CLIFFORD CHANCE

LEGAL CONSIDERATIONS AROUND SMART CONTRACTS:
CONTRACTS BETWEEN COMPUTER PROGRAMS

— THOUGHT LEADERSHIP

SEPTEMBER 2021

LEGAL CONSIDERATIONS AROUND SMART CONTRACTS: CONTRACTS BETWEEN COMPUTER PROGRAMS

Difficult legal issues arise when computer software purports to enter into a contract. Electronic contracting is not a new concept. However, the rise of artificial intelligence and smart contracting means that these issues will become more important. They therefore deserve analysis.

In this note, we consider two questions around capacity to contract and reversibility of performance which arise where two computer programs contract directly with each other, in circumstances where there is no separate written natural language contract and where there is no overarching contractual framework governing the interaction. While this is not a common scenario at present, it is likely to be seen more frequently as the use of electronic contracting becomes more common. These issues arise whether or not DLT is a feature of the underlying platform or software.

Capacity to contract

The first question is whether two pieces of software can in fact enter into a valid contract in the absence of human intervention. The key issue here is that English contract theory is based on the idea of agreement between parties. It is clear that a contract can come into existence where only one party is aware of the fact – this is the *Shoe Lane* principle[1], and is widely relied upon today in situations where software makes prices and offers available on the internet such that these offers can be accepted by users and therefore create contracts However, this principle does not apply where neither party is aware of the contract. Such contracts are common today on securities and coin exchanges. However, in these cases the thing which converts the computerised interactions into contracts is a rule of the exchange which has the effect of binding both parties. Where such interactions occur outside an exchange, it is by no means clear whether a valid contract can come into existence at all.

The issue here is whether this should be clarified in English law.

Reversibility of performance

The second question concerns the consequences of a subsequent challenge to such a contract once made. Where a contract is made between two pieces of software, it is generally executed immediately and automatically – indeed, the ability to do this is one of the primary reasons for employing such software in the first place. However, such instant execution is only useful if it has some sort of settlement finality protection where the underlying transaction remains valid even if the contract pursuant to which it has been effected is challenged. These protections are generally available in established financial markets, but are less common outside them.

The issue here is whether it should be possible for parties to bring themselves within such a regime by agreement or otherwise.

---

1   *Thornton v Shoe Lane Parking Ltd [1971] 1 K.B. 532*.

1. Contract formation

There are various ways in which electronic or smart contracts might be structured. This ranges from a written natural language contract, with some degree of automation as to its execution only, to a fully automated electronic or smart contract between autonomous computer programs, which captures all terms and conditions in code. Currently, most electronic contracts will either have written terms pre-agreed, either orally or in a "wrapper" written contract, or be entered into under an overarching framework which governs the process of contracting. For example, it is increasingly common for trading programs to enter into transactions directly with each other, but this generally happens within the legal framework provided by investment exchanges or other trading venues, where the creation of enforceable contracts is a result of the application of the rules of the exchange or venue. However, in a limited number of cases, computers may deal directly with each other outside such venues, or these overarching rules may not apply.

The problem these situations give rise to at English law is that, technically, such interaction cannot create a contract. The basic position at English law (as well as many others) is entirely clear – a contract is created when an offer made by one of the parties is accepted by the other to whom the offer is addressed, and that acceptance is communicated to the offeror[2].

The conventional analysis of computerised interaction resulting in a contract is based on what might be termed the *Shoe Lane* analysis. This is the case which established the principle that an offer made to the world at large may validly be accepted, and may result in a contract, even though the person making the offer does not know that it has been accepted, or by whom[3]. It is therefore perfectly possible to analyse the activities of a trading program as the making of a series of offers on behalf of the person who has set it in motion with the intention of being legally bound.

Those offers may be accepted, and may result in valid and enforceable contracts, even though the owner of the program only finds out after the event that he or she has in fact contracted.

