Lost and found in smart contract translation – considerations in transitioning to automation in legal architecture[1]

By Scott Farrell[2], Heidi Machin and Roslyn Hinchliffe, King & Wood Mallesons, Australia

"Smart contracts" can be different things to different people, from a theoretical term for self-executing computer code, to legal contracts which use automated processes to ensure performance. The relevance of smart contracts has evolved since their conception. The development of blockchain and other distributed ledger technology has enabled more sophisticated use of smart contracts, supporting their evolution from a theoretical possibility to a practical reality.

The transition to automation in legal architecture through smart contracts should improve transactional productivity, efficiency and risk management. However, this transition also involves a translation of legal rights, obligations and remedies into a framework which is, at least partly, digital. It is critical that this translation is conducted carefully, so that nothing is unknowingly lost with the change in expression. Further, it gives rise to issues of legal recognition which do not arise for traditional contractual frameworks and to which harmonized cross-border solutions need to be found.

In this paper, we set out what smart contracts are, legal considerations in their creation and issues of legal recognition and harmonization which need to be considered further if the potential for these contracts to transform cross-border trade and finance is to be achieved.

1 Smarter contracts

The term "smart contracts" is not new. Early references to "smart contracts" can be found in data and computer science papers from the late 20th Century. In 1996 Nick Szabo described smart contracts as being:

> *"a set of promises, specified in digital form, including protocols within which the parties perform on the other promises."*[3]

In other words, a digitally executable set of obligations which requires minimal external inputs to ensure performance. The architecture of the smart contract automates the performance of obligations. However, these smart contracts are more than mere automated processes for enabling performance, they also have the means of ensuring the performance.[4] Accordingly, there is confidence that the transaction can be completed. This is sometimes referred to as "tamper-proof" execution. It is this ability to perform "on its own" which makes the contract "smart". However, despite their name, these smart contracts may be created only

1 This paper in part draws on the remarks made to the combined meeting of the Financial Stability Board and the Committee of Payments and Market Infrastructure on Uses of Distributed Ledger Technology in Financial Markets and Issues for Financial Authorities at the Bank of England on 21 October 2016 and to the 2016 UNCITRAL Emergence Conference Regional Perspectives on Contemporary and Future Harmonization Agenda in International Trade Law on 13-14 December 2016, in each case by Scott Farrell.

2 Scott Farrell is a member of the Australian Government's FinTech Advisory Group. However, none of the comments made in this paper should be taken to have been provided in this capacity.

3 Szabo, *Smart Contracts: Building Blocks for Digital Markets*, 1996.

4 An example of this is a vending machine which completes transactions automatically by delivering the item purchased once the payment has been made. Completion of this transaction requires no input from the owner of the machine. Instead, the architecture of the system controls the timing and the manner of the parties' performance.

as processes for effecting transactions and it may not be integral to their design that they have any recognition at law.

The connection between smart contracts and legal contracts has arisen as new technology has allowed use of smart contracts to be expanded into more complex transactions and relationships. Blockchain and other distributed ledger technology has permitted smart contracts to be applied to a broader architecture where digital assets and digital value can both be maintained. In this more complex architecture, self-executing computer code can be used to effect transactions, exchanging the digital assets for digital payments. This has allowed smart contracts to be applied in a broader range of transactions, including those where legal enforceability is as important as operational execution. Some early definitions of smart contracts are not able to describe this expanded application, particularly as some parts of those transactions are too complex to be automated. A revised definition of smart contract is warranted, and one has been supplied by Clack, Bakshi and Braine:

> "*A smart contract is an agreement whose execution is both automatable and enforceable. Automatable by computer, although some parts may require human input and control. Enforceable by either legal enforcement of rights and obligations or tamper-proof execution."*[5]

This definition defines a contract to be a smart contract if at least *some parts* can be performed by a computer programme automatically, without human input, *whilst also* being enforceable either through the appropriate legal system or because the execution of the obligations by the computer cannot be undone. It is a contract which is effective either through self-execution of computer code or legal force. The coded contract and the legal contract are linked. This linkage between computer code and law is the basis for the issues discussed in this paper.

