Smart Contracts – Dumb Idea

Kieron O'Hara • *University of Southampton*

In the summer of 2016, the world of distributed public ledgers, blockchains, cryptocurrencies, and trustless trust was agog as a US$55 million hack unfolded on the Ethereum platform. Like a slow-motion car crash, it piqued interest for a few days, taking everyone's minds off the impending Brexit vote and the unedifying US election campaign. In the end, the hack was thwarted, but the money's final resting place is far less important than the issues it raises for the digital citizenry. Let's explore some of those issues here.

Smart Contracts

Techies like to develop solutions for problems that nobody ever noticed.[1] The rationale, in so far as there is one, is that a messy, scruffy real world needs to be tidied up by code, because tidiness is a virtue.

In *The Laws* (written about 350 BC), Plato wrote a little about contracts in a way that implied that they were a well-understood and familiar part of contemporary civic life in Athens. He talked about what might happen if a man (probably only men could undertake contracts then) reneged on a contract, and who should arbitrate between the disputing parties. He also suggested that not all contracts were valid — for example, where someone contracted to do something illegal, where one of the parties consented under pressure, or where the failure to carry out the contract wasn't the fault of the party concerned. He didn't go into much detail, but enough to indicate that the ancient Greeks had organized their affairs via contracts quite well for some time and didn't require the gods (not even Silicon Valley ones) to adjudicate, thank you very much.

Fast forward to the 1990s, when it began to be argued that e-commerce had created a need for contracts to be smarter.[2] The idea of a smart contract is surprisingly difficult to pin down, but a recent definition is "a mechanism involving digital assets and two or more parties, where some or all of the parties put assets in and assets are automatically redistributed among those parties according to a formula based on certain data that is not known at the time the contract is initiated."[3] By being party to a contract, agents agree to perform (or not perform) certain tasks, and the smartness resides in the use of technology to reduce uncertainty about and transaction costs of that agreement. In the ideal, the mechanism (program) of the smart contract would implement the contract. Neither side could renege, thanks to the remorseless logic-crunching of the machine, whose algorithm would execute, verify, and enforce itself.

The applications no doubt go well beyond contracts — any kind of agreement that requires the parties take on risk during coordinated movement of assets could be rebooted into the 21st century with such mechanisms (including criminal schemes[4]). Trust is unnecessary, because the execution of the agreement is no longer separate from the agreement itself. The code *is* the contract. Its execution is the undertaking.

This should give us pause. Part of the point of contracts, as with other types of constraining commitment such as promises and vows, is to increase trust and spread it more widely than local social mechanisms such as those based on acquaintance, kin, or tribe. Contracts are part of the panoply of means by which we globalize trust. In a society with the rule of law and in which contracts are generally respected, cooperation is easier and less risky (and therefore more common). Thus, it's odd to position a form of contract as reducing the need for trust. Let's

1089-7801/17/$33.00 © 2017 IEEE
Published by the IEEE Computer Society

park this thought for now — we'll return to it.

Building on Smart Contracts

Smart contracts are an old (in digital years) idea, but their implementation became practical with another more recent technology that doesn't trust trust: blockchains. The terms of the contract and the assets to be transferred can be arithmetically coded into the open source, consensus-based, timestamped chain, and its execution verified independently. Neither party can modify the code independently, and so a blockchain-enabled smart contract will simply chug on to its conclusion, no matter how long that takes or indeed whether either party changes its mind about the contract itself.

Now that smart contracts have an implementation mechanism, we can envisage how to build on them. For instance, an organization is basically a group of agents cooperating in the pursuit of some kind of goal or type of practice, and many organizations — private enterprise firms, most obviously — are structured using contracts, both to constrain the roles of owners, members/employees, and outside stakeholders (such as customers), and to assert property rights over buildings and other assets. Make those contracts smart, and you get an organization controlled not by a hierarchy (that's so 20th century, dahling!) of (irrational) humans using (untidy) law, but a more autonomous peer network interacting via the rational, inexorable blockchain protocol.

