THE POWER OF SMART CONTRACTS: FROM DEFINITION TO REAL-WORLD USE

Ede Józsa
PhD Candidate
University of Miskolc

Introduction

The European digital decade in which we currently live forms an essential part of the EU's digital policy, announced under the framework of Web 4.0. Technological development has an impact on people's lifestyles and work, on how content is created and shared, and on the operation, innovation, and production of businesses, as well as on the position of consumers. These developments present both new opportunities and, undoubtedly, risks that must be addressed.
It is essential to examine how *civil law*, and in particular the field of *contract law*, is affected by this evolution and what challenges arise as a result. Addressing this topic is undoubtedly a complex task that requires not only legal expertise but also a certain level of technological understanding. Accordingly, this study also touches upon the fundamental technical characteristics of *smart contracts*, with particular attention to the operation of *blockchain technology* and the specificities of program codes that enable automated execution.
In recent years, blockchain technology has attracted growing interest across a wide range of industries: from the financial sector[1] and healthcare[2] to energy and utility services[3], the real estate market[4], and even public administration[5]. The reason for this surge in attention lies in the fact that blockchain enables the execution of legal transactions and operations that previously could only function via trusted intermediaries. Due to its *decentralized* structure[6], such transactions can now operate without centralized control, while maintaining the same functions and level of security. This mode of operation was simply unattainable in the previous technological environment.
Smart contracts – self-executing software codes (*scripts*[7]) embedded on the blockchain – combine the above concepts and enable the lawful functioning of decentralized and automated workflows. Naturally, transitioning to a decentralized network model is not always advisable. In fact, even when such a shift appears justified, the specific features of a given application may impose requirements that a blockchain-based system is unable to fulfill.
The purpose of this study is to provide a detailed overview of the functioning of blockchain and smart contracts and to demonstrate how they may be applied in the fields of *insurance contracts*, and *real estate registration systems*.

1. Definition of Key Concepts Related to Smart Contracts

This chapter presents the definitions of key concepts essential for understanding the operational mechanism of smart contracts.
Since smart contracts are based on blockchain technology, we begin by introducing the concepts associated with this underlying technology. This will provide an overview of the various *types of blockchains* and the functioning of *consensus mechanisms*.

[1]Kelly and Williams, 2016, n.p.
[2]Kar, 2016, n.p.
[3]Lacey, 2016, n.p.
[4]Oparah, 2016, n.p.
[5]Walport, 2016, n.p.
[6]See Chapter 2.
[7]For more information about this: Tangem, n.d., n.p

1.1. The Blockchain

The blockchain is a *distributed data structure* that operates by being replicated and shared among the members of a network[8]. It can be envisioned as a time-stamped ledger in which data is recorded in grouped units called "*blocks*[9]". Each block possesses a *cryptographic hash identifier*[10] and references the hash value of the preceding block. This chaining of blocks establishes the connection between them, which gives rise to the term "*blockchain*." Any *node*[11] that has access to this backward-linked list of blocks can read from it the global state that reflects the current data exchanges within the network[12].

The nodes form a *peer-to-peer network*[13] in which the following mechanisms apply: users interact with the blockchain using *private/public key* pairs based on *asymmetric cryptography*[14]. The private key is used to sign their own transactions, while their public key serves as their identifier within the network. Each signed transaction is transmitted from the user's node to its directly connected neighboring nodes. These neighboring nodes verify the validity of the transaction and forward it only if it passes validation. Invalid transactions are rejected. Valid transactions are thus gradually propagated across the entire network.

Through the process described above, the network collects and validates transactions, which are then arranged in sequence within a specific time interval and packaged into a time-stamped block. This process is referred to as *mining*[15]. The node performing the mining returns the newly created block to the network[16].

The nodes then examine whether the proposed new block: a) contains only valid transactions; and b) properly references the hash value of the previous block. If both conditions are met, the block is appended to the chain, and the transactions it contains are incorporated into each node's local state representation. If the conditions are not fulfilled, the block is rejected. This completes one cycle of the process. It is important to emphasize that this is a recurring procedure and constitutes the backbone of the blockchain network's operation.

