# Why smart contracts need shrewder people

**"I've forgotten – woe is me! What the magic word may be."**

*Johann Wolfgang von Goethe, The Sorcerer's Apprentice (1797)*

By Professor Michael Mainelli, Z/Yen Group and Bob McDowall, States of Alderney

## How smart is your code?

So-called "smart contracts" are science fiction realised. Executable pieces of code stored on a mutual distributed ledger for future execution bind people and payments to actions and outcomes. Computer scientist Nick Szabo has promoted the term since the early 1990s and describes bringing the "highly evolved" practices of contract law to the design of electronic commerce protocols between strangers on the internet.

Interest in the term "smart contract" has risen recently in line with interest in Bitcoin and blockchain. Blockchains are immutable distributed ledgers that can store pieces of executable code for future execution. The many potential applications range from betting contracts to digital rights management, loans, or even "smart bonds" and "smart equities". Claims have been made that virtually all of finance can move to smart contracts when combined with an appropriate payment system, often a cryptocurrency.

Here we draw a distinction between smart contracts and "dumb code". Computer code can be extremely stupid. Bits of code are Sorcerer's Apprentices, doing what we thought was our bidding at the time we wrote the code, oblivious to changes in our intentions. Author Larry Niven explains: "That's the thing about people who think they hate computers. What they really hate is lousy programmers."

Code embedded in mutual distributed ledgers is real and useful, and sometimes dangerous. If that code is written to emulate real world contracts it should perhaps more appropriately be called a "code contract".

Smart contracts have a number of parts. The transactions must involve more than the mere transfer of a virtual currency from one person to another (i.e. a payment transfer) and involve two or more parties (as every contract must). Most critically, the implementation of the contract requires no direct human involvement after the smart contract has been made a part of the distributed ledger, which makes these contracts "smart", or autonomous. The code automates the "what if this happens" element of traditional contracts. Computer code behaves in expected ways without the linguistic ambiguity of human languages. The code is replicated on many computers and run by the network when events require it, typically the expiry of some time period.

"Sprite" is an old term for using more traditional coding languages to achieve similar ends. They are effectively little "ghosts" or "geists" that act autonomously. Three decades ago, sprites were commonly used to integrate graphics into video games. Sprites are still found on navigation buttons or adding visual appeal to web pages.

While smart contract coding languages try to "hobble" the code to ensure no unintended consequences, sprite coders are trying to use the power of traditional coding languages to release their potential, relying on control by coders who are presumed to be smarter than the code they let loose. Sprites are no more than code placed into a distributed, immutable data structure, and can be run from pieces of Python, Lisp or Go languages embedded in, and recursively writing to, a blockchain.  Sprites are often used to perform simple security functions such as key and password structures, reading and writing directly to their ledger.

Smart contracts, unlike sprites, tend to use specialist architecture. A number of programming languages and virtual machine software engines have been developed to ensure that smart contract code runs in a secure way.

Ethereum is a particularly popular a cryptocurrency and blockchain platform with programmes and protocols to facilitate the automated performance of a contract. Ethereum, though perhaps the strongest proponent of smart contract enabled blockchains, is moving away from using "smart contract" indiscriminately for pieces of code, towards using the term only when the code is directed at contemporary legal issues.

Ethereum plans to use its blockchain to conduct cryptocurrency transactions of some complexity. Augur is a decentralised, open-source prediction market platform built on the Ethereum blockchain for prediction markets. Ethereum hopes to satisfy complex contracts in areas such as betting, mortgages, and insurance. In theory, platforms can be created which enable financial firms to create programmable versions of traditional securities – "smart securities" – stored in a distributed ledger.

> *Code embedded in mutual distributed ledgers is real and useful, and sometimes dangerous.*

The benefits seem obvious. Faster and cheaper bureaucracy and administration. Fewer errors and disputes. The amount of paperwork to support transactions should diminish. Routine transaction processing jobs will disappear. Many mid-level jobs with routine judgemental tasks and controls can be automated. Securities issuance, transfer and tracking should be streamlined using unique identifiers and asset segregation through securities clearing and settlement. Asset servicing, allocating dividends and interest payments as well as corporate actions processing, should be automated.  Derivatives clearing might move to smart contracts.

