Published on 8 September 2021

COMING TO TERMS WITH SMART CONTRACTS PART 2 – ENCODING CERTAINTY AND ENFORCEABILITY IN CONTRACTS "*EX MACHINA*"

[2021] SAL Prac 26

TEO Yi-Ling
LLB (Liverpool); LLM (Northwestern University, Chicago);
Barrister-at-Law (Middle Temple); Advocate and Solicitor (Singapore);
Senior Fellow, Centre for Excellence for National Security, S Rajaratnam School of International Studies, Nanyang Technological University

> The contract is the keystone of issuance. Our innovation is to express all the salient details of an issuance as an unforgeable contract, unforgeably linked into every action within a payment system. In this way, financial innovation can develop along the lines it always has done – by means of innovation within contracts. By translating the institution of the contract into the digital domain, we build upon centuries' and even millenia' worth of experience in documenting, sharing and disputing the meaning of agreements between parties.[1]
>
> –Ian Grigg

I. Introduction

1 In the first part of this article,[2] the focus of discussion was on the security of smart contracts and practical matters around their adoption. This second part will attempt a deeper analysis of the nature of smart contracts and the contractual certainty, validity, and consequential enforceability of these. Fundamental questions arise as to the existence of a binding contract as regards smart contracts: Was there consensus *ad idem*? To what extent – if at all – can contractual intention be properly

1 Ian Grigg, "The Ricardian Contract" (1996) <http://iang.org/papers/ricardian_contract.html> (accessed 13 April 2021).
2 Teo Yi-Ling, :Coming to Terms with Smart Contracts Part 1 – Fintech Security Challenges and Considerations" [2020] SAL Prac 23.

encoded in such contracts, and what does this subsequently mean for performance? Upon whom does liability land when the functioning of smart contracts goes wrong?

2 Szabo's definition of a smart contract as "a set of promises, specified in digital form, including protocols within which the parties perform on these promises"[3] references two frameworks of rules governing behaviour in relation to carrying out obligations and enforcing promises – the functioning of code, and established rules of contract formation. This part will deal with the questions of how these two frameworks sit with each other in the context of smart contract functionality and examine whether there are sufficient points of convergence between their respective rules. Further, as the smart contract has been described as a form of "algorithmic contract", the formation and execution of which are automated, it will be instructive as well to examine the types, nature and characteristics of the same.

3 Ahead of further discussion, it may be helpful here to reiterate the technical definition of a smart contract for context: it is a computer protocol – a set of rules for the transmission of data between computers; it executes according to *how it is coded*, without any downtime, fraud, control, or interference from a third party. It is "intended to digitally facilitate, verify, or enforce the negotiation or performance of a contract".[4] It is the basic building block of the Ethereum decentralised platform and programmers can write smart contracts on the Ethereum blockchain, using programming languages.

4 Two current contrasting attitudes to the nature of smart contracts are:

(a) they encapsulate a legal agreement within its code (the "Lex Informatica" or "code of law" school of thought, which will be explained further along in this article) and

3 Nick Szabo, "Smart Contracts: Building Blocks for Digital Markets" (1996).
4 Ameer Rosic, "Smart Contracts: The Blockchain Technology That Will Replace Lawyers" *Blockgeeks* (2016) <https://blockgeeks.com/guides/smart-contracts/> (accessed 2 July 2020).

are properly enforceable contracts at law, therefore is no requirement for the intervention of the law or lawyers;

(b) it is code that merely automates the performance or enforcement of an existing or underlying legal agreement, whether in writing or otherwise.

5 For reasons which will be discussed, the prevailing attitude is mostly the latter. Whether or not smart contracts are capable of being categorically deemed "contracts at law" will come down to whether, in essence, all the requirements of a legally binding contract are satisfied. This then is evidence to demonstrate *consensus ad idem* – the requisite meeting of the minds, to obviate any uncertainty as to the contracting parties' intentions. These requirements are addressed in more detail below. As a starting point for discussion, the following assertions are made with respect to the function of contracts and contracting:

(a) Contracts provide the ability for people to express and indicate their respective values and preferences in the process of making mutual exchanges to arrive at agreement over the subject of the transaction.

(b) Conventional rules of contract law operate on the assumption that contracting parties consciously evaluate the terms and conditions in the process of transacting.

(c) When lawyers become involved in this process, their role is to assist in clarifying and articulating the intentions of the parties to the contract, mitigating where possible the risk of uncertainty.

(d) Contractual arrangements can be extremely complex, and drafting a contract that accounts for all probabilities, permutations, and contingencies is impossible.