When discussing the validation of transactions, a natural question arises: what qualifies as a valid transaction? To ensure that this decentralized environment remains functional and that the network can establish a unified global state – a so-called "*world state*" reflecting consensus among

[8]Christidis and Devetsiokiotis, 2016, p. 2293

[9]A block is the basic unit of the blockchain, containing a collection of validated transactions and the hash of the previous block, thereby ensuring the continuity and immutability of the chain within a decentralized, cryptographically secured system. Yaga et al., 2018, p. 7

[10]Cryptography is a specialized field that encompasses the principles, tools, and methods for transforming data in order to conceal its semantic content, prevent unauthorized use, and exclude undetected modification. Cryptographic hashing is a one-way mathematical process that generates a fixed-length, unique identifier (hash) from arbitrary data. In blockchain systems, hash functions ensure the chronological linkage of blocks: each block contains the hash identifier of the previous one, thereby making retroactive modification impossible.

[11]A node refers to a device that participates in the operation of a distributed ledger network, essentially enabling the functioning of the blockchain system. Any active electronic device can be a node – including computers, phones, or even printers – provided they are connected to the internet and therefore possess an IP address.

[12]Eris Industries, no date, n.p.

[13]The peer-to-peer (P2P) model in digital data exchange refers to a network structure in which participants connect directly to one another without a technical intermediary. This model allows users to carry out transactions without involving a financial institution. Due to the decentralized nature of the network, each node is equally entitled to initiate, receive, and validate transactions. The primary significance of this structure lies in eliminating the need for trust in a central party, replacing it with cryptographic principles – primarily hash-based proof-of-work – to ensure the integrity of the system's operation.

[14]Greenspan, 2015a, n.p.

[15]Mining is the process that ensures the operation of blockchain technology, during which transactions are validated and blocks are created through cryptographic computations, typically within an incentive (reward) system. Narayanan et al., 2016, pp. 131–138

[16]Who becomes a mining node and what gets included in the block is determined by the consensus mechanism employed by the network.

the nodes – each blockchain network must define specific rules governing the conditions under which database operations are considered valid.

1.2. Consensus Mechanism Employed

The nodes operating the blockchain must reach agreement on which transactions should be included in the next block and in what order they should appear. Without such consensus, the various copies of the blockchain would diverge, resulting in so-called *forks*[17]; nodes would reflect differing "world states," thereby preventing the network from maintaining a unified, reliable chronological order – in other words, a consistent blockchain – unless the fork is successfully resolved[18].
For this reason, every blockchain network requires a *distributed consensus mechanism* that ensures the formation of a uniform position among nodes[19]. The specific type of consensus mechanism depends on the *nature of the blockchain network* and the *attack vectors* considered by the entity operating the given network[20].
In an ideal scenario, all validating nodes would vote on the order of transactions to be included in the next block, and the accepted version would be determined by majority decision. However, in an open network – where anyone can participate – such a process could have catastrophic consequences due to the risk of *Sybil attacks*[21].

1.3. Overview of Blockchain Types

While this study often refers to "the blockchain" or "blockchain technology," it is important to clarify that there is not a single blockchain, but rather several known variants of it[22].
From a legal perspective, the distinction between *private and public blockchains* is of particular significance. A *private blockchain* is a system that is accessible only to specific, authorized individuals[23]. Before joining the blockchain, access is typically subject to approval by a central authority that verifies whether the participation requirements have been met. Users of private blockchains are often identifiable – usually by the controlling authority, but in some cases also by other users. Moreover, the nodes operating the blockchain are generally known, making the operation of the blockchain potentially subject to influence.
In contrast, a *public blockchain* is a system that anyone can join at any time, provided they download the publicly available client software and meet the necessary technical requirements[24]. In such networks, there is no central entity responsible for verifying access requirements or identifying participants.
Another key distinction can be made based on the *permission model*, allowing for the differentiation between *permissioned and permissionless blockchains*. This classification primarily addresses the issue of authorization. In *permissioned blockchains*, only designated participants are allowed to perform transactions, whereas in *permissionless blockchains*, anyone may initiate transactions.

