
Smart Contracts as Techno-Legal Regulation

Peter GL Hunn

Accord Project, UK
E-mail: peter@accordproject.org

Received 18 April 2019;
Accepted 27 June 2019

Abstract

Smart contracts on a blockchain network can be implemented to control digital value. A key question that arises is the extent to which smart contracts can, or should, operate as “smart legal contracts”. Simply put, can smart contracts meet requirements of validity at law and practical efficacy. In order to achieve the goal of value maximization, the efforts of policy-makers, standards organisations and regulators should be informed by first principles. Standards, and other regulatory activities, must be driven by consideration of the techno-legal functions of contracting. Blockchain-based smart contracts offer the potential to reduce transaction costs through new methods of stateful computation. When applied to commercial transactions, smart contracts can represent enforcement of an executed state. This paper argues that distributed ledger and smart contracts standards should seek to provide sufficient flexibility to facilitate contracting parties to coordinate in an optimal manner.

Keywords: Distributed ledger technologies, Blockchain, Smart Contracts, Digital value.

Journal of ICT, Vol. 7.3, 269–286. River Publishers

doi: 10.13052/jicts2245-800X.735

This is an Open Access publication. © 2019 the Author(s). All rights reserved.

1 Introduction

The proliferation in the implementation of distributed ledger technologies¹ over the last decade has resulted in an increased focus on, and realization of, the promise of “smart contracts”². The term is invariably used to refer to scripts that run in a persistent manner on a blockchain system, often controlling digital value, with the purpose of optimizing methods of economic coordination.

Beyond ‘cryptoassets’³, a notable line of analysis considers the extent to which smart contracts can, or should, operate as “smart legal contracts”⁴. Much of this assessment is predicated upon function following form; namely, the extent to which the functionality of a “smart contract”, so defined, can meet requirements of validity at law and practical efficacy⁵. Numerous standards⁶,

¹See Xu *et al.*, ‘Architecture for Blockchain Applications’, (Springer: New York, 2019), 45–59.

²The concept of “smart” or “computable” contracts is not particularly new, see: H. Surden, ‘Computable Contracts’ (2012) 46 UC Davis Law Review 629; N. Szabo, ‘Smart Contracts: Building Blocks for Digital Markets’ (1996) 16 *Extropy*, available at: http://www.fon.hum.uva.nl/rob/Courses/InformationInSpeech/CDROM/Literature/L_OTwinterschool2006/szabo.best.vwh.net/smart_contracts.2.html (Accessed May 2019).

³See Financial Conduct Authority, ‘Guidance on Cryptoassets’ (Financial Conduct Authority, CP19/3, 2019), available at: <https://www.fca.org.uk/publication/consultation/cp19-03.pdf> (Accessed May 2019).

⁴See: K. Werbach & N. Cornell, ‘Contracts Ex Machina’ (2017) 67 *Duke L.J.* 313; P. De Filippi & A. Wright, ‘Blockchain and the Law: The Rule of Code’ (Harvard University Press: Boston, 2018);

UK Jurisdiction Taskforce of the LawTech Delivery Panel, Public Consultation: The Status of Cryptoassets, Distributed Ledger Technology and Smart Contracts under English Private Law. Available at: <https://www.lawsociety.org.uk/news/documents/ukjt-consultation-cryptoassets-smart-contracts-may-2019/> (Accessed May 2019); Cardozo Blockchain Project, Research Report #2, “‘Smart Contracts’ & Legal Enforceability’ (16 October 2018). Available at: <https://cardozo.yu.edu/sites/default/files/Smart%20Contracts%20Report%20%232.0.pdf> (Accessed May 2019); International Swaps and Derivatives Association, ‘Legal Guidelines for Smart Derivatives Contracts’ (February 2019). Available at: <https://www.isda.org/a/23iME/Legal-Guidelines-for-Smart-Derivatives-Contracts-ISDA-Master-Agreement.pdf> (Accessed May 2019).

⁵See M. Raskin, ‘The Law and Legality of Smart Contracts’ (2017) 1 *Georgetown Law Technology Review* 304; M. Giancaspro, ‘Is a ‘Smart Contract’ Really a Smart Idea? Insights from a Legal Perspective’ (2017) 33(6) *Computer Law & Security Review* 825.

⁶ISO/TC307; British Standards Institute PAS 333:2019, Smart Legal Contracts; ETSI Industry Specification Group for Permissioned Distributed Ledgers; International Association for Trusted Blockchain Applications.

legislative⁷, and technology⁸ initiatives have been established in an effort to aid adoption. This paper argues that in order to achieve the goal of value maximization, these efforts should be informed by first principles. Standards, and other regulatory activities, must be driven by consideration of the techno-legal functions of contracting.

The first section introduces the notion of contracting as stateful computation, how distributed ledger technologies fit within this paradigm, and the implications for smart legal contracts. The second section considers potential models for smart legal contracts. The third section proceeds to address the implications for standardization initiatives. The fourth section concludes. Any assessment of architectural implementations are beyond the scope of this paper.

2 Contracting as Stateful Computation

Distributed ledger technologies offer considerable promise for the administration of commercial transactions owing to the congruity between contracts and blockchain systems as devices of stateful computation. The following subsections will unpack this relationship.