At least one platform, Ethereum (www.ethereum.org), has been developed with an eye to supporting smart contracts. A rival cryptocurrency to Bitcoin, the ether, underpins the mechanism, to encode the assets and pay for services and fees, but on top of this a more sophisticated scripting language facilitates a wider set of functions. Ethereum, at one stage apparently worth a cool billion dollars, has been growing at Bitcoin's expense as the latter struggles with expanding network capacity.[5]

The Holy Grail is an autonomous organization governed by smart contracts whose operation is as far removed as possible from the day-to-day input of a clique of its managers, members, or owners — a Decentralized Autonomous Organization (DAO). The definition's edges turn out to be fuzzy when you look at the details, but the idea is clear enough; as it's decentralized it lacks a bottleneck or single point of failure, while as it's autonomous it runs itself. A DAO isn't too different in principle from a complex multinational company; firms are often owned by other firms (which count as legal persons), and it can be hard to work out who owns and is responsible for conglomerates such as these. Some economies, such as Italy's or South Korea's, contain deep overlapping networks of cross-ownership. Can the actions of a single company within those economies be identified easily as the responsibility of a particular person or group of people? Not always. In that context, a DAO might seem the next logical step.

DAO Ker-ching!!

The most prominent DAO called itself, with impressive hubris,[6,7] The DAO. The DAO was (note the past tense) a capital investment fund run on the basis of peer-to-peer smart contracts, as well as a home for early adopters keen to show the concept's viability. Investors bought ether coins to join the fund, in which they received a vote proportional to their investment. A candidate for funding would put forward a business plan together with a smart contract to define its relationship with The DAO; investors would vote on whether to support the candidacy. A "no" would mean there would be no relationship; a democratic "yes" would trigger the smart contract, and under the rules that it set, funds would flow.[8] This, the largest crowdfunding campaign in history, raised $150 million in May 2016.

By June, it had collapsed. Although there had been skepticism and some cautionary voices amid the hype, they missed the proximate cause. Some had worried that investment decisions, properly taken, were time-consuming, and so the number of votes cast for each decision might be too small to leverage the wisdom of crowds. Withdrawing uncommitted money was simple to do, and so some commentators were concerned that the $150 million might disappear overnight. Still others wondered how the world's financial system would cope with a company that wasn't registered in any state, and which had no employees. As an article in *The Economist* opined, in the world of cryptocurrencies, faith and rationality go together like yin and yang.[8]

Yet the problem was even more obvious than any of these difficulties. A bug in the code was exploited. The system depended on smart contracts, and if these weren't secure (and research has located vulnerabilities in Ethereum's code[9]), there would be nothing standing between hackers and 150 million smackeroonies. Less than a month after the flurry of publicity, $55 million disappeared in the general direction of who knows who, via a replay attack in which the same transaction was repeated over and again. Fortunately (depending on your point of view) the hack required the money to be siphoned off into a subsidiary bank account where it sat for long enough for Ethereum's coders to devise and implement a hard fork to recover the cash[10] and restore it to the investors (who immediately and wisely took it out of the system, no doubt breathing heavily and mopping sweat from their brows).

Yet was this ethical, or principled? Recall, The DAO was premised on

smart contracts, whereby the code is the contract. The contract therefore couldn't be rescinded, and trust in the system wasn't needed — such was the rhetoric. Yet in the face of a loss that used the code as written, the smart contracts were indeed rewritten. A hard fork bifurcates the blockchain — it's a change in the rules for validating blocks that are the basis for consensus. After the rule changes, the chain diverges into two incompatible chains, one of which follows the new rules and the other clings to the old ways (and, it's hoped, withers away). The relationship between these two branches — and the asset allocations they encode — is nontrivial.[11] Many in the Ethereum community thought this played fast and loose with the ideology behind their innovation (which it does), and preferred to continue on the old fork (christened Classic Ethereum).

The problem is clear: if the code is the contract, then whatever the hackers did was permitted within the contract. The facts that all code is buggy, and that the Ethereum coders clearly didn't intend to license replay attacks, are neither here nor there. It was Ethereum that broke the contract, not the hacker, because The DAO, which held about one ether in seven in circulation, was deemed too big to fail. However, as we discovered in 2008, just because an entity is judged too big to fail, that doesn't mean that it won't fail.

The rules is the rules, except where they isn't. As the great philosopher Mike Tyson once said, "Everyone has a plan till they get punched in the mouth." How stands trustless trust now?

Indeed, if the code is the contract, could the hacker even sue for his or her money back? That would be a juicy case!

The Human Factor

If Ethereum can break its own unbreakable contracts, then the certainty that's supposed to be its unique selling proposition disappears. Yet smart contracts were always highly inadequate types of contracts.

Can you engineer humans out of contracts? Suppose a smart contract — in insurance, for example — is to pay out after a certain event (say, an extreme weather event). If it's distributed across the blockchain, what or who is responsible for retrieving and verifying the meteorological data? Which copy of the contract activates the process? Whenever anything happens in the nondigital world as a result of the smart contract, there will surely need to be a human in the loop if only as a tiebreaker — the trusted third party turns out to be inevitable after all.

