









**Law  
Commission**  
Reforming the law

# SMART CONTRACTS

---

Summary of call for evidence

<p><b>What is this project about?</b></p> 	<p>We are undertaking a scoping study on smart contracts to provide an accessible account of the current law and set out how it will, or may, apply to smart contracts. Our project is intended to inform public debate and seek a consensus about areas which may need further consideration or possible reform in future. The scoping study itself will not offer formal recommendations for reform.</p>
<p><b>What is a smart contract?</b></p> 	<p>For the purposes of the call for evidence we define a smart contract as a legally binding contract in which some or all of the contractual obligations are recorded in or performed automatically by a computer program deployed on a distributed ledger.</p>
<p><b>Why are we launching a call for evidence?</b></p> 	<p>This document is a summary of a longer call for evidence available at <a href="https://www.lawcom.gov.uk/project/smart-contracts/">https://www.lawcom.gov.uk/project/smart-contracts/</a>. The call for evidence is the first step in our smart contracts scoping study. We are seeking views about, and evidence of, the ways in which smart contracts are being used, and the extent to which the existing law can accommodate them.</p>
<p><b>Who do we want to hear from?</b></p> 	<p>We are keen to receive comments from as many stakeholders as possible, including:</p> <ul style="list-style-type: none"> <li>• people using or considering using smart contracts;</li> <li>• coders and other technologists developing smart contracts; and</li> <li>• lawyers and academics advising on smart contracts or with an insight into how the current law could apply to smart contracts.</li> </ul>
<p><b>What is the deadline?</b></p> 	<p>The call for evidence closes on 31 March 2021. Responses are particularly welcomed through our online form at <a href="https://consult.justice.gov.uk/law-commission/smart-contracts">https://consult.justice.gov.uk/law-commission/smart-contracts</a>.</p>
<p><b>What happens next?</b></p> 	<p>After reviewing all responses, we will produce a scoping study explaining how we consider the current law applies to smart contracts, and identifying any areas in which further work, or possibly law reform, may be required. We will also publish a summary of the responses received to the call for evidence.</p>



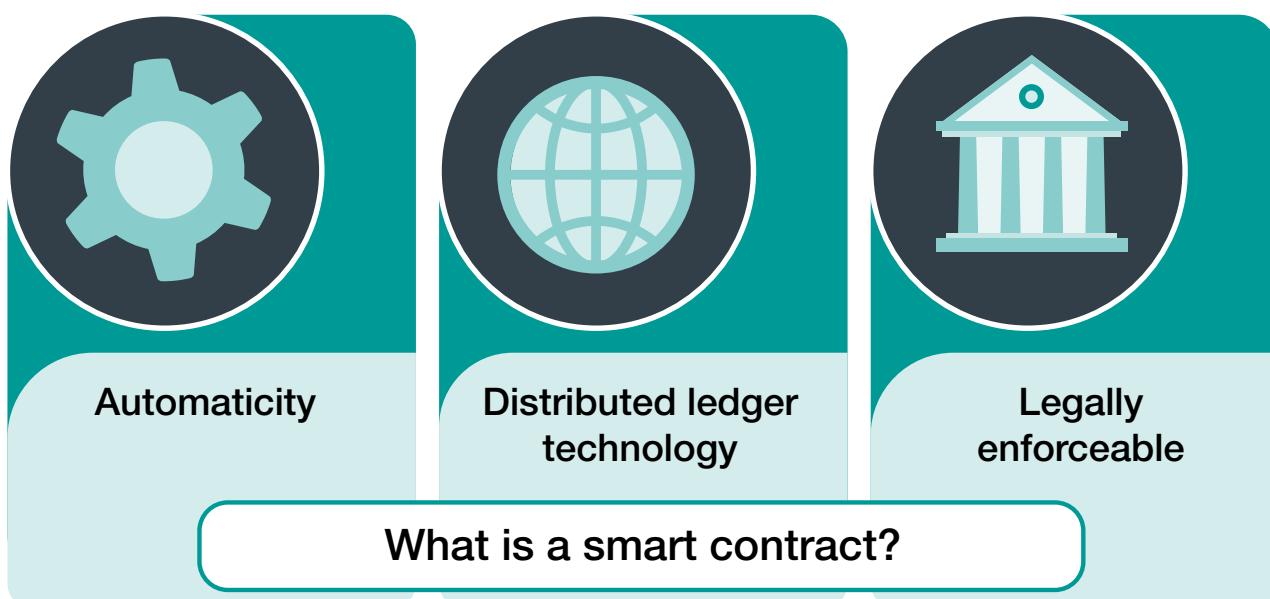
# WHAT IS A SMART CONTRACT?

Emerging technologies such as distributed ledgers are being promoted as a way to create “smart contracts”: computer programs which run automatically, in whole or in part, without the need for human intervention. Smart contracts can perform transactions on decentralised cryptocurrency exchanges, facilitate games and the exchange of collectibles between participants on a distributed ledger, and run online gambling programs. They can also be used to record and perform the obligations of a legally binding contract. It is this second category of smart contracts (sometimes referred to as smart legal contracts) which is relevant to our work. When we talk about smart contracts in this summary we are therefore talking about legally binding contracts in which some or all of the terms are recorded in or performed by a computer program deployed on a distributed ledger. These smart contracts are expected to increase efficiency, trust and certainty in business, and reduce the need for contracting parties to trust each other; the trust resides instead in the code.

## Automaticity

A distinctive feature of smart contracts compared to traditional contracts is that some or all of the contractual obligations are expressed in computer code and performed automatically by computer programs, without the need for human intervention. Contractual obligations which follow a conditional logic (if X, then Y) are good candidates for being coded. For example, a contract for sale of an asset could be coded so that title to the asset transfers automatically on receipt of a certain amount of money into a particular account.

The use of computer programs to automate the performance of contractual obligations is not a novel practice. Automated bank payments (such as direct debits and standing orders) and online shopping (including the purchase and download of digital content and the purchase and delivery of physical goods) all involve computer programs automating some element of contractual performance.



---

This automation, however, does not make the underlying contract a “smart contract” for the purposes of our call for evidence. This is because the automatic processes are under the control of at least one of the parties to the contract and can, therefore, be stopped by that party. In some cases, the process of performing such a contract will also involve human intervention at a variety of key stages. In the case of standard online shopping, for example, although order acceptance could be automated based on transfer of funds, the picking and delivery of the goods still requires human intervention.