II. Code is law *versus* code is under law

6 In Jewish Hasidic lore, there appears the story of a golem – an artificial creature animated by secret magical incantations, created by a rabbi in 16th century Prague. The golem was created

for carrying out specific tasks, among these protecting Jews from attacks by those that hated them. All was well as long as the rabbi correctly ensured the proper functioning of the golem, but a day came when he omitted to do so, and the golem ran amok, causing damage and threatening life. The rabbi was then forced to render the golem lifeless.[5] This story indeed holds a number of symbolic relevancies for the issues discussed in this article.

7 Code is a series of steps or set of instructions, written in a high-level computer language and automatically translated in order to be readable and executable by a computer. It can be characterised as being "low complexity, repetitive, or non-critical … and is straightforward to implement".[6] In his book *Code (Version 2.0)*, Lawrence Lessig describes two contexts of regulation: the code of constitutions, statutes, and case law that governs physical reality, and the code operating the software and hardware that is cyberspace; the online, virtual world. He cites others before him who have described the latter as the law that governs cyberspace as "Lex Informatica", *ie*, "code is law".[7] His point was that those engaged in writing application code and constructing software architecture layers do so by a process of making choices about values and outcomes, and in doing so, create rules for how these ecosystems should be regulated and operated.[8] The analogy is thus presented as coders replacing lawmakers in cyberspace; coders creating law and regulation for the virtual world.

8 In the first part of this article, a description of the Ethereum ecosystem was provided, explaining the various significant layers that constituted the same. It is true that in creating this ecosystem, the coders creating the layers in the software stack operating off the blockchain make such decisions about operationality. This perspective of a separate law of cyberspace is possibly in line with the libertarian vision of cyberspace being a place of freedom

5 Alden Oreck, "Modern Jewish History: The Golem". Also note that certain denominations or sects within Judaism regard themselves as separate from other denominations and mainstream society, and self-governing.
6 John Spacey, "Algorithms vs Code" *Simplicable* (7 August 2016).
7 Lawrence Lessig, *Code: Version 2.0* (Basic Books, 2006) at p 5.
8 Lawrence Lessig, *Code: Version 2.0* (Basic Books, 2006) at pp 6–7.

and self-governance; a place where coders were free to dream and bring into existence the means of self-defining action and direction, encoding in the values that should prevail in such an existence. Further, the coders use a scripting language – with its own syntax rules – to create the smart contracts. Hence, by operation of logic, the "code is law" argument extends to the manner in which smart contracts should be regarded – that their coding prescribes the only means by which they are regulated.

9 As Utopian as this may seem, it is also arguably naïve, given that the real world and that of cyberspace are virtually inseparable – events occurring in one realm will have effects in the other. In the Jewish story about the golem – the magically animated artificial being – that malfunctioned, its creator the rabbi had to de-animate it permanently to avoid causing further damage in the real world. To assert that "code is law" is to wholly ignore these realities:

(a) as stated above, events that occur in cyberspace have effect on the order of things in reality;

(b) society will demand that the *lex loci* must identify the real entity to which accountability for damage must accrue, on the basis of restitution;

(c) in the absence of affirmative choices being made about matters that have a legal bearing, a choice to abide by established default laws is effectively made, and this allows courts to provide predictable outcomes; and

(d) the existence of vitiating factors, including fraud and illegal agreements, alongside statutory and policy limitations upon the making of voluntary agreements.

10 In the first part, of this article, the DAO hack and the subsequent decision to create a hard fork to provide a remedy was discussed. Interestingly, someone claiming to be the hacker wrote an open letter to The DAO and Ethereum Community when it became public that the hard fork remedy was being considered and claimed that his activities were legal (upon consultation with

his lawyers).[9] Ironically, this individual threatened conventional legal action against any attempts to invalidate or circumvent the outcomes of his actions.[10] The proponents of the "code is law" position may wish to remind themselves that just as cyberspace as we now know it is subject to government intervention, determining the legality of smart contracts is very much within the *lex loci*. Automation of execution *does not imply* autonomy of regulation and self-governance.

III. Legal limitations of smart contracts

> Since there is no such thing as a reasonable computer, there is no point in approaching the interpretative exercise in relation to smart contracts by asking what a reasonable machine, or a reasonable user of human language would make of the language used.[11]
>
> –Adam Sanitt & Sarah Green

11 The discussion now turns to considering the practical legal issues faced by smart contracts; effectively, these mostly arise from the nature of their authorship – in code. Arguably, these issues present themselves as variations of the theme of lack of *consensus ad idem*. For illustrative purposes, below is an example of a smart contract, authored in Solidity, for the purpose of conference ticket purchases. The parameters of the contract are: allowing registered participants to buy tickets, allowing the organiser to set a cap on the number of attendees, and allowing for refunds.[12] It is written using the conditional <<if/then>> statement logic.[13]

---

9 "The Attacker", "An Open Letter" *Pastebin* (18 June 2016) <https://pastebin.com/CcGUBgDG> (accessed 14 April 2021).

