SMART CONTRACTS: OPPORTUNITIES AND CHALLENGES

The Rt Hon Lord Hamblen of Kersey

Justice of the Supreme Court of the United Kingdom

Introduction

1.1. The concept of a 'smart contract' was first developed by Nick Szabo in the 1990s. Identifying an opportunity to bring well-established principles of contract law into the design of electronic protocols running on public networks such as the Internet, Szabo defined a smart contract as "*a computerised transaction protocol that executes the terms of a contract*".[1]

1.2. There is no single or universally accepted definition of what a smart contract is. In its 2021 Report the Law Commission of England and Wales defined a smart contract as "a legally binding contract in which some or all of the contractual obligations are defined in and/or performed automatically by a computer program".[2] Its essential features are that it comprises computer code and that it is self-executing. Performance is automatic once certain conditions are met. It is most obviously concerned with conditional actions – if X then Y. Since this action is programmed it will occur without the need for human interaction.

---

[1] Szabo, Nick (1994) *Smart contracts*: http://www.fon.hum.uva.nl/rob/Courses/InformationInSpeech/CDROM/Literature/LOTwinterschool2006/szabo.best.vwh.net/smart.contracts.html

[2] Law Commission Report: Smart legal contracts Advice to Government at para 1.2: https://www.lawcom.gov.uk/project/smart-contracts/

1.3. Under Szabo's vision, smart contracts would not only satisfy common contractual conditions, minimise malicious and accidental exceptions, and reduce the need for trusted intermediaries, but it would also serve various economic goals including lowering fraud loss, arbitration and enforcement costs, and other transaction costs.[3]

1.4. To illustrate his concept of a smart contract at its most basic level, Szabo refers to "*the primitive ancestor of smart contracts*": the humble vending machine. In essence, a vending machine takes in coins, and via a simple mechanism, dispenses a chosen product (and change, if so required) according to the price displayed on the machine. This is what it is programmed to do, and it will execute that transaction automatically when the action of putting in the coins and choosing the product is carried out. To the extent that anybody with coins can participate in an exchange with the vendor and thus the vending machine is a contract with the bearer of the coins, Szabo conceptualises the vending machine as a smart contract.[4]

1.5. A second example to which Szabo refers is a hypothetical digital security system for cars, in which security protocols only give control of the cryptographic keys for operating the car to the person who rightfully owns it. So, if a car was being used to secure credit, when the owner failed to make a payment, the smart contract could invoke a security protocol that returned control of the car keys to the bank. Szabo recognises that "*it would be rude to revoke operation of the car while it's doing 75 down the freeway*", which is somewhat reassuring – but no doubt there would be various other mechanisms to enable a security protocol such as this to work in practice.

2. 'Smart contracts': the potential

2.1. Smart contracts have become of potentially wide-ranging application due to the development of distributive ledger technology. This can be defined as a digital system

[3] Szabo, *Smart Contracts*

[4] Szabo, Nick (1997) Formalising and Securing Relationships on Public Networks, First Monday, vol.2, no.9 (September): https://firstmonday.org/ojs/index.php/fm/article/view/548/469

that allows any number of computers to keep an identical record of information, without any central master copy.[5] It is a distributed and decentralised system: the ledger is shared across the network with no one person having right or responsibility to maintain it.

2.2. A common type of distributed ledger is a blockchain, which does what it says on the tin: it is a chain of blocks, where each block is a collection of data, made up of a number of transactions.[6] The most well-known example of the use of a blockchain is Bitcoin, a digital currency that doesn't rely on any intermediary institutions such as banks. The blockchain is a fundamental element in the Bitcoin architecture.

2.3. A Bitcoin transaction is a cryptographically signed statement on the blockchain, transferring Bitcoin tokens between two or more cryptographic private keys, which are then grouped together in a chain with other blocks. Because the system is distributed and decentralised, anyone can view Bitcoin's blockchain at any time (and all viewers will see the same version). The blockchain updates approximately every ten minutes, during which time transactions clear, value and title are transferred, and everybody's copy of the ledger updates automatically.[7]

2.4. What this means for smart contracts using distributed ledger technology is that the smart contract can be fully executed by the parties to it, without any third-party involvement, in a matter of minutes.