However, this analysis breaks down where the interaction is between two programs. If performance of a smart contract by a computer program is simply the coded execution of a series of instructions, then it is difficult to ascertain whether a computer program could distinguish between making an offer or accepting one. If all that the programs are doing is making offers to each other, then we have the problem that "cross-offers are not acceptances of each other"[4]. Put simply, an offer may be automated (for example, unilateral offers made on e-commerce websites, where it is widely acknowledged that offers may be automated and therefore made without the original "offeror" having express knowledge of an individual offer), but valid acceptance, as English Law stands, requires an act of human will. As a matter of legal theory, conduct only constitutes acceptance where that conduct is clear objective evidence of the fact that the accepter has made the necessary act of will to enter into the contract. If on the facts the accepter is ignorant of the very existence of the offer at the time it is made, this condition cannot be satisfied.

This analysis leads in a profoundly uncomfortable direction. If there is no acceptance at the time of the contract, then, in theory, there is no contract, and indeed it has been said that "there is a very strong argument for the unenforceability of algorithmic contracts"[5]. On this basis, the best analysis of what trading algorithms get up to at night when there is no one around is merely teeing up mutual offers, with acceptance occurring only when a human being comes into the office in the morning, reviews what the algorithms have done, and indicates acceptance of such offers. Such an analysis would be absolutely contrary to the beliefs and expectations of those engaged in this sort of trading, and cannot be accepted. There is

---

2   Chitty on Contracts, ed. H.G. Beale, Sweet & Maxwell 33rd ed. 2018 (Chitty) para 2.001.
3   *Thornton v Shoe Lane Parking Ltd [*1971] 1 K.B. 532.
4   *Tinn v Hoffman & Co.* (1873) 29 L.T. 271, 278, and see Chitty, 2-043.
5   Algorithmic Contracts, L Scholz, 20 Stan Tech L. Rev 128 (2017) at p.151.

surprisingly little academic analysis of this problem.

This point was elided in the most recent case involving computers dealing with each other – *Quoine Pte Ltd v B2C2[6] Ltd* – ("Quoine") – by the Singapore Court of Appeal (with the exception of Mance LJ dissenting) deciding that the contracts concerned could be said to have been "made" when the automated trading programs concerned were set in motion. This apparently nonsensical conclusion was reached because the facts of the case turned on the doctrine of mistake, and in order to apply that doctrine, it was necessary, following *The Great Peace[7]*, to establish whether the relevant mistake was operative when the contracts were entered into. By holding that the contracts were, in effect, entered into when the programs were set in motion, the court was able to conclude that the test for mistake should be applied at that point. However, that argument can be challenged. It seems illogical to argue that a party has accepted an offer before that offer is made, at a time when the party has no idea what offers may be made, or by whom. It could be argued that at best any such contract would be a contract to contract, and although such contracts may sometimes be enforceable at law, this will only happen where the terms of the future contract can be ascertained[8].

In response it may be argued that where a program purports to accept an offer made by another program, then acceptance could somehow be referred back to the act of the program owner setting the program in motion in the first place, due to the nature of code meaning that it will follow the specific instructions given. The developers of any smart contract acting in this way must have envisaged the scenarios in which they wanted the program to act, and outlined limits and boundaries as to how it should do so. For example, conditions for acceptance of an offer may be specifically built into the program using if/then programming. The expectation would be for the code to act in every situation where it can under those defined limits, and so it could be argued that the very act of starting the program and letting it run could be an external manifestation of consent. The challenge comes with certain complex 'black box' algorithms, where it is not possible to know which terms or conditions the algorithm will select in advance, and it may not always be straightforward or even possible to determine the exact basis on which the algorithm made any particular decisions. If, at the time that the programs were set in motion, the terms of any future transactions were both unknown and unknowable, this cannot be the correct analysis as the law currently stands. One solution therefore may be for legislation to be enacted that would allow the acceptance of an offer before it has been made or for mutual unilateral offers to produce a contract, in certain specific circumstances where it is clear from the actions of the parties that this is what was intended.

Another possible solution to this problem would be to accept the idea that no contract is ever made, but that because of their conduct the parties are estopped from denying that fact and must therefore proceed as if they had in fact contracted. This should be rejected – not least because such an estoppel only operates between the parties to the transaction. In a conventional trading situation, where A sells to B, who then sells it to C, if A proceeds against C directly for the recovery of the goods, C cannot rely on an estoppel available only to B. E-commerce does not operate on this basis – it operates on the basis that a transaction is a transaction, and that the rights and liabilities of the parties arise at the moment when the transaction is entered into the system. The conclusion that what appear to be transactions concluded within the system are not in fact transactions at all is radically incompatible with the intentions of the users of such platforms.