10 "The Attacker", "An Open Letter" *Pastebin* (18 June 2016) <https://pastebin.com/CcGUBgDG> (accessed 14 April 2021).

11 Adam Sanitt & Sarah Green, "Smart Contracts" *Norton Rose Fulbright* (November 2019).

12 ConsenSys, "A 101 Noob Intro to Programming Smart Contracts on Ethereum" *Medium* (29 October 2015).

13 ThinkAutomation, "A Beginner's Guide to "IF" Statements". Essentially, "if" statements are logical blocks of code used in programming. They are conditional statements that instruct a computer what to do when certain information is input. Looked at another way, they allow a programme to make "decisions" while operating.

```
contract Conference {
 address public organizer;
 mapping (address => uint) public registrantsPaid;
 uint public numRegistrants;
 uint public quota; // so you can log these events
 event Deposit(address _from, uint _amount);
 event Refund(address _to, uint _amount); function Conference()
{// Constructor
 organizer = msg.sender;
 quota = 500;
 numRegistrants = 0;
}
 function buyTicket() public returns (bool success) {
 if (numRegistrants>= quota) {return false;} // see footnote
 registrantsPaid[msg.sender] = msg.value;
 numRegistrants++;
 Deposit(msg.sender, msg.value);
 return true;
}
 function changeQuota(uint newquota) public {
 if (msg.sender!= organizer) {return;}
 quota = newquota;
}
 function refundTicket(address recipient, uint amount) public {
 if (msg.sender!= organizer) {return;}
 if (registrantsPaid[recipient] == amount) {
 address myAddress = this;
 if (myAddress.balance>= amount) {
 recipient.send(amount);
 registrantsPaid[recipient] = 0;
 numRegistrants–;
 Refund(recipient, amount);
}
}
}
 function destroy() {// so funds not locked in contract forever
 if (msg.sender == organizer) {
 suicide(organizer); // send funds to organizer
}
}
}
```

A. Inherent shortcomings of code and the construction of contracts

12 One question that immediately surfaces is whether the above code discloses the various instances upon which refunds can be made. Does it allow for refunds to be made for any reason? Or do the subsidiary "if–then" statements coded prescribe specific instances where refunds will be allowed? This illustrates the main issue of establishing certainty in smart

contracts: as such contracts are expressed in code, they are therefore mostly incomprehensible except to their coders. It would certainly be unreasonable and unrealistic to expect parties to a smart contract to be conversant with code. Notwithstanding the basic contract law tenet that a contract does not need to be in writing to be valid and enforceable, there must be some evidence of a meeting of the minds of the parties, arising out of the ability to reasonably comprehend and assess the terms of the contract. Can a contract written in a language that the parties objectively do not understand be valid and enforceable?

13 In the context of "clickwrap" Internet contracts, the argument for validity is borne out by the fact that a user has had the actual opportunity and ability to read the terms and conditions of the service provided – written in natural language – ahead of deciding whether to click "I agree" or "I disagree". Smart contracts in and of themselves do not have this inherent quality.

14 Further, it is unclear if smart contract code is able to articulate or automate matters that are commonly included in conventional contracts, such as limitation of liability, non-disclosure, termination, choice of law, and dispute resolution mechanisms. One is reminded here that in the absence of affirmative choices regarding these, resort will be had to whatever the *lex loci* provides as a default mechanism.

B. The problem of subjective performance

15 In conventional contract drafting, adjectival language is commonly used to describe the requisite standards of the performance of obligations by the parties – examples are "undertake best endeavours in", "acceptable to the other party", "to the satisfaction of", or "to a reasonable standard". The yardsticks here to measure fulfilment of performance are subjective ones, and in contracts where this is desired, smart contracts are severely limited in their ability to articulate this. As seen from the smart contract example above, its code relies on a certain precision of logic for execution. Sanitt and Green make the point that while humans possess context and social conditioning to aid in the interpretation of language, a computer

relies solely on the construction of the syntax of code. They provide a cogent example to show how humans and machines have different cognitive responses to language:[14]

> … take as a very basic example the following instructions:
>
> > 'Go to the shop and buy a newspaper. If there any eggs, get a dozen.'
>
> The most likely human response to this instruction is to buy a newspaper and, in the event that the shop has eggs, to buy a dozen eggs as well. A computer, on the other hand, presented with this instruction, would buy a newspaper and, in the event that eggs are also available, will buy 12 newspapers rather than one. In response to this instruction, it will not buy a dozen eggs because at no point has it been told to purchase eggs; the only object of the 'buy' mandate is a newspaper.

16 Is it possible for this cognitive deficiency on the part of computing to be remedied by thoughtful and intelligent coding? If not, and together with the fact that subjectivity does not lend itself to translation or reduction into code, significant doubt is cast upon broadening the utility of smart contracts. What this means in practice may well be that only relatively simple and straightforward transactions requiring objective standards of performance will lend themselves to being articulated in smart contracts, with a higher probability of certainty and enforceability.

