Arbitration in Smart Contracts – Code Naïve v Code-Savvy

Written by Hetal Doshi & Sankalp Udgata

Combining law, computer science and finance in unprecedented ways, "Smart Contract" is the latest addition to the unending list of Internet of Things. Unlike a traditional contract, which only lays out the terms of agreement for subsequent execution, a smart contract autonomously executes some or all of the terms of the agreement as it are usually based on Block-chain. It has the potential to reshape our understanding of contract and technology law. The shift from the code naïve to the code-savvy, has surfaced problems in dispute resolution beyond the existing legal perception which this article aims at analysing and resolving.

Working of the Smart Contract

By removing the need for direct human involvement, a smart contract is deployed on to a distributed Trustless Public Ledger. However, in order for the smart contract to work efficiently, exactly specified conditions for the execution of the contract are necessary, otherwise, it will be impossible to automate the process. Also, smart contracts receive information from outside block-chain platform through the use of **Oracle** programs that mediate with external databases and are entered into the block-chain technology.

A Hornet's Nest

Smart contract come with their own sets of limitation and drawbacks. Following are few of the many problems, inevitable in resolving disputes over smart contracts. Interestingly however, although these problems may be encountered by an Arbitral Tribunal, arbitration (with requisite checks) is the most efficient mechanism to deal with such problems.

Enforceability Quandary

1. A) Formal Enforcement

A very fundamental and critical impediment, Courts and Tribunals are consistently skeptical in enforcing such unconventional contracts. Although the

use of automated communication or system to conclude contracts or make it binding on the parties has been long accepted by the business community, a Tribunal is often troubled with disparity in validity of smart contracts over conflicting jurisdictions.

Secondly, Article 2.1.1 of UNIDROIT (PICC) undoubtedly includes automated contracting. However, problems may arise in relation to codes meeting the *in writing* requirement of UNCITRAL and the New York Convention.

1. B) Substantive Enforcement

The artificial nature of contracting deprives actions of the human touch. Complexities arise when there a subsequent smart contracts. For example, if there is a supplementary smart contract, consent for which is sought from the parent contract. Since it is the codes in the parent smart contract that initiate the subsequent contracts and transactions and the performance, can consent be said to have been given by a mere code and is such consent valid and enforceable against such code.

A Hitch in the Seat

Given the distributed nature of block-chain i.e. a ledger which is spread across the network among all peers in the network and the operation of Smart Contracts, it is important to agree a seat for the arbitration to avoid satellite disputes about the applicable seat and/or procedural law.

Problems in Execution- Irreversibility and Irremediability

Since they are theorized to be complete contract by focusing on *ex ante* rather than *ex post*, they eliminate the act of remediation, by admitting no possibility of breach. However, the *DAO case* was incomplete as it failed to anticipate the possibility that coding errors could result in unexpected wealth transfers. In addition, smart contract may deal with commercial scenarios so complex and unpredictable that the code will fail to embed all possible answers to all possible questions.

Further, if the smart contract contains a mistake, security flaw, or does not accurately capture the parties' intent, the smart contracts will be difficult to modify or change, due to a block-chain's resilient and tamper resistant nature.

The program will continue to blindly execute its code, regardless of the intent of the parties or changed circumstances. When the transaction is more complex, involving multiple players (humans or machines), multi-component assets and diverse jurisdictions, computer code *smartness* may easily turn into plain *dumbness*.

Needless to mention, a Tribunal or a Court will encounter several problems in executing a decision vis-à-vis a smart contract such as:

1. Lack of in-rem jurisdiction- Reversing a transaction on a decentralised ledger with several contributors that may not even be parties before the Tribunal.
2. Excusing future performance or specific performance- Since they operate automatically and are not flexible.

The Truth about Consent

Contracting also has issues such as duress, fraud, forgery, lack of legal capacity and unconscionability which require human judgement and cannot be scrutinised by a smart contract which simply functions on a series of binary inputs. Moreover, though it provides guarantee of execution to certain extent, it cannot verify whether the contracting parties have the legal capacity to get into legal relationships or business capacity to make an agreement.

It also does not care whether there truly exists *consensus as idem* between contractual parties, there is no possibility for the contract to be void or voidable. However, although codes are not natural language that might be vague or ambiguous, leaving space for interpretation. For a consensual dispute resolution mechanism like arbitration, the indispensable requirement of free consent and the evaluation of intention of parties cannot be comprehended by a smart contract that stands deprived of reason and morale.

This may be an issue in circumstances where the Smart Contract is entered into by a computer, is in code and/or and does not create legally binding contractual obligations under the applicable law. The solution to this can be that the Arbitration clause can become part of the *Ricardian contract* which like any other similar contract is a hybrid form of smart contract which is partly in human readable form.

The Catch in Imputing Liability in a Dispute

The code smart is sadly not insusceptible to security vulnerabilities and exploits like *forking*, which could cause a smart contract to operate unexpectedly and invalidate transactions, or worse, enable a third-party to siphon digital currency or other assets from contracting parties accounts. Scary, isn't it?

However, since a Tribunal is only an *in personam* jurisdiction, it can barely inspect or issue directions against such third parties. Such vulnerabilities might also jeopardise the secrecy that arbitration aims to achieve.

It is not unjust to say that such a contract is dangerous enough to attract *strict liability* in case of any harm caused due to an error in coding. That, juxtaposed with the existence of foreseeable risk in execution of smart contracts poses a potentially huge hurdle to the exponentially growing use of block-chain technology.

Furthermore, disputes, to summarize, may arise:

1. between the parties of a smart contract, or
2. between two conflicting smart contracts.

Since the code smart is a form of artificial intelligence replacing human involvement, it is the second set of disputes where a Tribunal or Court will be troubled with the attachment of liability.

Cutting the Gordian knot – checks and suggestions

Given our shift from *not so smart contracts,* we must keep an eye for the following checklist while dealing with dispute resolution in smart contracts.

Formality requirements

Parties should therefore ensure the arbitration agreement meets any formality requirements under the governing law of the arbitration agreement and Smart Contract, the law of the seat and wherever the award is likely to be enforced.

Choice of seat

Parties should base check whether in their chosen seat,

1. Domestic law does not render a Smart Contract illegal or unenforceable
2. The disputes likely to arise are arbitrable
3. The codified arbitration agreement in question will be upheld and enforced by the supervisory courts.

Tribunal with specialist technical knowledge

Some Smart Contract disputes will be fairly vanilla contract law disputes, but others will be of a highly technical nature, for example, where the code does not operate as expected. Pursuant to the novel nature of the smart contract the importance of having a tribunal familiar with the technology against the importance of having the dispute decided by experienced arbitrators becomes crucial.

Severable arbitration clause

Although the *doctrine of separability* protects the validity of an arbitration clause, the dispute resolution clause should always be kept independent of any smart codes.

Localised Termination Clause

Given the automated and perpetual nature of smart contracts, there should be an option to terminate the contract. Although non-amenability is an essential feature of a smart contract, the option to cede away from the distributed ledger (terminate the contract) should be sole switch available the each of the contributors. The code may prescribe conditions for pulling the plug, i.e. create joint switches. Therefore, a party shall not be able to terminate its obligations without assent from any of its debtor on the ledger. As a result, once the debt is settled either by payment of dues or by an award of a Tribunal, the parties may pull the plug.

Power of Pardon

Each party to a smart contract should be at liberty to excuse payment by a debtor in under a direction by a tribunal or a Court in case of a *force majeure* or any other scenario where performance is liable to be excused.

This list, although non-exhaustive, will certainly sustain best practices in arbitration until the next great invention in the sphere of technology and business

will live to fight another day.