This is, of course, only one instance of a widespread problem, that being that people generally act with shocking disregard for legal technicalities in their personal and commercial transactions.

---

6   [2020] SGCA(I) 02.
7   Great Peace Shipping Ltd v Tsavliris (International) Ltd [2002] EWCA Civ 1407.
8   Chitty, 2-143.

As Chitty ruefully notes[9], this is a significant gap between the technical legal analysis of offer and acceptance, and the modes of dealing of quite significant sections of the commercial world. This has, from time to time, led the courts to take a robust line that where the communications and the conduct of the parties show that they believe that they have made a contract, they should be treated by law as having done so, regardless of the presence, or absence, of an identifiable offer or acceptance[10]. However, as Chitty goes on to say, "such an outright rejection of the traditional analysis is open to the objection that it provides too little guidance for the courts (or for the parties or for their legal advisers) in determining whether an agreement has been reached… This approach is supported by cases in which it has been held that there was no contract precisely because there was no offer and acceptance"[11].

An alternative view expressed by certain academics[12] is that algorithms (often referred to as "software agents" or "electronic agents") could be conferred with some degree of legal personality, which would mean they effectively act as constructive agents for the principal contracting parties they serve. Under this analysis, a computer program could enter into a legally binding contract, relying on the law of agency for its validity. It has been argued that a software agent performing a smart contract for its principal with no human intervention should be akin to a human agent performing the same task, and thus should be treated equally to the human agent under the law. This would provide a basis for enforceability of smart contracts that solves the question of consent and contract validity by integrating with existing legal theories about contractual freedom and conclusion of contracts However, this theory does pose difficulties in justifying the attribution of legal personality to an automated entity based on at least three considerations: moral authority, social capacity and legal

convenience (which are effectively impossible with computers). Ultimately, the law of contract would also need to account for additional concerns as to the assignment of liability, including who would take responsibility for algorithmic outcomes accomplished in a way that algorithm creators could not have predicted. It would also need to appropriately address the 'mistake' concern, i.e. that where an agent makes a mistake in deciding whether to enter into a contract for his or her principal, it is the state of mind of the agent which must be tested, and this is not possible with an electronic agent.

It seems to us that what is needed here is a confirmation of the fact that, at English law, an interaction between two computer programs occurring without direct human intervention, which is intended to result in a legally enforceable arrangement to transfer rights or property, may be enforceable as a contract in the same way that it would have been had it been entered into as a result of human interaction.

This is not a case of redesigning the law of contracts. Smart contracts aim to translate a process currently undertaken by humans into something that can be performed mostly by machines. This reflects a change in method, not a change in the nature of contracting.

This, of course, gives rise to exactly the problem that Lord Mance raised in *Quoine* – what should the position be where the computers act in a way in which it is clear that no human being would have acted and/or is contrary to the intentions of those who set the programs in motion? Should the actions of a computer be subject to some sort of "reasonableness override", such that where a computer does something which is so plainly erroneous that a human would immediately recognise it as such, there should be a remedy based on that fact alone? We are inclined to agree with Lord Mance that a development of the

9   Ibid, para 2-118.
10 See e.g. *Gibson v Manchester City Council* [1978] 1 W.L.R. 520 at 523, *Finmoon Ltd v Baltic Reefer Management Ltd* [2012] EWHC 920 (Comm.), [2012] 2 Lloyds L.R. 388 at [22].
11 Ibid, para 2-119.
12 See for example Algorithmic Contracts, L Scholz, 20 Stan Tech L. Rev 128 (2017), Samir Chopra & Laurence White, Artificial Agents and the Contracting Problem: A Solution Via an Agency Analysis, 2009 University of Illinois Journal of Law Technology & Policy, 363.

Coys of Kensington[13] principle (that a person who receives a benefit by mistake should reimburse the value of that benefit to the other party if it is readily returnable without substantial difficulty or detriment) would satisfactorily address this issue. However, these positions are – properly – left to the courts. What cannot be left to the courts is the fundamental uncertainty which arises from the potential unenforceability of computer contracts under English law. This should be clearly addressed by policymakers.

