ANDRÉ FEITEIRO

HOW USEFUL ARE SMART CONTRACTS?

The mainstream interest in the topic grew with popularity of blockchain technology and cryptocurrencies. Interest, however, often leads to misconceptions and, as regards to smart contracts, it contributed to make a sweeping assumption that smart contracts are more than code that reads and writes on a blockchain.

The mainstream interest in the topic grew with popularity of blockchain technology and cryptocurrencies. Interest, however, often leads to misconceptions and, as regards to smart contracts, it contributed to make a sweeping assumption that smart contracts are more than code that reads and writes on a blockchain.

Distributed ledger technologies, and blockchain specifically, are as worthy to the provision of a service as the efficiencies – in number or quality – they offer to the provision of such service. Smart contracts are as useful as the simplicity that they bring to the table.

Opposing a sizeable optimism on the applicability of smart contracts blindly to all industries and services, the relevance of the vulnerabilities of smart contracts is of greater importance than its potential use cases.

Language and trust are two major issues: on the one hand, the language (or semantics) of code is formal and is therefore unable to replicate the flexibility of natural language; on the other, for certain transactions, the use of smart contracts must necessarily trust external sources, which poisons their decentralized and trustless character.

The rigidity of code language is a limitation, as binary code is unable to make a fair judgement on ambiguous terms, those to which one cannot regress into binary code, 0s and 1s, or logic gates, ifs and thens. If we take a contract for a repair service, for example, the performance of the service consists of the repair against a payment agreed between the parties. While the performance or non-performance of the payment obligation is ascertained in a logic gate of yes or no and then, the good performance of the repair requires social ontology elements that code cannot reach.

In general, smart contracts and blockchain technology enable a trustless environment. It is not that trust is missing, it is just that trusting a third party is not necessary, which is very relevant in the case where one eliminates intermediaries. This is true for on-chain transactions, such as a Bitcoin transfer of funds, because data on the Bitcoin price, account addresses, signatures, etc. is already on-chain. This is not the case, however, with off-chain transactions, because regardless of the security, immutability and disintermediation that the smart contract and blockchain technology provide, the data is provided from outside of such a trustless environment.

An optimist will find infinite use cases for smart contracts because he finds them alternative to traditional contracts. A realist understands that the utility of smart contracts lies where (i) transactions are on-chain and they do not require the flexibility of natural language, and (ii) transactions are off-chain, they do not require the flexibility of natural language, and, in addition, they trust – or they do not need to trust – external sources.