## **Distributed ledger technology (DLT)**

Smart contracts can be deployed on a distributed ledger so that contractual obligations expressed in computer code are performed by the computers on the network. Once the smart contract is deployed on the distributed ledger, the code cannot be altered by the parties and it will be performed automatically as soon as the conditions specified in the code are met. Performance of a smart contract is therefore ‘guaranteed’ in the sense that no party can intervene to prevent the performance of the computer program once it is triggered. As the ledger is immutable, the parties to the contract have the security that no one, including the parties to the contract, can tamper with the computer program or the data entries recorded on the distributed ledger to stop it from performing or to reverse transactions. These features also mean that parties can transact directly with one another without having to rely on traditional intermediaries such as banks and clearing houses.

---

Smart contracts can be performed automatically by computer programs without the use of DLT. For example, one party might deploy a program on its own computer system to automate the performance of its obligations under a contract or parties may run separate versions of a program on their own computer systems. However, given the unique benefits of DLT, much attention continues to focus on DLT-based smart contracts. They also give rise to potentially novel legal issues. For these reasons, we focus in our call for evidence on smart contracts which use DLT. In the call for evidence, we ask consultees if they agree with this approach.

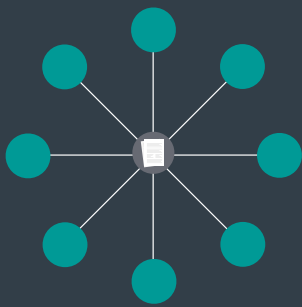
## **Legally enforceable**

The call for evidence focuses on smart contracts that record and perform the obligations of a legally binding contract. For a smart contract to be legally binding (and therefore legally enforceable) under the law of England and Wales it must satisfy various requirements for formation, discussed below.

## Distributed ledger technology (DLT)

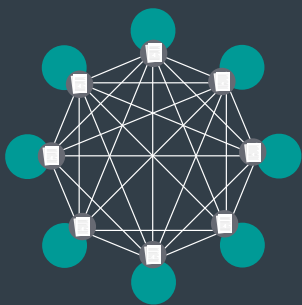
DLT is a method of recording and sharing data across a network. A DLT system comprises a digital database (a 'ledger') which is shared (that is, 'distributed') among a network of computers (known as 'nodes'). The ledger contains a record of data, such as a history of transactions, and each node holds a copy of the ledger on its system. When data is added to the ledger, every node's copy of the ledger is updated instantaneously. Therefore, at any point in time, every node holds an identical and up to date copy of the ledger. Examples of DLT systems include the Bitcoin blockchain and Ethereum.

The distinguishing feature of DLT compared to other shared databases is that the ledger is not maintained by a central administrator. Instead, the ledger is maintained collectively by the nodes on the network. No single node has the power unilaterally to add data to the ledger; a proposed change must be approved by all the nodes via a process called the 'consensus mechanism'. The consensus mechanism is typically designed so that, once data is added to the ledger, the data is very difficult to amend.



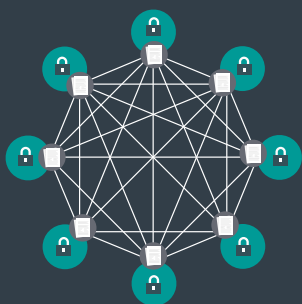
### Centralised ledger

A ledger is maintained by a trusted central administrator. Each network participant must reconcile their local databases with the ledger maintained by the central administrator.



### Distributed ledger (permissionless)

The ledger is maintained collectively by network participants. No single party is in control of the ledger. Data is only added to the ledger when network participants reach consensus on the validity of a proposed data entry. When data is added to the ledger, every participant's copy of the ledger is updated instantaneously.



### Distributed ledger (permissioned)

Parties must obtain permission from a central entity in order to join the network and propose additions to the ledger. Typically, parties must prove their identity before joining the network. Participation in the consensus mechanism may be restricted to a subset of participants, and the central administrator may have the power to make changes to the ledger. When data is added to the ledger, each participant's copy of the ledger is updated instantaneously.

---

## Three forms of smart contract

In practice, there appear to be three different forms that smart contracts can take. The form a smart contract takes can have an impact upon when and how it is formed, how it is interpreted and the remedies available to the parties if things go wrong.

### Form 1: Natural language contract with automated performance

This is a natural language contract in which the performance of some or all of the contractual obligations is automated by a piece of code deployed on a distributed ledger. The code itself does not record any contractual obligations but is merely a tool employed by the parties to perform those obligations.

We do not think that this form of smart contract raises any novel legal issues in the context of formation or interpretation. However, there may be challenges if a problem arises during the lifecycle of the contract and one or both parties seek to obtain a remedy as a result. Since some or all of the obligations will be performed automatically by code on a distributed ledger, the lack of a mechanism to stop performance of the code could have implications for the type of remedy sought. Would it be possible, for instance, to terminate the smart contract and bring its performance to an end?

---

### Form 2: Hybrid contract

A hybrid smart contract is a combination of code and natural language. At one end of the spectrum, the terms of a hybrid contract could be primarily written in code with natural language terms employed to add certain provisions (for example, governing law and jurisdiction clauses and dispute resolution mechanisms). At the other end of the spectrum, the terms of a hybrid contract could be primarily written in natural language and include, by reference, just one or two provisions written in code.

In the call for evidence we ask consultees for further information about how hybrid contracts are structured and about the relationship between the code and the natural language elements.

### Form 3: Solely code contract

This is a contract that consists only of code. No natural language version of the agreement exists. The parties agree the code and it executes on a distributed ledger. Alternatively, one party does all the coding and makes the program available on a distributed ledger where anyone with access to that ledger can run it. Our initial view is that this type of contract is likely to present the most challenges from a contract law perspective, in terms of determining whether and when a legal contract is formed, and how that contract can be interpreted.