2.1 Contracts and Transaction Costs

Contracts are agreements that provide for the performance of obligations, enforceable at law, upon defined events⁹. Normative contract theory assumes the goal of contracting is to maximize shared value between the parties¹⁰. At the level of first principles, contracting is an exercise in allocating and enforcing value transfer by computing the state of an agreement: a contractually relevant event, taken together with the current state of the contract, provides a new contractual state. An options agreement, for example, provides a right to buy or sell a particular asset at a later date at an agreed upon price.

⁷See, for example: 2019 CT H.B. 7310 (NS) March 7, 2019; H.B. 2417, 53d Leg., 1st Reg. Sess. (Ariz. 2017); 2019 NY S.B. 4142 (NS) March 1, 2019; 2019 ND H.B. 1045 (NS) March 8, 2019.

⁸Linux Foundation Accord Project (<http://www.accordproject.org>).

⁹See A. Burrows, 'A Restatement of the English Law of Contract' (Oxford University Press: Oxford, 2016); American Law Institute, Restatement (Second) of Contracts §1–95 (1981).

¹⁰A. Schwartz & R.E. Scott, 'Contract Theory and the Limits of Contract Law' (2003) 113 Yale L.J. 541.

Contractual negotiation and execution determines the relevancy of events¹¹ and the attendant state of the relationship: at the prescribed date, the option becomes exercisable.

Defining future states, and managing attendant business processes, produces externalities in the form of transaction costs. These may be numerous and diverse, including opportunity, enforcement, risk, and administration costs from contractual formation to completion or termination. According to Coesean principle¹², a firm is optimal size when marginal transaction costs that are a result of coordination via the hierarchy are equal to the marginal transaction costs resulting from coordination via the market. Combined, these transaction costs are both systemic and significant. The average lifecycle for ‘low complexity’ contracts is estimated to incur around \$7000 U.S. dollars in costs¹³.

Increasing enterprise digitalization has given rise to a growth in technologies aimed at reducing lifecycle transaction costs across by improving the formation, management, or understanding of commercial agreements¹⁴. Over the past decade, distributed ledger technologies have garnered considerable attention for the potential to do the same, notably through the application of “smart contracts”.

Much has been written around the relation, and interrelation, between “smart” and “legal” contracts¹⁵. It is sufficient for our purposes to consider a broad definitional paradigm comprised of three high-order concepts: “contract”, “smart contract”¹⁶, and “smart legal contract”. Importantly, the purview of each is not consistent with one another¹⁷. Contracts, as aforementioned,

¹¹Through the act of signature, for example.

¹²See, for example: M.C. Jensen & W.H. Meckling, ‘Theory of the Firm: Managerial Behavior, Agency Costs, and Ownership Structure’ (1976) 3 J. Fin. Econ. 305; F.H. Easterbrook & D.R. Fischel, ‘The Economic Structure Of Corporate Law’ (Harvard University Press: Boston, 1996); B.D. Baysinger & H.N. Butler, ‘The Role of Corporate Law in the Theory of the Firm’ (1985) 28(1) J. Law & Econ. 179.

¹³International Association for Contract and Commercial Management, The Cost of a Contract – IACCM Research Report, 18 October 2018. Available at: <https://www.iaccm.com/resources/?id=10392&cb=1539976689> (Accessed May 2019).

¹⁴Such as natural language processing, document automation, contract lifecycle management, and e-signature.

¹⁵See *supra* notes 4–5; J.G. Allen, ‘Wrapped and Stacked: “Smart Contracts” and the Interaction of Natural and Formal Language’ (2018) 14(4) European Review of Contract Law 307.

¹⁶See Szabo, *supra* note 2.

¹⁷Notably, these terms are not consistently used in the literature.

are agreements enforceable at law¹⁸. The notion of a “smart contract” is invariably used to refer to automated¹⁹ machine executable code that functions to perform operations on a blockchain or other distributed ledger system. Notably, a “smart contract” is capable of operating with or without any indicia of a contract at law. It does not follow that a “smart contract” is, necessarily, legally cognizable as a contract. Consequently, the term “smart legal contract” is increasingly used to emphasize the techno-legal qualities of a contract enforceable at law and often, although not exclusively²⁰, expressed in the form of a “smart contract”. Broadly stated, therefore, a “smart legal contract” can be considered to be an agreement constituting a valid contract at law, the documentation of which exists, at least in part, as a formal representation performing operations pertaining to the agreement through the medium of a computer system. The fundamental value proposition being that transactional efficiency can be improved by “automated *prima-facie* assessments of conformance with certain contract terms”²¹.

The capabilities of smart legal contracts are predicated upon the introduction of two attributes hitherto largely absent from contracts expressed in other forms: executability and statefulness. The former enables a contract document to be processed, in whole or in part, as a series of machine-readable instructions. In doing so, the contract document is able to operate as a dynamic artefact, responding to data inputs and generating data outputs. The latter enables the condition of the agreement to be stored based upon the resulting execution events.