---

[17]A protocol change that results in the creation of two or more divergent distributed ledgers / blockchain versions – for example, in cases where two or more different blocks have the same block height, meaning that each builds cryptographically on the same previous blockchain segment and thus each claims, in its own right, to be the valid continuation for subsequent transactions. ELI, 2022, p. 20
[18]Greenspan, 2015b, n.p.
[19]Christidis and Devetsiokiotis, 2016, p. 2294
[20]An attack model or "attack vector" refers to the set of methods aimed at exploiting vulnerabilities in blockchain systems, with the goal of compromising the network's security, operational functionality, or economic integrity, Conti et al., 2018, pp. 3416–3418
[21]A Sybil attack involves a single actor creating multiple identities, thereby gaining disproportionate influence over the functioning of the network – potentially allowing a minority entity to gain control. See: Douceur, 2002, pp. 251–260
[22]ELI, 2022, p. 25
[23]Ibid., p. 25
[24]'What are the 4 different types of blockchain technology', 2025, n.p.

Consequently, a blockchain can be categorized along two dimensions: public versus private, and permissioned versus permissionless. This results in four main types of blockchains: *(1) public permissioned blockchain; (2) public permissionless blockchain; (3) private permissioned blockchain; and (4) private permissionless blockchain.*
These categories will prove relevant in later sections when discussing the practical applicability of blockchain–smart contract technologies.

2. Smart Contracts

The use of smart contracts in contractual legal relationships primarily lies in the ability to execute agreements between parties in a pre-programmed, automated manner upon the occurrence of a specified event. As a result, contractual performance – under certain conditions – can be achieved without human intervention. This technology offers particular advantages in situations where the performance process can be clearly modeled and where post-contractual amendments are unlikely. For more on this, see the referenced literature[25].
In what situations and areas can smart contracts be usefully and effectively applied? Ideally, in any contractual agreement, all parties are interested in the smooth performance of the contract and in ensuring that execution occurs under the terms they have mutually agreed upon. Smart contracts essentially provide a solution to this objective: a platform that guarantees performance under predefined and transparent conditions, simultaneously observed by all parties.
Through automated mechanisms, smart contracts can enhance the dynamism of economic transactions, saving substantial time and effort. Future contracting parties may carry out legal acts, declarations of intent, and other actions with legal consequences even before the conclusion of the contract – actions that would otherwise require offline processing and consume time. In traditional procedures, such situations typically involve waiting – waiting for a person's response, legal declaration, or act.

2.1. Definition of Smart Contracts

From the perspective of conceptual clarification, the key question regarding the term "smart contract" concerns the first component: what makes a contract "*smart*," or why can a contract be considered smart?
The term "smart contract" was coined by Hungarian computer scientist and engineer Nick Szabo, whose name is also frequently associated with the invention of the cryptocurrency Bitcoin. According to his definition, a smart contract is "*a transactional protocol that executes the terms of a contract*[26]".
In Szabo's vision, basic contractual forms can be embedded into computer software and hardware, enabling the automation of contractual execution. He proposed that contractual provisions – such as guarantees or obligations – can be encoded and embedded into software or hardware tools, which are capable of autonomously enforcing the agreement upon fulfillment of the stipulated conditions[27]. As a result, the need for a trusted intermediary may be eliminated; trust between the contracting parties is substituted not by a third party, but by the underlying technology. Furthermore, this can significantly reduce the risk of contractual disputes.
Szabo considered the purchase from a vending machine a typical example of a smart contract, as the machine autonomously delivers the selected product. However, this is a misinterpretation. In this case, it is not the contract that is smart, but the machine, which performs the contractual obligation on behalf of the operator. While the example does not accurately reflect the substantive

[25]Turányi, 2018, n.p., Rohr and Wright, 2017, n.p.
[26]Csitei, 2019, p. 6; Szabo, 1994, n.p.
[27]Szabo, 1997, n.p.

concept of a smart contract, it does illustrate one aspect of digitalization – where a machine executes a contract instead of a legal subject.