C. Uncertainty as to identities

17 Technically, every smart contract and account residing on the Ethereum blockchain is identified by a unique address, through which Ether ("ETH") can be sent and received. Addresses in Ethereum are the identifiers of either normal accounts (usually managed by users) or other contracts.[15] In the article

14 Adam Sanitt & Sarah Green, "Smart Contracts" *Norton Rose Fulbright* (November 2019).

15 Sebastian Peyrott, An Introduction to Ethereum and Smart Contracts: a Programmable Blockchain" *Autho* (28 March 2017). A digital wallet is part of the mechanism of smart contract functioning, and it contains two cryptographic keys: a private key for user access to their accounts, and a public key for authentication and encryption of messages. The address of the account is a fragment of the public key. Both keys pair to allow for secure communication.

"Smart Contracts: A Boon or Bane for the Legal Profession", the author raises an issue that may affect the legal validity of smart contracts: uncertainty as to contracting parties.[16] The point is made that the current means of identification of parties by way of public addresses and their respective owners is deficient for two reasons:

> Firstly, the public address in a smart contract transaction may point to another smart contract instead of a wallet with a determinate owner. This information cannot be discerned from the public address alone because the public address of a wallet is often indistinguishable from that of a smart contract. Furthermore, it is also possible for a smart contract to participate in another smart contract transaction … Secondly, even if the public address belongs to a wallet, the owner of that wallet remains pseudonymous and cannot be discerned from the public address itself without extraneous information.

18 Unless and until this emerges as an actual point of contention for the courts to decide, the probability is that this issue will remain unaddressed, or it may find resolution in a technical intervention that rectifies this deficiency.

IV. Beyond code: the intervention of algorithms

19 Thus far, the discussion has focused on the limitations of smart contracts with regard to the nature of code and coding. There are situations where an algorithm is included in the programming of an automated transaction, which adds a layer of complexity in thinking about implications for contractual certainty. Although similar to code in that it is a sequence of steps used for problem-solving, performing calculations or completing tasks, an algorithm is characterised as being more complex and having a highly optimised or rigorous design.[17] Lauren Scholz has described these automated transactions as "algorithmic contracts" and has offered a definition of the same as follows:[18]

16 Gary Tse, "Smart Contracts: A Boon or Bane for the Legal Profession?" *Taylor Vinters Via* (24 September 2018).
17 John Spacey, "Algorithms vs Code" *Simplicable* (7 August 2016).
18 Lauren Scholz, "Algorithmic Contracts" (2017) 20 Stan Tech L Rev 128 at 134.

> Algorithmic contracts are contracts in which one or more parties use an algorithm to determine whether to be bound or how to be bound. More specifically, algorithmic contracts are contracts that contain terms that were determined by algorithm rather than a person. An algorithm is a process or set of rules to be followed in calculations or other problem-solving operations, especially by a computer.

20 Scholz identifies three manifestations of algorithmic contract in practice: high frequency trading (which gave rise to the action in *Quoine Pte Ltd v B2C2 Ltd*[19] ("*Quoine*")), dynamic pricing, and Ethereum smart contracts.[20] Further, she presents a taxonomy of algorithmic contracts in order to assess their nature and function in determining parties' rights and obligations in a contract – acting as either a gap-filler or a negotiator in the process of contract formation:[21]

> In gap-filler algorithmic contracts, parties agree that an algorithm, that operates at some time either before or after the contract is formed, will serve as a gap-filler, determining some term in the contract. In negotiator algorithmic contracts, one or more parties use algorithms as negotiators before contract formation … The algorithm chooses which terms to offer or accept, or which company to do the deal with.

21 A common example of a gap-filler algorithm in action is when a person makes a purchase on Amazon.com. While Amazon has standard terms and conditions for purchasers, it employs proprietary algorithms to determine an exact price at any given time, per purchaser.[22] As regards negotiator algorithmic contracts, an example would be high frequency trading of financial products via algorithms that generate real-time trading strategies to take advantage of market conditions, better than humans are able to.[23] High-frequency trading was the context

19 [2020] 2 SLR 20.
20 Lauren Scholz, "Algorithmic Contracts" (2017) 20 Stan Tech L Rev 128 at 137.
21 Lauren Scholz, "Law and Autonomous Systems Series: Toward a Consumer Contract Law for the Algorithmic Age" *University of Oxford* (17 April 2018). Scholz has also discussed the purchasing of airline tickets as an example of dynamic pricing in her article referenced at fn 17.
22 Lauren Scholz, "Law and Autonomous Systems Series: Toward a Consumer Contract Law for the Algorithmic Age" *University of Oxford* (17 April 2018).
23 Lauren Scholz, "Law and Autonomous Systems Series: Toward a Consumer Contract Law for the Algorithmic Age" *University of Oxford* (17 April 2018).

within which the recent landmark case of *Quoine* unfolded. The Singapore Court of Appeal was seized with determining whether a high-frequency cryptocurrency trading agreement formed purely through the operation of algorithms could be unilaterally cancelled for mistake.