A prototypical contract document embodies execution and state as static, coterminous, concepts: the state of the agreement is recorded at the point of *legal* execution²². The relationship between contracting parties will often diverge from the recorded state of the agreement, either owing to operation of law, the terms of the contract, and/or extraneous physical world events. Operationalizing the contract as a computable entity holds promise for

¹⁸See *supra* note 9.

¹⁹M. Finck, ‘Blockchain Regulation and Governance in Europe’ (Cambridge University Press: Cambridge, 2019), 24–25, 75–78.

²⁰See Allen, *supra* note 15. The term “computable contract” may be more inclusive to form, providing greater definitional accuracy: Surden, *supra* note 2.

²¹Surden, *supra* note 2, at 664. May be extended to operations without dependence upon a discrete assessment of conformance.

²²R.J. Gilson *et al.*, ‘Text and Context: Contract Interpretation as Contract Design’ (2014) 100 Cornell L. Rev. 23, 26.

reducing the variance between the recorded and true states of economic relationships, contributing to a reduction in transaction costs.

Take the example of a sales agreement for a commodity or perishable good. Invariably, that contract will provide for the goods to be transported in certain environmental conditions, such as within specified temperature and humidity parameters²³. A smart legal contract encoded with a set of programmatic instructions responsive to that environmental data is capable of computing, storing, and reusing that state over time enables subsequent contractual states to be generated. Those consequential states may be changes to contractual values such as the price of the goods dependent upon breach of the specified parameters, changes in counterparty rights such as the right to terminate, or the creation of legal interests and obligations such as to render payment for the goods. That state may then be used to initiate transactions using the updated values as computed between the parties, as well as within enterprise systems of record. By extension, transactional practices and business models become feasible that depend upon these attributes as technological or practical prerequisites.

2.2 Code as Regulation

As interest in distributed ledger technologies have grown, so have the breadth of implementations. Various models exist, each with differing technical qualities²⁴. Distributed ledgers, including blockchains, are typified as an append-only database of cryptographic, transactions executed and distributed across a fault-tolerant, consensus-based, peer-to-peer network²⁵. Crucially, blockchain technologies provide a mechanism to both sequentially link state changes according to a transaction-based state machine, and to verify state without reliance upon a central authority. By contrast, in a centralized database, transactions are generated by a single entity with permission to write updates to the state of the database. Accuracy of the state of the database is emanates from, and coextensive with, the trust extended to the maintainers that the aggregate of all changes is valid.

²³See, for example: <https://templates.accordproject.org/perishable-goods@0.11.1.html> (Accessed May 2019).

²⁴See Xu, *supra* note 1, at 27–44.

²⁵For technical features see: *Id.*, at 45–58. The Bitcoin blockchain is used herein as a prototypical frame of reference. For the characteristics of a “blockchain” as a data structure, see: S. Nakamoto, ‘Bitcoin: A Peer-to-Peer Electronic Cash System’ (2008). Accessible at: <https://bitcoin.org/bitcoin.pdf> (Accessed May 2019), at 2.

Blockchain transactions may be created by any permitted user²⁶ subject to the constraints governing state transitions. The requirements for a valid state transition are endogenously defined at a protocol or system level²⁷. Transactions are specified instances of these constraints conditioned and controlled by “smart contracts”²⁸. The validity and form of transactions are defined at the protocol or system level. Valid bitcoin transactions, for example, are smart contract transactions that comply with the Bitcoin protocol²⁹. The network state is the canonical history of all valid transactions within the network³⁰.

Combining (1) a deterministic *state machine*³¹ with (2) a *consensus protocol* that provides network level agreement on the same *sequence* of operations, it is possible to perform replicated computation of state³². The capability to programmatically condition and enforce transactions in this manner is often promulgated as effectuating “code as a form of law”³³. In this paradigm, protocol rules may be seen as providing a form of public law, and “smart contracts”, a form of private law³⁴. From first principles, contracts are perhaps better seen as a form of regulation through stateful computation.

²⁶Distributed ledgers may be “permissioned” and “permissionless”: Xu, *supra* note 1, at 27–44.

²⁷Distributed ledger protocols are the aggregate of all technical constraints in the network, *cf.*: Nakamoto, *supra* note 25 and note *infra* 74; G. Wood, ‘Ethereum: A Secure Decentralised Generalised Transaction Ledger’, Ethereum Project Yellow Paper EIP-150 Revision. Accessible at: <http://gawwood.com/paper.pdf> (Accessed May 2019) and note *infra* 62.

²⁸Compare the scope of permissible logic-based state transitions in Bitcoin (Turing-incomplete) and Ethereum (Turing-complete), *Id.*

²⁹Nakamoto, *supra* note 25.

³⁰See the Bitcoin transaction history (‘UTXO set’): <https://www.blockchain.com/charts/utxo-count> (Accessed May 2019).

³¹See L. Lamport, ‘The Implementation of Reliable Distributed Multiprocess Systems’ (1978) 2(2) Computer Networks 95.

³²F. B. Schneider, ‘Implementing Fault-Tolerant Services using the State Machine Approach’ (1990) 22(4) ACM Computing Surveys, 299.