Other authors[28] have defined smart contracts as "*self-executing autonomous computer protocols*," or as "*agreements between two or more parties that are programmed in such a way that their execution is ensured through blockchain technology*." Of particular note is the definition provided by Ágnes Juhász, who describes the technology as follows: "*In the simplest and most concise formulation, a smart contract is nothing other than a self-executing agreement.*[29]"

The various definitions offered by these authors emphasize different attributes of the technology, yet a common element is the self-executing nature of the contract.

It is reasonable to consider and use as a starting point the definition provided in the European Union's legal framework for blockchain and smart contracts, published in September 2019, which may be viewed as a consensus-based definition. According to this document, a smart contract is "*computer code stored on a blockchain and accessible by one or more parties (…) often self-executing.*[30]"

Taking into account the above characteristics and definitions, the present study adopts the following definition of a smart contract: a piece of code containing an if–then logical instruction that executes autonomously once the predefined condition(s) are met[31].

2.2. Forms of Smart Contracts

In 2022, the European Law Institute published a document outlining principles[32] and guidelines concerning blockchain technology, smart contracts, and consumer protection. Principle 2 addresses the different *forms of smart contracts* and their legal characteristics. According to this principle, several *types of smart contracts* can be distinguished: 1) a code-only construction without a legal agreement (in such cases, only a technical transaction occurs); 2) a tool for executing a legal agreement that exists *off-chain*[33]; 3) a legally binding declaration of intent – such as an offer or acceptance – which may itself constitute a legal agreement; 4) a hybrid form in which the smart contract is integrated with the legal agreement, and therefore exists simultaneously *on-chain*[34] and off-chain[35].

The concept of smart contracts has been widely discussed in legal scholarship, partly due to misunderstandings arising from the terminology itself. When lawyers speak of a "contract," they generally refer to a *legally binding agreement*, whereas for developers, this is not necessarily the case.

From a technical perspective, smart contracts are pieces of program code that execute conditional "if–then" logic. From a legal standpoint, however, the central question is whether such code can be regarded as a contract under civil law.

No universally applicable rule can be given to answer this question; each case must be evaluated individually, taking into account the type of blockchain involved, the parties concerned, and the

[28]Kost, in: Corrales – Fenwick – Haapio, 2019, p. 5.: self-executing, autonomous computer protocols that facilitate, execute and enforce commercial agreements between two or more parties; Wattenhofer, in**:** Corrales – Fenwick – Haapio, 2019, p. 5: an agreement between two or more parties, encoded in such a way that the correct execution is guaranteed by the Blockchain.

[29]Juhász, 2020, p. 2

[30]Lyons et al., 2019, p. 22

[31]For example, an operation executed by a smart contract could be as follows: if a certain amount is received in a bank account, then 10% of that amount is automatically transferred to a second account for long-term savings purposes.

[32]ELI, n.d., n.p.

[33]Located, executed, or operated outside the blockchain system. Off-chain operations are not directly recorded on the chain, making them faster but potentially less secure. Wang et al., 2019, pp. 2266–2277

[34]Located, executed, or operated within the blockchain system. On-chain operations are public, immutable, and validated through decentralized consensus., Xu et al., 2019.

[35]In the latter case, it must be determined whether the given agreement should be considered on-chain or off-chain in nature. Therefore, if the smart contract has merged with the legal agreement, it is up to the parties to decide whether they wish to treat it as an on-chain or off-chain agreement.