2.5. As to its potential, the World Economic Forum has predicted 10% of global GDP will be stored on the blockchain by 2027. There have already been end to end blockchain bonds issued – i.e. bonds created, allocated, transferred and managed throughout by blockchain.[8] Use cases identified by the Law Commission included insurance,

---

[5] Werbach, Kevin & Cornell, Nicolas (2017) Contracts Ex Machina, *Duke Law Journal*, 67(2) 325

[6] Green, Sarah (2018) Smart contracts, Interpretation and Rectification, *Lloyd's Maritime and Commercial Law Quarterly*, Vol. 2018, No. 2, 10.05.2018, p. 234-251.

[7] Ibid, p. 236

[8] https://www.icmagroup.org/Regulatory-Policy-and-Market-Practice/fintech/new-fintech-applications-in-bond-markets/

finance, decentralised finance, real estate, supply chain, peer to peer and intellectual property.[9]

2.6. Potential contractual benefits include speed and reduced enforcement costs (due to self-execution); cost efficiency (due to the lack of an intermediary); transparency (as there can be a single store of information/documentation); increased trust (due to automatic validation, monitoring and performance), and reduced ambiguity (due to code replacing natural language).

2.7. ISDA has been at the forefront in carrying out work in relation to how smart contracts may be of assistance in derivatives trading. A digital ISDA CDM or Common Domain Model has been developed which provides a digital representation of a blueprint of events and actions that occur throughout the lifecycle of a derivatives trade. Various instructive papers have been produced discussing how smart contracts could improve the efficiency of the derivatives market by automating performance.[10]

2.8. As with all technological developments, however, enthusiasm must also be met with scrutiny. Smart contracts are still in their infancy and there are a number of potential challenges that still need to be met. For example, what happens if one or both parties change their minds? Where are assets held? Is a smart contract confidential? What happens if there is an event which should automatically bring the contract to an end? How is an event of default to be acted upon? What governing law applies to a smart contract? How are they to be regulated?

2.9. Today, I propose to illustrate the potential opportunities and challenges of smart contracts by focusing on five areas. First, I am going to give a brief overview of the principal models of smart contracts. Second, I will outline some of the differences between smart contracting and traditional contracting. Third, I will identify some of the ways in which smart contracts may go 'wrong' such that smart contract disputes arise. Fourthly, I will address the ways in which judicial remedial action may operate

[9] Law Commission Report at para 2.86

[10] https://www.isda.org/2019/10/16/isda-smart-contracts/

in the context of such smart contract disputes. Finally, I will address some practical implications of smart contracts for financial services and banking.

3. Models of 'Smart contracts'

3.1. For some the term 'smart contract' is a misnomer because it is neither smart nor a contract.

3.2. A smart contract is not 'smart' because it involves no intelligence in the sense of being able to convert unstructured information into useful knowledge. It can simply examine whether certain states have occurred and, if they have, trigger a pre-determined action.

3.3. An example of the difference between human intelligence and computer pre-determined action is as follows:[11]

> The instruction is given: "Go to the shop and buy a newspaper. If there are any eggs, get a dozen".
> If the shop has eggs, the human result is (or should be) the purchase of one newspaper and a dozen eggs.
> The computer result is, however, the purchase of a dozen newspapers.

3.4. A smart contract is not, or may not be a 'contract', because a document written entirely in code may not satisfy the requirements of a legal contract under the applicable governing law.

3.5. That is why the term 'smart legal contract' is sometimes used instead of 'smart contract'. There are three main models of such a contract: the 'external' model; the 'internal' model and the 'solely code' model.

[11] Green, Smart contracts, Interpretation and Rectification, p. 234

3.6. In the external model the contract is written entirely in natural language, but certain conditional elements of the contract would be coded so that the required actions would happen automatically when the relevant conditions were met. The code would not be part of the contract; it would simply provide a mechanism for automatic performance of aspects of the contract. The parties' agreement would be as set out in their natural language agreement rather than the code, but it would, of course, be important to ensure that the code accurately reflects that agreement.

