to be used as an all-encompassing label for 'legally binding digital contracts that are smarter than conventional contracts'.[43] LawTech goes on to differentiate between 'digital contracts', i.e. legal agreements in digital format, and 'smart legal contracts', which it intends to associate with legally enforceable contracts where some (or all) of the natural languages are represented by machine-readable code. LawTech deems this as a subset of 'digital contracts' which it states must be distinguished from '"smart contracts" that are simply coded persistent scripts'.[44]

The development from paper-based contracts to digital contracts (of varied iterations) has highlighted a common thread, namely that change is constant. The development (legally and relative to terminology) must therefore also be dynamic and adaptive. The definition of 'smart contract' offered by the Law Commission, whilst carrying a high-level understanding of the software used in smart contracts, does not identify all six different concepts and types uncovered by our research, as explained in section 3.3 below.

2.3 Origin of the Term 'Smart Legal Contract': a Misnomer?

The term 'smart legal contract' seems to have originated from Stark,[45] who observed in 2016 that the term 'smart contract' was defined in various manners: for example, as 'autonomous machines'; as 'contracts between parties stored on a blockchain'; or as 'any computation that takes place on a blockchain'. Stark noticed that, at times, the term was also used to refer to 'a specific application of that technology: as a complement, or substitute, for legal contracts'. In such a light, we should, Stark proposed, name these 'smart legal contracts'.[46] The first set of examples provided by Stark, are operational in nature, involving software agents.[47] Clack, Bakshi and Braine suggested that the word 'contract' in this sense indicates that 'these software agents are fulfilling certain obligations and exercising certain rights and may take control of certain assets within a shared ledger'.[48] They further point out that there was no consensus then (and it is argued here that there is still none now) on the definition of the computer code / software agent use of the term 'smart contract'.[49] In this regard we concur with Clack, Bakshi and Braine that each definition is indeed different in

[42] https://lawtechuk.io/programmes/smarter-contracts accessed 1 March 2023.
[43] https://resources.lawtechuk.io/files/introduction.pdf accessed 1 March 2023.
[44] https://resources.lawtechuk.io/files/introduction.pdf accessed 15 June 2022.
[45] Josh Stark, 'Making Sense of Blockchain Smart Contracts' (CoinDesk, 4 June 2016) <www.coindesk.com/markets/2016/06/04/making-sense-of-blockchain-smart-contracts/> accessed 15 June 2022.
[46] ibid.
[47] Computer programs that involve some autonomous action.
[48] Clack, Bakshi and Braine (n 11) 9.
[49] ibid 2.

subtle ways;[50] some of these are unpacked in section 3 below. Stark attempted to rename these software agents as 'smart contract code'.[51]

Stark used the term 'smart legal contracts' to label how contracts (used in its true legal sense) are expressed and executed in code – he perceived these 'as a complement, or substitute, for legal contracts'.[52] He reserved this label for smart contracts that deal with operational aspects, issues relating to how contracts are written and how 'legal prose' should be interpreted.[53] We disagree with this labelling, for reasons to which we now turn.

2.4 A Conflation

We submit that the term 'smart legal contracts' is legally tautological.[54] A 'contract' in law is binding. Thus, referring to a 'smart *legal* contract' is redundant. Similarly, to refer to 'a legal contract', while not completely incorrect, is not only superfluous but also confusing for the public. In their eyes, the result of the status quo is that a smart contract is, or can be, a legally enforceable agreement, but simultaneously, it can also be computer code (see section 6).

Compounding the issue further, and demonstrated below in section 4, a smart contract, as used by the computer science community, does not imply only one genre of code: it does not follow one set of criteria, but can be represented by several coding practices, and does not necessarily involve a third party to render it a smart contract code. Thus, at its most basic explanation it may be either interactive (engages with third parties) or non-interactive (does not engage with third parties). But in both instances, computer scientists would still refer to this code as a 'smart contract'.

Considering the above, it is submitted that the Law Commission should perhaps have taken the opportunity to disabuse the term 'smart contract' of its coding implications and adopted the term as having legal import, if for no other reason than to avoid obfuscation of the issue. In addition, and as mentioned above, the definition the Law Commission promotes does not cater for the different constructs identified in the coding environment and literature, highlighted in sections 3 and 4.

---

[50] ibid.
[51] Stark (n 45).
[52] ibid.
[53] Clack, Bakshi and Braine (n 11) 2.
[54] Wilkinson and Giuffre state that the term would be a tautology to lawyers, and while we agree with this statement, we disagree with the authors' approval of the justification of the term being used to differentiate the concept from a smart contract. They introduce the term 'smart contract' as though it is a definitively accepted single-use term (they do not posit which field). For the reasons demonstrated in this paper, it is not, and therefore the justification is not accepted (Susannah Wilkinson and Jacques Giuffre, 'Six Levels of Contract Automation: Further Analysis of the Evolution to Smart Legal Contracts' in Allen and Hunn (n 29)).

Clack, Bakshi and Braine, finding no clear consensus on the terminology used, identified the same dichotomy between 'smart contract code' and 'smart legal contract' and fostered the following definition:[55]

> A smart contract is an automatable and enforceable agreement. Automatable by computer, although some parts may require human input and control. Enforceable either by legal enforcement of rights and obligations or via tamper-proof execution of computer code.

They reasoned that such definition is 'sufficiently abstract to cover both "smart legal contracts" (where the agreement is a legal agreement, at least some of which is capable of being implemented in software) and "smart contract code" (which is automated software that may not necessarily be linked to a formal legal agreement)'.[56] Oddly, they then capitulated and adopted Stark's suggestions of the terms 'smart contract code' and 'smart legal contract' in their discussion.

The conundrum with this capitulation is two-fold: the first is that the term 'smart contract code' has no fixed definition and, as already shown, can mean more than one thing. Second, it begs the question, if code used in this context is a 'smart contract code' and a legally enforceable automated agreement is a 'smart legal contract', then what is a *smart contract?*

The challenge continues when one examines the literature where authors have favoured either one or the other or adopted their own 'hybrid definition'. Mik, for example, states that the term is used 'inconsistently'[57] and then proposes that 'smart contracts are technological means for the automation of payment obligations or obligations consisting in the transfer of tokens or cryptocurrencies'. This definition in our view is too narrow as it limits the capacity of smart contracts to automated payment systems. Smart contracts are capable of much more, and their technological competence is increasing. Mik takes the view that smart contracts 'are not agreements but means of performing obligations deriving from other agreements'.[58] That only takes us so far; what are the means? Did Mik intend computer code? If so, it would have been clearer to state such. Mik states that because of the lack of a standard definition of 'smart contract', 'it is impossible to generalise or make assumptions as to the universal attributes of "smart contracts"'.[59] We disagree and maintain a diametric opposing view. Mik adds that the UK Jurisdiction Taskforce (UKJT) did not define the term at all and criticises the Taskforce's 'legal evaluation of smart contracts', stating that 'it lacks a concrete point of reference'.[60] While we

[55] Clack, Bakshi and Braine (n 11) 9.
[56] ibid.
[57] Eliza Mik, 'Smart Contracts: A Requiem' (2019) 38 *Journal of Contract Law* 70.
[58] ibid. See also Alan Cohn, Travis West and Chelsea Parker 'Smart After All: Blockchain, Smart Contracts, Parametric Insurance, and Smart Energy Grids' (2017) *Georgetown Law Technology Review* 273, 276.
[59] Mik (n 57) 2–3.
[60] ibid.

acknowledge that this latter criticism is ambiguous and possibly subject to a different interpretation, we take the view that there is an important and relevant point made by the UKJT: the issue of automaticity.[61] In a later paper, Mik correctly posits that this research area (we add in both the legal and technical fields) is 'plagued by vague terminology, conflicting objectives and many untested assumptions'.[62] That should not, however, impede attempts to attach some defining agnostic characteristics to communally build a universal understanding of smart contracts. The UKJT's emphasis on automaticity as a characteristic is indeed a case in point.