³³L. Lessig, ‘Code and Other Laws of Cyberspace’ (Basic Books: New York, 1999), 3–8; L. Lessig, ‘Code is Law: On Liberty in Cyberspace’ (2000) January-February, Harvard Magazine. Available at: <https://harvardmagazine.com/2000/01/code-is-law-html> (Accessed May 2019); P. De Filippi & S. Hassan, ‘Blockchain Technology as a Regulatory Technology: From Code is Law to Law is Code’ (2016) 21(12) First Monday. Note that this is not an argument for code *being law*: K. Werbach, ‘The Blockchain and the New Architecture of Trust’ (MIT Press: Cambridge, 2018), at 153–160.

³⁴Wider regulatory issues of social governance in blockchain systems (e.g. protocol upgrades, modifications) may be considered.

A blockchain transaction can only modify state where the specified constraints are met. By restricting conditions on state machine transitions, “smart contracts” can therefore be used to define the conditions precedent for computing a valid future state and enforcing of state transitions, as defined by the protocol level. Each transaction is capable of cryptographic verification³⁵. As a result of these technical attributes, it is possible to determine both past state and conditions for state modification. Conditioning modification of state upon consensus-based transitions provides a mechanism for logic to mediate transactions between parties that may not trust execution of the logic or the state of a database when operated in a centralized fashion. Taken together, the scope of reliance upon current and future states is commensurate with reliance upon the rules governing state transitions as between the parties. Validity, and therefore value, is derived from the enforcement and verification of computational state, rather than the legal institutions of the State³⁶. Smart contracts introduce a narrow regulatory framework³⁷, a source of rights and obligations, in technological form by bundling computation and enforcement of state together.

Contracts entail a necessarily wider regulatory scope. A contract, by definition, cannot exist within a legal vacuum³⁸. Contractual state is a function of the coordination between the parties within this regulatory paradigm. Just as a “smart contract” defines the regulatory parameters for the validity of transactions on the network, a contractually relevant transaction is defined by legal doctrine. The execution and state of a contractual transaction is necessarily dependent upon, and subject to, the manner in which the law impinges upon the technical mechanism³⁹. Matters become interesting when the technical and legal validity of a transaction is purported to be coextensive⁴⁰.

³⁵See Nakamoto, *supra* note 25, at 4; Xu, *supra* note 1, at 21–24.

³⁶Werbach, *supra* note 33, at 160–162.

³⁷See Finck, *supra* note 19, at 66 *et seq.*

³⁸See *supra* note 9; Allen, *supra* note, 15 at 17; *Id.*, at 85–86 *cf.* A. Savelyev, ‘Contract Law 2.0: “Smart” contracts as the beginning of the end of classic contract law’ (2017) 26(2) Information and Communications Technology Law 116.

³⁹It does not necessarily follow that regulation must adhere to jurisdictional norms: Lessig, *supra* note 33; D. Sventsson, ‘The Holy Trinity of Legal Fictions Undermining the Application of Law to the Global Internet’ (2015) 23(3) International Journal of Law and Information Technology 219. On the notion of “trustlessness”, see: M. Zou *et al.*, ‘In Code We Trust? Trustlessness and Smart Contracts’ (2019) Society for Computers and Law Journal.

⁴⁰For example, a “smart contract” operates, or intends to operate, as a contract at law.

Within this paradigm, a contract is necessarily legally valid, but the underlying state transition does not necessarily provide a guarantee of the validity of the execution of the transition or the state, either in its executory or executed form. A transaction may be technically valid without being legally valid. A bitcoin transfer as consideration within the context of a wider contractual agreement may, for example, be vitiated by a range of factors⁴¹ that do not encumber the technical execution of the transaction.

The application of smart contracts *qua* contracts is fundamentally reducible to regulatory considerations. Blockchains are valuable precisely because of the technical qualities that engender trust in the state of ledger. If reliance upon stateful computation within the system is mitigated, regulatory trust is substantially shifted to dependencies outside of the system. As a corollary, the nature of the regulatory environment of “smart legal contracts” is necessarily predicated upon a techno-legal assessment.

Contracts are often not conducive to expression as state machines⁴² due to the *ex-ante* “costs of identifying obligations for many different states of the world”⁴³. Even when those state contingencies can be identified⁴⁴, contractual relationships are characterized by incompleteness⁴⁵, ambiguity⁴⁶, and a lack of determinism⁴⁷. Frequently, this is employed by contracting parties as an

⁴¹Burrows, *supra* note 9, at 169 *et seq.*

⁴²See *supra* note 31. This does not, of course, mean that contracts could not be purposefully expressed as such.

⁴³R.E. Scott & G.G. Triantis, ‘Incomplete Contracts and the Theory of Contract Design’ (2005) 56(1) *Case Western Reserve Law Review* 187, at 197.

⁴⁴C.J. Goetz and R.E. Scott ‘Principles of Relational Contracts’ (1981) 67 *Va. L. Rev.* 1089, at 1091; Gilson *et al.*, *supra* note 22, at 56; O. Kvaløy and T. Olsen, ‘Endogenous Verifiability and Relational Contracting’ (2009) 99(5) *American Economic Review* 2193.