3.7. In the internal model, much of the contract would remain in natural language but parts of it would be in code, either solely in code or linked to a natural language representation. It would effectively be a hybrid agreement. Instead of being written out in the contract the code could refer to a standardised form of code set out elsewhere, which is one of the areas that ISDA has been investigating.

3.8. In the solely code model, the agreement is in code without being housed in any natural language architecture.

3.9. Under English law, a valid contract is one where an offer has been made and accepted, consideration has been given from one party to another, the parties have intended to create legal relations, and the parties have certainty of terms. In November 2019 the LawTech Delivery Panel's UK Jurisdiction Taskforce concluded that all models of smart contract are capable of satisfying those conditions of forming contracts under English law, in the same way as a traditional or natural language contract, and that a smart contract is therefore capable of having contractual force such that it can be enforced in English courts[12]. The Law Commission reached the same conclusion in its Report (see Chapter 3).

4. Smart contracting

[12] '*Legal statement on cryptoassets and smart contracts: UK Jurisdiction Taskforce'*, published by The LawTech Delivery Panel (November 2019)

4.1. Having considered what smart contracts are, I next propose to consider the process by which smart contracts are entered into.

4.2. There are many differences between smart contracting and traditional contracting, but I shall focus on three of the principal differences: (i) the process of translating natural terms into code; (ii) the possibility of supplementary contracts such as data threshold agreements; and (iii) the potential for new roles, such as 'legal engineers' and 'human oracles'.

Processable terms and machine-readable code

4.3. A first key difference in the process of reaching a smart contract agreement is that in order for a smart contract to be partly or wholly written in code, its terms will need to be recorded in computer-readable form.

4.4. This means that in deciding to enter into a smart contract, the parties will need to ensure that their 'natural language' terms are translated into 'code' form. This requires the process of contractual expression to fundamentally re-orientate itself away from ordinary language, towards highly structured data that is amenable to computer processing.[13]

4.5. Some clauses will be more conducive to this translation exercise than others. The definition of a price, for example, will generally be relatively easily codified: the natural language term "*this option expires on December 31, 2020*" has a single and unambiguous meaning that lends well to being codified:

<Option_Expiration_Date: 31/12/2020>[14]

4.6. Operational clauses, which generally embed some form of conditional logic, will also be relatively conducive to being codified. Operational clauses are those which determine that upon the occurrence of a specified event, or at a specified time, a

[13] Surden, Harry (2012) Computable Contracts, *UC Davis Law Review*, Vol. 46, No. 629, p.642
[14] Ibid, p. 649

particular deterministic action is required. For example, on receipt of £10,000, party A shall transfer title in the goods to party B.

4.7. Derivative transactions are promising territory for the application of smart contract methodology because many of their terms are dependent on conditional logic. For example:

- a clause requiring an amount to be payable on a payment date equal to the product of a calculation amount, a floating rate and a day count fraction;

- a clause that provides that one party to the contract is to pay the other an amount equal to the difference between the settlement price and a forward price, with the party required to make such payment being determined by whether the settlement price exceeds the forward price or vice versa; and

- a clause that requires a party, on a particular date, to transfer assets that have a value equal to the amount by which a required credit support amount is less than the value of collateral provided, subject to certain formulaic haircuts and adjustments.

4.8. However, many contracts will also include non-operational clauses that are open to interpretation and not easily codified. One example would be the use of 'best' or 'reasonable' endeavours.

4.9. If the term to be drafted in code is in natural language terms such as "*Prior to exchange of contracts, party A warrants that it shall use its reasonable endeavours to ensure that conditions X and Y are satisfied*", then the 'reasonable endeavours' action would need to be more explicitly codified. To be read and enforced by a machine, it would need a list of specific actions that would constitute the action of using such 'endeavours' so that the algorithm could determine whether such endeavours had or had not been met for the purposes of satisfying conditions X and Y.