A. Brief facts of Quoine Pte Ltd v B2C2 Ltd

22 Both parties were market-makers active in cryptocurrency trades, creating liquidity on the trading platform ("Platform"), and both used deterministic algorithms to close trades, *ie*, with no direct human intervention. Due to errors in Quoine's algorithm, a sequence of events was set in motion which triggered margin calls against counterparties, who had borrowed ETH from Quoine to buy Bitcoin ("BTC"). Quoine's algorithm automatically sold ETH to the counterparties in exchange for BTC, so as to repay the ETH loans. However, the software error caused ETH to be traded at around 250 times the going market rate, resulting in a windfall for B2C2. Realising this, Quoine unilaterally cancelled and reversed these trades, and B2C2 claimed against Quoine for breach of contract.

23 One of the defences raised by Quoine was that the underlying trades were void for unilateral mistake at common law or voidable in equity, which entitled Quoine to reverse the trades. The main consideration for the Court of Appeal was determining what type of knowledge the non-mistaken party possessed. The trade contracts were formed and executed by deterministic algorithms, with neither party possessing any knowledge of their formation or terms prior to the same being concluded. One of the mistaken beliefs Quoine asserted that the counterparties had was that they were purchasing ETH for BTC at prices that accurately reflected the true market value of ETH with respect to BTC.

24 As regards this assertion, the court held that it was "wrongly characterised" and that as the parties had already agreed that the prices would be derived by the parties' respective algorithms, the actual basis for the mistake was the belief that the platform would not malfunction. Even then, the court held this was a mistaken assumption as to the circumstances under

which the trades were entered into, and not a mistake in respect of a contractual term. The common law doctrine of unilateral mistake failed accordingly.

B. A deterministic algorithm as the crux of the decision

25 The Court of Appeal judgment did not venture into the general questions of validity of algorithmic contracts; it appeared that this matter was not germane to its decision on the basis that validity was not an issue between the parties. In *Quoine*, the algorithm in question was a deterministic one, in that it "is one which will always produce precisely the same output given the same input … the Trading Software will do just what it was programmed to do and does not have the capacity to develop its own responses to varying conditions".[24] The fact that it was deterministic was material in assisting the court in arriving at its decision. Staying within the four corners of the case, it did not avail itself of the opportunity to comment or provide an exposition on the nature of non-deterministic algorithms, and what this could have meant for the outcome of this case had the algorithms been non-deterministic. The decision would certainly have been different, and from an anticipatory point of view, it is worth engaging in some prospective deliberation as to possible ramifications of using such algorithms in practice.

26 With respect to the use of a non-deterministic algorithm, it may result in a consequence that is neither expected nor anticipated by the creators of the algorithm. This is so as non-deterministic algorithms provide different results for the same inputs on different executions.[25] In essence, this effectively means that the party responsible for the algorithm has no idea how it operates and is unable to predict its behaviour. These algorithms are also known as "black box" algorithms, and Scholz is of the opinion that algorithmic contracts employing the use of

24 [2020] 2 SLR 20 at [15].
25 "Non-Deterministic Algorithm" *technopedia* (29 August 2019).

the same presents the most difficulty for classic contract law jurisprudence, in respect of certainty, let alone validity:[26]

> These algorithms have decision-making procedures that are not functionally human-intelligible before the program runs – and often cannot even be parsed after the program runs.

27 A chorus of questions arises in terms of whether such transactions are valid and enforceable; if there is actual meeting of the minds: Do contracting parties need to understand what the algorithms are doing on their behalf? What if parties decide to trust the algorithms to act in their best interests? If parties delegate their decision-making to an algorithm, and it gives them a result that they disagree with or do not expect, what recourse is there and with whom does accountability lie? How far backwards along the chain or workflow of intervening actors (and their actions) should accountability travel? What are sensible and natural legal rationalisations in response to these questions? In the event that the performance of the contract gives rise to a result that is manifestly unfair, shouldn't equity find a solution, in the absence of any reasonable legal theory? Some of these questions were raised in *Quoine*. While the court did not determine if it could exercise its equitable jurisdiction in respect of the alleged mistakes in question, it ventured an exploration of outcomes on the assumption that it could. As a starting point, at both common law and in equity, the required knowledge of the mistake must crystallise before the contract is formed, in order for the doctrine of mistake to operate. The majority decision held that in ascertaining knowledge, reference would have to be made to *the state of mind of the algorithm's programmer at the time of creating the algorithm*. If the programmer, at the time of creating the algorithm, contemplated or ought to have contemplated a future party being mistaken *and then wrote the algorithm to take advantage of this*, only then would the doctrine of unilateral mistake apply. Another situation where the doctrine would apply would be if in the *intervening period between algorithm creation and formation*, such knowledge accrues to the programmer, who *still initiates the operation of the algorithm in order to benefit from*