Before considering those issues, it is worth describing why smart contracts warrant consideration. In summary, it is because they improve the efficiency, speed and performance of contracts. Efficiency is improved because of the automation of contractual actions, which reduces the need for human involvement and, as a result, the potential for human error. Speed is improved as actions can occur in real time as information is collected and verified. Performance is improved as the terms are unambiguous and results predictable and auditable. This has the potential to reduce the risk of disputes. These factors are similar to those which drive the use of financial market infrastructure for critical transactions. Indeed, when appropriately used with distributed ledger technology, smart contracts have the ability to perform the same functions as centralised financial market infrastructure – which is particularly important for those transactions where such infrastructure is not currently available, or impossible to obtain.[6]

It is fundamental to a legal examination of smart contracts to consider whether it is possible for a contract which is set out in computer code to be valid at law. This is considered in the next section.

2 A contract in code

As described above, a smart contract achieves efficiency, timing and performance improvements because of automation of the contract's terms. This automation is effected through the computer code which governs the automated performance. This raises a critical issue – whether contractual provisions which are expressed in computer code can be valid and effective under law. This is

[5] Clack, Bakshi and Braine, *Smart Contract Templates: foundations, design landscape and research directions*, 2016.

[6] This comparison is considered further in section 5 of this paper.

sometimes regarded as a difficult hurdle to clear for the widespread use of smart contracts in complex and regulated areas, such as finance. Often this issue is considered in the context of a contract which is expressed entirely in computer code. In such cases, basic contractual formation issues can cause concerns, such as the identity of parties, the identification of terms, the time of creation and the governing law.

However, before considering these issues, it is important to keep in mind that not all of a smart contract needs to be set out in code. Some terms of contracts which are more complex than the immediate transfer of value and property are likely to not be efficiently encoded. This is because computer code (like mathematics) is well adapted to represent terms which are expressions of logic but not terms which are based in concepts such as reason or conscience. Further, they are not useful to represent terms which are based on the exercise of discretion that is outside of clearly defined frameworks.

For example, code could be used to represent the contractual agreement that, if an event happened:

> "*the price is to be adjusted by subtracting the product of x and y.*"

This provision can be coded easily because it is an expression of logic. However, code would not be useful to accurately represent that, if an event happened:

> *"the price is to be adjusted by the party in a commercially reasonable manner."*

Or:

> *"the price is to be adjusted by negotiation between the parties in good faith."*

These two provisions are based in the exercise of reason, conscience and discretion in the future. They cannot be easily coded because their meaning is not able to be comprehensively expressed as a matter of logic. An attempt to do so would create the risk of *divergence* expressed in natural language between the meaning of the original contractual provision and its expression in code.

Real commercial and financial contracts are a mixture of contractual provisions based in logic, reason, conscience and discretion. Accordingly, there are a mixture of provisions which are able to be efficiently coded and those which are not. It follows from this that if smart contracts are to be used meaningfully in commercial contracts then they will need to be blends of both coded and natural language terms. As the logical provisions are usually applicable during the normal life-cycle of contracts (as opposed to the provisions which apply when unexpected events occur) then this is where the most efficiency should be obtained from using coded terms.[7]

This context changes the nature of the fundamental legal issue with contracts expressed in code. As complex contracts will need to be blends of natural language and code the primary concern is no longer whether a contract can be created because the natural language elements of the contract should be able to satisfy these requirements. Instead, the concern should be whether a contract can be valid if part of it is expressed in natural language and part in computer code. An alternative to a blend of coded and natural language terms is to retain

[7] This separation in contractual architecture between provisions applicable in the ordinary life-cycle of transactions and those applicable in other circumstances can be best seen in the contractual frameworks used in the international derivatives market. Life-cycles provisions are contained in transaction confirmations whilst other provisions are contained in the master agreements which govern them.

the entire contract in natural language and use code to separately perform those terms. However, this duplication creates the risk of a discrepancy, the risk that the actions being performed by the code do not match the legal meaning of the natural language contract.