2. Reversibility of performance

The second key issue for automated contracting between two computer programs is the question of the consequences of a subsequent valid challenge to such a contract once made.

Where a contract is made between two pieces of software, it is generally executed immediately and automatically. However, such instant execution is only useful if it has some sort of settlement finality protection where the underlying transaction remains valid even if the contract pursuant to which it has been effected is challenged. This idea is sometimes summarised as the notion that "the code is law" – meaning that what is done through code should be irreversible. These protections are currently not widely available outside specific financial markets contexts, although DLT-based smart contracts may bring certain additional practical difficulties for unwinding a contract.

The utility of a legal provision of this kind can be illustrated with a simple example. Imagine two businesses, both of which operate entirely automated online businesses. A thief hacks into the software of both companies and fraudulently induces them to enter into a contract under which goods are sold by one to the other. The goods are delivered and the recipient, believing them to be validly acquired, sells them on to third parties, again through automated contracts. The contract is subsequently found to be void. The rule of *nemo dat* means that those third-party buyers do not have legal title to the goods that they have bought, even though the contracts by which they purchased those goods

are valid, and they have no way of knowing that the contract by which the seller purchased the goods was challengeable. The position would be somewhat different if the initial transfer had been of merely equitable title, since in equity a purchaser for value without notice of the legal estate will prevail. This, however, raises another set of questions. If the ultimate purchasers purchased through automatic contracts, could they show that they had acted "in good faith"? This in turn takes us to the question of whether a clear and demonstrable lack of bad faith is sufficient to make a purchaser a "purchaser in good faith"?

In *Quoine*, it was established that if a contract made between dealing programs was void for mistake, the transfers of property made pursuant to that contract would be reversed. The Court of Appeal was faced with the difficult choice between upholding the validity of a contract which was clearly mistaken, or retrospectively reversing executed contracts.

The issues that arise when a contract is retrospectively voided are an extreme example of this problem. However, they also arise in a number of other, less extreme, cases. For example, what is the position if a piece of software transfers property in circumstances which are clearly incompatible with the terms of the contract pursuant to which it does so? In particular, if a contract is for the sale of A, but the software transfers B, should that transfer be extinguished?

The essence of the cyber-purist view is that it should not. The idea that "the code is law", reduced to legal concepts, is precisely that what is done automatically and without human intervention by a program should not be capable of being reversed or varied because of human intervention. The use of "immutable" DLT technology underlying a smart contract may lend weight to this argument, although there are ways to address this. However, with a few exceptions, this is not currently possible under English law.

---

13 *McDonald v Coys of Kensington (Sales) Ltd* [2004] EWCA Civ 47.

It is instructive to begin by considering the exceptions. The simplest is the case of real estate. A transfer of real estate is done using two documents: a contract and a conveyance. Because the conveyance is executed as a deed, it has a legal effect which is independent of the contract by reference to which it is created. Thus, the avoidance of a contract for sale of real estate does not itself void a conveyance of that real estate. A similar outcome can be produced in several other situations. For example, where two parties contract that a bill of exchange shall be created, and such a bill is created, the bill itself is valid regardless of any subsequent challenge to the relevant contract. Further, as mentioned above, the settlement finality regime as it applies to financial contracts has the effect of preserving certain transfers of financial assets in the face of a challenge to the underlying contract.

The problem is that, outside of these identifiable exceptions, English law is incapable of separating contract and conveyance. This is in contrast to other legal systems – in German law, for example, the principles of separation and abstraction have the effect that a rescission of a contract does not itself affect the validity of a transfer made under it. At English law, if a contract is voided, the transfer made under it is equally voided and, in a contract which is not subject to special rules, this will be true whether or not the transfer of property is separately documented. For example, where a contract for the sale of a ship is documented using a bill of sale to transfer ownership of the ship, the bill of sale does not have independent legal effect, and derives its legal consequences entirely from the contract pursuant to which it is made.

The basic legal proposition is lucidly set out in the "Legal Statement on cryptoassets and smart contracts" of the UK Jurisdiction Taskforce[14]. This correctly states two important preliminary points[15]. The first of these is that the use of software cannot produce legal obligations between persons other than through the operation of the law of contract.