4.10. Other examples of commonly used non-operational clauses include the following:

- A clause specifying what law should govern in the event of any dispute;
- A clause specifying what jurisdiction any disputes may be brought in;
- A clause providing that the written legal document represents the entire agreement between the parties;
- A representation that a party's obligations under the legal agreement constitute legal, valid and binding obligations;
- A clause that dictates that when making a decision or a determination, the person doing so must act in good faith or in a commercially reasonable manner; and
- A clause that provides that all transactions entered into under a master agreement form a single agreement between the parties. [15]

4.11. The accuracy of the process of translating natural language terms or intentions into machine-readable code is obviously going to be very important. It may well be that the parties agree on a term in 'natural language' but that their agreed mutual intention does not translate into the term as codified, resulting in an asymmetry in understanding between the substantive authors of the contract (human beings) and their intended audience (the computers responsible for executing the contracts).[16]

Data threshold agreements

4.12. A second key difference is that the parties might choose to enter into a supplementary agreement in order to agree their approach to data prior to entering into the smart contract.

4.13. In his work on 'computable contracts', Harry Surden suggests that one way that contracting parties can endow computer data with shared interpretations is through a 'data-meaning' threshold agreement. These are traditional, written language documents that parties agree to before engaging in data-oriented contracting. Such an

---

[15] ISDA Linklaters (August 2017) Whitepaper: Smart Contracts and Distributed Ledger – A Legal Perspective, https://www.isda.org/a/6EKDE/smart-contracts-and-distributed-ledger-a-legal-perspective.pdf

[16] Green, Smart contracts, Interpretation and Rectification, p. 239

agreement would serve as a legal foundation for subsequent data-oriented contracting by establishing, from the outset, important topics such as the meaning of data or the processes by which the parties will handle unanticipated exceptions.[17]

4.14. Another approach would be to incorporate existing data standards, created by a centralised body and publically available. This would allow both parties' computer systems automatically to indicate to each other that they are sharing a common interpretation for the data by reference to a public standard. This approach is already used in the context of other digital agreements (for example, electronic financial contracts are commonly composed according to various sets of pre-defined data standards) but the principle arguably applies to smart contracts also.

4.15. Either way, it is likely that smart contracts will need to be supplemented by some form of underlying agreement – whether it is a separate threshold agreement entered into by the parties for the purposes of entering into a smart contract, or the incorporation of an existing set of standards. This is a further important consideration for the interpretation of smart contracts in the event of a dispute. The courts may need to interpret a smart contract in conjunction with an underlying agreement or set of standards. This is, of course, a process that the courts are already well equipped to do but the need to define an approach to code prior to entering into a smart contract is an important element of smart contracting that needs to be considered.

Expert input

4.16. A third key difference is that the contracting process may be shaped by a different set of expertise.

4.17. In a traditional contract, it would typically be the role of lawyers to draft, negotiate and agree terms to reflect the intention and agreement of the parties. In a smart contract, lawyers will need to draw upon coders and/or legal engineers. Coding is a specialist skill, and at this point in time, there are very few lawyers who are also

[17] Sunden, Computable Contracts, p. 651

coders. Hence, until we update the legal professional skillset to include coding, lawyers will no doubt need to draw upon coders and legal engineers to assist them in drafting smart contracts. As Sarah Green has said, "perhaps the most significant change that smart contracts augur is that the most marketable lawyers will in the future be as fluent in the language of coding as they are in the language of the law. Until that time, such external translation will be crucial to the interpretative exercise."[18]

4.18. Smart contracts will sometimes need to rely on external data feeds. In the distributed ledger community, these separate data sources are known as 'oracles'. For example, a conditional term may depend on whether a particular stock price has reached a certain level, and in order for the smart contract to determine whether this stock price has been reached, it will need to reference stock price data. This external data source would be the 'oracle' to which the smart contract code refers.

4.19. Unlike the blockchain itself, these oracles are not fully decentralised. This means that in entering into a smart contract which requires an oracle, the contracting parties will – to some degree – have to trust the operator of the oracle and the authenticity of the data.[19] Hence, in the smart contracting process, expertise may be required to ensure that the appropriate and agreed oracles are incorporated into the smart contract.

