Do we need programmable money?

Programmable money can be designed to flow as easily as email without sacrificing regulatory controls, monetary policy or personal privacy. – IBM 2018

If the US dollar is to remain the world's primary reserve currency in the unfolding digital century, how can it remain an analog instrument and unit of account for things increasingly programmable and denominated as digital tokens?– Digital Dollar White Paper 2020

Contrary to the beliefs of many fintech enthusiasts, we already live in a world of "digital money". The most common forms of money exist only as records in the computers of commercial and central banks. Those systems make decisions about money, whether to move it, how much to move and whether conditions required for its movement have been met e.g. sufficient funds are available to pay a standing order. Even when a human presses the button, the mechanics are mostly carried out by computers. So why the growing interest in "programmable money" from central banks, financial and technology firms?

Robert Sams (founder and CEO of Clearmatics, a firm building decentralised market infrastructure) identifies two distinct definitions for programmable money. "In non-crypto fintech, 'programmable money' seems to refer to leveraging open banking APIs to build new, automated use-cases over legacy bank payment infrastructure but in our crypto world, the 'programmable money' refers to digital cash hosted on a blockchain, where cash can be placed under the control of a smart contract."

"Open banking APIs" relates to two overlapping concepts, 'open banking' and 'application programme interfaces (API)'. Open banking consists of a series of regulatory driven initiatives (most prominently associated with the European Union's second Payment Services Directive PSD2, which came into force in 2018) intended to increase competition in financial services. Notably by making banks share customer information with other providers of financial services, if requested by those customers. An API is a technical concept that dates back decades. It is a software layer created around a system to allow other systems (both within the organisation and from outside) to access the system's functionality and data. Access to APIs is strictly controlled in the financial sector and having an API does not mean external systems can access *all* data or functionality. An API allows permissioned systems to retrieve data, update data and/or initiate the operation of a business logic e.g. marking a payment. APIs can be built that go far beyond the requirements of open banking legislation, for instance a bank may allow third party apps to make (and act on) decisions regarding a customer's savings, investment or level of insurance.

Date originally posted: 2020-09-14

Permalink: https://blogs.lse.ac.uk/businessreview/2020/09/14/do-we-need-programmable-money/

Blog homepage: https://blogs.lse.ac.uk/businessreview/

Citigroup payments expert Tony McLaughlin, in the paper "*Modernizing payments: tokens or accounts?*", expresses high hopes for the open banking/API model. "It could be argued that APIs, rather than tokens are the route to a programmable financial system, and a programmable digital economy more broadly….A great wave of innovation would be unleashed were it possible to access the banking system through standardised, secure APIs." However, in the same paper he recognises the commercial and organisational bottlenecks. Many banks treat open banking as a regulatory burden and make no effort to go beyond providing minimal APIs. Other banks worry that if they give more control of a customer's funds to third parties it will reduce their own revenues streams.

Understanding the cryptocurrency model of programmable money is somewhat more complicated. In the world of cryptocurrencies (and other security like "tokens" built on blockchain technology) it is possible for funds to be held by people, organisations or computer programs. Depending on the type of blockchain, that computer program (usually referred to as a smart contract) can make decisions about what to do with funds or how to respond to requests from others relating to the funds. In general, in blockchain platforms such as Ethereum, two things make it possible for smart contracts to have complete control over the funds. One is that the most common "public" blockchains are "de-centralised". No single party (and definitely no regulator) has control over transaction processing, operation of the system, or the code that is run on the system. With no outside party able to control the system no one can stop or reverse transactions generated by a smart contract.

Another factor is that cryptocurrencies are not like conventional digital money such as funds deposited in banks (commercial bank money) or by banks at central banks (central bank reserves). Conventional digital money is based on the concept of double entry accounting. There is no discrete pile of digital banknotes assigned to each customer. Funds deposited represent a liability (funds owed) by the bank that are balanced by the bank's assets i.e. claims on others such as loans made and bonds owned. Cryptocurrencies lack a connection to any assets of value (not any significant usage apart from speculation and some specialised areas of crime). This means that cryptocurrencies can literally be "locked up" for prolonged periods with no real impact on the real world. Real world money cannot genuinely be locked up by banks. For funds in a bank to be accessible to customers, banks have to constantly work to make sure assets are worth more than liabilities (i.e. they are solvent) and that they have sufficient assets such as central bank reserves that are acceptable to fulfil obligations to other banks (i.e. they are sufficiently liquid).

These unique features have allowed the rapid creation for a large and complex crypto ecosystem where smart contracts have a high degree of control. Arguably this ecosystem has not worked very well. The first large scale investment strategy run by a smart contract, the distributed autonomous organisation (DAO), ended in major fraud. A wave of security-like "token" issuances in the initial coin offering (ICO) craze led to large scale waste and even larger frauds. More recently a complex series of pyramid scheme-like investments called DeFi (decentralised finance) have grown up. Described by author David Gerard as "a worked example of the hazards of programmable money — incomprehensible financial derivative instruments, bots front-running everything a human does, hackers stealing everyone's money through badly-written code."