The confusion has reached the point that companies are now proffering their own definitions of 'smart contract'. For example, Ethereum states on its website that smart contracts 'are computer programs stored on the blockchain that allows us to convert traditional contracts into digital parallels'.[63] Ethereum simultaneously attempts to appropriate them: '[a] "smart contract" is simply a program that runs on the Ethereum blockchain. It is a collection of code (its functions) and data (its state) that resides at a specific address on the Ethereum blockchain'.[64] Ironically, these two definitions seem to differ in meaning. The first refers to legal implications and the second to code. The conundrum is still not solved. In any event it should not be left to industries to try to set definitions for legally employable terms. This will create (or deepen) confusion, and at the same time lacks objectivity.

As mentioned, smart contracts are neither purely legal nor purely technological, but instead straddle the two disciplines. To enable an appropriate reconstruction both must consequently be considered.[65] We now turn to use of term by computer scientists' to unpack the technical understanding of it.

3. Decomposition of the Use of the Term 'Smart Contract' by Technologists

The Law Commission defined a smart contract as 'computer code that, upon the occurrence of a specified condition or conditions, is capable of running automatically according to prespecified functions'.[66] For a computer scientist, 'computer code' usually refers to 'source code', namely text that adheres to the syntax of a programming language that humans can understand, and that dictates the actions of a computer (see Figure 4). It can also refer to source code after it has been translated by a compiler and is set up for execution on a computer. The latter is referred to as

[61] We discuss this and other important characteristics in section 6.
[62] Eliza Mik, 'Contracts in Code Law' (2021) 13(2) *Innovation and Technology* 478.
[63] 'Introduction to Smart Contracts' Ethereum <https://ethereum.org/en/smart-contracts/> accessed 25 June 2024.
[64] ibid.
[65] Both must be considered to avoid what Herian refers to as the 'fetishisation of the contractual form by smart contract designers'. He adds that the problem is that 'smart contract designers consider the electronic form alone to be perfect' when it is not and herein lies the risk of 'fetishisation' (Herian (n 13) 268).
[66] Law Commission (n 2) vii.

'object code', or sometimes 'compiled source code', and is the meaning that the definition of the Law Commission is referring to. Broadly speaking, any code (not just smart contract code) will execute according to the way it is programmed (i.e. its 'prespecified functions'). If the code interacts with an environment (e.g. via inputs) then it reacts on its inputs in the way it has been programmed (i.e. checks for 'occurrence of a specified condition or conditions'). From a computer scientist's viewpoint, the Law Commission's definition refers to 'interactive object code', but it does not necessarily signify a smart contract.

In this section we unpack the uses of smart contracts by computer scientists. We discover that the computer science community uses the term 'smart contract' to describe six different types of artefacts in relation to tamper-proof executions of computer code, as listed below. We have adopted the convention of writing the six terms in bold, accompanied by the acronym 'SC' to indicate where the term 'smart contract' is used indiscriminately by technologists and their research community.

- **SC Source Code:** where 'smart contract' refers to source code prior to deployment to a blockchain (see Figure 4).
- **SC Object:** where 'smart contract' refers to compiled source code deployed to a blockchain but not yet executed within a transaction (see Figure 5).
- **Executed SC Function:** where 'smart contract' refers to code on the blockchain that has been executed within a transaction (see Figure 5).
- **Agreed SC Exchange:** describes the full set of **Executed SC Functions** needed to complete an agreement.
- **Contract SC:** where 'smart contract(s)' refers to a binding contract in the legal sense.
- **Generic SC:** used when referring to an abstract concept or general capabilities of a 'smart contract'.

The following section unravels these six types as follows. First, as technology is about application for practical purposes, we start by introducing, in section 3.1, a working example, called 'InPerpetuity', which we developed for the Ethereum blockchain for this paper. Although the example aligns to Ethereum, it captures the general concepts behind most blockchain platforms. Section 3.2 describes how to interact with the working example, thereby identifying the first four uses of the term 'smart contract' listed above. Finally, section 3.3 identifies all six types of uses of the term 'smart contract' in the computer science literature. The literature review classified occurrences of the term 'smart contract' within abstracts of a random sample of relevant technological papers. It revealed these different and indiscriminate uses of the term 'smart contract' in the literature, which even surprised the 'techies' amongst the authors.

3.1 Authoring and Reading our Example Smart Contract Code: `InPerpetuity`

Before our analysis, we describe the three main steps that typically occur when establishing an interactive system running on a computer (see Figure 1). First, a programmer writes source code to describe the behaviour of the system adhering to

the syntax of a programming language that humans can understand. Secondly, the source code is compiled to object code, a format that a computer can understand and execute, and deployed to a computer system. The third step is a suitable interface that allows a user of the system to interact with it. The second step meets with the definition of 'smart contract' proffered by the Law Commission, although below we work through an example of (specifically) a smart contract (not any generic code).

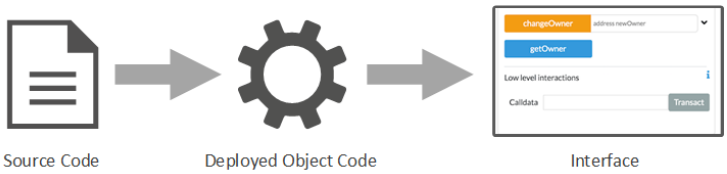

Figure 1: Typical parts to establish an interactive computer system

Figure 2: The hash "3841184ac44ed1ca31375cb2742dde47bb916d8e6" is generated from the artistic image seen here

The example used is of a non-fungible token (NFT), a specific example of an **SC Object**. NFTs are usually linked to digital assets.[67] For our NFT, the digital asset is the image seen in Figure 2, which is represented by a sequence of symbols generated from the image file using a cryptographic process (called hashing) to certify authenticity succinctly. The **SC Source Code**, `InPerpetuity` (see Figure 4), provides structure and functionality around this representation of the image, to store and transfer

[67] An NFT is a unique digital identifier stored on a blockchain, usually representing a unique digital asset (like artwork or music). NFTs are used to certify ownership and authenticity, where ownership can be transferred by the owner, allowing NFTs to be sold and traded.

ownership of the (still to be generated) NFT. The **SC Source Code** is then compiled into *compiled source code*, aka *object code* (see Figure 3).[68]

```
608060405260405180604001604052806 00d81526020017f496e205065
72706574756974790000000000000000000000000000000000000008152
50600190816200004a919062000349565b5060405180606001604052 80
602981526020016200092260299139600290816200007591906200034 9
565b506000600355348015620000885760008 0fd5b5033600080610100
0a81548173ffffffffffffffffffffffffffffffffffffffff02191690
8373ffffffffffffffffffffffffffffffffffffffff16021790555062
000430565b600081519050919050565b7f4e487b7100000000000000000
000000000000000000000000000000000000000000000600052604160045260
246000fd5b7f4e487b7100000000000000000000000000000000000000000
000000000000000000600052602260045260246000fd5b600060028204
905060018216806200015157607f821691505b60208210810362000167
... <continues for 71 more lines>
```

*Figure 3: Compiled **SC Source Code**, aka object code, in hexadecimal notation*

Only after the compiled **SC Source Code** is deployed to the blockchain does it convert to an **SC Object** which is able to certify ownership – et voilà, the NFT is born! Through this, the owner of the NFT can transfer ownership, allowing the NFT to be sold.