5. How could the smart contract go wrong?

5.1. The smart contract, or at least the smart legal contract, is unfortunately not so smart that it may not go wrong. There are three broad categories of how a smart contract may go wrong that I would like to highlight.

5.2. The first category of potential smart contract issues includes the most basic way in which it may go wrong such as computer glitches – i.e. where some event external to the code affects the smart contract's ability to self-execute. This may be due to a

---

[18] Green, Smart contracts, Interpretation and Rectification, p. 246
[19] Werbach and Cornell, Contracts Ex Machina, p.336

system failure or a technical fault. In such cases, any dispute arising out of such circumstances must be capable of adjudication.[20]

5.3. The second category of smart contract issues includes the more ‘technically complex’ ways in which a smart contract may go wrong: for example, where the smart contract does self-execute, but it is claimed that the code does not accurately reflect the agreement. If one or both parties raises a dispute with the action performed by a smart contract, or is dissatisfied that the outcome is not as anticipated, it could be found that the code does not properly reflect what the parties agreed and the code may need to be rectified.[21]

5.4. It may seem counterintuitive to illustrate such a case by casting minds back to the 19th century - when the technological innovations of that time were steam and electricity as opposed to blockchain - but for reasons that will become clear, I would like to illustrate this point by reminding you of the 1864 case on contractual mistake, *Raffles v Wichelhaus,* more commonly known as “*The Peerless*”[22].

5.5. The “*Peerless*” case arose out of a dispute between two parties who contracted for the sale and purchase of a quantity of cotton. The claimant, Mr Raffles, entered into a contract to sell 125 bales of Surat cotton to the defendant, Mr Wichelhaus, at the rate of 17 ¼-d. per pound. Pursuant to the agreement, it was agreed between the parties that the cotton would arrive by the ship ‘Peerless’, sailing from Bombay to Liverpool.

5.6. In fact, unknown to the parties, there were two ships named ‘*The Peerless’* and both parties had a different Peerless ship in mind when they entered into the contract. The defendant pleaded that he had meant the *Peerless* ship that sailed from Bombay in October; and the claimant pleaded that he had meant the *Peerless* ship sailing from Bombay in December. The effect of this was that when the Surat cotton arrived in Liverpool on the ‘second’ ship, the defendant refused to pay on the basis that (in his mind) it was by that time months late.

---

[20] LawTech Delivery Panel, *Legal statement on cryptoassets and smart contracts* at §136
[21] LawTech Delivery Panel, *Legal statement on cryptoassets and smart contracts* at §154
[22] *Raffles v Wichelhaus* [1864] EWHC Exch J19, (1864) 2 H & C 906; 159 ER

5.7. Standard contract law held this agreement unenforceable and found for the defendant: the doctrine of mutual mistake excuses performance when both parties were mistaken about an essential fact. The question for us to have in mind today, however, is what would have happened had this same agreement been made as a smart contract?

5.8. As long as the smart contract code was satisfied that the conditions of the contract had been met (i.e. that a *Peerless* ship had delivered the correct quantity of goods at the right place) then the code would go ahead and automatically execute the contract regardless, probably on the basis of whichever *Peerless* arrived first, transferring the payment from buyer to seller automatically upon the ship's arrival.[23] The self-executing element of the smart contract is a bit like a runaway train: once the smart contract has been entered into, there is no stopping it or even slowing it down; it will execute the agreement regardless and the parties will have to raise an issue further down the line if it appears that the code does not properly reflect what they had agreed.

5.9. The third category of potential smart contract issues includes those more 'traditional' ways in which a smart contract may go wrong, whereby a common law or equitable issue arises under the contract, e.g. duress, fraud, misrepresentation or illegality. In such cases, just as with any contract, a court would need to intervene. As was recognised by the LawTech Delivery Panel's legal statement on smart contracts, "just because a contract is a smart contract does not mean that the normal and well-established rules do not apply."[24]

6. Judicial remedial action

6.1. So, if a smart contract were to go wrong for one of the reasons I have just outlined, how would the courts go about remedying a smart contract dispute?