⁴⁵O. Hart & J. Moore, ‘Foundations of Incomplete Contracts’ (1999) 66(1) *Review of Economic Studies* 115 *cf.* ‘obligational completeness’: G.G. Triantis, ‘Contractual Allocations of Unknown Risks: A Critique of the Doctrine of Commercial Impracticability’ (1992) 42 *U. Toronto L.J.* 450, 464–468; R.E. Scott, ‘The Law and Economics of Incomplete Contracts’ (2006) 2 *Annual Review of Law and Social Science* 279.

⁴⁶R.A. Posner, ‘The Law and Economics of Contract Interpretation’ (2004) 83 *Tex L. Rev.* 1581. Often purposely: B. Bernheim & M. Whinston, ‘Incomplete Contracts and Strategic Ambiguity’ (1998) 88(4) *American Economic Review* 902.

⁴⁷O. Hart & J. Moore, ‘Contracts as Reference Points’ (2008) 123(1) *Quarterly Journal of Economics* 1; I.R. Macneil, ‘Contracts: Adjustment of Long-Term Economic Relations under Classical, Neoclassical, and Relational Contract Law’ (1977) 71 *Nw. UL Rev.* 854; Gilson *et al.*, *supra* note 22, at 56–58.

endogenous optimization apparatus by the parties⁴⁸ to shift state determination to *ex-post* enforcement mechanisms⁴⁹.

As a result, contractual state is generally not capable of canonical determination in the same manner as within a blockchain system, either *ex-ante* or *ex-post*. Consequently, the scope for a contract to operate *as a* smart contract, in whole or in part, is bounded by the feasibility of expressing potential future states and determining prevailing state. These bounds place constraints on the form and functions of smart legal contracts to the extent that they cannot be contracted away⁵⁰ or regulatory recourse to resolve state is not practically, economically, or legally prohibitive.

Fairly obviously, the composition of a smart legal contract will depend upon the feasibility of capturing the “human” agreement in formal representation, wherein issues of intention⁵¹, contract interpretation and construction⁵² are particularly pertinent. Furthermore, smart legal contracts need to incentivize state management as a form of coordination between the parties. It is not to be assumed that computing state at the agreement level, rather than at the transaction level, is universally beneficial. In circumstances where this is not practicable, a contractual agreement will, at least partially, require expression in natural language rather than a formal representation⁵³.

Consequently, contracting provides ample regulatory scope for some prevailing state to exist that is not appropriately reflected in the executed state of a transaction⁵⁴. Examples can include states created by operation of law, such as misrepresentation or mistake, as well as by the internal operation, or interrelation, of contractual provisions. Reprising the example above, assume that the sales agreement includes a typical *force majeure* provision. Upon occurrence of a qualifying event, that provision may create a prevailing state

⁴⁸Scott & Triantis, *supra* note 43; Bernheim and M. Whinston, *supra* note 46; P. Aghion & R. Holden, ‘Incomplete Contracts and the Theory of the Firm: What Have We Learned over the past 25 Years?’ (2011) 25 *The Journal of Economic Perspectives* 181.

⁴⁹R.E. Scott ‘Theory of Self-Enforcing Indefinite Agreements’ (2003) 103 *Colum. L. Rev.* 103 1641 at 1644–1645. See Restatement, *supra* note 9, at §33, Comment a.: (“The actions of the parties may show conclusively that they have intended to conclude a bargain, even though one or more terms are missing or are left to be agreed upon. In such cases courts endeavor, if possible, to attach a sufficiently definite meaning to the contract”).

⁵⁰Code *as* contract, see II(A).

⁵¹Such as comprehension of formal representation.

⁵²Posner, *supra* note 46.

⁵³See Allen, *supra* note 15.

⁵⁴Finck, *supra* note 19, at 27 (“Whereas the role of law is to fill gaps in agreements with default rules, smart contract code contains no gaps allowing law to intervene”).

not represented in the execution of agreement, either as the contract is not capable of determining the prevailing state and/or the state cannot be enforced so as to encumber the technical operation of the contract. The latter may occur, for example, where the *force majeure* provision is expressed in natural language and thus cannot preclude the technical execution of a provision, such as a digital asset transfer, the performance of which may be impacted by the former.

Mandated enforcement is assumed desirable within the technical fabric of blockchain systems. Indeed, smart contracts are revered for this very quality. Porting this regulatory feature to contracts, however, elides optionality of enforcement. It does not follow that because parties contract for a given state that they will elect to enforce that state. Often enforcement is a decision driven by commercial exigencies, not the form of the agreement or manner of performance⁵⁵. Whereas enforcing computed state provides the mechanism for efficient coordination⁵⁶ in a blockchain system, the same value and operational efficiencies are not necessarily inherent within, or incentivized by, other forms of economic organization. In many circumstances it will be infeasible or suboptimal to bundle computation and enforcement of state together.

3 Models of Smart Legal Contracts

In view of the above, smart legal contracts may be modelled as one of two core high-level variants: (1) “contract as code”; or (2) “contract as code and natural language”. The former results in a contract being documented exclusively in a formal representation such as a programming language. The latter, by contrast, consists of a composite of human-readable natural language and machine-readable representation.