26 Lauren Scholz, "Law and Autonomous Systems Series: Toward a Consumer Contract Law for the Algorithmic Age" *University of Oxford* (17 April 2018).

the mistake. It was held that as B2C2 had not appeared to have engaged in such contemplation, it therefore did not have the required actual or constructive knowledge for the operation of the doctrine.

28 It is submitted that the above reasoning as respects knowledge of the programmer of the algorithm comes across as somewhat contrived and artificial; an attempted squaring of a circle. Is it realistic or reasonable to expect a programmer to have such far-sighted omniscience? Further, such programmer would unlikely be acting independently; he would most probably be taking instructions from employers or persons requiring the use of the algorithm. How then is the programmer realistically a party potentially fixed with relevant intention? Further, it is noted that there was much attention given to the deterministic nature of the algorithm in question: while this process did produce leverage for the majority decision, it disregarded the contextual aspects of algorithm authoring and training which are complex, and the environmental conditions surrounding the use of such algorithms.

29 The dissenting judgment centred on the notion that equity could operate to set aside a contract where a reasonable person, having awareness of prevailing circumstances, clearly perceives that a fundamental mistake has occurred, regardless of when the same took place. Mance IJ stressed the point that notwithstanding that algorithmic contracts devoid of human intervention did not square tidily with conventional principles of unilateral mistake, what was very clear in this situation was the fundamentality of the mistake. He was of the opinion that a court is at liberty to intervene where there is substantive unfairness of the contract, and choose to focus instead on whether the contract is exceedingly onerous to one party. This is the essence of the *obiter* discussion in the case of *Chwee Kin Keong v Digilandmall.com Pte Ltd*[27] on the role of equity in addressing unilateral mistake. Despite the arguable fundamentality of the mistake in *Quoine*, the majority holding instead emphasised the strict common law principle of the non-intervention of the courts as respects

27 [2005] 1 SLR(R) 502.

an agreement freely entered into by the parties. This was so despite the parties knowing they would not even be aware of the formation of a contract or its terms, in this immediate case. It is tempting to ask the question as to what was in effect being protected here – the sanctity of a contract formed with no human involvement, or the interests of the persons affected by the outcomes of erroneous execution of such contract.

C. An agency argument for algorithmic behaviour?

30 Further, when parties allow or agree to a non-deterministic algorithm to act on their behalf, is this tantamount to accepting the risk that it may result in an unexpected outcome and that there will be no explanation as to why that outcome was reached? Does this mean that parties are estopped from asserting breach of contract? In terms of accountability, is it reasonable – or even sensible – to imply a theory of agency on the part of the non-deterministic algorithm and does that extend to the party making the algorithm available as well as the creator of the algorithm? Scholz has argued that in the context of business-to-business transactions, algorithmic contracts are enforceable on the basis that the algorithm can be deemed as a constructive agent for the entity deploying it, and whatever acts are undertaken by the algorithm demonstrate the entity's intention. She makes the following comments about the treatment of "black box", as opposed to "clear box" algorithms where it is possible to anticipate how they will behave and operate:[28]

> Black box algorithmic contracts inherently introduce a gap between the objectively manifested intent of the party using the algorithm and what the artificial agent does. Unlike in typical contracts, where we assume that a 'sophisticated party' knows what it is doing enough to bind and be bound, black box algorithms by definition engage in emergent behavior that cannot be anticipated by a principal. The presumption of deference to general acts showing an intent to be bound, even of a sophisticated party using algorithms, must be relaxed in the case of black box algorithmic contracts, and this relaxed presumption could potentially result in a contract being unenforceable.

28 Lauren Scholz, "Algorithmic Contracts" (2017) 20 Stan Tech L Rev 128 at 136.

31 It is submitted that the agency argument, while relatively sustainable in the context of "clear box algorithms", may not be so with "black box" algorithms, given the extreme degree of uncertainty. Then again, if parties have agreed to accept whatever outcomes these algorithms produce in the execution of the transaction, *ie*, accept the risk of using such algorithms and agree to not hold each other accountable for unwanted outcomes, would such a position be tenable? In other words, can parties use estoppel to limit their contractual liability in such a situation? Inasmuch as it is open to the parties to determine the scope and nature of their agreement, this is still subject to the inherent power of the courts to determine if such agreement is contrary to public policy and unreasonably exclusionary and strike down the same accordingly.