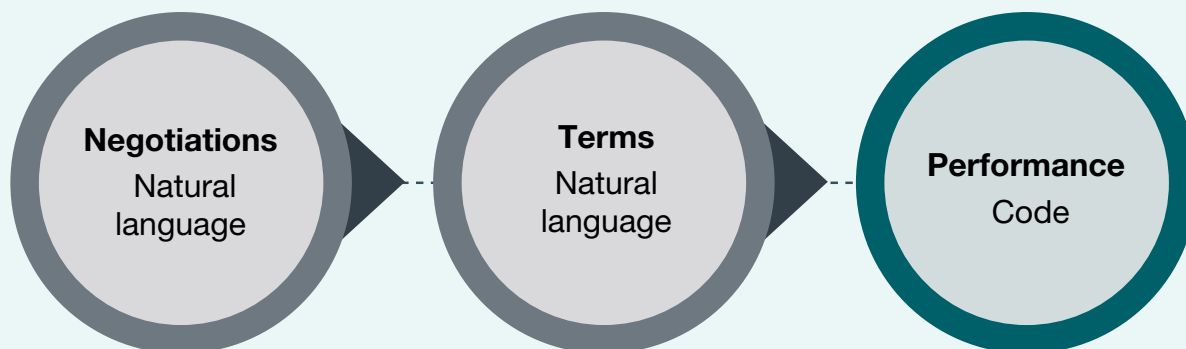
---

**We ask questions about when and how each of these three forms of smart contract are being used or developed**

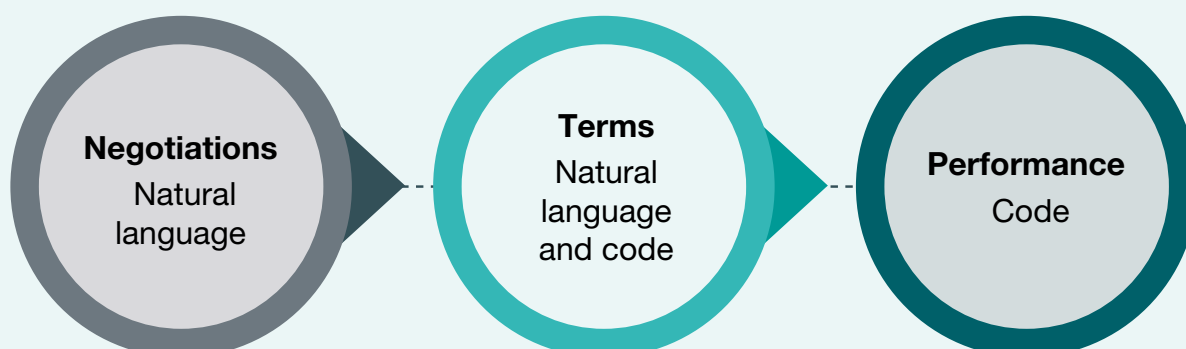
---

## The three forms of smart contract compared

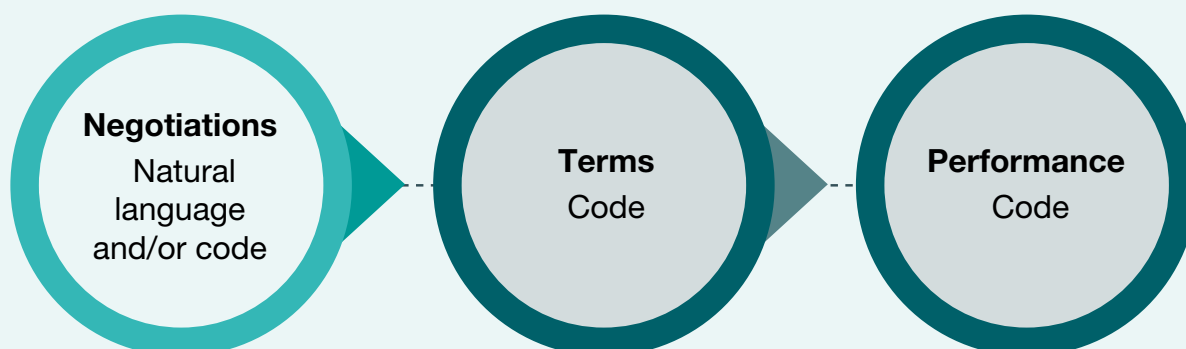
### Form 1: Natural language contract with automated performance



### Form 2: Hybrid smart contract



### Form 3: Solely code contract



---

## Use cases

Smart contracts are increasingly being considered as a means of automating specific processes within conventional legally binding contracts, including:

- Payment of insurance claims.
- Payment of royalties in respect of intellectual property rights.
- Transferring rights in property.
- Managing supply chains.
- Underpinning emerging commercial and peer to peer arrangements such as the transfer of cryptoassets and tokens representing real world assets.
- Facilitating “DeFi” or decentralised finance, which aims to disrupt conventional banking and securitisation arrangements.

Some of these use cases are already in operation or in the development (“proof of concept”) stage, while others are currently theoretical only.

---

## Potential benefits and cost savings

Our preliminary research and discussions with stakeholders have identified the following potential non-monetary benefits and cost savings associated with the use of smart contracts which utilise DLT:

- *Increased efficiency and lower transaction costs:* Because every participant has an up to date copy of the ledger and therefore real time details of performance of the smart contract, and contracts are performed without the need for human intervention.
- *Lower enforcement costs:* Because the code executes automatically, the need to take enforcement action for failure to perform obligations under a smart contract may be less common as compared with a traditional contract.
- *Reduced risk of fraud:* The consensus mechanism and immutability of a distributed ledger mean that contracting parties can trust in its veracity and transact with one another in confidence, even where there is no central administrator.

---

**We ask questions about existing and planned use cases for smart contracts and for consultees’ views about the potential benefits and costs of smart contracts compared with traditional natural language contracts**

---



---

# FORMATION OF SMART CONTRACTS

The term “smart contract” is often used to describe a computer program which is not a contract in any legal sense. In the call for evidence, we are concerned only with computer programs which record or perform the obligations of a legally binding contract. In Chapter 3 of the call for evidence we consider the requirements for the formation of a legally binding contract: agreement, consideration, certainty and completeness, intention to create legal relations, and compliance with formalities (including in relation to deeds). We are interested to understand whether consultees foresee any potential uncertainties in applying these requirements to smart contracts.

## Agreement

The first requirement for the formation of a legally binding contract is an agreement, comprising an offer to be bound on specified terms, and an acceptance of those terms. We expect that, in most cases, entry into a smart contract will typically be preceded by natural language negotiations or other communications between the parties: email correspondence, instructions to coders, exchanges of draft agreements, or oral conversations. In these cases, the question for the court would be the same as in a traditional contracting scenario: did the words and conduct of the parties reveal an offer and acceptance and therefore an agreement?