3.1 “Contract as Code”

The formally represented component of the contract document may exist either ‘on-chain’ and/or ‘off-chain’. In an ‘on-chain’ model, execution is performed, and state maintained, in the runtime environment of a distributed ledger system. Conversely, an ‘off-chain’ model executes the code in an environment other than that of a distributed ledger. A ‘on-chain’/‘off-chain’ hybrid model

⁵⁵It is at least arguable, however, that using the mechanism of a smart contract *qua* contract exhibits the intention to enforce.

⁵⁶Enforcing the protocol: *supra* note, 27.

borrowed from each of these variants, separating the execution of the code and/or persistence of state between a distributed ledger and an ‘off-chain’ system⁵⁷.

By modelling contracts *as* code we ascribe to the notion that regulation through stateful computation provides a rational means of contractual coordination. Bitcoin provides this characteristic as the protocol constraints imbue value into the unspent transaction outputs resulting from smart contract state transitions.

The calculus shifts where collapsing state and execution does not provide the same value, namely that the state of the system *is* the coded regulation. In other words, the contract is definitively represented by the code, and the executed state is definitively accurate. It follows that we necessarily assume that contractual state can be accurately computed but, interpreted strictly, that the computed state is accurate absent erroneous inputs or computation, and that the state of the agreement is not impacted by other forms of regulation. Outside of the walled garden of a blockchain system this is obviously impracticable. Instead, one should whether assessment of the regulatory context for a given contract produces an aggregate net benefit in terms of transaction costs. This may, for example, be as a result of reduction of administrative overhead, reduction of transactional risk, or operational efficiencies.

Automated enforcement is a regulatory feature of Bitcoin that generates value. The latitude for state to be impacted by extraneous factors outside of the discrete computation of a smart legal contract is far greater. On-chain execution may reduce transaction costs pertaining to enforcement, management, execution risk, and legal costs. Equally, however, it may itself produce these costs including computational execution risk, operational reliance upon inaccurate state, and the costs of unwinding transactions where technical state does not reflect the legal realities.

The appropriate form of a “contract as code” is an important factor in this regulatory assessment. A smart legal contract may be embodied as a simple blockchain-based smart contract that conditions a transfer upon a physical

⁵⁷ A comprehensive discussion of the techno-legal factors behind the architecture of smart legal contracts is beyond the scope of this paper. See, for example: C. Molina-Jimenez *et al.*, ‘On and Off-Blockchain Enforcement of Smart Contracts’ in G. Mencagli *et al.* (eds) Euro-Par 2018: Parallel Processing Workshops. Euro-Par 2018. Lecture Notes in Computer Science, vol 11339; J. Eberhardt and S. Tai, ‘On or Off the Blockchain? Insights on Off-Chaining Computation and Data’ in F. De Paoli *et al.* (eds) Service-Oriented and Cloud Computing. ESOC 2017. Lecture Notes in Computer Science, vol 10465.

world event⁵⁸. Equally, state may be computed off-chain and a transaction executed on-chain pursuant to this computed state⁵⁹ or auditable state instantiated on distributed ledger⁶⁰. Such an architecture may be more appropriate where the contractual context benefits from a reduction in administrative overhead but would not, for example, value automated enforcement without manual verification by the parties. In many circumstances computation and auditability of state without enforcement may be a sufficient balance.

3.2 “Contract as Code and Natural Language”

Given the inherent limitations of instantiating agreements entirely in code, increasing attention is given to architecting smart legal contracts “in a language that is both human-intelligible and machine readable, whose text incorporates an algorithm which automates some or all of the performance of the agreement”⁶¹. The natural language operates as a device to produce the state of the agreement where it is infeasible to do so using computational methods for the reasons discussed above.

Optimal stateful computation for smart legal contracts is a function of combining an appropriate structure of the formal representation with the appropriate form of natural language, bound together using an appropriate method.

The ‘happy path’ is that the natural language records the genesis state of the agreement, and the computational components evolve the state, including any programmatic performance, in step. The result is accurate stateful computation across the components. Whereas the ‘contract as code’ model purports to create a canonical state by bundling execution and enforcement together,

⁵⁸See AXA Parametric Insurance (<https://www.axa.com/en/newsroom/news/axa-goes-blockchain-with-fizzy>). (Accessed May 2019).

⁵⁹See, for example: Clause, Stellar Transfer Action (<https://docs.clause.io/en/articles/55-stellar-transfer-action>) (Accessed May 2019).

⁶⁰See *infra* note 67.