This is an area where local laws will be particularly relevant and there is a role for harmonisation of those laws to facilitate cross-border transactions. Key issues for consideration include:

- ***Legitimacy of multilingual contracts.*** In one sense, a contract which includes provisions which are expressed in code is similar to a contract which includes provisions which are expressed in different languages. If contracts which are expressed in more than one language are not effective under local laws, then it is likely that contracts expressed partly in code and partly in natural language may not be effective either.

- ***Understanding of coded terms.*** If a contract is partly expressed in code then the understanding of that code by the parties is relevant. This relevance can be a matter of regulation, such as whether particular parties (such as consumers) can be bound by terms which they do not understand, or it can be a matter of fundamental contract law, such as whether there was sufficient mutual understanding of the terms to form the contract at all.

- ***Evidence of coded terms.*** Even though the parties can agree to express specific terms of their relationship in computer code, it is important that that expression is admissible in any judicial and arbitrary proceedings which arise out of that relationship. An inability to admit this record of the parties' agreement would impair its legal effectiveness.

These issues can be seen in context if the circumstances are changed only slightly, by replacing the reference to provisions expressed in code with provisions expressed in a foreign language. If part of a contract was to be performed in a foreign country then it could be more efficient to express that part of the contract in the language of that foreign country, so that the people having to perform can understand the terms in their own language. Each of the issues described above would also apply to such a bi-lingual contract. Hopefully, they are able to be solved under local laws in exactly the same way.[8] Whatever the solution reached under local laws, there would need to be consistency across jurisdictions which seek to be involved in an international smart contract marketplace.

Of course, there is a difference between the creation of a smart contract comprised of both natural language and coded terms and ensuring its enforceability. The self-executing nature of the coded provisions does not guarantee their effectiveness at law. This is because the law does not accept that everything which has been done must have been lawful. The fact that the terms are in code, does not mean that the code has become the law. This is discussed in the next section of this paper.

3 The code is not law

The effect of self-executing performance of smart contracts has been described as "the code is law", because the coded provisions have effect without external input or control.[9] However, this is usually not intended to be a comment on the

[8] Assuming the local law is meant to be technologically neutral.

[9] Indeed, that is the very expression used by some leading commentators, such as Lawrence Lessig. Lessig's original work was in fact called *The Code is Law and other laws of cyberspace (*1999). Lessig published a second edition of this in 2006 called *Code 2.0.*

applicability of the law. Of course, smart contracts do not change or replace law and the law still applies to a smart contract regardless of the code.[10] This means that a key issue in translating contracts into smart contracts is to determine which laws need to be contemplated in the smart contract design, because they will apply regardless of the smart contract's code. Building this into the smart contract architecture allows the smart contract to work with the law, instead of trying to work against it. Two categories of such laws are those which interrupt or reverse performance and those which change the contract's terms.

Laws which interrupt or reverse contractual performance

Laws can interrupt the performance of contracts, or cause the reversal of performance of contracts, for different reasons. In essence, these laws have this effect because they express the public policy that the result of interrupting or reversing the parties' private contract is more important than compelling those parties to fulfill their obligations under that contract in accordance with its terms. It is not possible to contract out of these laws; they apply regardless of its terms or its performance. Accordingly, it is not possible to "code" out of these laws either.

One example are laws which render obligations under a contract to be void or voidable. This can happen because of the insolvency or bankruptcy of one of the parties, the contract is found to be unauthorized, improper or fraudulent or the purpose of the contract, or the conduct of the parties is contrary to regulation or otherwise unlawful. If a smart contract contains coded terms which effect payments and deliveries between the parties over a period of time then those terms could conflict with those laws (for example, if one of the parties becomes insolvent before all of those payments and deliveries have been made).

If the smart contract is to remain efficient, it is important that it is sufficiently flexible in order to operate in compliance with these laws. This is particularly the case where the technology on which the coded parts of the smart contract operate is, from a technology perspective, immutable. This would be the case for smart contracts held on a blockchain or other distributed ledger technology.[11] In these circumstances, there needs to be included in the smart contract architecture a method for relieving the contract from the constraints of that technology. If there is a difficulty with including this in the coded terms then the flexibility could be provided through the natural language provisions which also form part of the smart contract.