underlying interests. The primary doctrinal point of departure is the following: smart contracts can, under certain conditions, constitute legally binding contracts in the sense of civil law.
A contract is legally binding if it meets the validity requirements established under civil law[36]. In other words, a contract is formed when there is a meeting of the parties' *declarations of intent* – that is, when the offer and the acceptance correspond in substance[37].
For an offer to be legally effective, it must be sufficiently specific, contain the essential terms of the agreement to be concluded, and reflect a clear intention to be bound. By contrast, an invitation to treat (*invitatio ad offerendum)* does not express a binding intention and thus does not constitute an offer. A valid acceptance must also correspond exactly to the terms of the offer; any deviation renders it ineffective.
In the first category of smart contracts – those that operate solely as code – no legally binding contract is formed. Smart contracts execute conditional "if–then" instructions, and in some cases, they simply result in a change of state on the blockchain. While this may have real-world consequences, it does not necessarily produce legal effects. Such smart contracts are considered technical processes rather than contracts under civil law[38].
The next form involves a smart contract used as a technical means to perform a legally binding agreement concluded off-chain. In this case, the off-chain agreement specifies the rights and obligations of the parties, as well as the fact that blockchain technology – specifically, the smart contract – will be used to facilitate performance. Here, the blockchain serves exclusively as a tool for execution and technical implementation.
A smart contract may also function as a legally binding declaration of intent – such as an offer or acceptance – or even as the contract itself. During the pre-contractual "formation phase," it is possible that only part of the contractual process is conducted using blockchain technology. In such cases, the smart contract may represent only one component of the legal transaction. For example, one party might make an offer in on-chain form through a pre-deployed smart contract, while the other party accepts it off-chain, such as through a written or verbal declaration. Conversely, the offer might be made off-chain in the form of *source code*[39], and the acceptance could occur on-chain when the other party compiles the code into *bytecode*[40] and deploys it on the blockchain.
In the latter scenario, a legal question arises as to whether deployment itself qualifies as contractual acceptance, and if so, what the content of the resulting agreement is. If the contract involves a counter-performance obligation – such as the transfer of funds to the deployed smart contract – and the offering party fails to fulfill this obligation, such failure may constitute a breach of contract, provided that the contract was validly formed under applicable legal standards.
This leads to the question of whether a program code – such as a smart contract – can express a declaration of intent. Based on the principle of private autonomy, it can be concluded that such

[36]Welmann, 2014, pp. 21–24; Romanian Civil Code (Rptk.), Article 1178 *Freedom of Form*; Hungarian Civil Code (Ptk.), Section 6:63 [Conclusion and Content of the Contract].
[37]The meaning of offer and acceptance corresponds to the interpretation used in classical contract law.
[38]Example of such a smart contract: In Decentralized Autonomous Organizations (DAOs), the voting method is determined by the membership model. In many cases, DAOs operate in an open and permissionless manner, where decisions are made using governance tokens. Anyone holding such a token may vote. The technical basis for voting is typically a smart contract designed for that purpose. During the process, a token holder submits a proposal, and other participants vote by temporarily assigning their token to the smart contract – either in support or opposition. After the vote, all tokens are returned to their owners. Such a smart contract performs a purely technical operation and does not qualify as a legally binding agreement.
[39]Human-readable program code written in a programming language, which must be translated or interpreted into a machine-executable form.
[40]*An intermediate, machine-readable code generated from the source code by a compiler and executed by a virtual machine (e.g., the Java Virtual Machine).*

code may indeed serve as a medium for expressing the will of the parties and, consequently, may give rise to legal effects[41].

Taking the following criteria into account, a smart contract deployed on the blockchain may meet the requirements for a valid offer[42]:

1) Definite content: in a contract based on "if X, then Y" logic, the performance triggered by the smart contract is clearly determined at the moment of deployment (e.g., transfer of cryptocurrency), thereby ensuring specificity of content;
2) Intention to be bound: the act of deploying a smart contract to the blockchain typically demonstrates an intent to be legally bound, particularly because such deployment is generally irreversible.

Accordingly, a smart contract, as a set of unilaterally pre-established contractual terms, may legally constitute an offer that can be accepted through implied conduct. In summary, regarding the third category, a smart contract may independently or partially give rise to a legally binding agreement, provided that the validity requirements of contract law (offer and acceptance) are satisfied under the applicable legal system[43].

Finally, in certain cases, the technical implementation of the smart contract and the substantive legal agreement are so closely integrated that the contract consists of both on-chain and off-chain elements. In such hybrid arrangements, it is essential to determine whether the legally binding and enforceable terms of the contract are expressed on the blockchain (on-chain) or through separate, for example, written declarations (off-chain)[44].