## Contract limitations – information and time

We see two particular problems in the extreme smart contract scenarios, data sources and deposits. First, smart contracts of substance rely on external data sources of many kinds, ranging from Libor to FX rates to interest rates, to meteorological information.

Smart contracts have been proposed, for example, to handle prediction markets on US elections. Have the programmers forgotten "hanging chads"? Is there some "ticker tape" of US Supreme Court decisions the programme can access to decide who won the bet on the US election? No. So these types of programs are not self-contained. They rely on outside information and some of this outside information can be unreliable. If any market gets big enough, it is worth "gaming" – remember Libor and FX scandals? Data sources can be unreliable for more prosaic reasons, such as a meteorological station being out of action. What does a smart contract do then? Wait? Infill, i.e. guess? Revert to human intervention?

So surely it's great to move financial structures into code on blockchains?

Parties can see clearly what they are committing to, and let the code decide outcomes and run when it determines it needs to.

In reality, the mathematical and academic disciplines of provable code are in their infancy.

Further, these structures are computationally expensive and complex. The financial services industry needs to be able to articulate why a decentralised system of data storage and computation is worth the additional cost and complexity.

## Many ways to "go legal"

Traditionally, it has been more efficient and cost effective for one organisation to act centrally as a "trusted third party" running the storage and computing platform in a formal or informal "club" arrangement. Customers or members could log in, strike a bargain and rely on the trusted third party to validate the bargain and assets, safeguard the transactions and preserve the transaction records. Sometimes the trusted third party enforces arrangements, sometimes enforcement is left to the legal system.

But "the legal system" is diverse. As well as litigation, there are many other means of resolving disputes, for example, expert determination where an independent third party makes a final and binding determination in a dispute, often used in contracts that require a valuation or technical assessment of who did what how well.

Mediation is a "without prejudice" process that helps both parties reach a resolution yet often takes into account how a court might interpret the situation.

Arbitration is dispute resolution by a private third party, effectively a private court, often needed in complex international situations or where the parties favour speedy resolution.

This diversity within the "legal system" reflects the many different ways commerce can go wrong and the need for a variety of ways to put things back on track.

Proof or guarantee of execution is not possible under some business models or transaction forms. For example, no guarantee of execution is possible where execution is dependent on service levels or variable fee rates. Such models beg the question how do I know the code will do what it says it can do. And when the code isn't doing what I wanted it to do, how do I stop it, and if needed, move the problem into the shrewd hands of experts, mediators, arbitrators, and lawyers.

> *The financial services industry needs to be able to articulate why a decentralised system of data storage and computation is worth the additional cost and complexity.*

Contracts over long periods of time that have material payment considerations may need to hold money in escrow, "on deposit". This limits "liquidity", i.e. lots of money winds up being held on account and unusable. This can be solved by creating netting and insurance vehicles but then we've gone round in a circle and recreated the central financial third parties we were supposedly disintermediating. There are some interesting ways of addressing these issues, but financial services and technologists are at an early stage of exploring these.

## Not so fast

So what do we predict? At least in the near term "dumb and short term contracts" will prevail over "smart long term contracts", for three reasons.

First, if an executable contract has a life of a day or so, then the mutual distributed ledger is not open to long-term sabotage or disruption.

Second, most realistic smart contracts seem to rely on the existence of persistent external data sources, which means the contracts become complicated quickly, or wind up relying on human intervention, rather defeating their purpose. So contracts that depend only on the ledger and perhaps a timing source have an advantage.

Third, smart contracts that involve payments requiring the posting of collateral are going to be seriously restricted. Locking-up collateral would lead to a serious reduction in leverage and pull liquidity out of markets. Markets might become more stable, but the significant reduction in leverage and consequent market decline would be strongly resisted by market participants.

Radical innovation, overturning the accepted order of business models and processes, needs to be tested over time. The starting point is simple experimentation focused on application of smart contracts to simple tasks and processes. Simple tasks and processes will limit information dependencies and financial, reputational, and operational risks.

Once smart contracts are let loose in a commercial world, it is difficult to rein them in without considerable risk cost and embarrassment to say the least. Even simple tasks should be confined simple near-term transactions.