For the reader to gain an understanding of the authorship that implements the functionality of ownership and transfer of the NFT (step 1 in Figure 1), we explain the functioning of the **SC Source Code** below.

[68] While there is a technical difference between compiled **SC Source Code** and its deployed **SC Object**, for simplicity we refrained from introducing a separate name for compiled **SC Source Code**.

```
1  // SPDX-License-Identifier: GPL-3.0
2
3  pragma solidity =0.8.18;
4
5  contract InPerpetuity {
6
7      address payable owner;
8      string name = "In Perpetuity";
9      string hash = "3841184ac44ed1ca31375cb2742dde47bb916d8e6";
10     uint value = 0;
11
12     constructor() {
13         owner = payable(msg.sender); // Set the owner to sender address
14     }
15
16     function setValue(uint newValue) public {
17         require(msg.sender == owner, "Caller is not owner");
18         value = newValue;
19     }
20
21     function transferOwner() payable public {
22         require(value > 0, "Value still set to 0");
23         require(msg.value >= value, "Submitted value not high enough");
24         owner.transfer(msg.value);
25         owner = payable(msg.sender);
26         value = 0;
27     }
28
29     function getOwner() external view returns (address) {
30         return owner;
31     }
32 }
```

*Figure 4: **SC Source Code** written in the programming language Solidity*

The code starts on lines 1 to 3 with information needed for the Solidity[69] system to function correctly.[70] On line 5 the keyword `contract` and its name `InPerpetuity` are indicated. Four variables are then defined and used by `InPerpetuity` to store the required values for its correct execution, namely:

- `owner` (line 7) stores an address[71] that represents the current ownership of the NFT;
- `name` (line 8) stores the name of the NFT;
- `hash` (line 9) stores the hash of the related image (assures authenticity);
- `value` (line 10) stores a number to be set by the current owner as the price at which they are willing to sell the NFT.

The code listing in Figure 4 introduces four functions that define the behaviour of `InPerpetuity`, i.e. its interactive functionality when translated into an **SC Object**. To interact with an **SC Object**, it must be initialised. This initialisation occurs when the **SC Object** is deployed to the blockchain, as denoted by `constructor()` on line 12 in Figure 2. For `InPerpetuity`, the ownership of the NFT was set to

[69] Solidity <https://docs.soliditylang.org/en/v0.8.18/> is the popular smart contract programming language often used when writing **SC Source Code** for Ethereum and related blockchain platforms.
[70] The Solidity system includes a compiler to translate source code to object code.
[71] The address refers to a user's account on the blockchain.

whoever creates the `InPerpetuity` **SC Object**, that is, the address of whoever deploys its compiled **SC Source Code** to the blockchain.

The function `setValue()` on line 16 in Figure 2, allows the current owner (seller) to set a price. As with all **SC Objects** stored on a public blockchain, any user registered on the blockchain can try to interact with the NFT. Similarly, any user can use the `setValue()` function and pass a number `newValue` with it, aiming to set a price for selling the NFT. However, the code performs a test on line 17 which fails if the user account using `setValue()` differs from the value stored in `owner`. If this happens, the use of the function `setValue()` will also fail with no change at all of the blockchain system. If this test is successful, i.e., the code verifies the owner, the variable `value` is updated on line 18 to the number stored in `newValue`, the execution completes successfully and the blockchain system is updated with the new parameters.

Similarly, any user can use the smart contract function `transferOwner()` on line 21 from their user account address. In doing so, they also need to include some amount of Ether in their interaction, which is indicated by the keyword `payable` on the same line. The function then performs two tests on lines 22 and 23, namely that the current owner, i.e. the seller, has previously executed the `setValue()` function to set the variable `value` to a number bigger than `0`, and that the amount of Ether included in the execution of `transferOwner()` is at least as big as the content of the variable `value` held line 10 (the price). If one of those tests fails, the use of `transferOwner()` aborts with no change to the blockchain system.

If both tests are successful, the execution continues by passing the Ether on to the current owner, changing ownership to the user address who is using `transferOwner()`, and resetting value to `0`. The first step of this process is coded on line 24: `owner.transfer()` refers to the built-in mechanism for transferring Ether to the address `owner`; `msg.value` is a built-in variable containing the amount of Ether included in the use of `transferOwner()`. On line 25, the variable `owner`, which indicates the current owner, is updated to the address of the user account which has used `transferOwner()`, i.e., the purchaser. Finally, the variable `value` is reset on line 26, the execution completes successfully, and the parameters in the blockchain system get updated.

Execution of the last smart contract function, `getOwner()`, on line 29, reveals the current owner by showing the address stored in the variable `owner` when called.

Now that `InPerpetuity` as a **SC Source Code** is introduced, we turn to explain its use on the Ethereum blockchain system.

3.2 Using and Interacting with our Example Smart Contract: `InPerpetuity` on the Blockchain

To illustrate the use of `InPerpetuity` on the Ethereum blockchain system we consider two protagonists commonly used in the 'techie world', Alice and Bob: Alice

owns the work of art in Figure 2 and wants to turn it into an NFT; and Bob wishes to buy the NFT.

The live Ethereum Network is available to the reader via web links for each of the described concepts. For example, Alice's Ethereum account can be viewed at https://bit.ly/Alice-account,[72] and Bob's account at https://bit.ly/Bob-account.[73]

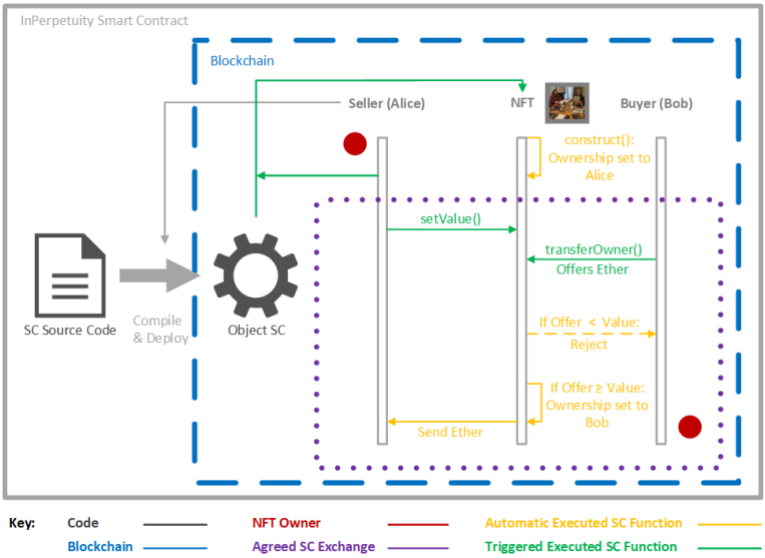

Figure 5: The process of creating, deploying and using smart contract code from a computer scientist's point of view

[72] Abbreviates https://etherscan.io/address/0xA11ce253Ea16b02b1F8e785D55247116adA28442.
[73] Abbreviates https://etherscan.io/address/0xB0b11B89957519F8278F85a8c7883D23da5a2899.

The use case for our example `InPerpetuity` is illustrated in Figure *5*. On the far left is the **SC Source Code** as described in section 3.1. Before being placed on the blockchain this **SC Source Code** is compiled and deployed into an **SC Object**, i.e., it has changed from a file that can be edited (by a programmer) to a file that is executable (by a user), like an application on a PC or phone. `InPerpetuity` is executable; it can be used by a non-technical user, by calling any of the available functions in the code. This is done by pressing a button on a software interface like an app; see Figure 6 for the user interface of the `InPerpetuity` **SC Object** on the Ethereum network.