V. Bridging the comprehensibility divide

32 In a *Forbes* article, David Black succinctly summarises the key issues with smart contracts, by comparing sequence steps in a smart contract creation and execution with the real-world processes of contractual formation and execution.[29] Citing issues of inflexibility regarding smart contract platforms, wallets, and choice of cryptocurrency, as well as security, privacy, immutability of smart contracts, comprehensibility, and enforceability (some of which have been discussed in the first part of this article), his overriding reaction is one of doubting whether the promises of efficiency, cost-effectiveness, and ease of doing business in using smart contracts actually hold up. In the absence of any accompanying natural language version of the smart contract, questions as to determination, performance, and enforcement may arise, especially if there are coding errors, or if it is executed in a manner contrary to the parties' expectations. Also, there may be questions about whether the "if-then" protocol of coding will sufficiently address the requirements for legal contractual formation, and whether its logic will allow for flexibility in

29 David B Black, "Blockchain Smart Contracts Aren't Smart and Aren't Contracts" *Forbes* (4 February 2019) <https://www.forbes.com/sites/davidblack/2019/02/04/blockchain-smart-contracts-arent-smart-and-arent-contracts/?sh=2942f8191e6a> (accessed 2 April 2021).

performance. Taking a sober view of all these considerations together – the dependencies on variable programmer execution ability and the technical execution environment – they represent the potential for compounding errors within the final output. The attainment of *consensus* will logically be unattainable. Any time or cost saved by automating the contracting process would be eroded by remediation efforts. It then stands to reason that the fewer the intervening translations or interpretations are, the greater the likelihood of achieving *consensus*.

33 In considering how the divide between traditional contractual conventions and those presented by smart contracts can be bridged, one approach that has been cited elsewhere is that of using Ricardian contracts as a link. The other suggested here could be the adoption of a mechanism similar to the Creative Commons "three-layer" licence design framework.

A. Ricardian contracts

34 First introduced in 1995 by programmer Ian Grigg, Ricardo is a financial cryptography system for "effecting payment in a wide range of instruments, including money, financial instruments and gaming units across the Internet".[30] The Ricardian contract was invented to effect secure contracts over the Internet, and enables any contract "that represents assets to become the basis for an Internet payments system".[31] Grigg's own definition of a Ricardian contract is as follows:[32]

> … a single document that is a) a contract offered by an issuer to holders, b) for a valuable right held by holders, and managed by the issuer, c) easily readable by people (like a contract on paper), d) readable by programs (parsable like a database), e) digitally signed, f) carries the keys and server information, and g) allied with a unique and secure identifier.

35 Ricardian contracts identify issuers who are the signatories, who are able to include any terms and clauses that

30 Systemics, "Ricardo: An Executive Summary".
31 Systemics, "Ricardo: An Executive Summary".
32 Ian Grigg, "The Ricardian Contract" (1996) <http://iang.org/papers/ricardian_contract.html> (accessed 13 April 2021).

they wish for the purposes of lending the document contractual validity and enforceability.[33] They are formatted as text files that can be accessed easily either in soft copy or print forms and must be "readable by people and parsable by programs".[34] In setting out his proposition for the use and usefulness of Ricardian contracts, Grigg specifically made mention of the importance of enabling valid and judicially-recognised Internet contracts:[35]

> The full story is right there in textual form, in parsable parameters, and in the signature chain. Thus, within a dispute, a hostile legal attack has less room to manoeuvre, and can only confirm the facts as laid out in the contract.
>
> Our intent is that the contract is the beginning and the end of the discussion; we call this principle *the rule of one contract*. The legal fraternity refers to 'the contract being bounded by *the four corners of the page*.' By showing how we have carefully laid out a readable document, with a verifiable digital signature, and an unforgeable identifier linking to every record, we can more readily ask the judiciary to accept that the single document which is being presented is indeed the valid contract agreed to by the parties.
>
> [emphasis in original]

36 Due to Ricardian contracts being program-readable, they possess the characteristics of smart contracts and can be deployed as such – but unlike smart contracts, they are flexible and scalable, allowing for the incorporation of new terms.[36]

B. The Creative Commons "three-layer" licence design framework

37 Creative Commons ("CC") is a non-profit organisation (of which Lawrence Lessig is a co-founder) that works to provide

33 Ian Grigg, "The Ricardian Contract" (1996) <http://iang.org/papers/ricardian_contract.html> (accessed 13 April 2021).
34 Ian Grigg, "The Ricardian Contract" (1996) <http://iang.org/papers/ricardian_contract.html> (accessed 13 April 2021).
35 Ian Grigg, "The Ricardian Contract" (1996) <http://iang.org/papers/ricardian_contract.html> (accessed 13 April 2021).
36 Blockchain Simplified, "Smart Contracts on Blockchain vs Ricardian Contracts" (17 August 2020) <https://blockchainsimplified.com/blog/smart-contracts-on-blockchain-vs-ricardian-contracts/> (accessed 13 April 2021).