In some instances, the program code and the traditional legal agreement cannot be meaningfully separated, but together form a single contractual construct containing both blockchain-stored (on-chain) and conventional (off-chain) components. These are referred to as hybrid contracts, and this dual nature may influence the legal interpretation of the contract's formation, content, performance, and enforceability.

According to the European Law Institute – a position also endorsed by the present author – in the event of a conflict between the on-chain and off-chain versions, the off-chain text shall prevail[45]. From a legal interpretative standpoint, the human-readable content is authoritative, even if the execution of the agreement is partially automated and conducted on-chain.

2.3. Operation of Smart Contracts

In the context of blockchain, smart contracts are scripts stored directly on the blockchain[46]. Since they reside on the blockchain, each smart contract is assigned a unique address. To activate a specific smart contract, a transaction is sent to its address, triggering the contract to execute autonomously and automatically.

---

[41]However, this must in no way result in a weakening of the protection of market participants, particularly consumers – remedies (e.g., protection against unfair terms, consumer protection regulations) must also apply in such cases.

[42]Ptk., § 6:64 [Binding nature of an offer]; Rptk., art. 1188 [Contractual offer].

[43]A smart contract may constitute a legally binding offer, with the associated transaction serving as acceptance. A typical example of this is an Initial Coin Offering (ICO), as well as similar decentralized financial structures such as decentralized exchanges or cryptocurrency-based lotteries. In the context of an ICO, a smart contract is generally deployed on the blockchain which – in simplified terms – contains the following provision: "If someone transfers 0.1 (a predefined unit of value) to the address of the smart contract, they will receive a newly issued token in return." At first glance, it is apparent that the deployment of such a smart contract may be considered an offer, as it contains the fundamental elements required for the conclusion of a contract – in particular, the consideration (the price of the token) and the definition of the subject matter of performance (the newly issued token).

[44]An example of this is when the parties record the automated conditions for payment and performance in the form of a smart contract, while the rules on liability or available legal remedies are included in a separate written annex. In such a case, certain provisions of the contract are implemented on-chain, while other elements operate off-chain, and thus the legal assessment must separately consider which component carries the essential content of the agreement.

[45]ELI, 2022, p. 41

[46]Christidis and Devetsiokiotis, 2016, p. 2296

This process is executed across all network nodes in a predefined manner based on the data included in the transaction. Accordingly, a blockchain that supports smart contracts can be seen as a distributed virtual machine, where each node runs a local instance of that virtual machine.
To illustrate the operational mechanism of a smart contract, let us consider the following practical example[47]: imagine a blockchain network in which Alice, Bob, and Carol participate, trading digital assets of types X and Y; Bob deploys a smart contract to the network containing the following functions:

a) a "deposit" function that allows him to deposit X units into the contract,
b) a "trade" function that sends back 1 unit of X (from the previously deposited amount) in exchange for every 5 units of Y received, and
c) a "withdraw" function enabling Bob to withdraw the entire asset balance held by the contract.

In this example, the "deposit" and "withdraw" functions are written in such a way that only Bob (via his own cryptographic key) can activate them. This was Bob's choice at the time of coding the contract and is a reasonable one in this scenario; however, in theory, these functions could be written to allow any user on the network to trigger them.
Next, Bob sends a transaction to the smart contract address, activating the "deposit" function and transferring 3 units of X into the contract. This transaction is recorded on the blockchain. Subsequently, Alice – who holds 12 units of Y – sends a transaction to invoke the "trade" function, transferring 10 units of Y to the contract in return for 2 units of X. This transaction is also recorded on the blockchain. Finally, Bob sends a signed transaction to the "withdraw" function. The contract verifies the signature to ensure that the withdrawal is indeed initiated by its rightful owner, and then returns the remaining assets (1 unit of X and 10 units of Y) to Bob.
Based on the example above, the following key observations can be made:

1) A smart contract has its own blockchain account and can hold assets, acting as an independent entity within a distributed database.
2) It encodes business logic – contractual terms – into executable code, e.g., "1 unit of X is transferred for every 5 units of Y received."
3) A well-designed contract handles all execution outcomes, such as rejecting indivisible inputs (e.g., 12 Y), or partially fulfilling the offer (accept 10 Y, return 2 Y, and send 2 X)[48].
4) The contract is data-driven: e.g., Bob sends a transaction that states, "Send 5 Y to receive 1 X in return."
5) Smart contracts are triggered by transactions (messages) sent to their blockchain address.
6) They are deterministic: the same input always yields the same output, ensuring network-wide consensus[49].
7) The contract code is fully transparent and viewable by all network participants[50].
8) All interactions are digitally signed and recorded, creating a cryptographically verifiable audit trail[51].