In the financial sector that means leaving long-term financial instruments such as swaps and most bonds till much later, when there is stronger empirical proof that such contracts can be reliably written.

In practical terms, that means a focus on simple tasks, such as security keys or timestamping or archiving, with simple ledger interactions and data dependencies limited to narrow, reliable data sources, perhaps the ledger itself and some universal time clock.

For the immediate future, a "get out of smart contract clause" will invoke human intervention. As well as "legal jurisdiction", "human intervention" will need to be "written into" so-called smart contracts for the foreseeable future.

Contracts that require human intervention or intermediation by way of arbitration, mediation, or expert determination, will be unsuitable for smart contracts for some time.

Some examples of application areas that might be suitable now include:

- **Trading ownership of digital assets in self-referential or token-based online marketplaces**

Ownership of digital property over the internet can be established in a peer-to-peer decentralised environment. This environment extends to pre-sale tokens representing ownership of tickets, merchandise, products and subscriptions. Smart contracts or sprites may find their "smartest" application areas here in the near term.

- **Trading voting rights**

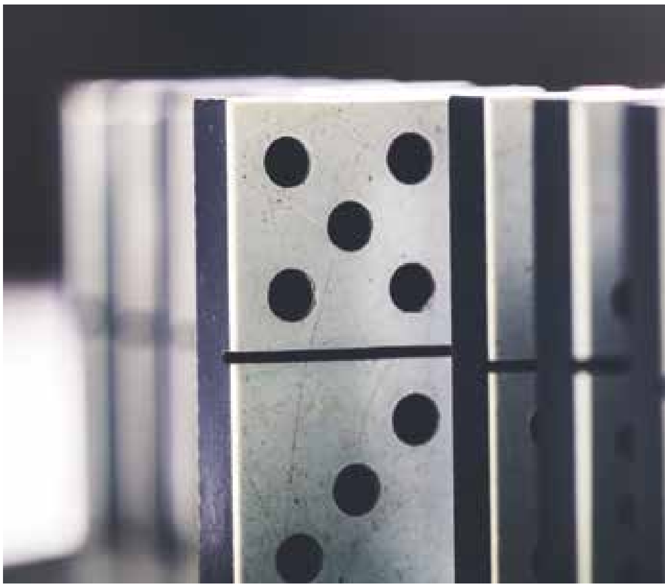
Corporate and social enterprises and even political parties have proposed the creation of blockchain-based systems to build a fairer and more transparent voting environments.

Nasdaq will test blockchain technology to "better manage and streamline the proxy voting process." Smart contracts or sprites could provide in significantly better corporate governance as institutional agents (pension funds) distribute voting rights to the ultimate beneficiaries (pensioners) to hold corporate management to account.

- **Identity management**

Smart contracts or sprites can help know your customer (KYC), anti-money laundering (AML), ultimate beneficial ownership, or health information applications use mutual distributed ledgers to transmit authenticated or notarised documents, recording their use, and structuring key management.

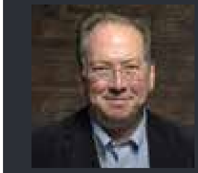
Blockchain underpins smart contracts because the implementation of the contract requires no direct human involvement after the smart contract has been made a part of the blockchain, which makes these contracts "smart". Until there has been much more extensive experimentation with smart contracts, for practical commercial purposes they should be confined to a restricted set of short-term digital transactions. *BT*



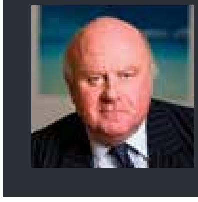
**"Be thou as thou wert before! Until I, the real master call thee forth to serve once more!"**

*Johann Wolfgang von Goethe, The Sorcerer's Apprentice (1797)*

## ABOUT THE AUTHORS



Professor Michael Mainelli is executive chairman of Z/Yen Group and principal advisor to Long Finance. His latest book, *The Price of Fish: A New Approach to Wicked Economics and Better Decisions*, written with Ian Harris, won the 2012 Independent Publisher Book Awards Finance, Investment & Economics Gold Prize.



Bob McDowall is an Associate of Z/Yen and additionally chairman of the policy and finance committee, States of Alderney, Channel Islands.