Blockchain: a global supercomputer that only answers to code?

Written by

Andrei Kirilenko

Published

6 June 2017

Category

Finance

Key topics

Cryptocurrency, Finance

Dr. Andrei Kirilenko, Director of the Centre for Global Finance and Technology, highlights how object-orientated programming bypasses distrust

A blockchain is usually described as a distributed ledger: a database of records shared by all clients with access. This description is not incorrect, but it leaves a lot to be desired. It's like calling a car a "horseless carriage", or a handheld personal computer a "smartphone".

A blockchain is really a computer: a finite-state machine. Currently, it is not a very good computer. It is very slow – it takes minutes to complete a change of state. It is not exact – the change of state is probabilistic. It is also very expensive – it uses a lot of power to complete a change of state. Yet, it is a truly global computer that does not reside in any particular physical or virtual machine. And – crucially – it allows anonymous users to share their private computing power and memory capacity for a reward.

Many such computers currently exist: Bitcoin, Ethereum, and Ripple, to name a few. Reminiscent of the earliest computers, such as ENIAC (Electronic Numerical Integrator and Computer), they are not very good finite-state machines; they are more like calculators than computers. The biggest problem with ENIAC was that it was not designed to store both data and program in memory, so any additional operation had to start with reloading the original data along with the execution code.

> One of the definitions of trust is a “firm belief in the reliability, truth, or ability of someone or something”. This is exactly what financial institutions have lost after asking to be rescued with hundreds of billions of taxpayers’ money, but what objected-oriented programming provides by its very design

In contrast, a blockchain relies on the brilliant concept of object-oriented programming. The main premise of object-oriented programming is that both data and execution code are stored together in the same place, which is called an object. For example, an object in Bitcoin or Ethereum blockchain consists of data (user profile) and code (commands to send and receive payments). And – most importantly – objects have IDs, but once they are created, what’s inside them can be kept completely anonymous and immutable – a black box. Originally this was done to reduce the number of bugs in the code, so an object could not be mistakenly altered by a sloppy programmer.

But what engineers have created to keep the code reliable had been rediscovered after the global financial crisis in a much more general incarnation – trust. One of the definitions of trust is a “firm belief in the reliability, truth, or ability of someone or something”. This is exactly what financial institutions have lost after asking to be rescued with hundreds of billions of taxpayers’ money, but what objected-oriented programming provides by its very design.

> The trust in a blockchain comes from its native object-oriented architecture. Users inside objects can remain anonymous; they do not need to know or “trust” each other

So, how does it work? Once an object is defined, it is only allowed to do things that are defined by its communication interface – messages that it can send to or receive

from other objects. For Bitcoin or Ethereum blockchains, messages are transactions. Transactions involve the exchange of “value” quantified in cryptotokens or cryptocoins like, for example, Bitcoin. Transactions then become available for “mining.” Mining involves solving a crypto “hash” puzzle – a Sudoku-type exercise that’s very difficult to complete (that’s why it takes minutes), but easy to check. Thousands of competitive miners use “gigahashes” of computing effort to brute-force their way through these puzzles in the expectation of getting valuable rewards. Mining a single transaction is typically not rewarding enough, so each miner composes several transactions into a block and then mines that block.

As miners compete, often several of them arrive at a solution at about the same time. When that happens, there is a need to achieve consensus among the miners about who exactly mined the last block and, thus, who gets to keep the reward. This consensus protocol is also what makes the change of state probabilistic – the state of a blockchain gets modified as another block is “sealed” and attached to the chain, but which miner’s solution to the crypto hash puzzle – and, thus, which transactions end up being included in the next block – is not known in advance. This is why blockchain is a probabilistic finite-state machine.

> Does blockchain have the potential to fundamentally change computer-based economic interactions by connecting buyers and sellers of computing resources?

Now you can see how the trust in a blockchain comes from its native object-oriented architecture. Users inside objects can remain anonymous; they do not need to know or “trust” each other. Transactions between them only execute if it is confirmed that the users possess the funds that they claim they do. There is no need for “trusted” gatekeepers, validators or reconcilers. The processing of transactions and the validation of blocks is outsourced to a distributed network of fiercely competitive miners, who hack their way through crypto hash puzzles to chain blocks together for a reward, which they receive in the form of crypto tokens – irrespective of the identities of the users.

To sum up, a blockchain is a quite trustworthy, but not a very good, computer. ENIAC was also not a very good computer in 1946; yet we know how fundamentally computers have changed the world since. So, the question is: Does blockchain have the potential to fundamentally change computer-based economic interactions by

connecting buyers and sellers of computing resources?

My colleagues and I at the Centre for Global Finance of Technology at Imperial Business School are working on answering this trillion-dollar question. As financial economists, we think that the answer boils down to designing a blockchain that minimises net operating cost – while also creating flexible reward incentives that maximise the use of distributed computing resources. We think it exists. Let’s find it.

Written by

Andrei Kirilenko

Published

6 June 2017

Category

Finance

Key topics

Cryptocurrency, Finance

Share

About Andrei Kirilenko

Senior Research Fellow and Director, Centre of Global Finance and Technology
Andrei Kirilenko was the Director of the Centre for Global Finance and Technology, a Senior Research Fellow at the Brevan Howard Centre for Financial Analysis, and Visiting Professor of Finance.

Monthly newsletter

Receive the latest insights from Imperial Business School

Sign up now

Bankers knew the risks they were taking before the 2008 crisis

Top executives of US banks that experienced huge losses in the 2008 crisis sold their own shares well before the crisis hit

Read more