Identifying an agreement may be more difficult where there is limited or no natural language negotiation or communication between the parties and their conduct may be the only evidence of an agreement between them. For example, one party might deploy a piece of code on a distributed

ledger and another party might interact with that piece of code, causing the code to execute a transaction, or the parties might each deploy programs on a distributed ledger which subsequently interact and execute transactions. In these cases, the court may have to consider novel questions about the circumstances in which deploying or interacting with code on a distributed ledger can give rise to an agreement.

The pseudonymous nature of some DLT systems may make it comparatively more common for parties to enter into smart contracts without knowing the real identity of their counterparty. There is no requirement under the law of England and Wales for contracting parties to know each other’s real identities so the anonymity or pseudonymity of one or both of the parties will not preclude them from reaching an agreement. It may, however, pose challenges later on if there is a dispute and the jurisdiction of the smart contract is in question, or if one of the parties seeks to obtain a remedy against the other.

## Consideration

Under the law of England and Wales, contractual promises cannot as a general rule be made ‘gratuitously’ – that is, for nothing in return. The exception is a promise made by deed, which does not require consideration to be legally binding. Our initial view is that smart contracts are unlikely to create particular challenges in identifying relevant consideration. In most cases, the consideration (such as payment in money or tokens) could be identified from the terms of any natural language agreement, or the operation of the code itself.

---

## Certainty and completeness

An agreement must be “certain and complete” to constitute a contract. Where a smart contract consists of a natural language agreement that is performed automatically by a piece of code, this requirement could be assessed in the ordinary way by examining the provisions of the natural language agreement.

In a hybrid agreement, a potential source of uncertainty is that the natural language and coded terms may conflict with one another. However, conflicts between the code and natural language could be resolved by applying the principles of interpretation: a court will only hold an agreement to be uncertain as a last resort.

In a solely code agreement, the behaviour of the code may itself provide a strong indication that the agreement is certain and complete, because only complete and syntactically correct code will be executed by a computer.

## Intention to create legal relations

English law presumes that parties intend to create legal relations when they make an express agreement in a commercial context. An express agreement is an agreement expressed in words, rather than inferred from the parties’ conduct.

Where commercial parties conclude a natural language agreement which is then performed automatically by code, the presumption is likely to apply so that there will be little difficulty in proving that the parties intended to create legal relations. However, difficulties may arise if the parties’ agreement is made as a result of interaction on a distributed ledger where the agreement is inferred from the parties’ conduct rather

than as a result of an express agreement. Here, the presumption may not apply so it might be necessary to prove that the parties actually intended to create legal relations. An intention to create legal relations may be difficult to prove, for example, if the shared understanding of participants on a DLT system is that transactions on the ledger are not intended to be legally binding. The development of DLT was in part driven by a desire to exclude institutional influence, including legal systems

Some parties may expressly deny an intention to create legal relations, for example, by including a provision to this effect in the natural language component of their agreement. Although they want the agreement to be performed, they are content to rely on the operation of the code and the relevant DLT protocol to do this and don’t wish to have recourse to the courts if something goes wrong: “the code is law”. The courts have given effect to clauses which expressly deny an intention to create legal relations.

## Formality requirements and deeds

The general rule is that contracts need not be made in any particular form. However, some statutes require certain contracts to be made “in writing” and “signed”. Deeds are subject to additional formality requirements.

---

## Can a smart contract satisfy an “in writing” requirement?

Some contracts are required by statute to be in writing. The Interpretation Act 1978 defines writing as including all modes of “representing and reproducing words in a visible form”. If the terms of a smart contract are set out in a natural language document, then the smart contract would satisfy an “in writing” requirement. The more difficult question is whether contractual terms recorded in code could satisfy this requirement. This depends on whether the code is in a form which can be read by a person.

We understand that drafting code involves two steps. First, the code is drafted in a high-level programming language, known as source code. Source code uses a combination of words and symbols and can be read by an expert coder. Second, the source code is compiled into machine readable ‘object code’, which is written in binary form and is impossible even for an expert coder to read.

If the terms of a smart contract reside in the source code, then it is arguable that those terms can be read and therefore that the smart contract would satisfy an “in writing” requirement. However, if the terms of the smart contract reside in the object code, then it is more difficult to argue that the contract is “in writing”, because object code cannot be read.

---

## Can a smart contract be “signed”?

In most cases, contracts governed by the law of England and Wales do not require a signature. Where the law does require a contract to be signed but is silent on the form that the signature must take, the common law generally adopts a pragmatic approach and does not prescribe any particular form or type of signature.

Where the terms of a smart contract are recorded in a natural language document, the smart contract could be signed in the ordinary way. Where a smart contract consists solely of code, the parties could sign the contract electronically, for example by using a digital signature to authenticate a piece of code deployed on a DLT system. A digital signature is generally capable of satisfying a statutory requirement for a signature. This is the case save where the contrary is provided for in relevant legislation or contractual arrangements, or where case law specific to the document in question leads to a contrary conclusion.

---

## Can a smart contract be used to create a legally binding deed?

Where a deed is executed by an individual, it must be signed in the presence of a witness who attests to the signature. If the terms of the deed are recorded in a natural language agreement, performance of which is automated by code, the deed could be signed, witnessed and attested in the ordinary way.

However, where the terms of the deed are recorded wholly or partly in code, the question arises as to how a signature authenticating the coded terms of a deed could be witnessed and attested. The current law does not support witnessing other than by the witness being physically present when a deed is signed. If a witness is physically present with an individual when they digitally sign the code, then the witnessing requirement could potentially

be satisfied. In *R (Mercury Tax Group Ltd) v Her Majesty's Commissioners of Revenue and Customs*, Mr Justice Underhill (as he then was) said that in the case of a deed “the signature and attestation must form part of the same physical document”. This requirement may be challenging if existing technology does not allow a witness to record on the smart contract that they have observed the execution of that contract.

---

**We ask questions to identify whether consultees are aware of, or foresee, any difficulties in applying the principles of contract formation (including in relation to deeds) to smart contracts**

---





---

# INTERPRETATION OF SMART CONTRACTS

A court may be asked to interpret a contract where the parties disagree as to the meaning of the terms of that contract. Such a dispute would usually arise where one party has done something or has failed to do something that another party considers a breach of the contract. Chapter 4 of the call for evidence considers how the principles of contractual interpretation could be applied if a court were asked to interpret a smart contract. We highlight particular areas where challenges may arise but we are interested to hear from consultees as to whether they have any particular concerns in relation to these areas or to any other aspect of interpretation.