The argument that the use of software giving rise to automatic outcomes means that there is no need for a party's obligations under a smart contract to be legally enforceable is nonsense – if only because property transferred otherwise than under a legal contract can be recovered back. The second is that, once it is understood that the relationship between the parties to a smart contract must, in law, be a contractual relationship, there is then no good reason for treating smart contracts as being different in principle from conventional contracts.

This is the crux of the issue between the conventional legal analysis and the view of the cyber-purists. In this context, when the cyber-purists say that they want a world in which "the code is law", what they mean is that they want a transaction effected through a software operation to be recognised as valid regardless of whether the contract pursuant to which the transaction is effected is challengeable or challenged.

It is insufficient to reply to this by saying simply that this is not the law as it stands today. The question is whether the law should accommodate this. Settlement finality has been accorded to other types of transaction in other contexts, and there is nothing in the idea which is inherently repugnant to English law.

One way of approaching this might be to ask whether what we need is simply an extension of the existing law of bills and notes to payment instructions embedded in software programs. However, this would underestimate the scope of the problem. The law of bills and notes is, by definition, restricted to payments. The role of software is not so limited. For example, imagine a situation in which two software programs have purported to enter into a contract with each other. One of them sends a software instruction to make a payment, the other sends a software instruction to deliver whatever it is that has been sold. It would be absurd to conclude that the outcome of the operation of one piece of software should be challengeable, whilst the other should be protected. If protection of this kind is to be extended to certain actions which

14 Published in November 2019 and available at **https://technation.io/lawtech-uk-resources/**
15 See paras 136 – 148.

are the outcome of the operation of a type of software, it would be absurd to suggest that that same protection should not be afforded to other such actions, without at least producing a coherent theoretical basis for that distinction. The fact that one relates to money, whilst the other relates to things, is not an adequate theoretical basis in this regard.

As between the parties, there is a simple solution – the intention of the parties. However, the question with which we are concerned here is the position as regards third parties. Assume that three trading programs are interacting with each other on an exchange. Trading program A sells Z to trading program B, which in turn sells it to trading program C. Can program C safely sell Z to program D? That depends on whether the transaction between A and B is valid. Can C determine whether that transaction is valid? With the law as it stands today, the answer is no.

Viewed from this perspective, it seems clear that the promotion of market efficiency weighs heavily in favour of the creation of some sort of protection for transactions where the universal expectation is that transfers should not be reversible. However, such provisions have traditionally been confined within narrow bounds, on the basis that it is important that it should be entirely clear both to the contracting parties and to third parties what the rules are which apply to any particular transaction. With bills of exchange, for example, very exact rules as to their content and form were imposed precisely in order to try and create a clear division between instruments which were enforceable as bills and instruments which were not.

It is our view that a degree of protection should be afforded by law to payments and transfers made through the operation of software where the circumstances of the contract are such that it is clear both to the parties and to the world at large that that is the common intention. Strict "code-is-law" supporters would argue that the mere fact that a contract has been set up to be fulfilled by software is itself sufficient evidence of this. We do not agree. We think that, in the same way that settlement finality protections are only available under the existing legislation for certain types of transaction identifiable as such, a similar regime should be put in place such that it should be clear when a particular arrangement is intended to be final in this way.

Finally, we note that the availability of settlement finality protections does not necessarily significantly reduce the rights of the parties. For example, in securities transactions, a party who has lost out by reason of a mistaken transfer may sue for damages for breach of contract, or for compensation in unjust enrichment, or potentially under a number of other causes of action. The introduction of a degree of protection for transactions of this kind need not necessarily result in any diminution of the rights of any person.

CONTACTS

Kate Gibbons
Partner
London
T: +44 207006 2544
E: kate.gibbons@
     cliffordchance.com

Simon Gleeson
Partner
London
T: +44 207006 4979
E: simon.gleeson@
     cliffordchance.com

Kate Scott
Partner
London
T: +44 207006 4442
E: kate.scott@
     cliffordchance.com

CLIFFORD
CHANCE

CLIFFORD
CHANCE