⁶¹Allen, *supra* note 15 at 313. Also: Sir Geoffrey Vos, ‘Cryptoassets as Property: how can English law boost the confidence of would-be parties to smart legal contracts?’, Joint Northern Chancery Bar Association and University of Liverpool Lecture, 2 May 2019. Available at: <https://www.judiciary.uk/wp-content/uploads/2019/05/Sir-Geoffrey-Vos-Chancellor-of-the-High-Court-speech-on-cryptoassets.pdf> (Accessed May 2019); C.D. Clack *et al.*, ‘Smart Contract Templates: Foundations, Design Landscape, and Research Directions’, CoRR abs/1608.00771 (2016), 2. I. Grigg, ‘On the Intersection of Ricardian and Smart Contracts’ (July 2016). Available at: https://iang.org/papers/intersection_ricardian_smart.html (Accessed May 2019) (“We can now see that the real challenge between smart contracts and Ricardian Contracts or legal documents is not to choose, but to incorporate.”).

the hybrid model potentially introduces a divergence between the natural language and the computational state of the contract. The manifestation of this divergence results from the interrelation between the two components. In one instance, the code performs business processes embodied in the natural language, such as calculations, transactions, and the like. The code may be expressed or determined, through canons of construction and interpretation, to be subordinate to the natural language. In another instance, the code replaces natural language text or operates in a superordinate manner.

The method of binding is important to optimize stateful computation between the two components, thereby maximizing the coordinational efficiencies of a smart legal contract.

A ‘light’ form of binding may be achieved by adding natural language as comments interspersed with the contract code⁶² or by referencing external scripts within the natural language⁶³. Both models offer a form of minimal viable integration by prescribing a referential basis between the components. In many circumstances, neither will attain optimal stateful computation⁶⁴.

A more substantial form of integration involves binding at the content level by associating the variables within the prose to the execution logic⁶⁵. This may be achieved by providing defined values from the textual components into an executable representation⁶⁶ that trigger execution either on-chain and/or off-chain, as required. For example, a request-response structure to provide the flexibility to compute state off-chain and use that state to execute on-chain operations such as storing audit logs or executing transactions⁶⁷. Alternatively, variable data may be passed from the textual components into function calls to smart contracts. Instead of integration by reference, the contract can be structured in reference to a data model⁶⁸. The data model may act as a shared

⁶²See, for example: NatSpec Format, Solidity Documentation, Ethereum Foundation (<https://solidity.readthedocs.io/en/develop/natspec-format.html>) (Accessed May 2019).

⁶³‘Dual Integration: Putting the Contracts in Smart Contracts’, Monax, (https://monax.io/learn/dual_integration/) (Accessed May 2019).

⁶⁴J.M. Skarloff, ‘Smart Contracts and the Cost of Inflexibility’ (2018) 166 U. Pa. L. Rev 263, at 296 (“smart contracts cannot create a transaction-costless environment”).

⁶⁵Grigg and Clack *et al.*, *supra* note 61; Template Specification, Accord Project (<https://docs.accordproject.org/docs/accordproject-specification>) (Accessed May 2019).

⁶⁶See the sample instance of the above sales agreement: <https://templates.accordproject.org/perishable-goods@0.11.1.html> (Accessed May 2019).

⁶⁷See Clause, Hyperledger Fabric Actions (<https://docs.clause.io/en/articles/62>) and Clause, Kaleido Actions (<https://docs.clause.io/en/articles/57-kaleido-integration>) (Accessed May 2019).

⁶⁸See Accord Project, *supra* note 65.

resource for both components to optimize the expression and computation of state⁶⁹. Optimization will often be a contextual assessment facilitated by techno-legal standards.

4 Toward Techno-Legal Standards

Technical standards play a crucial role in regulating the application and adoption of technologies across both established and nascent industries⁷⁰. At a macro-level, better external standards enable firms to make improved use of market economies⁷¹, such as through subcontracting and outsourcing. At a micro-level, standards may reduce transaction costs⁷² by making “it simpler for all parties to a deal to recognise what is being dealt in”⁷³.

As discussed, blockchain systems provide a regulatory framework that facilitates efficient coordination by optimizing transaction costs through stateful computation. Bitcoin limits the regulatory surface area by defining a narrow set of operations⁷⁴ for value exchange within the constraints of the protocol. Contracting has developed a corpus of practical standards⁷⁵ that may be seen to provide informal coordination mechanisms to the same end.

The introduction of computational functionalities into contracts offers the opportunity to benefit from improved stateful computation, but doing so broadens the regulatory considerations at a technical and legal level. A tight bundling of execution and enforcement of state is often not contextually appropriate for legal contracts. Consequently, the scope of regulation needs to

⁶⁹*Id.*

⁷⁰C.F. Cargill, ‘Why Standardization Efforts Fail’ (2011) 14(1) *Journal of Electronic Publishing*. DOI: 10.3998/3336451.0014.103; K. Blind *et al.*, ‘The Impact of Standards and Regulation on Innovation in Uncertain Markets’ (2017) 46(1) *Research Policy* 249.

⁷¹See K. Blind, ‘The Economics of Standards: Theory, Evidence, Policy’ (Edward Elgar: Cheltenham, 2004).

⁷²D.C. North, ‘Economic Performance Through Time’ (1994) 84(3) *American Economic Review* 359.

⁷³David P., ‘Standardisation Policies for Network Technologies: The Flux between Freedom and Order Revisited’ in Hawkins *et al.*, ‘Standards, Innovation and Competitiveness: The Politics and Economics of Standards in Natural and Technical Environments’ (Edward Elgar: Cheltenham, 1995) at 22.