## The principles of contractual interpretation

Contractual interpretation is the process by which a court determines the meaning of the language used by the parties in the express terms of a written agreement. The courts of England and Wales take an objective approach to contractual interpretation. The court does not ask what the parties themselves meant by the language they used. Rather, the court asks what the language would have meant to a reasonable person, equipped with all the background knowledge available to the parties at the time the contract was made.

The language of the contract is therefore given primacy in the interpretation of the contract, with other information (such as business common sense and context) only serving to assist with the objective interpretation of the language used. Evidence of the subjective intentions of the parties (including evidence of their prior negotiations) as to the meaning of the words used is not admissible.

## Challenges for applying the principles of interpretation to smart contracts

Novel interpretation issues are unlikely to arise where the smart contract takes the form of a natural language contract with automatic performance by code. The natural language contract will be treated as containing the terms agreed by the parties and it will be those terms that the court will be called upon to interpret. The court will only look to the code if it is asked to consider whether the code correctly implements the terms of the natural language agreement.

However, where the terms of a smart contract are recorded partly or solely in code, this potentially poses difficulties in the interpretation exercise. The principles of interpretation have developed in response to courts seeking to interpret natural language terms. Can the existing principles be used to interpret terms recorded in computer code? While the court has the tools to allow it to interpret a smart contract, do those tools need to be employed slightly differently in a smart contract context (as distinct from a natural language contract context)?

## Should a court base its interpretation on what the contract means to a computer?

As computer code is designed for the special purpose of instructing computers, one potential approach to interpreting coded terms is to ask what the computer would do upon receiving the coded instructions. In almost all cases, this question will permit only one answer as, from a computer's perspective, a coded term has exactly one 'meaning', or none at all.

---

However, this approach could be said to pay insufficient regard to the intentions of the parties. This is because, although code may be unambiguous from the computer's perspective, there may be situations in which the code behaves in ways not expected by the parties to the smart contract. This could occur where parties instruct a third party to produce the code, but the resulting code is not as the parties intended because the instructions were insufficient, misunderstood, or not properly transposed. The outcome of the running of the code could also be affected by external influences and therefore differ from what both the parties and the coder expected. For example, the code may execute in an unforeseen way due to a malfunctioning oracle, a system failure on the platform on which the code is deployed, or interference by malware (such as a virus). Finally, limiting interpretation of code simply to observing the performance of that code will not give the court the opportunity to consider the context in which a coder used it.

### **Should a court base its interpretation on what the contract means to a reasonable person with knowledge of code?**

A potential alternative approach to interpreting coded terms is to ask what a reasonable person with knowledge of code would understand them to mean. An expert coder would be able to assist the court by translating the code, in much the same way as an expert may translate another contract written in a language not familiar to the court. However, for the purposes of assisting the court in interpreting the contract, it is unlikely to be sufficient for an expert coder merely to translate the code into natural language. This is because the court is unlikely to be familiar with the way instructions in code are interpreted by a

computer, or the way a coder might arrange instructions in order to elicit a particular outcome from the running of a code.

Instead, as well as translating the code, a coder would need to give the court their reasoned opinion as to what the code appears to instruct the computer to do. This approach arguably shifts the role of interpretation further from the judge and towards experts. However, it does have the benefit of giving an insight into what the code appeared to instruct the computer to do, regardless of what the performance of the code actually achieved. This is arguably more relevant than the actual performance of the code to what the parties intended the code to do.

### **Inconsistencies in a hybrid smart contract**

The general approach to potential inconsistency between different terms of the same contract is that the court should try to reconcile the two, having regard to the contract as a whole. This approach applies whether the clauses of a contract are found in a single document or in two or more separate documents which together make up one contract. It is particularly relevant in the context of hybrid smart contracts, which include a natural language element and a code element.

Parties to a hybrid contract may employ drafting techniques to try to clarify their intentions and deal with any inconsistencies which may arise. A hybrid contract could include a term setting out an order of precedence to deal with situations where coded terms and natural language terms conflict. The natural language element may also set out how the parties intend the code to operate. This could be in the form of a detailed term or simply a broad statement of intention.

---

We are interested to understand whether parties to hybrid smart contracts are employing drafting techniques to deal with any potential inconsistencies between natural language and coded terms.

### **Should a court take into account prior negotiations?**

Coding remains a specialist skill and it is unlikely, at least in the short to medium term, that many commercial parties will be able to read, write or understand code. Parties who wish to enter a smart contract will have to engage a coder to write the coded elements of the contract. Where parties cannot understand code, it may therefore be difficult to view the code as an expression of their intention, although we could say that the coder's intentions are a proxy for the parties' intentions.

There are likely to be natural language interactions which precede the smart contract, even where the contract itself is expressed solely in code. This could