This is unsurprising, but there's a deeper point — the notion of a smart contract rests on a fundamental misunderstanding of what a contract is there to do. Contracts aren't mechanisms to make specific things happen. They're social arrangements — voluntary constraints not unlike,[12] but not identical to[13,14] — promises, backed by the machinery of law.

Contracts have a social function (which is why they have legal underpinning). They enable cooperation, and help spread habits of warranted trusting around an economy. Society benefits from regular and reliable exchanges of goods and services; it's hard to imagine the money economy functioning without a contract (money itself is another type of promise). Whereas many kinds of agreements receive support from rich networks of norms (of friendship, or kinship, for instance), parties to contracts often have little in common beyond the contract. Hence the trust-building function is key to the social value of the institution.[15]

Furthermore, the social good of contracts is promoted by the possibility of the contract being broken. That might seem paradoxical, and if the whole point of a contract was that contracted action A_1 be performed by agent X, then X's failure to perform it would be a moral outrage. Yet the law's remedies aren't generally punitive, and don't enforce the performance of A_1. In most jurisdictions all that other parties to the contract can expect to receive from X by way of damages is the expected value of A_1. Furthermore, the law also expects other parties to take steps themselves to minimize the damage caused by X's default.

The result of this is the optimization of the social benefit of X's resources. If X can get better use of her resources by doing something other than A_1, then she's able to, and is better off even after compensating the other parties to the contract (who are no worse off). This is called the doctrine of efficient breach.[16]

Won't this foster a culture of opportunism and betrayal? It hasn't yet; contracts are respected, and the law could always be changed if this ceased to be the case. The point made by supporters of efficient breach is that making the compensation equal to the losses following from the breach of contract encourages efficient breach while discouraging inefficient breach; any more is mere paternalism. Note also that even in this limited sense, a contract still goes beyond a promise; a promisor neither opens herself up to legal scrutiny, nor takes upon herself responsibility for correcting harms done by failure to deliver.[13]

Perhaps most important of all, a contract has built into it the presumption that interpretation and flexibility will be needed, partly to deal with failures to agree on the meanings of particular commitments, partly because of the immense complexity of some contracts (for example, governing major pieces of infrastructure), and partly because things change and both parties might want and expect the contractual terms to evolve over time. The courts can adjudicate here,

and can strike down unfair contracts, such as usurious loan rates or an unjustly one-sided employment arrangement.[17] Contracts are also rarely in one direction; they generally involve reciprocity or exchange, and so have the additional complexity that brings.[15]

There's a balance between the words that each party signs to (the textual interpretation), and what each party wants out of the agreement (the intentional interpretation). Naturally there are arguments for each interpretation, and the courts seek a balance. This isn't a bug, as techies might think, but a feature. On the other hand, there's no way back from the smart contract (other than the hard fork, impractical as a general remedy for obvious reasons) if parties have misunderstood the specification of the code, if the code is poorly written, or if one party has been coerced or misled into taking on an unfair obligation.

There are, no doubt, several important roles that could be filled by smart contracts. In some places, the rule of law may be shaky, or courts may be congested. Routine or short-term commitments might be better served by algorithms than contracts, and we could easily imagine arrangements within an organization being made using a blockchain. Intra-entity trust is a less-pressing issue than inter-entity trust, and so the inflexibility characteristic of the smart contract is less likely to cause long-term problems if used to allocate resources within a single organization. There's also more likely to be agreement about terms.

But smart contracts are dumb contracts, and the best contracts are fallible and human. In his influential book *Code*, Lawrence Lessig drew our attention to various means of constraining human behavior — regulations, social norms, economic incentives, and code or architecture.[18] This important argument has resonated in many contexts, and has been a key axiom for 21st century digital politics.

However, it also has led to a dangerous fallacy. Just because we can imagine different types of mechanisms being used to constrain behavior, it doesn't follow, as many assume, that these mechanisms are interchangeable. It makes an enormous difference if we replace a legal constraint with software.

In the first place, the law can be challenged, whereas in software the forbidden option is irreversibly grayed out and inaccessible. Second, the law is developed and administered transparently by our democratically elected representatives and the courts; software development, even open source, is opaque, and concentrated in a small programming community, many of whom are in the pay of a few oligopolistic corporations directly accountable to no external party.

Third — and most important from my point of view — we can break the law. There are consequences when we do, and the system would break down if we disobeyed it all the time. But the law is consistent with maximal liberty; it can't compel obedience (though it can disincentivize it). Code, on the other hand, won't allow behavior inconsistent with itself.

