



مجموعة جامعة
الجميع الذكية
للتعليم القابضة

MIDOCEAN
UNIVERSITY

Smart Contracts

by Samara Jasim

LAW614

Lecture 5

A. DEFINITION

A "smart contract" is computer code that automatically does parts or all of an agreement. It's saved on a platform that uses blockchain technology.

A. DEFINITION

The code can be the only agreement between the parties or it can work together with a traditional contract written in text. It can do certain things like moving money from one party to another. The code is copied on many parts of the blockchain, so it benefits from the security, permanence, and unchangeability of the blockchain. Every time a new block is added to the blockchain, the code is executed.

If the parties start a transaction and meet certain conditions, the code will do the actions connected to those conditions. But if there's no transaction, the code won't do anything. Most smart contracts are written in programming languages like Solidity that are made for these types of programs.

The term “smart contract” was first introduced by computer scientist and cryptographer Nick Szabo some 20 years ago as a graduate student at University of Washington.

New 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.

When Szabo compares smart contracts to paper contracts and puts quotes around the word "smart," and when he emphasizes the absence of artificial intelligence, it carries significance. Smart contracts may be considered "smarter" than paper contracts because they can automatically execute predetermined steps. However, they should not be mistaken for intelligent tools that can interpret the more subjective aspects of a contract.

Szabo's classic example of a smart contract is a vending machine. Once a buyer fulfills the conditions of the "contract" (e.g., inserting money into the machine), the machine automatically fulfills the terms of the unwritten agreement and dispenses the snack.

Smart contracts also have their roots in Ricardian Contracts, which were introduced by Ian Grigg and Gary Howland in 1996 as part of their work on the Ricardo payment system for asset transfers. Grigg saw Ricardian Contracts as a link between textual contracts and code, characterized by several parameters.

These parameters include a single document that serves as a contract offered by an issuer to holders for a valuable right, easily readable by both people (like a paper contract) and programs (like a database), digitally signed, carrying keys and server information, and associated with a unique and secure identifier.

A blockchain is a special kind of database that groups records into blocks, similar to organizing them onto a single sheet of paper. Each block is then connected to the next block using a cryptographic signature. This linking process forms a chain of blocks, hence the name "blockchain." The remarkable feature of blockchains is that they can function as a shared ledger, accessible to anyone with the necessary permissions, allowing for verification and validation.

The significance of blockchain technology has also been acknowledged by the International Monetary Fund (IMF). They recognize the potential benefits of virtual currencies, such as faster and more efficient payments and transfers. The IMF states that the distributed ledger technology that underlies some virtual currency schemes offers advantages that extend beyond the currencies themselves.

By implementing blockchain technology in smart contracts, these contracts become self-executing and self-enforcing, eliminating the need for intermediaries. Additionally, every transaction carried out within blockchain-based smart contracts is automatically recorded in the distributed database. As a result, these smart contracts can be referred to as "decentralized smart contracts" since there is no central database or register governing them.

When automating certain parts of a legal agreement using computers, it involves writing code that needs to be verified and validated. In this context, "verified" refers to an internal process where the computer code is checked for technical accuracy (is the programmer building the code correctly?). On the other hand, "validation" is an external process where previously verified code is examined to ensure it aligns with the agreement (is the programmer building the right code?).

A computer can only execute instructions in an executable language consisting of sequences of binary digits (0 and 1), which is very different from the human-readable code produced by programmers. The conversion from human-readable code to executable code goes through several intermediate layers in the language stack, each with its specialized language(s).

Errors and misunderstandings can occur during the conversion process as the code goes through the layers of the language stack until it becomes the final bits that control the computer. It's important to understand why and how these errors and misunderstandings can happen.

When considering legal documentation, the language stack begins with the natural language used in the written documentation, rather than the code written by the programmer:

- ***Natural language:*** At the top of the language stack is the natural language (e.g., English) in which aspects of the agreement, including definitions, conditional logic, and actions to be performed, are expressed.

- ***Specification language:*** In computer science and electronic engineering, "specification languages" have been used for many years to describe complex software and systems. These languages can also provide formal descriptions (known as "specifications") of legal agreements. A specification language is typically not a natural language but can be a mathematical or logical formalism or similar to a programming language, supporting computer simulations of contract performance.

Despite their name, smart contracts are not always strictly legal contracts. Different jurisdictions have varying approaches to the formalities required for a contract, but generally, a legal contract must include elements such as offer, acceptance, and the intention of the parties to enter into a legal agreement.

The recognition of smart contracts as legally binding is essential to ensure that the outcomes of self-executing smart contracts are legally effective and enforceable in a court of law. For example, if a smart contract is unable to legally transfer ownership of an asset as it claims, much of the potential usefulness of smart contracts would be lost.

One potential solution to address this issue is to draft a traditional legal contract and "translate" it into code. In this approach, the legal contract acts as a "wrapper" for the automatic performance of the legal obligations carried out by the smart contract. This can be achieved by requiring the contracting parties to accept natural language terms that confer binding contractual effect on the transaction performed by the code.

This acceptance could be obtained through actions like clicking "I agree" to a set of terms. Once launched, the smart contract would self-execute with legal effect according to the coded terms. By ensuring that the legal formalities and elements of a contract are satisfied, it is highly likely that a smart contract and the outcomes of its self-execution could be considered legally binding.