---

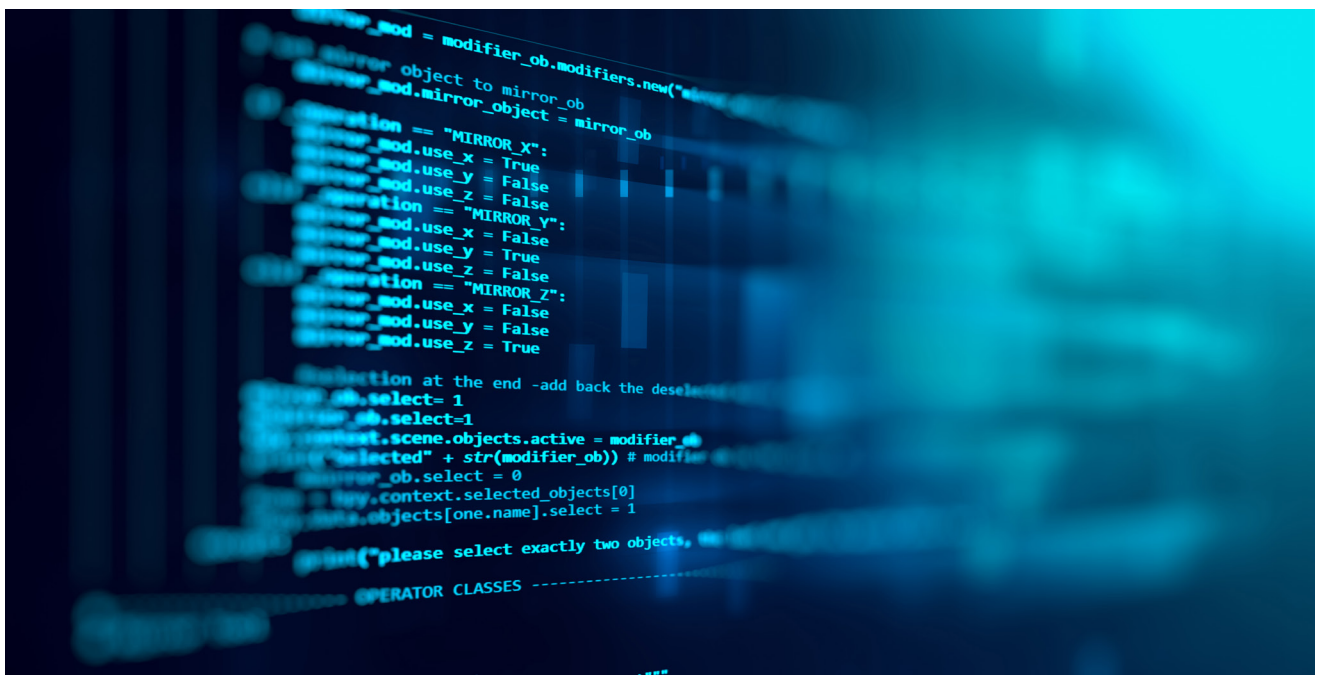
include the business process document provided by the parties to a coder, emails or conversations between the parties as to their intentions, or a description on a DLT platform of how the smart contract is to operate.

However, evidence of the parties' pre-contractual interaction is generally inadmissible to prove the meaning of a contract (the so-called 'exclusionary rule'). There is an argument that, when seeking to interpret the coded terms of a smart contract, the court should be able to examine the parties' interactions prior to entry into the smart contract. This could provide the court with a better insight into what the parties intended the code to do, especially where parties are not code literate and may have completed all negotiations in natural language.

---

### **We ask questions to identify any particular concerns in relation to interpretation of smart contracts**

---



---

# REMEDIES AND SMART CONTRACTS

Various problems can arise in the lifecycle of a contract, and in response to these problems the law provides a range of remedies. In Chapter 5 of the call for evidence, we discuss the problems that might arise in the context of smart contracts, the remedies that the parties might seek, and how a court might award those remedies in practice. We ask consultees if they are aware of, or foresee, any difficulties in awarding these remedies in the smart contract context.

## Rectification

A court can rectify a written contract where the terms of the contract do not accurately record the parties' agreement. In a smart contract context, the parties might decide to record some or all of the terms of their contract in code. If the coded terms do not reflect what the parties in fact agreed, would the remedy of rectification be available to amend them?

If code can be considered a form of "writing", then in principle the remedy of rectification might be available to amend the coded terms of a smart contract. The remedy may be particularly relevant where the parties enlist the services of a coder to 'translate' their bargain into code. Errors in translation may result in the code failing to give effect to the parties' actual common intention at the time of contracting. In these circumstances, rectification might be available to amend the code so that it is consistent with that common intention.

A court may face practical difficulties in rectifying coded terms. For example, it may not be possible for the coded terms to be amended if they are immutably recorded on

a distributed ledger. Another difficulty is that a party may only discover the error in the code after the code has executed. In these circumstances, rectification may not provide an effective remedy for a claimant, who will want to reverse the effects of the code's execution.

## Vitiating factors and restitutionary remedies

The law recognises various 'vitiating factors' that render a contract defective. These include mistake, misrepresentation, duress and undue influence. A vitiating factor may render the contract 'void' (of no effect from the start) or 'voidable' (liable to be set aside from the start). Where a contract is void or a voidable contract has been set aside, the court may award restitutionary remedies to restore the parties to their pre-contractual positions.

## Mistake

A contract can be rendered void if one or both of the parties laboured under a mistake when entering into the contract. In the smart contracting context, the parties may hold beliefs or assumptions about how the code will execute. If those beliefs or assumptions turn out to be mistaken, might the smart contract be void on the ground of mistake?

Mistake has a narrow scope under English law and it may be difficult to establish in a smart contracting context. Where both parties are mistaken about a matter relevant to the code's execution ('common mistake'), the mistake will only render the contract void if it makes performance of the contract or achievement of the contractual purpose impossible. That the mistake renders



---

performance of the contract significantly more onerous for one of the contracting parties is not sufficient. Further, mistake will not be arguable if the smart contract contains provisions which, on their proper interpretation, allocate the risk of the mistake between the parties.

Where one party is mistaken about the code's execution ('unilateral mistake'), the mistake will only render the contract void if the mistake relates to a term of the contract and the other party knew of the mistake at the time of contracting. A mistake about circumstances surrounding the execution of the code, even if known to the other party, will not be sufficient to void the contract. The mistake must relate to a term of the contract. Difficult questions may also arise in proving a party's 'knowledge' of a mistake, where a smart contract is entered into by computer programs on the parties' behalf.

### **Misrepresentation, duress and undue influence**

A contract is voidable if a party is induced to enter the contract by a misrepresentation made by the other party. We do not anticipate that smart contracts will give rise to novel legal issues in this area. Like traditional contracts, the entry into a smart contract will typically be preceded by a period of negotiation between the parties. Whether a party made a misrepresentation, by their words or conduct, which induced the other party to enter into the smart contract, can be determined by applying the existing law.

A contract is voidable if a party is induced to enter the contract by an illegitimate threat made by the other party, or where a party enters into the contract under the undue

influence of the other party. We do not anticipate that smart contracts will give rise to novel legal issues in these areas. As in the case of traditional contracts, the question will be whether the claimant entered into the smart contract because of illegitimate threats made by the other party or in the context of a relationship of undue influence.