Importantly, this is not a question of changing those laws so that they do not impact on smart contracts. The laws exist to achieve a policy outcome which is deemed more important than individual contract certainty. This should apply equally to traditional contracts and smart contracts (technological neutrality works "both ways"). What is important from an efficiency perspective is that the contract is designed so it has a means to operate other than in conflict with such laws.

Laws which change contractual terms

Laws can also change the terms of a contract. This can happen by terms being implied by law (such as under consumer protection legislation), terms being

[10] This point was expressly noted by Lessig. In *Code 2.0* Lessig notes: "Of course, for the computer scientist codes is law." but "Code is not law, any more than the design of an airplane is law. Code does not regulate, any more than buildings regulate. Code is not public, any more than a television is public." Another way of expressing could be that code is law for machines, but law is code for legal entities. They each work in different frameworks.

[11] The immutability of a blockchain arises from its "append only" nature. This means that new information can be added, but the existing information cannot be altered.

found void and removed, or even by rectification if a court finds that the terms do not reflect the true agreement between the parties.

This is also a significant consideration in translating contractual provisions into a "smart" form. If a subsequent change in terms required by law is not able to be included in the operation of a smart contract, then there is a significant risk that the automated performance of the contract will not be consistent with the terms of the real agreement between the parties – there will be a risk of a "discrepancy" between agreement and performance.

Theoretically, it is possible for the coded terms of a smart contract to be drafted so as to automatically respond to certain laws. For example, Marino and Juels[12] have proposed the inclusion of standards in smart contracts dealing with rescission by court, which have the effect of halting automatic performance of the smart contract if an order is made by an appropriate court, and providing for the contract to automatically compensate partial performance. However, given the immense complexity of laws and potential factual circumstances, it would not be feasible, nor computationally efficient, to include such automated standards covering all potentially applicable laws and events, particularly where the underlying transaction or surrounding facts are complicated.

Accordingly, it is also important from an efficiency perspective that the design of the smart contract is sufficiently flexible to be able to respond to a change in its terms imposed by law without creating an unavoidable discrepancy between those terms and its automated performance.

Flexing the code

It is noted above that smart contracts need to be designed with sufficient flexibility to work with the laws which will apply to them, and to other contracts. However, this need for flexibility in a smart contract can compete with the certainty provided by the coded parts of that contract. The efficiency of a smart contract arises from its automatic operation and there are practical limits to the flexibility which can be included in such automation. Current technology does not facilitate an efficient incorporation of all of the possible events which could occur into the fabric of the code of a smart contract. This leaves the question as to how the necessary flexibility can be achieved.

One possibility for achieving the effect of a reversal or change in terms is to create that result through the creation of a new smart contract which, when added to the existing contracts, has the effect of the desired reversal or change. An example of this would be a new transaction which exactly offsets an existing transaction, negating its effect. However, this solution is not perfect as the very reason for the reversal or change in performance could also legally prevent such new transactions from being legally effective. This could apply in the insolvency of one of the parties where the new transactions themselves could be void if they are entered into after insolvency has commenced.[13]

An alternative approach is for the architecture of the smart contract to include an "off ramp", which allows the contract to be governed by the natural language terms agreed between the parties instead of the code. This would allow the code to govern the performance of the contract whilst it is still accurately reflecting the parties' relationship and then ceases to govern performance where it does not. The flexibility can be left to the natural language terms in this circumstance. This mechanism can be useful in other circumstances too, such as where there is a

[12] Marino and Juels, *Setting Standards for Altering and Undoing Smart Contracts*, Rule ML, 2016 pp. 151-166.

[13] Where this is needed for important financial market infrastructure, such as clearing systems, it usually needs legislation for these effects to be given a "safe-harbour" from other laws.

breach or termination of the contract. This is considered in the next section of this paper.