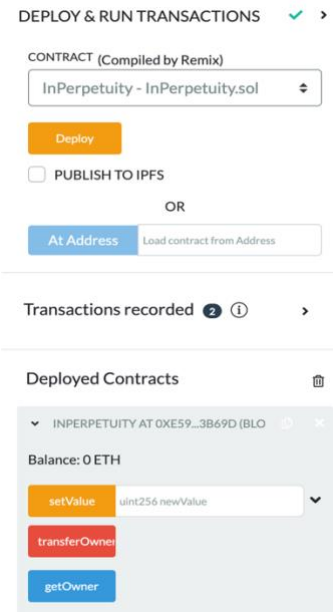

Figure 6: Ethereum network interface, with buttons for deploy and contract functions

In this case, the first user interaction of `InPerpetuity` happens when Alice deploys the compiled **SC Source Code** to the Ethereum blockchain, which creates the **SC Object** as an executable instance with a specific address on the blockchain (executable instance https://bit.ly/NFT-account[74]), i.e. the NFT. This executable **SC Object** provides the blockchain with the behavioural capabilities to automatically carry out several functions, as described in detail in section 3.1. Within the blockchain,

[74] Abbreviates https://etherscan.io/address/0xe59dbdd3d48059d2d9736e4b86eb73f40773b69d.

* Monica Vessio is Lecturer in Law, University of Reading; Arnold Beckmann is Professor of Computer Science, Swansea University; Matt Roach is Associate Professor in Computer Science, Swansea University; Séverine Saintier is Professor of Commercial Law, Cardiff University; Rhys Clements is Senior Cloud Compute and Storage Engineer, Swansea University; and Anton Setzer is Reader in Computer Science, Swansea University.

in Figure 5, the green and orange arrows represent the execution of a smart contract function, hereafter known as **Executed SC Functions**.[75]

In Figure 7, the full sequence of **Executed SC Functions** involved in the creation and trading of the `InPerpetuity` NFT is shown in a table. This table represents part of the ledger stored on the blockchain.[76] The combination of steps 2 and 3 is defined as an **Agreed SC Exchange** (depicted within the purple dashes in Figure 5). Following creation of the `InPerpetuity` NFT as an **SC Object** (step 1 in Figure 7), as the first step, Alice is setting a price of Ξ1,[77] using the `setValue()` function (step 2). Bob, willing to pay that price, sends the required amount to the NFT address, using the `transferOwner()` function (step 3). Bob is now the owner of the NFT and can set a new price – in this example he is offering the NFT for Ξ100 (step 4).

[75] We distinguish between two types of Executed SC Functions: triggered and automatic. 'Triggered' refers to the part that originates from a user account, 'automatic' to the part originating from a **SC Object**.

[76] To be precise, the table displays the transactions that Alice and Bob are sending to the Ethereum Network via the interface, and which are recorded on the Ethereum ledger. They all result in green edges in Figure 5.

[77] The symbol for the cryptocurrency *Ether* is still under debate; here we will use Ξ.

Executed Function Ledger		Worldstate			
Step	Action Executed SC functions	NFT Owner	NFT Value	Alice Account	Bob Account
1	Alice deploys InPerpetuity[78]	Alice	0	0	1
2	Alice setValue(1 Ether)[79]	Alice	1	0	1
3	Bob transferOwner() & 1 Ether[80]	Bob	0	1	0
4	Bob setValue(100 Ether)[81]	Bob	100	1	0

Figure 7: The steps of an example interaction between two actors deploying and transferring ownership using the `InPerpetuity` *smart contract code on a blockchain*

This specific implementation and interaction of the `InPerpetuity` code has demonstrated the many ways in which the term 'smart contract' can be used. To complete this analysis, we now turn to the use of the term 'smart contract' in the computing literature

3.3 Decomposition of the Use of the Term 'Smart Contract' in the Computing Literature

To consider how the term 'smart contract' is used within the wider computer science community, we analysed trend data from peer-reviewed papers published in computer science literature in the years 2019–2024. The results confirm the indiscriminate use of the term 'smart contract' by technologists.

[78] https://bit.ly/Alice-deploysSC; abbreviates https://etherscan.io/tx/0x73cb23ea356b08ab1546cc13d26e675766dcb1bdcfdf261ad3505ef878dfc87c.
[79] https://bit.ly/Alice-setValue; abbreviates https://etherscan.io/tx/0xc574ee186bf46afaa45339cb4fc859d1a094cb33313dc216907c71c1f14d2100.
[80] https://bit.ly/Bob-transferOwner; abbreviates https://etherscan.io/tx/0xe39df9c53dc7679017883263c027271c33890df5bdbf2423ba44bdde403b42aa.
[81] https://bit.ly/Bob-setValue; abbreviates https://etherscan.io/tx/0x3a01fbee52ce3edcbb4b1261f95e97987b1921c5505b93092ad68d07554104e6.

$$\text{TITLE} - \text{ABS} - \text{KEY}(\text{"smart contract" OR "smart contracts"})$$
$$\text{AND PUBYEAR} > 2019 \text{ AND PUBYEAR} < 2025$$
$$\text{AND (LIMIT} - \text{TO(SUBJAREA, "COMP"))}$$

Figure 8: Search term for Scopus database

The review was carried out through the Scopus database with a refined search term as defined in

Figure 8. A total of 10,607 papers were identified, from which a pseudo-random 5% sample of 542 papers were selected. Four reviewers analysed the title and abstract of each paper within the sample to determine for each occurrence of the term 'smart contract' which of the more detailed defined concepts this use was referring to. After reviewing several articles together, to calibrate and confirm the understanding of those definitions, each reviewer reviewed papers independently, conferring with colleagues if there was an edge case or a difficult classification to be made. Each use of the term 'smart contract' was counted and classified. However, the results presented inFigure 9 were normalised per paper; that is, each paper was weighted equally no matter how frequent the use of the term 'smart contract' therein.

Below are the concept definitions that were introduced in section 3.2, with additional examples discovered during the reviewing process, providing the trend data:

- **SC Source Code** (Figure 4): Typical uses found in the literature include: '…given a high-level description of a cross-transaction can automatically generate smart contracts in Solidity…',[82] and '…present a synthesis tool called XCHAIN that […] can automatically generate smart contracts in Solidity'.[83]
- **SC Object** (Figure 5): Typical uses found in the literature include: '…shielded computations in Smart Contracts Overcoming Forks',[84] and 'Smart contracts are abstract pieces of codes'.[85]
- **Executed SC Function** (Figure 5): Typical uses found in the literature include: '…smart contract executions are consuming blockchain resources',[86] and 'the

[82] Narges Shadab, Farzin Houshmand and Moshen Lesani, 'Cross-chain Transactions', (2020) IEEE International Conference on Blockchain and Cryptocurrency, ICBC <https://ieeexplore.ieee.org/document/9169477> accessed 25 June 2024.
[83] ibid.
[84] 25th International Conference on Financial Cryptography and Data Security, FC 2021, Lecture Notes in Computer Science, 12675 LNCS.
[85] Vishakh Rao, Ankur Singh and Bhawana Rudra, 'Ethereum Blockchain Enabled Secure and Transparent E-Voting' (2021) 1290 *Advances in Intelligent Systems and Computing* 683 <https://link.springer.com/chapter/10.1007/978-3-030-63092-8_46> accessed 25 June 2024.
[86] Serdar Metin and Can Özturan, 'Max–min Fairness Based Faucet Design for Blockchains' (2022) 131 *Future Generation Computer Systems* 18 <https://doi.org/10.1016/j.future.2022.01.008> accessed 25 June 2024.