### **Restitutionary remedies**

We anticipate that restitutionary remedies may be particularly relevant in the smart contract context. In a traditional contracting scenario, parties are likely to cease performing the contract when they discover the factor rendering the contract void or voidable. However, in a smart contract, some or all of the terms are performed automatically by code on a distributed ledger. Depending on the nature of the distributed ledger, there may be no mechanism for the parties to stop the execution of the code. The code may continue to execute (and fully execute), regardless of the fact that one or more vitiating factors might have arisen in relation to it. In these circumstances, the parties are likely to rely on restitutionary remedies to recover benefits transferred by the code under the defective smart contract.

We do not anticipate that English courts will encounter novel legal issues or practical difficulties in awarding restitutionary remedies if a smart contract is void. The voidness of the contract could provide a foundation for a claim in unjust enrichment, the remedy for which is restitution. The court could identify the benefits transferred by the code under the void contract, value those benefits in money, and then order the parties to make restitution to each other to the value of those benefits.

---

Where a smart contract is voidable, it may be set aside so long as the parties can be restored to their pre-contractual positions. English courts do not insist on literal restoration; instead the court aims to do ‘practical justice’ between the parties. In some cases, it may not be possible literally to restore benefits transferred by code if those transfers are immutably recorded on a distributed ledger. However, the court could achieve practical justice between the parties by, for example, valuing the benefits transferred by the code in monetary terms, and ordering the parties to make restitution of those benefits.

## **Breach of contract**

It is often said that smart contracts dramatically reduce the possibility of breach of contract. However, this conclusion may be premature. For example, if the code is merely a tool used by the parties to perform their obligations under a natural language contract, then a party may be liable for breach of the natural language contract if the code fails to perform those obligations correctly. The party in breach may have to pay damages to place the other party in the position they would have been in had the code executed correctly. Exceptionally, the remedy of specific performance may be available if damages are inadequate. A court could order the party in breach to deploy a new piece of code, which corrects the defective execution of the old piece of code.

Where the code is not merely a tool for performing the parties’ contractual obligations, but instead records and defines those obligations, it may be more difficult to establish a breach of contract. If the ‘meaning’ of code is what the code instructs a computer to do, then it is difficult to see how a coded obligation could be breached by the computer that performs it.

---

A functioning computer will generally always perform in accordance with its instructions: it will “simply do what it is programmed to do”. On the other hand, if the coded terms are interpreted by reference to what a reasonable coder would understand the code to mean, it may be possible to establish a breach of contract, if the performance of the code diverged from that meaning.

## **Frustration**

In the smart contract context, there is a risk that events beyond the parties’ control may affect the execution of the code. For example, there may be a system breakdown on the platform on which the code is deployed, which means that the code cannot execute or that the code executes in a different way than the parties anticipated. External data sources or ‘oracles’, which relay information to the smart contract, may also break down, affecting the performance of the code.

We anticipate that, in many cases, parties to smart contracts will deal with these risks by including a ‘force majeure’ provision in the natural language component of the smart contract. This will typically identify a range of events that might affect the performance of the contract, specifying their effect on the contract and the remedies available to the parties in those circumstances. In that case, the question of frustration will not be reached because the consequences of the subsequent event will be governed by the force majeure provision.

However, in the absence of a force majeure provision, a party might argue that the smart contract is ‘frustrated’ at common law, meaning that performance has become physically or legally impossible, or something “radically different” from what was contemplated by the contract. In

---

principle, a smart contract could be frustrated where, by reason of a subsequent event, further performance of the code becomes impossible. For example, if the platform on which the code is deployed is shut down due to some unforeseen event, this could be seen as a case of impossibility of performance due to the destruction of the subject matter of the smart contract. In other cases, the subsequent event might not prevent the code's performance, but cause the code to execute in a way which is "radically different" to that contemplated by the contract.

If a smart contract is discharged for frustration, the parties may recover benefits transferred by the code prior to frustration under the Law Reform (Frustrated Contracts) Act 1943. If the code continues to transfer benefits after the frustrating event, those benefits could be recoverable under the law of unjust enrichment.

## Illegality

A concern that is sometimes expressed about smart contracts is that they may facilitate illegal activity. Some DLT systems

---

enable the parties to transact using pseudonyms, without disclosing their real identities. Further, DLT enables participants to transact directly with one another, without the need for intermediaries, such as banks, who would traditionally play a role in detecting illegal activity. Finally, the immutability of data on a DLT system may make it difficult for authorities to halt the code's performance once illegal activity is detected. If, however, a party were to bring a claim in relation to a smart contract tainted by illegality, a remedy might be denied, in the same way as it would in relation to a conventional contract, if to do otherwise would harm the integrity of the legal system.

---

**We ask questions to discover whether consultees are aware of, or foresee, any difficulties in awarding remedies where problems arise in the formation or performance of the smart contract**

---



# CONSUMERS AND SMART CONTRACTS

Most of the legal issues raised in the call for evidence apply to all smart contracts, whether business to business commercial contracts, peer to peer arrangements such as transfers of bitcoin or Ether tokens, or business to consumer (“B2C”) contracts. In Chapter 6 of the call for evidence we look at existing consumer law protections in three broad categories:

- transparency and fairness requirements
- consumers’ right to treat the contract as at an end
- information rights.



We consider how these measures may assist consumers who enter smart contracts and ask consultees what challenges they foresee in applying existing consumer protection laws to smart contracts. We ask to what extent smart contracts are already being used or developed in the context of consumer contracts, and whether any smart contract specific protections may be required if B2C smart contracts become more prevalent.

## Transparency and fairness

A particular challenge in the B2C smart contract context is that consumers may be required to agree to a trader’s standard terms and conditions without understanding some or any of what they are asked to agree to because they are not code literate.

There are already statutory requirements for traders and service providers to ensure that terms of a consumer contract are “transparent” and unfair terms and commercial practices (including misleading actions and omissions) are unenforceable. Where the terms of a smart contract are wholly or partly in code, the trader may have to consider taking further steps to comply with these requirements. This could include, for example, providing pre-contractual literature to explain coded terms and conditions or including a natural language element in the smart contract which sets out the terms and conditions in full.

## Information rights

There are also statutory requirements which require traders offering goods or services to consumers in certain circumstances to set out, in advance of the consumer entering the contract, the key terms in a clear and



---

comprehensible way. It therefore appears that businesses are already under an obligation to provide a clear, natural language explanation of any coded terms in a consumer contract.