4 Code-breaking

As has been discussed, an attractive feature of smart contracts is that they can *ensure* performance through their self-executing nature. The "tamper-proof" nature of their execution means that performance can, as a practical matter, be counted on. There isn't a need to consider what happens in the case of breach as the opportunity for it occurring is limited, assuming the proper operation of the smart contract. However, this issue is more complex when smart contracts are used for more than the simplest transactional relationships.

If a contract contains executory provisions which are to be performed at some time in the future, then it is always possible that they won't be able to be performed. As described in the previous section of this paper, it won't be possible to legally compel performance if one of the parties is insolvent – even if the contract is self-executing as a matter of process. Also, if there is a failure to perform some of the obligations of a contract, for whatever reason, the other party may want to terminate. This means that the provisions of a smart contract still need to deal with breach and its consequences.

It would be possible to include automated provisions to deal with some of these events. For example, if there were a failure to perform some obligation then the code could ensure that no further obligations need to be performed, make some sort of calculation as to the damage which is payable as a result and also effect that payment. However, this will not be sufficient to deal with termination rights which arise as a result of law, or termination rights which are discretionary rather than automatic. This is important because under many laws, if a party to a contract defaults then the other party is not compelled to terminate and it may choose not to do so. That choice could be made on the basis of other factors which are entirely beyond the scope of the contract itself – such as the impact on other contracts or relationships or information on the circumstances of other, unrelated entities. Attempting to comprehensively catalogue the consequence of breach in the code is either going to over-simplify the existing rights of a non-defaulting party under a contract (the divergence risk described earlier), or involve such a heavy use of code to contemplate the range of possibilities so as to make its use inefficient.

There is a further consideration related to breach. At common law, it is possible to willingly breach a contract. There are consequences which arise as a result, such as an obligation to compensate the other party, usually through the payment of damages. Nevertheless it is an option which parties to a contract have, and in certain circumstances (such as pending insolvency), it is an option which can prove to be important. It arises because of the reluctance which common law courts often have to granting orders for specific performance in the case of ordinary breach of contractual terms. This right to deliberately breach a contract is not consistent with the self-execution of coded terms. Such coded terms do not allow for deliberate breach because the terms are performed automatically. This represents another possible divergence between a traditional contract and coded terms of a contract.

For each of these two issues, there is a solution which would avoid the risk that the terms of the smart contract were different to those of the traditional contract on which it was based. That would be to include the flexibility to allow the terms of the contract to be governed by its natural language provisions instead of the coded provisions (the "off-ramp described in the previous section). This would seem conceptually simple to implement in the case of breach of the contract. In the case of providing the same uniliateral right to breach which exists under traditional contracts, this could be achieved by providing each party with the right to move the contract from its coded terms to its natural language terms at any

time, and through this providing the ability to exercise the discretion not to perform.

5 Recognizing smart contracts

The previous sections of this paper have identified legal challenges in translating traditional contracts into smart contracts. Many of these can be addressed in the design of the smart contract architecture rather than changes in law. However, as smart contracts increase in relevance and scope, there is likely to also be a need to consider developments in law to recognize smart contracts themselves. Two examples of this include recognition as financial market infrastructure and as some sort of entity at law.

Smart contracts as financial market infrastructure

Smart contracts can perform the same functions as financial market infrastructure. Indeed, some of the initial use cases, such as clearing and settlement and payments, are in the performance of those functions. Smart contracts and financial market infrastructure have a shared purpose, being to provide confidence to their users in the performance of transactions. Financial market infrastructure achieves this through the regulation and oversight of the operator and the legal protection given to its rules and regulations. Smart contracts achieve this through their self-executing nature, reinforced by their resilience when they are held on a blockchain or distributed ledger. [14]

This comparison of function and purpose between financial market infrastructure and distributed smart contracts can disguise a fundamental difference between them in their relationship with law. A key foundation of the transactional certainty enjoyed by users of financial market infrastructure is the legal protection which insulates those transactions from the local laws which would otherwise interfere with them. For example, local laws often protect the operation of clearing houses from the impact of the bankruptcy or insolvency of participants. Smart contracts on their own do not have equivalent protection and, as described earlier in this paper, are subject to the full application of local laws despite their self-executing nature.