transaction being executed by a smart contract needs to be reverted to avoid undesirable consequences'.[87]

- **Agreed SC Exchange** (section 3.2): Typical uses found in the literature include: 'blockchain-based peer-to-peer sustainable energy trading in microgrid using smart contracts',[88] and '…smart contracts are executed among telecom providers as bidders, and government authorities as auctioneers'.[89]
- **Contract SC**: Typical uses found in the literature include: '[t]he proposed solution […] enables the owner of the Will to deploy a smart contract mentioning his wishes'[90] and '… legal advantages and disadvantages of the legal regulation of using smart contracts in civil circulation from the perspective of applicable law'.[91]
- **Generic SC** (argued in section 3 as a confused use of the term): Typical uses found in the literature include: '…Healthcare Record (EHR) system that is layered on the Ethereum blockchain platform and smart contract in order to eliminate the need for third-party systems',[92] and '…smart contract of blockchain as a trusted authority to fairly evaluate contributions and allocate rewards'.[93]

---

[87] Lu Liu, Lili Wei, Wuqi Zhang, Ming Wen, Yepang Liu and Shing-Chi Cheung, 'Characterizing Transaction-Reverting Statements in Ethereum Smart Contracts' Proceedings - 2021 36th IEEE/ACM International Conference on Automated Software Engineering, ASE 2021 (2021) 630–641, <https://doi.org/10.1109/ASE51524.2021.9678597> accessed 25 June 2024.

[88] ICREST 2021 – 2nd International Conference on Robotics, Electrical and Signal Processing Techniques (2021).

[89] Farnazbanu Patel, Pronaya Bhattacharya, Sudeep Tanwar, Rajesh Gupta, Neeraj Kumar and Mohsen Guizani, 'Block6Tel: Blockchain-based Spectrum Allocation Scheme in 6G-envisioned Communications' 2021 International Wireless Communications and Mobile Computing (IWCMC), Harbin City, China, 1823 <https://doi.org/10.1109/IWCMC51323.2021.9498854> accessed 18 October 2023.

[90] Jainam Chirag Shah, Mugdha Bhagwat, Dhiren Patel and Mauro Conti, 'Crypto-Wills: Transferring Digital Assets by Maintaining Wills on the Blockchain' in Bansal, J, Gupta, M, Sharma, H and Agarwal, B (eds), *Communication and Intelligent Systems. ICCIS 2019. Lecture Notes in Networks and Systems* vol 120 (Springer 2020) 407 <https://doi.org/10.1007/978-981-15-3325-9_31> accessed 18 October 2023.

[91] VV Popov and DP Strigunova, 'Advantages of Smart Contracts in Civil Circulation and Their Legal Regulation Disadvantages' in Ashmarina, SI, Mantulenko, VV and Vochozka, M (eds), *Proceedings of the International Scientific Conference 'Smart Nations: Global Trends In The Digital Economy'*. Lecture Notes in Networks and Systems, vol 397 (Springer, 2022) 81 <https://doi.org/10.1007/978-3-030-94873-3_11> accessed 18 October 2023.

[92] Tomilayo Fatokun, Avishek Nag and Sachin Sharma, 'Towards a blockchain assisted patient owned system for electronic health records' (2021) 10(5) *Electronics* 580, 1–14 <https://doi.org/10.3390/electronics10050580> accessed 18 October 2023.

[93] Chunxiao Li, Xidi Qu and Yu Guo, 'TFCrowd: a Blockchain-based Crowdsourcing Framework with Enhanced Trustworthiness and Fairness' (2021) *Eurasip Journal on Wireless Communications and Networking 2021* article 168 <https://doi.org/10.1186/s13638-021-02040-z> accessed 18 October 2023.

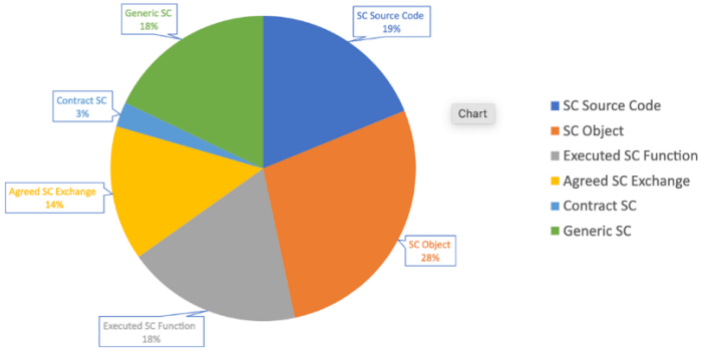

Figure 9: Uses of the term 'smart contract' in computer science literature, 2019–2024

The results of this analysis are illustrated in Figure 9. The usage distribution is surprisingly even: approximately 20% for the majority of definitions. Many authors (18%) use the phrase abstractly, i.e., **Generic SC**. The largest use of the term (28%) refers to **SC Object**. One might have expected **Agreed SC Exchange** to be the most frequent since this is the essence of a completed automated smart contract, without explicitly recognising the legal requirements to make it a legally binding agreement. Surprisingly this only represents 14% of the usage. In numerous instances (16.7%), distinct uses of 'smart contract' within a given paper referred to different concepts. As seen in Figure 9there are very few uses (3%) with an explicit appreciation of the legally binding associations with the term 'contract', that is, **Contract SC**.

The lack of consensus over the meaning of the term 'smart contract' within the computer science community is now beyond doubt as it is clearly illustrated through our `InPerpetuity` use case and the trend data analysis. Both instances show that the computer scientists use the term 'smart contract' to name at least six different types of contract codes. Other smart contract concepts beyond the six types identified here are likely to be identified over time. For example, one already perceivable concept could be called *Opaque SC Objects*, which would be a variant of **SC Objects** where all or part of the object code is represented by its hash. For such an *Opaque SC Object*, a successful interaction would require passing the object code matching the hash.[94] We have decided to not list this and other perceivable concepts as part of our findings, as such a list would never be complete (new concepts will be discovered over time). Instead, we followed a strict methodology to identify concepts based on our `InPerpetuity` example and the uses in scientific literature.

[94] A related feature is implemented in Bitcoin's 'pay to script hash' <https://bitcoinwiki.org/wiki/pay-to-script-hash> accessed 6 June 2024.

This array of uses is regrettably not shown by the Law Commission. Although it does provide some discussion on code[95] its explanations of, for example, source code and object code does not demonstrate a holistic understanding of the various iterations of smart contracts.

Now that the lack of consensus between the legal and computer science community is exposed, we turn to how society has reacted or absorbed the term 'smart contract'.

4. Trends in the Use of the Terms 'Smart Contract(s)' and 'Smart Legal Contracts'

Current widely accepted definitions of 'smart contracts' can be found on Wikipedia:[96] '[a] smart contract is a computer program or a transaction protocol which is intended to automatically execute, control or document legally relevant events and actions according to the terms of a contract or an agreement'. Additionally, Bitcoin and Ethereum are included in the page's 'list of blockchain platforms supporting smart contracts'. Presently, and we suggest tellingly, no corresponding Wikipedia page exists for 'smart legal contract'.

Figure 10 shows a comparative display of the Google searches (interest shown over time[97]) carried out worldwide on the terms 'smart contract' (orange) and 'smart legal contract' (blue) from 1 January 2015, until 6 June 2024, from Google Trends.[98] This roughly shows what people (i.e. society at large) are searching for in relation to smart contracts. They are certainly not searching for 'smart legal contract', as the statistics in Figure 10 show. We capture and discuss the public's qualitative understanding of these terms in section 5 below.

