# MAKE YOUR CONTRACTS SMART

## THE BASIC RULES FOR CREATING SMART CONTRACTS NOW

*António de Macedo Vitorino*

In 1996, Nick Szabo coined the term "smart contract" in a revolutionary article on the introduction of digital technology in the realm of contracts. In that article, Nick Szabo stated: "[*n*]*ew institutions, and new ways to formalize the relationships that make up these institutions, are now made possible by the digital revolution. I call these new contracts "smart" because they are far more functional than their inanimate paper-based ancestors. No use of artificial intelligence is implied. A smart contract is a set of promises, specified in digital form, including protocols within which the parties perform on these promises*."[1]

Since 1996, the emergence of blockchain technology has allowed the creation of a new form of legal ledgers which are run in a decentralised manner.

There are various classifications of smart contracts and smart legal contracts depending on whether they are fully or partially automated and recorded in a blockchain system or not.[2] In this article, we will use the expression "smart contract" in its simplest and widest form.

We define smart contracts as computer instructions that represent the intention of the parties to create an obligation, make a payment, purchase an asset or service or trigger an event that has a legal consequence.

The primordial example of a smart contract is a vending machine. By inserting a coin in a vending machine, the person who inserted the coin purchases a snack or a beverage from the vending machine owner or operator. Automated ticketing machines are also old automated or self-executing contracts, whereby a person acquires the right to use a transport service, enter a movie theatre etc.

More recently, online contracts with Amazon and the like are also forms of automated purchase agreements that fall into the concept of "smart contract" because they allow clients to purchase goods and services by giving automated instructions through a machine.

In any of the above examples, there is an underlying "natural language" contract, that is, a language used by people and not coded instructions given to a computer. When we enter an online contract, we are often required to accept a standard contract, sometimes in a foreign language that we may not fully understand. This raises issues of the validity of those rules concerning general and standard clauses. We will not be concerned with these implications in this article; we will focus on the issue raised by "software coded contracts", which is the widest possible definition of smart contracts.

"Software coded clauses" or "smart clauses" are computer instructions using software language that may constitute the performance of an obligation or trigger the verification of contract conditions. Software coded obligations are self-executing

[1] Nick Szabo -- Smart Contracts: Building Blocks for Digital Markets (uva.nl).

[2] UK Law Commission - Advice to Government (Smart-legal-contracts-accessible.pdf).

because, in most cases, they do not need human intervention. Software coded contract provisions are now used in all online contracts because every service or product purchased online triggers legal consequences, such as the obligation to pay and the obligation to deliver a service or good.

The entry and performance of online contracts have occurred without disrupting legal systems. Generally, around the world, legislations accept the validity of online contracts. Disputes have been resolved in a fairly satisfactory manner because suppliers that value their clients are ready to solve complaints in a friendly way. Customers dissatisfied with suppliers simply cease to purchase from those suppliers. The rules of supply and demand that govern mature and fair markets take the stress out of the system. However, some terms and conditions and the conduct of many online suppliers may not be legal on many occasions.

The buzz about smart contracts goes much deeper than simple online B2C online agreements, where software coded information is translated into service and purchase orders and money transfer instructions.

Smart clauses coded in self-executing software language can include complex payment terms and formulas, trigger conditions, default clauses and security enforcement provisions. Smart contracts can be used in all types of contracts, including very complex B2B contracts such as credit agreements, mergers and acquisitions, issuance, acquisition and disposal of securities, IPOs, financial derivatives, futures markets, restructuring agreements, construction contracts etc.

Also, matters of technical nature can be linked to the contract clauses through smart clauses, that is, technical parameters stated in legal contracts involving the operation of telecommunication networks, power grids, power production, software requirements etc. can be linked to the legal language of contracts and trigger contract consequences. Presently, many of those technical issues are left in the contract's fringes and put into more vague and elusive contract terms, such as "best efforts" and reasonableness judgements, or left to mediation or litigation when they become unsolvable by the people dealing with the day-to-day operation of the contract.

There are five basic practical rules for ensuring that smart clauses are enforceable and do not raise even worse issues than natural language clauses.

**First, smart clauses should be translated into a natural language**. This means that every smart clause should have an equivalent natural language clause. The natural language clause should be as objective as the smart clause and not include open concepts that cannot be translated into the coded clause.

The need for this rule does not result from the Law but serves a practical need: clauses should be understandable by people without very deep knowledge of the law and the technical aspects of the contract. Judges and business decision-makers should be able to understand the key points of the contract.

Suppose the main clauses of the contract, like the performance of the contract, are dependent on a coded clause (with no natural language equivalent). In that case, it won't be easy to understand why the parties chose that particular solution instead of another. Sure, there are contracts with heavy and complex technical issues, be that the legal formulation of the clause or the commercial and technical aspects. But, in most cases, the economic, commercial and technical terms have a "natural language" formulation inscribed in the agreements and understood by the persons present at the negotiation table, even when they include complex technical schedules.