Application of the legal protection given to financial market infrastructure would solve a number of the "translation" issues already described in this paper. If the terms of the coded contract were applied despite the operation of other laws, then the need to consider flexibility in smart contracts would be reduced. However, this protection is usually offered only to the most important systems in a jurisdiction, such as its payment systems, exchanges and clearing houses. These are highly regulated. Also, the availability of that protection is usually dependent on meeting the international principles applicable to financial market infrastructure.

The international benchmark used for determining what is important to the safety of the financial market infrastructure is the *Principles of Financial Market Infrastructures.*[15] This forms a common language across jurisdictions to determine what is important and how it needs to be protected. These *Principles* identify legal standards as being critical to the "safety and efficiency" of the financial market infrastructure and they identify other principles which depend on legal constructs for their effectiveness.

Where this leaves smart contracts is that if they perform systemically important functions of the same nature as financial market infrastructure then they should warrant the same legal protection as is provided to that infrastructure. However,

[14] The decentralization means that a smart contract which effects clearing and settlement can operate like a clearing system without a clearing house.

[15] Published by CPSS-IOSCO, April 2012.

it will be important that before they do so, they meet the same regulatory requirements including the international standards for the provision of that protection. Of course, the consideration of smart contracts in this context cannot be completely separated from consideration of the blockchain or distributed ledger on which they are held. This technology must contribute to the satisfaction of these requirements.[16]

Smart contracts as legal entities

In the sphere of technology and data science, a smart contract can be referred to as an *entity* to which payments and deliveries can be made and from which payments and deliveries can be received. This is most relevant in multi-party smart contracts such as the decentralized autonomous organization (DAO).[17] From the perspective of traditional contract law this would seem non-sensical as the smart contract is the relationship between legal entities and is not a legal entity itself. However, this deserves further consideration.

A multi-party smart contract, like the DAO, can operate as a business vehicle which facilitates economic co-operation between those who participate in it. Instead of articles of association and shareholders agreements, they operate according to bylaws expressed in the interlocking software code. They have no registered offices, directors or employees. Once the smart contracts are created and deployed on a blockchain or other distributed ledger technology then human involvement is limited as the operation, management and control is automated. Decisions can be made by collective votes made by persons who hold tokens, which represent investments which have been made (these investments are made by contributing digital currency "to" the smart contract).

From an historical perspective, there are a number of types of entities which are now supported by some level of recognition, but which originally were formed as agreements between persons to jointly undertake commercial enterprises. These include partnerships, companies and trusts. Legal recognition was made available to clarify their relationships with other entities, and facilitate their regulation.[18] If these multi-party smart contracts increase in use, then these same issues may arise. For example, it may become important for such arrangements to have in place governance arrangements which are beyond the code and regulation through legal recognition could be a means of achieving this.[19] Accordingly, it is not impossible that multi-party smart contracts could be recognized in the same way.

There are a number of ways in which such recognition could be provided under local law if it were decided to be beneficial. Of course, these entities are unlikely to be constrained by international borders and the issues of recognition become more complex when the conflicts of laws issues which arise with cross-border transactions are added. This is considered in the next section of this paper.

---

16 Consideration of the manner in which blockchain and distributed ledger technology can meet the PFMIs is a worthy subject of a separate paper.

17 The DAO was a smart contract created on a distributed ledger platform intended to automatically facilitate the investment of digital currency into projects. Its terms were said to be limited to its code. It failed with spectacular effect when the code was found not to be consistent with many participants expectations when one participant was able to withdraw funds well in excess of those contributed by it.

18 The process of recognition of company status under 19th Century English law, through unincorporated associations to deed of settlement companies, bears some interesting comparisons.

19 Indeed, the absence of an external governance arrangements was a contributing factor to the difficulties experienced in connection with the DAO's failure. There is an analogy to the need for natural language provisions in bilateral smart contract to complement the coded provisions of a smart contract.