Still the potential to create systems based on smart contracts, but operating within the conventional financial system, has created a great deal of interest. However working within the existing framework essentially means throwing away many of the key features of distributed ledger technology/smart contract based systems.

Table. Features of distributed ledger technology/smart contract-based systems

Date originally posted: 2020-09-14

Permalink: https://blogs.lse.ac.uk/businessreview/2020/09/14/do-we-need-programmable-money/

Blog homepage: https://blogs.lse.ac.uk/businessreview/

Feature	Incompatibility with Real
Anonymous bearer assets	The long running trend in finance has bene to restrict the use of physical cash because of evasion and money laundering.
Lack of centralised control over the operation of the system	In general regulators want clear accountabili infrastructure. Including the ability to check c
Censorship Resistance (transactions cannot be stopped or reverse)	The ability to stop or even reverse transacti both banks and law enforcement. Notably in heist.
Nobody can stop the operation of a smart contracts	As demonstrated in the 2016 DAO hack, it is stop the operation of a buggy smart contract
Use of "native" cryptocurrencies that are not linked to real world bank accounts	Cryptocurrencies are highly prone to mani excessive volatility. Most regulators and centr them.
Visibility of all transactions to all participants	Public blockchains (and even many private b of transactions visible to all parties with acce

Even after these elements have been discarded there is still the challenge of finding the right kind of real world money to make "programmable". Commercial bank money has major limitations because it is economically equivalent to a deposit at a bank. This may be acceptable for individuals or companies but not if a smart contract is managing flows of money between banks. Modern financial systems create the illusion that a pound or dollar in a central bank reserve account, a bank with high credit rating and a bank with lower credit rating are equal in value. However this "illusion of fungiblity" is created by the design of modern payment systems including the willingness of central banks to act as last resort provider of liquidity. If a payment from a smart contract involves paying Bank A funds from Bank B (by effectively increasing Bank B's debt to Bank A) it would rapidly run into the credit limits banks have in place with each other.

Using central bank reserves as a basis for programmable money avoids these problems and some central banks have shown interest in issuing forms of digital money that is accessible to a wider range of parties than central bank reserves. Generally this is referred to as central bank digital currency (CBDC). CBDC can be based on conventional or distributed ledger technology. CBDC using a conventional technology does not really make life any easier for those building programmable money than integrating with existing central bank systems. CBDC based on distributed ledger technology or blockchain tokens issued by a third party but backed by central bank reserves (a stable coin) are potentially more useful for creating programmable money. However even this is problematic because the viability of programmable money depends on the blessing of central banks. There are also potential liquidity problems with locking up funds for a prolonged period inside a smart contract.

Even if these problems for the various forms of programmable money can be overcome, it still leaves the question of what is programmable money for? Robert Sams points to the general potential for innovation, "More likely are the use-cases that don't even exist today and can't exist without programmable money. Use-cases where the contractual form of the deal is changed due to the capabilities of programmable money." Aleksi Grym (head of digitalisation at the Bank of Finland) has a less optimistic view. "Generally, I'm not a fan of new words for old concepts, so in this case I'm asking myself, what would a normal person call 'programmable money'? I think the answer is 'conditional payment'."

Date originally posted: 2020-09-14

Permalink: https://blogs.lse.ac.uk/businessreview/2020/09/14/do-we-need-programmable-money/

Blog homepage: https://blogs.lse.ac.uk/businessreview/

Overall there probably is scope to create more mechanisms for adding more conditionality in the financial system, locking up funds until an event happens or creating more easily accessible escrow arrangements. Whether this requires the increased adoption of digital ledger technology, smart contracts and tokenisation is far from clear. What is clear from the experiences of initial coin offerings and decentralised finance, is that making an already complex financial system harder to understand and control would not be such a good idea.

Notes:

- *This blog post expresses the views of its author(s), not the position of LSE Business Review or the London School of Economics.*
- *Featured image by Bermix Studio on Unsplash*
- *When you leave a comment, you're agreeing to our Comment Policy*

Martin Walker is director of banking and finance at the Center for Evidence-Based Management. He is the author of the book *Front-to-Back: Designing and Changing Trade Processing Infrastructure* and contributed to the book *Evidence-Based Management: How to Use Evidence to Make Better Organizational Decisions*. He has also published several papers on banking technology. Previous roles include global head of securities finance IT at Dresdner Kleinwort and global head of prime brokerage technology at RBS Markets. He has been asked to provide evidence to two parliamentary inquiries, "Digital Currencies" and "IT Failures in the Financial Services Sector". He received his master's degree in computing science from Imperial College, London and his bachelor's degree in economics from LSE.

Date originally posted: 2020-09-14

Permalink: https://blogs.lse.ac.uk/businessreview/2020/09/14/do-we-need-programmable-money/

Blog homepage: https://blogs.lse.ac.uk/businessreview/