### **A consumer's right to treat a contract as at an end**

An important protection for consumers is their right to treat a contract as at an end in certain circumstances. These include where a consumer receives goods they have ordered but they are not as described, or the goods are not delivered within an agreed period. Consumers also have a right to withdraw an offer or cancel certain distance contracts without giving any reason and to receive a reimbursement by the trader.

Ensuring that consumers can exercise these rights in a smart contract context may pose a challenge. Since smart contracts perform automatically and are immutable where recorded on a distributed ledger, the ability to end a contract at any point is not a natural fit. It will be necessary, therefore, to consider how to ensure that a consumer can effectively exercise their rights to treat a smart contract as at an end, withdraw an offer, or cancel a distance contract.

---

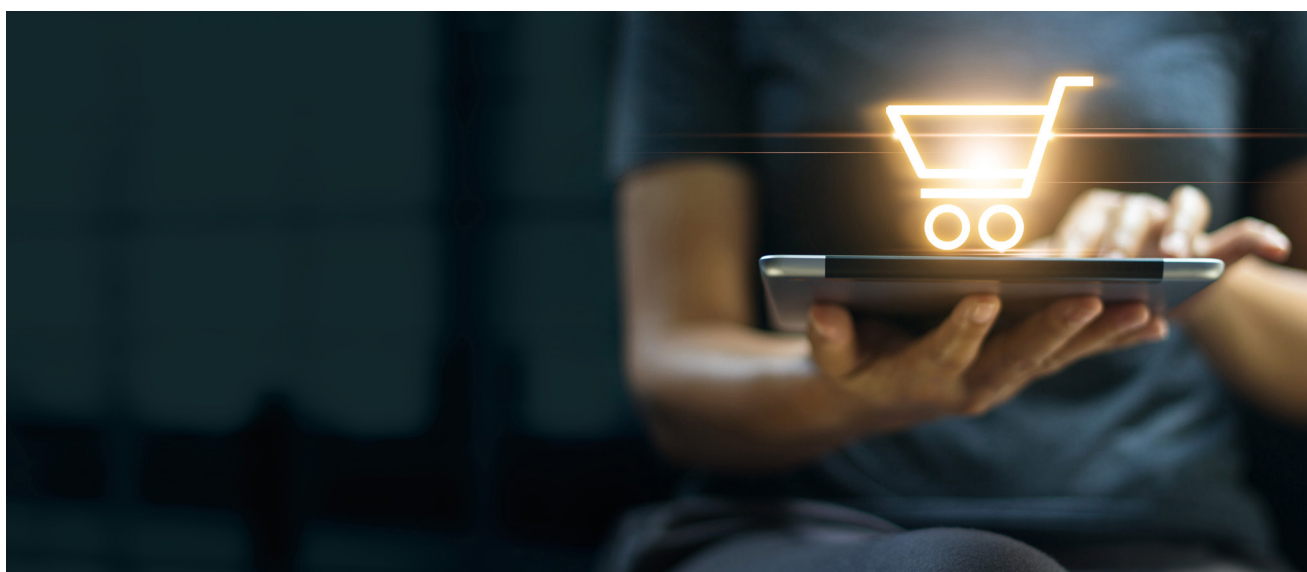
### **Is smart contract specific protection required?**

Existing consumer protections were not designed with smart contracts in mind. In some instances, however, they appear flexible enough to accommodate smart contracts and deal with the challenges these may pose to non-code literate consumers. However, there are other important areas of consumer protection, such as consumers' rights to treat a contract as at an end in certain circumstances, which may not be so easily accommodated by smart contracts.

---

**We ask consultees questions about any challenges they foresee in applying consumer protection laws to consumer smart contracts and whether specific protections may be required if consumer smart contracts become more prevalent**

---



---

# JURISDICTION AND SMART CONTRACTS

When problems arise in relation to cross-border contracts and the parties have not included a choice-of-court clause in the agreement, the rules of private international law determine which national courts have jurisdiction to hear and adjudicate upon the parties' claims. In Chapter 7 of the call for evidence we consider the various factors that can determine whether the courts of England and Wales will have jurisdiction to hear disputes under those contracts. Of these various factors we identify four which we think may be challenging to apply in a smart contract context. We ask consultees questions about these and whether they foresee any other challenges in determining jurisdiction where a smart contract does not have a choice-of-court clause.

## Domicile and presence of contracting parties

One factor which may be relevant to determining jurisdiction is the domicile or even the physical location of the defendant. The pseudonymous nature of some DLT systems may make it comparatively more common for parties to enter into smart contracts without knowing the real identity of their counterparty. This poses obvious challenges for determining the applicable jurisdiction regime and for making out bases of jurisdiction that rely on a defendant's domicile or presence.

## Circumstances of contract formation

The place of formation of the contract may also be relevant to determining jurisdiction. Conventionally, a contract is formed at the moment when, and in the place where, the acceptance of an offer is communicated to the offeror by the offeree (or their agent). Identifying the place of formation may be more challenging in a smart contract context where there is little or no natural language interaction between the parties or where computer programs interacting autonomously on a distributed ledger form a smart contract.

## Governing law

A contract's governing law can be:

- a basis upon which jurisdiction is founded;
- a factor in determining the comparative appropriateness of a particular court; or
- a necessary precursor to identifying some other basis of jurisdiction (such as the contract's place of performance).

Parties can agree upon a governing law and express this in a contract. However, we query whether it is possible to express a governing law clause in code. This could cause challenges for smart contracts which are written solely in code. Another peculiarity of smart contract is that parties may wish to choose for their obligations to be governed by the protocol of a DLT platform rather than by a system of national law. However, it is unlikely that this is a valid choice of governing law.

---

In the absence of party choice, the law applicable to a contract will be determined by its connections to different legal systems. A significant challenge for private international law will be that of identifying meaningful connecting factors for a form of contract that is designed to be dispersed across participants all around the globe.

## **Performance, acts and enrichment**

Jurisdiction rules are often based on the identification of a substantive connecting factor between a contractual dispute and a particular legal system. Such a connection can be found in the place of contractual performance, the place of acts giving rise to a liability to make restitution, or the place of enrichment.

---

Smart contracts may pose certain unique challenges when seeking to identify the geographical location of performance, actions and enrichment, such as where the obligations under a smart contract are performed on a distributed ledger rather than involving any physical performance in the real world.

---

**We ask consultees questions about our analysis of these factors and whether there are other novel jurisdictional difficulties which should be considered.**

---