With smart contracts, this rules out the desirable economic gain of efficient breach. But in the wider context, the law's openness to breach allows many vital liberties, not least of which is the civil disobedience that helped shape our civilized liberal order from Thoreau to Gandhi to King. The next time we consider replacing law with code to tidy up a scruffy world, let's remember that.

Acknowledgments

This work is partly supported by SOCIAM: The Theory and Practice of Social Machines, funded by the UK Engineering and Physical Sciences Research Council (EPSRC) under grant EP/J017728/2.

References

1. E. Morozov, *To Save Everything, Click Here: Technology, Solutionism, and the Urge to Fix Problems That Don't Exist*, Perseus, 2013.
2. N. Szabo, "Formalizing and Securing Relationships on Public Networks," *First Monday*, vol. 2, no. 9, 1997; http://firstmonday.org/ojs/index.php/fm/article/view/548.
3. V. Buterin, "DAOs, DACs, DAs and More: An Incomplete Terminology Guide," *Ethereum Blog*, 6 May 2014; https://blog.ethereum.org/2014/05/06/daos-dacs-das-and-more-an-incomplete-terminology-guide.
4. A. Juels, A. Kosba, and E. Shi, "The Ring of Gyges: Using Smart Contracts for Crime," *Proc. 2016 ACM SIGSAC Conf. Computer and Comm. Security*, 2016, pp. 283–295; http://initc3.org/files/Gyges.pdf.
5. T.B. Lee, "Ethereum, Explained: Why Bitcoin's Stranger Cousin Is Now Worth $1 Billion," *Vox*, 24 May, 2016; www.vox.com/2016/5/24/11718436/ethereum-the-dao-bitcoin.
6. H. Green, "Introducing the DAO: The Organisation That Will Kill Corporations," *City A.M.*, 3 May 2016; www.cityam.com/240198/introducing-the-dao-the-organisation-that-will-kill-corporations.
7. A. Cleary, "Here Is How the DAO Will Soon Become the Greatest Threat Banks Have Ever Faced," *Frontera*, 25 May 2016; https://fronteranews.com/news/dao-will-soon-become-greatest-threat-banks-ever-face.
8. "The DAO of Accrue," *The Economist*, 21 May 2016; www.economist.com/news/finance-and-economics/21699159-new-automated-investment-fund-has-attracted-stacks-digital-money-dao.
9. L. Luu et al., "Making Smart Contracts Smarter," *Proc. 2016 ACM Sigsac Conf. Computer and Comm. Security*, 2016, pp. 254–269; www.comp.nus.edu.sg/~loiluu/papers/oyente.pdf.
10. V. Buterin, "Hard Fork Completed," *Ethereum Blog*, 20 July 2016; https://blog.ethereum.org/2016/07/20/hard-fork-completed.
11. T. Rapp, "How to Deal with the Ethereum Replay Attack," *Medium.com*, 17 July 2016;

https://medium.com/@timonrapp/how-to-deal-with-the-ethereum-replay-attack-3fd44074a6d8#.vu14sdqyl.
12. C. Fried, *Contract as Promise: A Theory of Contractual Obligation*, 2nd ed., Oxford Univ. Press, 2015.
13. R.E. Barnett, "Contract Is Not Promise: Contract Is Consent," *Suffolk Univ. Law Rev.*, vol. 45, no. 3, 2012; http://suffolklawreview.org/barnett-consent/.
14. S.V. Shiffrin, "The Divergence of Contract and Promise," *Harvard Law Rev.*, vol. 210, 2007, pp. 709–753.
15. D. Kimel, "Personal Autonomy and Change of Mind in Promise and in Contract," *Philosophical Foundations of Contract Law*, G. Klass, G. Letsas, and P. Saprai, eds., Oxford Univ. Press, 2014, pp. 96–115.
16. R.A. Posner, *The Economic Analysis of Law*, 9th ed., Wolters Kluwer Law & Business, 2014.
17. S.V. Shiffrin, "Paternalism, Unconscionability Doctrine and Accommodation," *Philosophy and Public Law*, vol. 29, no. 3, 2000, pp. 205–250.
18. L. Lessig, *Code and Other Laws of Cyberspace*, Basic Books, 1999.

Kieron O'Hara is a senior lecturer and a principal research fellow in the Web and Internet Science Group in the Electronics and Computer Science Department at the University of Southampton. His research interests include trust, privacy, open data, and Web science. O'Hara has a DPhil in philosophy from the University of Oxford. Contact him at kmo@ecs.soton.ac.uk.