[95] See Law Commission (n 2) 'An Introduction to Code' 8–11.

[96] Wikipedia (2024). The authors do not propose Wikipedia as a source of academic authority, rather it is used here as indicative of global societal trends outside of strict legal and technical paradigms.

[97] Numbers represent search interest relative to the highest point on the chart for the given region and time. A value of 100 is the peak popularity for the term. A value of 50 means that the term is half as popular. A score of 0 means that there was not enough data for this term <https://trends.google.com/trends/explore?date=all&q=smart%20contract,smart%20legal%20contract> accessed 12 July 2022.

[98] Google Trends is an open site owned by Google that publicises searches made by users from 2004 <https://trends.google.com/trends/> accessed 12 July 2023.

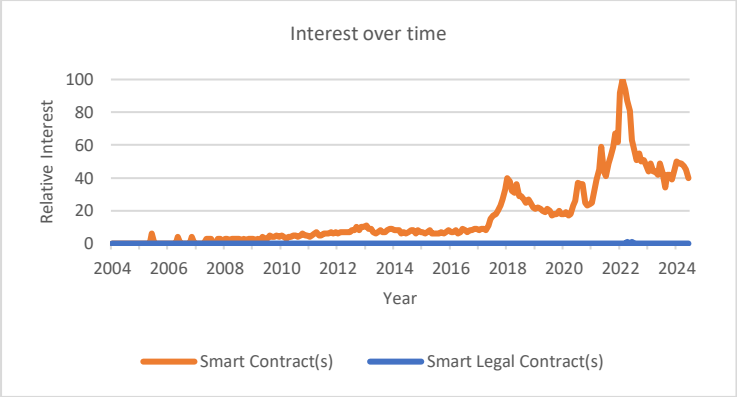

Figure 10: Google searches carried out worldwide for 'smart contract' and 'smart legal contract' from 1 January 2004 to 6 June 2024 (from Google Trends)

Following on from general use trends, we examined the history of the use of the two terms in academic literature using Elsevier's Scopus abstract and citation database. The precise search terms used are found in Figure 11.

$$\text{TITLE} - \text{ABS} - \text{KEY}(\text{"smart contract" OR "smart contracts"}) \text{ AND PUBYEAR} > 1979 \text{ AND PUBYEAR} < 2025$$

$$\text{TITLE} - \text{ABS} - \text{KEY}(\text{"smart legal contract" OR "smart legal contracts"}) \text{ AND PUBYEAR} > 1979 \text{ AND PUBYEAR} < 2025$$

Figure 11: Scopus Search Terms used for 'smart contract(s)' (top) and 'smart legal contract(s)' (bottom)

Figure 12 shows the usage of both terms over time. Of note, the scale of the terms' usage is substantially different (three orders of magnitude), with 'smart contract(s)' (in orange) peaking at 2,553 papers (left vertical axis) papers in 2021 and 'smart legal contract(s)' (in blue) peaking at eight papers (right vertical axis) in the same year. Szabo's seminal paper in 1996 is the first occurrence of the term 'smart contract' in peer-reviewed literature, followed by a period of little usage before a significant increase from 2016 onwards following the introduction of Ethereum.[99] These findings differ from the findings relating to 'smart legal contract(s)'; this term was not used until 2019, even then only 18 documents that used this term were found in the database.

[99] Similar deduction drawn by Mik (n 57) 71.

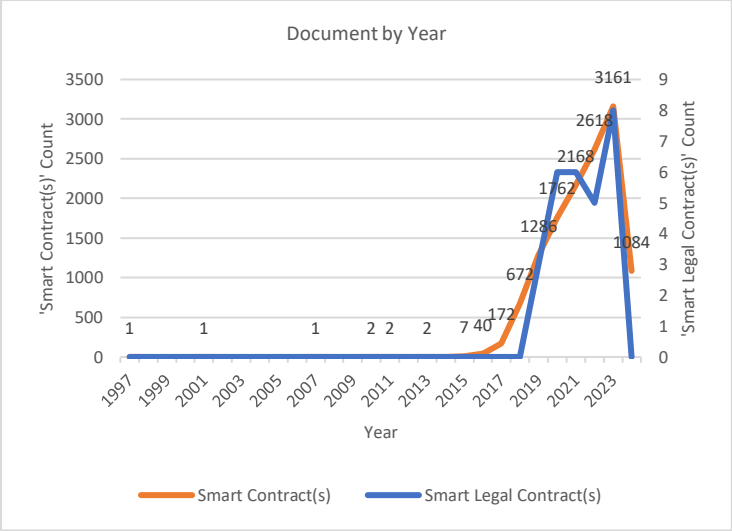

Figure 12: Usage of the terms 'smart contract' and 'smart legal contract' in scientific literature in computer science (note different scales)

When comparing the use of each term in relation to subject area (see Figure 13), differences also appear. While both show their use to primarily be within the computer science area ('smart contract' has 37.2% of its use and 'smart legal contract' has 28.4% of its use here), the second highest use of 'smart legal contract' was found to be within the field of social sciences, whereas for 'smart contract' it is within the field of engineering. This is notable as the term 'smart contract' only has 4.2% of its use within the field of social sciences, whilst 'smart legal contract' has 23.2%.[100]

The statistics show that 'smart legal contract' is relatively nascent and that 'smart contract' is used inconsistently within the computer science fields. This important finding supports the argument that the terms are not yet entrenched in the literature. There is therefore time to disabuse the incorrect use of the term 'smart contract' in the computer science field and appropriate the term to the legal field, removing the need for tautological labelling and harmonised use.

[100] Another noteworthy finding is how each respective term is used in the UK. Statistics show that of the documents found that use the term 'smart legal contract', only 38 originate from the UK. However, of the documents that use the term 'smart contract', 157 originate from the UK.

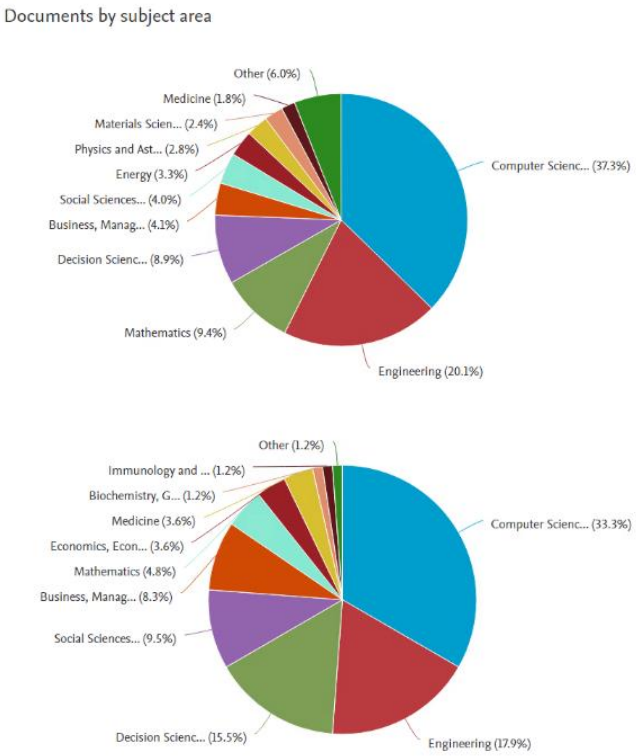

Figure 13: Usage of the terms 'smart contract(s)' (top) and 'smart legal contract(s)' (bottom) in academic literature in various subjects

We now turn to the general public's understanding of the term 'smart contract', the final element of our enquiry.