6 Conflicts beyond between code and law

This paper has described circumstances in which the interaction between smart contracts and laws which will apply to them cause a need either for care to be taken in the design of the smart contract's architecture or for consideration to be given to adjustments to local laws. However, commercial contracts often operate in a cross-border context with the result that there is a need to consider more than just one country's laws. This needs to be taken into account in smart contract architecture and expands the consideration of adjustments in law into a need to consider harmonization of those adjustments and laws.

It might initially be thought that the issues of conflict of laws arise because part of the smart contract is expressed in code. For example, one of these issues could be that the governing law of the contract could be difficult to ascertain from the code. However, this issue is likely to be able to be solved with smart contracts which have both coded and natural language provisions as the natural language provisions could record the parties' agreement as to the governing law in the same manner as is used for traditional contracts.

Far more complex conflict of law issues arise because of the combination of smart contracts with blockchain and distributed ledger technology. This is because the resulting distributed nature of the smart contract means that its coded provisions exist, and are being performed, simultaneously in multiple places, some of which could be in different legal jurisdictions. Importantly, this is not an issue which arises in other cross-border commerce. Two examples of the issues which require consideration are:

- ***The location of the smart contract itself*.** A contract's location could be relevant for different reasons, including in connection with the applicability of regulation or taxation. Also, it could be important for dealing with the proprietary aspects of contractual rights, such as transfer of those rights or granting a security interest over them. Local laws do not always provide that the location of a contract for these purposes is a matter of the parties' agreement. It can be a matter of objective determination based on the place of performance. Of course, tests have been developed in local laws to work this out for traditional contracts. However, these tests are not likely to operate effectively when applied to a smart contract which is partly held on, and effected through, a distributed ledger located in multiple places equally and simultaneously.

- ***The location of property evidenced by the contract*.** The terms of a smart contract are able to execute a transaction in, and evidence the ownership of, other property. The property could be currency, securities or other assets which are recorded on a blockchain or distributed ledger. The location of that property is likely to be relevant to the application of a number of local laws, including those related to taxation, duties and taking security. The concept of applying these laws to assets which are recorded on a register is not new and the possibility of multiple registers also exists with some current asset holding systems, such as for dematerialized and intermediated securities. However, what is different about property held in smart contracts on a distributed ledger is that there is no hierarchy between the different records. They are equal in status and synchronized. It is not the same as the holding of property in different layers of custodian and clearing system accounts where a "chain of title" can be tracked through the different registers.

Accordingly, the tests used for a hierarchy of accounts are not likely to be effective when applied to multiple, unsubordinated, registers. [20]

This is not a comprehensive list of potential cross border issues. Others include legal issues relating to privacy, data regulation, anti-money laundering laws and licensing for regulated services. However, these alone demonstrate a need for a solution which is more than the terms of the parties' agreements. Instead, a harmonized approach is needed across jurisdictions which seeks to take advantage of the efficiency, speed and performance benefits provided to commerce by smart contracts.

7 Lost and found in translation

This paper submits that legal challenges in translating traditional contracts into smart contracts for use in commercial transactions requires care in ensuring that nothing is lost, and that some new solutions be found.

Ensuring nothing is lost requires flexibility beyond the use of computer code. This is needed because of the inability of the logic expressible in code to describe the richness of all of a contract's provisions or the law which might be applicable to it. This does not mean that smart contracts have no application beyond the simplest of transactions. Instead it means that there needs to be significant care in the design of the smart contract's architecture to provide the flexibility required for real world operation.

The solution to be found is a contribution by law itself, particularly when it comes to cross-border application. This is because the normal principles used to determine applicability of laws do not contemplate an architecture which can operate independently of the parties and which can operate in multiple jurisdictions equally and simultaneously. This contribution is in the form of local law recognition of particular elements of smart contract architecture and cross-border harmonization of those local laws.

The depths of legal analysis needed to conduct both local law recognition and cross-border harmonization of these laws effectively may seem beyond the scope of the current proofs-of-concept and pilots which are being developed for smart contracts and the blockchain and distributed ledger technologies on which they are maintained. However, given the expanding use cases, and the time needed for that analysis to be completed and implemented, it would seem that there is little time to lose.

[20] For example, applying "PRIMA" (place of the relevant intermediary approach) in these circumstances would be challenging.