solutions for the sharing of digital resources.[37] CC's public copyright licences is one such solution that it has innovated, and it takes the form of a "three-layer" design. The goal of this layered design is clarifying the scope of permitted uses for licensors and licensees and helping them understand the terms and conditions of the licence better. These layers are:

(a) the Legal Code (employing conventional legal language and formats most familiar to lawyers);

(b) the Commons Deed (also known as the human-readable version for laymen) which acts as a reference for those seeking and offering licences, explaining and summarising key terms and conditions; and

(c) the machine-readable version of the licence, in text and formatting that software systems and search engines can read and understand. This was accomplished through CC developing a standardised syntax-free protocol known as CC Rights Expression Language (ccREL)[38] that software can comprehend.[39]

38 CC offers six standard licences, each providing a different set of permitted uses and conditions. Copyright holders then choose from this range of options in deciding how they would like to make the work available to the public. What is key is that once a licence is applied to a work, it cannot be revoked and will last for as long as the work it is applied to is in copyright, and this will be the case even if the rightsholder ceases public distribution of the work.[40] Once a rightsholder has decided which licence to apply, notification of the particular licence and a link to it needs to be made on the work, or by embedding the licence using the HTML code associated with it, on a website.[41] Notably, the CC licence framework also establishes that the Legal Code will govern the legal relationship between the parties.

37 Creative Commons, "What We Do".
38 Hal Abelson *et al*, "ccREL: The Creative Commons Rights Expression Language" (3 March 2008) <https://wiki.creativecommons.org/images/d/d6/Ccrel-1.0.pdf> (accessed 13 April 2021).
39 Creative Commons, "About the Licenses".
40 Creative Commons, "About CC Licenses".
41 Creative Commons, "About CC Licenses".

39 The proposition offered by the CC model is a linked-alignment approach combining the values of standardisation, user-centricity, and explainability with a means of technical translation for clear and effective contracting outcomes. Beyond the licence-centric application of this, it worth thinking about whether there are certain types of transactions facilitated by smart contract that *lend themselves to and are made more effective with a measure of standardisation*, taking inspiration from the ethos and functionality of the CC licence model for improving comprehensibility and certainty.

VI. Conclusion

40 It has been seen that the key concern with the enforceability or validity of algorithmic contracts is their comprehensibility and certainty, which may have a bearing upon their enforceability. The preferred view is that the categorical "code is law" approach does not square with legal contractual realities, as the users of algorithmic contracts are more often than not laymen with no access to the technical rubric of the "contract" code, let alone the ability to decipher and understand it. Although in theory coders appear to assert the power to shape and delineate the digital frameworks within which people transact, the reckoning is whether there will be a real acceptance and market demand for the same, with real customers requiring bespoke algorithmic contract creation. Practitioners should thus be positioning themselves to take advantage of this potential new market.

41 There is a realisation that with the advent of smart contracts, people are thinking about and are open to different ways to transact. While it is quite certain that algorithmic contract processes will see continued use in specific contexts where the transactions involve a simple if-then logic, they may well be far more problematic (as opposed to providing a solution) when transactions scale and become more subjectively complex. If the intention is to make the use of algorithmic contracts more mainstream and accessible, employing complementary "mediators" such as Ricardian contracts and a process similar to the Creative Commons "three-layer" licence design framework to bridge humans and machines may be of some utility. This

is one context where the disciplines of law and technology can continue to interact and learn from each other, instead of taking adversarial positions which may not provide the best practical and informed outcomes.

42 Instead of treating algorithmic contracts as a confrontation to established practices and a threat to the legal profession, a more helpful approach for practitioners may be in using these mechanisms as a new lens to view and reconsider classical contract law. This could also be an opportunity to interrogate prevailing paradigms in contract drafting for currency, relevancy, and even sensibility. Finally, given the speed at which technology advances and affects how life and business are transacted, it may make sense to proactively investigate the shortcomings and limitations presented by algorithmic contracts, and to devise means and measures towards increasing utility and enforceability. This could be more interesting and profitable than waiting for the occasional pronouncements of adjudication – which although do provide necessary clarification of legal principles as applied to a specific set of facts, are *ex post facto* assessments of what went wrong, how it could have been avoided, and what remedies should be provided. While algorithmic contracts provide a glimpse as to how the present can change, any change is not motivated purely by dint of its existence – it will be down to how legal principles and Singapore's legal, political, social, and commercial institutions collectively decide to treat them. Utility and value will be moot unless businesspeople endorse them, the legislature recognises them, and the courts are able interpret them.