---

[23] Werbach and Cornell, Contracts Ex Machina, p. 369

[24] LawTech Delivery Panel, *Legal statement on cryptoassets and smart contracts* at §155

6.2. Common law systems have a great advantage here because of the inherent flexibility of the common law. This means that judges are able to apply and adapt existing principles of law to new situations as and when they arise, without having to necessarily wait for – or depend upon – legislative intervention. As a result, the common law has historically been able to adapt to technological and business innovations, "including many which, although now commonplace, were at the time no less novel and disruptive than those with which we are now concerned."[25] Smart contracts are no exception, and in my view, when a smart contract dispute does make its way to the courtroom, the court will be well-equipped to deal with it using ordinary and well-established legal principles. This issue was addressed by the Law Commission which considered in detail how rules governing, for example, rectification, contract vitiating factors, breach of contract, frustration and illegality may apply to smart contracts (see Chapter 5).

6.3. A major challenge for the courts will be the process of smart contractual interpretation. When a dispute arises because it is claimed that the code does not accurately reflect the agreement a judge will have to interpret the smart contract, looking at the contract as a whole (which may include coded terms and natural language terms and may also incorporate a supplementary contract), and will have to take into account any admissible evidence, in order to ascertain what the parties objectively intended their rights and obligations to be.

6.4. This amounts to a relatively complex exercise of smart contractual interpretation, keeping in mind the various differences that I have already spoken about: whether the code was drafted by a lawyer, third-party coder, or legal engineer; whether the court may be assisted by an underlying threshold agreement or incorporated set of standards; whether the smart contract refers to an oracle, and so on. If the code contains ambiguities, or the court needs to understand how the coded elements and the natural language elements of the smart contract fit together, expert evidence may be required. In effect, at this stage, the court would need to look to the interactions and

[25] LawTech Delivery Panel, *Legal statement on cryptoassets and smart contracts* at §3

communications *outside* the code in order to ascertain whether the code correctly implemented the agreement.[26]

6.5. To illustrate what a judge might be faced with, a piece of code could look like this:

```
function checkExpired(uint campaigned) returns (bool expired)
    {
        expired = false;
        var campaign = campaigns[campaigned];
        if (campaign.deadline > 0 && block.timestamp > campaign.deadline)
                for (unit I = 0; I < campaign.num_contributors; i++) {
                        send(campaign.contributions[i].contributor,
                                campaign.contributions[i].amount);
                        delete campaign.contributions[i];
                }
                delete campaign;
                expired = true;
        }
}
```

6.6. This piece of code is an extract from a simple program that allows users to contribute directly to an online fund. This part of the program tells the computer that, on receipt of a donation, it must check the fund's deadline, and if the deadline has passed, the funds are to be returned to the donor and the fund itself must be deleted.[27]

6.7. The fundamental challenge for a judge interpreting coded terms will be ascertaining what a section of code like this one above *means*.

6.8. Under English law a contract is interpreted by asking what meaning the language of the contract would convey to a reasonable person having all the background knowledge which would reasonably have been available to the parties at the time the contract was made. As pointed out, however, in the Law Commission Report at para 4.33: "Natural language terms are designed to be read by human persons, and so it makes sense to ask what a reasonable person would have understood those terms to mean. However, code is not written with a reasonable person in mind. It is directed at

---

[26] LawTech Delivery Panel, *Legal statement on cryptoassets and smart contracts* at §152

[27] Green, Smart contracts, Interpretation and Rectification, p. 239

a computer. Asking what a reasonable person would understand a coded term to mean is unlikely to assist in ascertaining the meaning of such term".

6.9. The conclusion of the Law Commission was that: "interpretation of a coded term should be determined by asking what the term would mean to a reasonable person with knowledge and understanding of code – that is, a "reasonable coder". The answer to this question will be determined by reference to what the code, in that person's reasoned opinion, appeared to instruct the computer to do" (para 4.48).