**Second, the content of smart clauses should be open and auditable**. This means that the acceptance of the code should be made by experts of each of the parties. This requirement aims to give an equal footing to the parties. Smart clauses must be understood and controlled by the parties.

There should not be one party controlling the code and the consequences of the instruction generated by it. In present natural language contracts where the legal and commercial technicalities can be generally understood by the parties, each party should have its own legal advisors. In coded language contracts, an asymmetry in knowledge can be more damaging than not having legal counsel. Expert advice and verification procedures are necessary.

**Third, smart clauses should be protected**. Integrity is a key element of any contract. In natural language contracts, one of the parties cannot change the wording of clauses. This is ensured by putting the contract in writing and other formal requirements imposed by law or agreed by the parties. Tampering with the words of a contract means falsifying the contract content. The same rules apply to smart clauses and smart contracts.

However, because smart clauses are generally self-executing, the consequences of possible tampering with the contract software code are more direct and can unlock a chain of events that may be impossible to stop. For this reason, the integrity of smart clauses should be ensured.

Blockchain systems are an acceptable form of ensuring the integrity of contracts because the blocks of a blockchain cannot be changed without the agreement of the nodes of the system (all or a significant number of participants in the system depending on the type of consensus algorithm that is used). This ensures the integrity of the contract in a manner that is as efficient, if not more, than current government-sponsored or other centralised ledger systems. However, blockchain is not the only way to ensure integrity. The parties may appoint an independent third party to store and protect the code or even monitor its application.

**Four, smart clauses triggering events that require human intervention should not be left to the parties**. While many smart clauses rely on machine self-execution mechanisms, there are cases where human intervention is necessary to fill gaps or interpret the data. The parties should not take this decision.

For instance, if the contract stipulates that one of the parties must make a payment to the other if the temperature reaches 45 degrees and two official computer records state different temperatures, one 44.9 degrees and another 45 degrees, a decision will have to be made as to whether the payment condition was met or not.

This example underpins the fact that minor discrepancies in digital records or the non-existence of an independent digital record may require human intervention to verify or certify smart contracts trigger conditions. When that is the case, the person making that decision should be independent of either party.

In Ethereum, the contract parties may appoint persons, named “oracles”, to take decisions that will trigger or not trigger smart contract events. This is an adequate solution for smart contracts based on Ethereum. For smart contracts outside a blockchain, the parties may use independent entities and grant them the power to fill in the missing data or resolve digital records inconsistencies.

**Five, contracts with embedded smart clauses should include effective and fast dispute resolution mechanisms**. Smart contracts should contain dispute resolution mechanisms because smart clauses can add a layer of complexity for which courts are unprepared.

In existing b2c online contracts, few disputes are resolved in courts because their value is low. The consumer often simply refrains from purchasing from the vendor that failed to deliver. This is not the ideal way of resolving a dispute, and a more effective transnational redress system must be created.

In more valuable contracts, disputes can be taken to courts, but the time for resolution may be too long to remedy the harm suffered by one of the parties. Parties will trust more in contracts that contain safe and expedient means of resolution. These may include mediation and arbitration mechanisms that allow for taking over control over the code, stopping wrongful use of the code, correcting an improper use of the code and repairing or correcting the code that proved not to attain the parties' goals.

Smart contracts are one of the greatest inventions of the turn of the century. They improve efficiency increase speed and quality of delivery.

Embedding smart clauses in natural language contracts and creating adequate dispute resolution systems backed by robust control and verification mechanisms are the first steps to make your contracts smart.

### ABOUT MACEDO VITORINO

MACEDO VITORINO is a leading Portuguese law firm. We advise domestic and foreign clients in a wide range of business sectors, including banking, distribution, industry and projects. We are known for our professional and client-oriented approach to complex and challenging matters.

Since the firm's incorporation in 1996, we have been involved in several high-profile transactions in all of the firm's fields of practice, including banking and finance, capital markets, corporate and M&A, energy, real estate, project finance, and complex disputes restructurings.

### ABOUT **MV**LAB

**MV**LAB is MACEDO VITORINO's program for developing legal technology and the digitalisation of law, legal services, the judicial system, the government and public services.

António Vitorino is a founding partner of MACEDO VITORINO and the partner responsible for **MV**LAB. António also leads the firm's knowledge program «Saber Mais» and its Banking and Capital Markets Group.

**NOTICE**

The opinions expressed in this article are of a general nature and should not be considered as professional advice. Should you need legal advice on these matters, you should contact a lawyer. If you are a client of MACEDO VITORINO, you may contact us by email addressed to mv@macedovitorino.com.