5. Public Understanding of the Term 'Smart Contract'

We carried out a survey in November and December 2022 to elicit the general public's understanding of the term 'smart contract'. Data was collected by approaching members of the public in five UK cities (London, Bristol, Swansea, Exeter and Plymouth) and online outreach. The age range of respondents is illustrated in Figure 14; 44% of participants were female, 53% were male, and 'non-binary', 'prefer to self-

describe' and 'prefer not to say' represented 1% each. A total of *n=105* responses were collected.

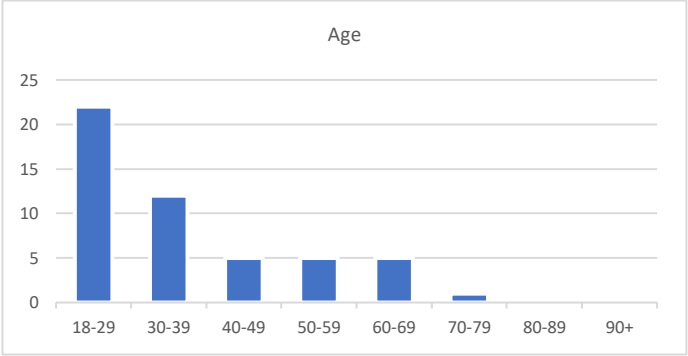

Figure 14: Age demographics of those surveyed

The first question posed was 'What do you understand by the word "Contract"?' A thematic analysis of the free text responses revealed that:

- 67.2% of responses understood it as an act of 'agreement';
- 56.9% explicitly referred to it as 'legal' or 'binding' (often legally binding);
- 35.3% of responses identified 'two parties' or a 'relationship';
- 23.5% referred to a contract as a 'document'.

Two less common concepts were the contract as an 'obligation' (13.7%) and the definition of 'terms' (15.7%).

We also asked, 'Do you believe that a "Contract" is legally binding?'. The results displayed are in Figure 15 and discussed later in this section.

The public was also asked about smart contracts. Knowledge of the term was low, with 88% declaring they did not know what it was, and 62% unsure if it would affect them in the future. The public's understanding of smart contracts was sought; we asked them: 'With your current knowledge, please describe what you understand a "Smart Contract" to be.' Those that did not know what it was were asked: 'From the wording of "Smart Contract", please provide your best guess on what the term might mean'. From the majority that guessed (self-declared did not know what a smart contract was) two strong conceptual themes emerged: smart contracts were technical or digital in nature (27.5%), and they were flexible or personalised (23.5%). Many respondents remained unsure or could not provide a response to this question (35.3%). From the few respondents who self-declared that they knew what a smart contract was, the responses ranged from one that simply stated, 'A contract' – resonating with the lawyers' views – to '[a] program which when executes makes

changes on the blockchain' – a more recognisable response for the computer scientists. Responses also included erroneous understandings e.g. 'A digital contract that help reducing environmental impact'.

The public was also asked whether a contract was legally binding, and the responses are shown in Figure 15. The public understanding of the legally binding nature of a contract can be seen in blue, with the vast majority (78%) declaring they believe a contract is often or always legally binding. For smart contracts (shown in orange) a significant minority of people (42%) declared they did not know if a smart contract is legally binding. However, it appears from the remainder that 'smart contract' and 'contract' are understood in similar ways.

The hypothesis (H_0,) was tested that the public's understanding of the legally binding nature of 'contract' and 'smart contract' is independent (different). With the alternative hypothesis (H_1), the public's understanding of the legally binding nature of 'contract' and 'smart contract' is dependent (the same). Since non-parametric categorical data was used, a Chi-squared test was applied with a confidence threshold set at 99.9%. The Chi-squared test static returned is 73.29 with a critical value of 44.314, resulting in a p-value less than the threshold set ($p = 0.000001 < \alpha = 0.001$, $n = 105$). In this test, the evidence to reject the null hypothesis H_0 and conclude that the distribution of the public's responses to the two questions is dependent was present. Therefore, it is proposed that this confirms that 'smart legal contract' is tautologous since the public's understanding of 'smart contract' already includes legal inference, as is the case with 'contract'.

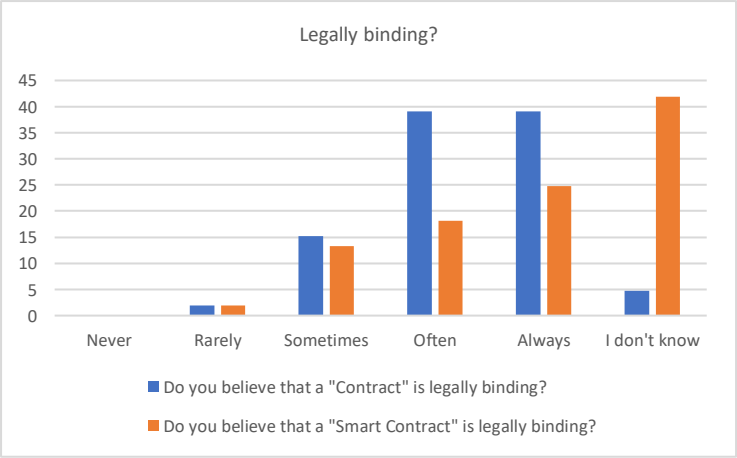

Figure 15: Public's perception of the legally binding nature of 'contract' and 'smart contract'

We started this enquiry with the one question, what is a smart contract? Given the multiplicity of answers we have highlighted, and to make sense of this field, which straddles two disciplines, we need to find a common lexicon that combines and represents a common understanding between the computer and the legal community. Let us turn to it.

6. A Synergistic Understanding: Mining the Consensus

As shown above, there is no single definitive use of the term 'smart contract' by the computer science and legal communities, as well as society at large. The responses to the questions we posed to the public showed that while most people in general understand what is intended when using the term 'contract' (i.e. that there is a legally binding element), most people do not have an understanding or are unable to explain what a smart contract is. However, a significant percentage of respondents are aware that smart contracts may affect them in the future.

In the computer science community, the use of the term 'smart contract' has been indiscriminate; as shown in section 3, it has been used to refer to several concepts in generic, source code, object code and so on. These uses are equally significant, making it impossible to prioritise them in order to justify assigning the term to that iterative aspect.

Whilst it may be possible to fully realise a legally binding agreement through code, none of these concepts (as used in the science community) are capable of individual legal enforcement as contracts, although the various iterations of the 'smart contract' implementations may form sub-parts of a legally enforceable digital agreement.

There have been many attempts to digitise contractual agreements. Even with blockchain and other cryptographic technological capabilities, the complete smart contract (self-implementing) has yet to be born. There is hardly any doubt, however, that through technological advancements the possibility of automating contractual terms activated by meeting conditions will formulate more efficient autonomous contracting environments.[101] This paper has not attempted to solve the technological difficulties, but rather to contribute a plausible start to solving the problems of terminology. As the computer scientists have appropriated the term 'smart contract' but used it sporadically for different functionalities, the legal community too has attempted to introduce alternative terms such as 'smart legal contract', 'digital contract' and, most recently, 'smarter contracts' as speculative but, as argued here, inappropriate solutions.

[101] The Law Commission has indicated that smart contracts are presently only likely to be useful in respect of rudimentary agreements, for example transferring cryptocurrency. However, it posits that, as technology becomes increasingly sophisticated, it will be able to accommodate a greater range of obligations, 'resulting in these contracts becoming increasingly more complex and able to perform a greater range of tasks' (Law Commission (n 2)) 1.

It is this obfuscation that our work attempts to clear. We indicated that 'smart contract' should only be used to describe a contract and should therefore, irrespective of the degree of automaticity or level of cryptography (which we posit as plausible characteristics), be legally enforceable. This understanding underpins the general public's comprehension of what contracts are as social constructs. The need to clarify the labelling use of 'smart contract' now is not only important for research and for the public, but also for industry players.