Smart contracts pose challenges within the existing legal frameworks of many jurisdictions. Their use in automatically executing and guaranteeing legal obligations raises complex legal questions.

Code is deterministic, meaning that a given input will always produce the same programmed output. The programming language used in smart contracts follows this principle. Consequently, smart contracts must be drafted using structured and unambiguous language that computers can understand. For simple agreements like "A shall pay £10 to B on 1 January 2018," this poses no issues.

However, many contracts are more complex as they involve legal concepts and expressions that are challenging to translate into code. For instance, a clause that triggers certain consequences in the event of a "material adverse change" cannot easily be inserted as code into a smart contract.

Fundamental legal concepts such as "good faith," "negligence," and "reasonableness" are also difficult to encode. These concepts exist precisely to provide flexibility to contracting parties by not precisely defining in advance the exact obligations.

Translating legal contracts into code as self-executing programs would entail losing much of the functionality of traditional legal language. Consequently, smart contracts may have limited applicability and may be suitable only for contracts that are easily executable, while contracts requiring more nuanced functioning may not be suitable.

One potential solution to this problem could be to create a "hybrid" or "split" contract where only certain terms are encoded in code, and human input is required for clauses that necessitate legal interpretation. In the future, artificial intelligence and natural language computing may also help overcome this barrier.

Blockchains operate as a globally distributed network of nodes and are not bound by specific jurisdictions. Similarly, the execution of a smart contract does not neatly align with traditional territorial jurisdiction. This creates challenges in determining which jurisdiction's laws should govern contractual matters related to a specific smart contract and which court would have jurisdiction to handle any legal disputes that may arise.



The lack of clarity in this regard can result in additional disputes as parties litigate over the applicable jurisdiction's laws and the appropriate forum for resolving their claims. It is crucial for contracting parties to be able to determine their obligations and liabilities under a contract with a reasonable level of certainty.

Overcoming these complex jurisdictional issues is essential for the effective use of smart contracts. One approach to address this is for the parties to include choice of law and choice of court clauses in the contract itself. By doing so, these issues can be predetermined and agreed upon before the smart contract is launched, providing clarity and avoiding potential conflicts.

While smart contracts aim to provide certainty and predictability, they are still susceptible to software bugs and errors, which can lead to unintended consequences. In cases where a smart contract self-executes in a manner that the parties didn't anticipate, or when vulnerabilities in the code are exploited (as seen in the DAO attack), it raises questions about the available remedies and who can be held accountable.

These issues are further complicated by the immutable and irreversible nature of the blockchain. While the permanence of the blockchain is a valuable feature, it poses challenges when it comes to traditional error correction. Once a smart contract's program code is stored on an immutable blockchain, it cannot be debugged or modified. This makes it challenging to achieve corrective justice in such situations.

To address these concerns, due diligence of smart contracts will likely become a significant aspect of a lawyer's work to ensure there are no flaws before launching a smart contract. One approach to ensure error-free code could involve running the smart contract in a simulation environment to test its performance and identify any potential issues before deployment.

Although smart contracts offer advantages, they are not immune to disputes regarding their performance. As mentioned earlier, various types of disputes can arise, such as challenges to the legal binding nature of a smart contract, disagreements over the governing law, or allegations of misconduct leading to claims for damages.

Due to the absence of a central enforcement authority and established precedents, it is challenging to predict how these issues will be addressed and resolved definitively.

To mitigate potential conflicts, it is advisable for parties to include a dispute resolution or arbitration clause when entering into a contract via a smart contract.

By doing so, they can establish a mechanism for resolving any disputes that may arise during the execution of the smart contract. This allows parties to pre-determine the forum and process for resolving conflicts, providing them with a clearer path for dispute resolution.

As automation and digitization continue to expand across industries, it is important to consider the impact on the legal profession. The emergence of self-executing contracts represents a new frontier that may bring significant changes to the legal landscape, and lawyers must adapt accordingly.

The nature of legal work is likely to undergo significant transformation. In the long term, certain areas such as conveyancing, wills, and estates may become more automated, reducing the need for manual processes. Transactional lawyers may also find that a significant portion of their work can be executed automatically through smart contracts. However, this does not necessarily mean that lawyers will become obsolete.

Automation is best suited for repetitive and time-consuming operational tasks. This means that lawyers may have more time to focus on complex legal issues and provide higher-value services to their clients. Additionally, algorithms and automation can efficiently handle high-volume, low-risk tasks, allowing lawyers to manage larger portfolios of work.

While automation may change the landscape of legal practice, it also presents new opportunities for lawyers to adapt their skills and expertise. By embracing technology and leveraging its capabilities, lawyers can enhance their efficiency, expand their scope of work, and provide greater value to their clients.

In conclusion, smart contracts offer the potential for increased efficiency and automation in legal agreements. However, challenges such as legal complexities, jurisdictional issues, and the need for dispute resolution mechanisms must be addressed. The legal profession will likely undergo significant changes, with certain tasks being automated while lawyers focus on more complex legal issues. Embracing technology and adapting to the evolving legal landscape will be crucial for lawyers to thrive in this new era.





مجموعة جامعة
الجميع الذكية
للتعليم القابضة

MIDOCEAN
UNIVERSITY



Thank You