6.10. Once the process of smart contractual interpretation has been completed, the court will need to consider the remedies available to the parties in a smart contract dispute. An important difference to bear in mind at this point is that in most cases a smart contract will have already been performed before the dispute arises. Unless, unusually, the parties appreciate that there is a problem before the time for performance arises, or the contract involves repeat performance, the self-executing nature of a smart contract means that performance will have taken place and one will therefore primarily be looking at post-performance remedies.

6.11. It has been argued that the equitable remedy of rectification may well become more widely used because of the fact that smart contracts are self-executing.[28] This may well be the case where the parties become aware that the code does not reflect what was agreed before performance or where it is relevant to continuing contractual relations.

7. Smart contracts, financial services and banking:

7.1. Having discussed how smart contracts work, and the benefits of smart contracts as tamper resistant, self-executing and self-verifying, as well as the technical and legal issues that may arise, I will now outline some of the practical applications of this technology. The financial services and banking sectors have already recognised the

[28] Generally: Green, Smart contracts, Interpretation and Rectification

promise of this technology, but implementation and scalability of this technology are difficult.

7.2. One application of smart contracts is in the insurance claim sector. The process of assessing an insurance claim's legitimacy is drawn out. Insurance contracts are often paper contracts, stored offsite. Manually counterchecking the terms of a contract and validation of a claim takes time and is prone to human error. Compounding this complexity is the number of parties involved, where consumers, brokers, insurers and reinsurers all need to be informed of the terms of the contracts and any settlements to be made pursuant to them. Smart contracts offer great potential to aid this sector and improve this process. In theory almost every insurance payout might be automated with the use of smart contracts. Companies like ScienceSoft are providing these services and note that smart contracts can intake and process data from an insurer's corporate systems and relevant third-party sources to accelerate underwriting and claim resolution cycles.[29] For example, in the car insurance context, it is possible to involve a certified mechanic to provide for automatic indemnity to the policyholder if the vehicle is repaired by that mechanic, with the mechanic itself confirming this by sending a transaction to the smart contract.[30] These improvements may reduce costs and improve efficiency.[31]

7.3. Another application of smart contracts is in the decentralised finance sector. As a counterweight to the traditional, heavily centralized, financial market infrastructure, decentralised finance uses smart contracts and ledger technology (usually Ethereum[32]) to provide financial services on a bilateral basis, between the service providers and their clients. For instance, Uniswap[33] is a company allowing individuals seeking to earn interest on their savings to use decentralised finance protocols to lend money to individuals that seek financing on a peer-to-peer basis, without using an intermediary. Likewise, owners of crypto assets can earn interest by

---

[29] ScienceSoft, "Smart Contracts in Insurance", https://www.scnsoft.com/insurance/smart-contracts
[30] Gatteschi V et al (2018) Blockchain and smart contracts for insurance: is the technology mature enough? Future Internet 10(2):1–16
[31] CBI Insights, "How blockchain is disrupting insurance", 19 July 2022, https://www.cbinsights.com/research/report/blockchain-insurance-disruption/
[32] Decentralized finance (DeFi) | ethereum.org
[33] UniSwap, https://app.uniswap.org/swap

providing liquidity to online market places (liquidity pools) where such crypto assets can be traded.

7.4. Smart contracts are also being adopted in heavily regulated financial sectors. The use of smart contracts in derivative transactions is seen as a solution that can enable more efficient monitoring and execution of complex and large derivative contracts that are usually based on comprehensive and heavily standardised master agreements. For instance, payment related provisions of a derivative contract, that require one party to pay a certain amount to another party upon the occurrence of certain events, can be coded into a smart contract that enables automatic execution. External information (like exchange rates necessary for calculation of payment amounts based on pre-defined calculation methodologies) can be incorporated into a smart contract via application programming interfaces (APIs) which are also deployed on the distributed ledger, meaning that external information may be incorporated into the computer code of the smart contract in a completely automated way. This streamlines the process significantly, and can reduce the counterparty risk associated with such trades. ISDA[34] has recognised the importance of smart contracts for derivative markets, and has published a number of papers so far analysing various legal and regulatory aspects of the use of smart contracts for the purposes of automation of derivative contracts.