The way that 'smart contract' is used in computer science is essentially a subpart of the term 'contract', namely, a way to refer to 'pieces' of technology. During this research, the computer science authors have come to acknowledge, and the legal academics concur, that unless it is a fully formed contract (i.e., a legally binding one), then computer scientists should avoid the use of the term (in whichever and all iterations previously used). The computer scientists have made suggestions as to what these functionalities could be labelled (section 3).

The Law Commission originally attributed three features to a smart contract.[102] The first is that some or all the obligations under the contract are performed automatically by a computer program, identifying automaticity as a characteristic. The second is that the contract must be legally enforceable; here, it gives weight to the word 'contract' and with it its legal import. The third relates distributed ledger technology DLT.[103] In the call for evidence, the Law Commission confined its paper to smart contracts which use DLT; however, after considering consultee responses, it revised its approach and no longer considers DLT to be an essential feature of smart contracts.[104] In its final paper the Law Commission referred to 'smart (legal) contracts', but as explained above , we do not agree with this label. We agree that 'limiting the definition to DLT is unnecessarily restrictive'.[105] We believe that smart contracts should be technology-agnostic, i.e. regulation developed for digital automated contracting should not be tied to any type of technological iteration, such as but not limited to DLT.

We demonstate the point by using DLT as an example: a company may offer a centralised smart contract that is not on a DLT network. For example, car rental

[102] Law Commission, *Smart Contracts A Call for Evidence* (December 2020) para 2.4.
[103] At its very basic, DLT is a technology that allows concurrent access, validation and record updating across a networked database. The Law Commission defines DLT as a 'digital store of information or data, that is shared or distributed amongst a network of computers which may be available to other participants' and asserts that 'the distinguishing feature of DLT compared to traditional, centralised databases is that the ledger is not maintained or controlled by a central administrator or entity' (ibid 14).
[104] Law Commission (n 2) 1 para 2.32 and 2.47.
[105] ibid 17. The elements identified in the Law Commission consultation paper have not been adopted in its *Advice to Government* (n 2). While the latter paper goes to great lengths to explain the features it does not bring the law in the UK any closer to having a definition for smart contracts.

Company A offers its services at an automated teller at the airport. Person B approaches the teller, enters the agreement, payment is transferred, and a deposit is taken. At this point, person B is provided with a code that will open an on-site security box from which B can retrieve the keys for the vehicle. Person B does so and drives off. The contract is terminated when the car is returned by person B to an approved location, the keys are locked in the indicated safe and the deposit released (presume camera and onboard AI approve that no damage/accident or speeding occurred). There is no need in such an automated contractual environment for participants to approve and synchronise additions to the ledger through any form of agreed consensus mechanism.

In other words, a DLT environment is not necessary to facilitate this smart contract. It is our contention that similar rules of application and interpretation will apply to this type of contract that would apply to a contract that is on DLT, thereby obfuscating the need for DLT as a necessary characteristic. It is further suggested that the technology should also remain independent of blockchain. This is because automated contracting technology has the potential to evolve; and whilst a different type of technology that facilitates automated contracts may not currently be conceivable, establishing an adaptable regulatory environment while the opportunity exists would be valuable, particularly considering the flexible nature of the common law (or prevention of legislative tie-in with civil law systems). A forward-thinking approach would circumvent practices from becoming quickly outdated.

Our proposal is therefore that the most logical meaning that can be assigned to smart contracts is one that incorporates only universal technology-agnostic characteristics. Of these, we recognise three. First, that a smart contract is a digital representation of terms and conditions (unlike a static pdf). This allows fluidity of data.[106] Second, that its performance is automated or part-automated. Third, that it is a legally binding agreement – as seen above, the law assigns specific meaning to the term 'contract'. Therefore, at its essence, *a smart contract is an automated or part-automated digital contract*. This definition is sufficiently broad whilst adequately flexible to cater for standard contractual transactions as well as sector-specific application and future technological innovations.

7. Conclusion

This paper was born out of robust discussions over the use of the term 'smart contract' by lawyers and computer scientists. Whereas the computer science community use it as a description of something static (noun) – the item that performs automatic agreement and makes a record of it – the lawyers use their various iterations to describe a dynamic thing (verb) – the act of agreeing between two parties. This lack of consensus between the lawyers and the scientists, far from being

[106] Explained at section 2.1 above.

purely of academic interest, has wider ramifications. Indeed, given the steady increase of the deployment of smart contracts in society, it is important, from a normative and regulatory viewpoint, to have a clear and common understanding of the meaning of the term for the public as well as industry stakeholders. A common term, legally sanctioned, serves as a robust instrument for accuracy, consistency and alignment across various disciplines. Given the dual nature of smart contracts, it is important for the putative definition to reflect the distinct nature of the socio-technical relationship that the law attempts to regulate.[107] This was the main aim of this paper.

Linked to this, it was also important to shift the universal view to recognise the misappropriation of the term 'smart contract', due to its use in diverse ways by the computer science community, as well as to remove the tautology of the use of the word 'legal' by some of the legal community. During our research, we found, however, that to stop here was not sufficient. It was necessary to go beyond fostering a common lexicon between the legal and computer science communities, to establishing the foundations of that very nomenclature. This was necessary to combine our understanding of the legal and technological approaches and intervene in an emerging ecosystem.[108] In the absence of such harmony, these two aspects would battle to take the upper hand and one would have to submit before the other takes over.[109] Schrepel suggests that this 'would push smart contracts to develop under the primary influence of either law or technology, depending on which one dominates during a given period'[110] and would lead to smart contracts acquiring unbalanced characteristics, for example, by ignoring legal constraints altogether. What we have suggested here is a mutual covenant.

Completely discarding the term 'smart contract' because it promises 'too much in both respects – being sometimes not smart and sometimes unlike a contract' is not, in our view, constructive.[111] Rather than pulling at the term, splitting it up and inserting redundancies, we have proposed, through findings from harmonised cross-discipline collaboration, that the term 'smart contract' be appropriated to its more natural field: the legal one. This avoids depriving the word 'contract' of its implicit enforceability implications. At the same time, it realigns computer scientists' labelling, moving away from the use of the term for myriad coding iterations, and for technological functionalities that are not contractual in nature, nor which resemble contractually enforceable arrangements.

---

[107] Brownsword states that the best approach for the law is to be technocratic (Roger Brownsword, 'Law disrupted, Law Re-Imagined, Law Re-Invented' (2019) 1 *Technology and Regulation* 10, 15).

[108] Idea adopted from Schrepel (n 12) 14.

[109] ibid.

[110] ibid.

[111] Felten refers to smart contracts as 'virtual objects' that are 'nothing more or less than mindless mechanisms' (Ed Felten, 'Smart Contracts: Neither Smart nor Contracts?' (Freedom to Tinker 20 February 2017) <https://freedom-to-tinker.com/2017/02/20/smart-contracts-neither-smart-not-contracts/>.

Without the proposed realignment, the smart contract matrix will continue to have the two disciplines pull at each other with sustained efforts at labelling and relabelling by jurists as they try to untangle the conundrum of the recurrent multiplications of use. We have therefore recommended that smart contracts be recognised through universal and technology-agnostic characteristics. That is, that the term 'smart contract' be used to mean *a digital representation of legally binding terms and conditions, the performance of which is automated or part-automated*. This allows for an application to a myriad of contractual environments, both private (banking, commercial, insurance etc.) and public (government, procurement etc.), and a harmonised but supple application of the term, one which can absorb future technological incarnations.