FIELD LAW

October 2018

The Medium

Intellectual Property
+ Technology Law Newsletter

SUBSCRIBE | FORWARD | CONTACT | WEBSITE

How do "Smart Contracts" Fit With "Traditional Contracts"?

By **Richard Stobbe**

Placing Smart Contracts in Context

A "smart contract" is really a set of computer programs designed to automatically execute certain transaction steps, provided certain conditions are met. It's not so much a contract, in the legal sense, as it is a way to execute a transaction, using software. These have also been referred to as "programmatically executed transactions" or PETs.

It's worth emphasizing that smart contracts or PETs are merely one element of the whole permissioned ledger ecosystem, using blockchain (or distributed ledger) technology. The smart contract enables and implements certain important transactional steps, but those steps fit within the broader context of a matrix of contractual relations between the participants.

Many of those relationships will be governed by "traditional" contracts. This traditional contract architecture enables the smart contract workflow. The take-home point here is that traditional contracts will remain a part of these business relationships. Here's an example: the Apple iTunes ecosystem contains a number of programmatically executed transactions using sophisticated rights-management software. When a consumer chooses a movie rental, a song download or a music subscription, the order fulfilment and payment processing is entirely automated by software. However, users cannot participate in that ecosystem, nor can Apple obtain content from content producers, without an overarching set of traditional contracts: end user license agreements, royalty agreements, content licenses, agreements with payment providers.

Those traditional contracts enable the PET, just as the PET enables the final transaction fulfillment.

Changing Smart Contracts

Once a programmatically executed transaction is set loose, we think of it as a self-actuating contract: it cannot be changed or altered or stopped by humans. The inability of humans to intervene is seen as a positive attribute - it removes the capriciousness of individuals and guarantees a specific pre-determined

machine-driven outcome.

But what if the parties decide (humans being humans) that they want the contract to be suspended or altered, due to changes in circumstance, price, or other variables that were not contemplated at the time the original smart contract code was written?

Where humans control the progression of steps, they can decide to change, stop or reverse at any point in the workflow. Of course we're assuming that this is a change or reversal to which all parties agree. But what is the mechanism to mutually hit "pause", or change a smart contract once it's midflight? That remains a challenge of smart contracts, particularly as PET workflows gain complexity using blockchain-based technologies.

- One solution may be found within those traditional contracts, which can be drafted in such a way that they allow for a remedy in the event of a change in circumstances to which both sides agree, even *after* the PET has started executing the steps it was told to execute. In other words, the machine may complete the tasks it was told to do, but the humans may decide (contractually) to control the ultimate outcome, based on a consensus mechanism that can override the machine during or after the fact. This does have risks - it injects uncertainty into the final outcome. It also carries benefits - it adds flexibility to the process.

- Another solution may be found in the notion of "hybrid contracts" which are composed in both machine-readable form (code) and human-readable form (legal prose). This allows the parties to implement the consensus using a smart contract mechanism, and at the same time allows the parties to open up and change the contract terms using more traditional contract methods, applying traditional legal principles.

Terminating Smart Contracts

Finally, consider how one party might terminate the smart contract relationship.

If the process is delegated to self-executing blockchain code, how can the relationship be terminated? Again, where one party retains the ability to unilaterally terminate a PET, the final outcome is uncertain, and one of the chief benefits of smart contracts is lost.

Too much flexibility will undermine the integrity of the process. On the other hand, too much rigidity might slow adoption of certain smart-contract workflows, especially as transaction value and complexity increases.

A multilateral permissioned mechanism to terminate the smart contract must be considered within the system.

Participants in a smart contract permissioned ledger will also have to consider what happens with the data that sits on the (permanent, immutable) ledger after termination. When building the contract matrix, consider what is "ledgerized", what remains in non-ledgerized participant databases, and what happens to the ledgerized data after contract termination.

If you need advice in this area, please get in touch with our *Emerging*

Technology Group

Calgary
400 - 444 7 AVE SW
Calgary AB T2P 0X8
T 403-260-8500
F 403-264-7084
1-877-260-6515

Edmonton
2500 - 10175 101 ST NW
Edmonton AB T5J 0H3
T 780-423-3003
F 780-428-9329
1-800-222-6479

Yellowknife
601 - 4920 52 ST
Yellowknife NT X1A 3T1
T 867-920-4542
F 867-873-4790
1-800-753-1294

Manage Preferences

FIELD LAW
▲▲
Because Clarity Matters®