⁷⁴See Bitcoin Developer Reference, Bitcoin.org, <https://bitcoin.org/en/developer-reference#opcodes> (Accessed May 2019).

⁷⁵K.A. Adams, ‘A Manual of Style for Contract Drafting’, 4th ed. (American Bar Association: Chicago, 2018); Electronic Signatures in Global and National Commerce Act 15 U.S.C. §7001-7006 (2018).

be cognizant to the techno-legal requirements that will facilitate adoption. An absence of appropriate standards is liable to act as an impediment to adoption, often contributing to transaction costs. The exact nature of these standards is a question for industry and far beyond the scope of this paper. From the perspective of first principles it is possible, however, to derive two guiding principles.

First, standards should facilitate appropriate forms of techno-legal computation. This is a contextual enquiry requiring assessment of the circumstances surrounding a given contractual transaction. As a framework, standards should optimize along two axis: execution and expression. The former considers the degree of flexibility between ‘on-chain’ and ‘off-chain’ execution of code. The manner of execution may be seen as a function of how tightly, at least in traditional conceptions of blockchains, execution and enforcement should be bound within the contractual agreement. The latter considering the technical and legal attributes required for the hybrid representation of contracts in code and natural language. The manner of expression should facilitate that execution by enabling optimal techno-legal expression of the agreement between the parties. A plethora of factors play into this assessment, including how tightly natural language and formal logic should be bound together, issues of safety such as how smart legal contracts should be verified and tested⁷⁶, the permissible range of executable operations⁷⁷, interrelation between runtime environments, and issues of exception management⁷⁸ to name but a few.

Secondly, standards need to address the social aspects⁷⁹ of contracting. Contracts are, at least today, predominantly formed and executed between human parties, representatives, or agents. Implementation of standards should, therefore, impact transaction costs in the most direct way. Overhead in the use

⁷⁶See domain modeling, see: Accord Project, *supra* note 68; Digital Asset Modelling Language v0.13.5 (<https://docs.daml.com/daml/reference/structure.html>) (Accessed May 2019); International Swaps and Derivatives Association, ‘ISDA Common Domain Model Version 1.0 Design Definition Document’ (October 2017). Accessible at <https://www.isda.org/a/gVKDE/CDM-FINAL.pdf> (Accessed May 2019). Formal verification of computed contract state: K. Bhargavan *et al.*, ‘Formal Verification of Smart Contracts’ (2016) Proceedings of the 2016 ACM Workshop on Programming Languages and Analysis for Security, 91. DOI: 10.1145/2993600.2993611; Accord Project, Ergo (<https://docs.accordproject.org/docs/ergo>) (Accessed May 2019).

⁷⁷Turing completeness, *cf.* Ergo, *Id.*; DAML, *Id.*; Solidity, *supra* note 62; Bitcoin Script, *supra* note 74.

⁷⁸Dispute resolution: P. Ortolani, ‘Self-Enforcing Online Dispute Resolution: Lessons from Bitcoin’ (2016) 36(3) Oxford Journal of Legal Studies 595.

⁷⁹Cargill, *supra* note 70.

of standards-based contracting is liable to be an impediment in the same way as the more discrete technical factors. Aside from legal factors⁸⁰, a range of practical factors warrant attention. Considerations here include minimizing changes in contracting practices, particularly where this creates market or legal uncertainty, such as by impacting drafting conventions, for example. The coordinational impact of extending execution beyond e-signature also need to be addressed. Factors here include interoperability, so as to limit the introduction of transaction costs in technical coordination, future-proofing to limit transaction costs created by changes that impact execution, and portability to enable contractual content to be reusable across environments. Whether through a common format⁸¹ or guidelines, standards should ensure that any post-formation reduction costs are not realized at the expense of additional costs in formation and modification.

5 Conclusion

Blockchain-based smart contracts offer the potential to reduce transaction costs through new methods of stateful computation. When applied to commercial transactions, smart contracts represent a shift from recording state at contractual execution to enforcement of executed state. This is undoubtedly beneficial in circumstances where regulation occurs exclusively as a technological conception. Expansion into generalized contractual relations raises a host of techno-legal considerations that require wider regulatory assessment. As part of that regulatory framework, standards initiatives need to be cognizant of the various factors that impinge upon that analysis. The optimal model for smart legal contracts is a contextual enquiry. Distributed ledger and smart contracts standards should seek to provide sufficient flexibility to facilitate contracting parties to coordinate in an optimal manner.

⁸⁰See *supra* notes 4–5.

⁸¹See Accord Project, *supra* note 65.

Biography



Peter Hunn is a UK-trained lawyer, and the founder of both technology startup Clause and the non-profit Accord Project hosted by the Linux Foundation. The Accord Project operates in collaboration with the leading industry bodies, enterprise blockchain providers, standards organisations, and law firms to develop an open source technology stack for smart legal contracts. The Project is DLT independent and agnostic, meaning the protocol can operate with any existing or future DLT platform, and is devised specifically for enterprise and legal application. Clause is a leading provider of enterprise smart legal contract infrastructure. Peter is also a member of the UK Jurisdiction Taskforce of the UK Lawtech Delivery Panel and contributes as an expert to ISO TC307.