7.5. Smart contracts can also be used in the trade finance and supply chain documentation space. Because they are more efficient than paper-based systems, smart contracts can reduce processing times of supply chains and trade finance. While digitizing letters of credit and bills of lading should reduce the opportunity for forgery, blockchains can also be used to secure public receipts and transactions and ease workflow management with digital signatures. Bank of America, Barclays Corporate Bank, Standard Chartered, and the Development Bank of Singapore are all in the process of testing the use of smart contracts to automate log change of ownership and payment processes for their organisations[35].

---

[34] ISDA, "ISDA Smart Contracts", https://www.isda.org/2019/10/16/isda-smart-contracts/

[35] Deltec Bank, "Smart Contracts and Financial Services", https://www.deltecbank.com/2022/02/15/smart-contracts-and-financial-services/, 2023

7.6. Smart contracts can also be used to create efficient equity settlements that prevent discrepancies and save costs. An Accenture survey of eight banks found that clearing and settling costs of transactions could be reduced by $10 billion USD through blockchain technology[36]. Wall Street has successfully tested smart contracts for clearing and settlement[37] and is looking to expand the application of this technology. However, the road to integrating these new technologies is not straightforward. Adoption of these technologies requires numerous parties to be onboard to novel platforms quickly, outlay significant technology adoption costs and manage implementation carefully. In the banking and financial services sector, companies and startups working with blockchain technology will also have to overcome significant regulatory and legal hurdles before there is industry-wide adoption.

7.7. These barriers can be difficult to overcome. By way of example, in December 2022, the Australian Securities Exchange and the Depository Trust & Clear Corporation abandoned plans to develop a smart contract based clearing and settlement system after failing to implement the technology with a spend of $150 million dollars[38]. These difficulties demonstrate that whilst the technology may work, implementation can be problematical.

8. Concluding remarks

8.1. The challenges that I have addressed today may seem a little premature. It may be some time before a court is tasked with the exercise of interpreting a smart contract. In spite of the fact that smart contracts are at the forefront of technological innovation in the commercial field (in particular in financial services) they are yet to become widely used. Nevertheless, it is important that these questions are addressed as early as possible.

---

[36] FT, “How can blockchain platforms make currency trading cheaper”, https://www.ft.com/content/10d1d7ed-6ae2-457a-a901-ff1937ec0f8f, 2 November 2023

[37] Forbes, “Will Blockchain replace clearinghouses”, 2 December 2020, https://www.forbes.com/sites/philippsandner/2020/12/02/will-blockchain-replace-clearinghouses-a-case-of-dvp-post-trade-settlement/

[38] Financial Times, “Australian stock exchange apologies for dropping botched blockchain upgrade”, https://www.ft.com/content/029dd01f-eaf5-493c-b195-299408b62469

8.2. The legal statement issued by the UK LawTech Delivery Panel, and the Law Commission's Report, were widely welcomed as providing much-needed legal certainty. Their conclusion that smart contracts are capable of satisfying the requirements of contracts in English law and are thus enforceable by the courts – may well boost confidence in the adoption of new technologies and the use of smart contracts. The Law Commission's Report also includes at Appendix 3 a helpful list of issues which parties may wish to provide for in their smart contracts.

8.3. There is a school of thought that considers smart contracts to be outside the scope of the traditional legal system and that their self-executing nature means that they will not require the intervention of lawyers, or the adjudication of judges. As is illustrated by some of the issues discussed, I very much doubt this.

8.4. There are many ways in which smart contracts are revolutionary, but fundamentally, they are contracts enabled by advanced technology. Technology is not faultless and agreements are never entirely free of ambiguity. If distributed ledger technology had been available in the 19th century, and the case of the *Peerless* had been drafted as a smart contract, it may still have run into the same problems: the term referring to the relevant ship could still have been open to two interpretations even if it had been written in code and the smart contract had been self-executing. Smart contracts are revolutionary, but, as you will no doubt be pleased to hear, not so revolutionary as to remove the need for lawyers and judges.[39]

---

[39] I am very grateful to my judicial assistants, Gemma McNeil-Walsh and Mannat Malhi, for their research in relation to this paper and help in its presentation.