[47]Ibid., p. 2296
[48]It is important to note that these functions – which govern the execution of the contract – depend on the developer of the smart contract and the intent of the party commissioning its application.
[49]On a properly designed blockchain platform, writing non-deterministic smart contracts is either impossible – by restricting developers to deterministic programming languages and constructs – or, while technically possible, the network will reject the deployment of such contracts. For more information, see: Solidity Documentation, no date, n.p., Cachin et al., 2016, p. 2
[50]The content of the contract and the conditions of its formation can be examined and accessed by anyone. If the public key of the contracting parties becomes known – by being linked to an identifiable person – feigning the existence of a contract becomes impossible.
[51]Thus, the record of transactions is accessible to everyone, making it possible to track the input and output activities of a given public key.

What types of conditions can be incorporated into the code of smart contracts? Depending on the will of the person who determines the condition for the conclusion of the contract, conditions can be classified into three categories: *contingent, mixed*, or *potestative* conditions.
A *contingent* condition means that the occurrence of a future, uncertain event is independent of the will of the parties involved. E.g., if during the month of May the temperature drops below -2°C for ten consecutive days, the insurer will pay the insured an amount of $1,000. In this case, payment – performance by the insurer – is conditional upon the ten-day period with temperatures below -2°C, which is independent of the will of either party. This form of condition does not affect the validity of the legal transaction.
A *mixed* condition means that the occurrence of the future, uncertain event depends partly on the will of the interested party and partly on the will of a specified third party. For example, the insurer promises to pay a sum of money to the insured if, within a given month, the inflation rate of the dollar rises by 5%, resulting in the largest supplier demanding a 20% price reduction. In this case, the occurrence of the condition depends both on the will of the interested party and the declared intent of the other person (the specified client). Defining such a condition is legitimate and permissible for achieving specific goals, and does not affect the validity of the legal transaction.
The third type of condition – the *potestative* (self-imposed) condition – can take two forms: *purely potestative* condition and *simply (or mixed) potestative* condition. The essence of a potestative condition is that its occurrence depends on an act arising from the will of the interested party (entitled or obliged party), meaning it depends on the action of a specific person. In cases where the potestative condition depends solely on the obligor's voluntary will (purely potestative condition), without being tied to any external event, the legal transaction is null and void. For example: "I will pay you compensation if I feel like it." Conversely, if the entitled party has control over the purely potestative condition, the transaction remains valid. A classic example is a conditional sales contract containing a clause that the legal effect of the sale occurs if the product suits the buyer's taste – this is a valid contractual provision.
In my opinion, only contingent conditions can be considered in the context of smart contracts, since conditions tied to human conduct – namely, mixed and potestative conditions –currently lack any database (as of today) that could make their occurrence traceable. While weather changes (such as ten consecutive days in May with temperatures below -2°C) are recorded in numerous databases and can be monitored, similar data processing systems for human actions do not yet exist. Therefore, the inclusion of such conditions – in my view – is not possible in smart contracts, or at the very least, they are not trackable.

3. Potential Applications of Smart Contracts

Blockchain networks that support *Bitcoin-type* transactions enable the transfer of assets between parties who do not trust one another. Blockchains that support smart contracts go a step further: they allow for the execution of complex, multi-step processes – so-called "interactions" – even in the absence of mutual trust.
Parties engaging in a transaction or legal act involving a smart contract can rely on the following key aspects: 1) by reviewing the contract code in advance, they can assess whether they wish to contract under the given terms, anticipating the legal and economic consequences; 2) execution is guaranteed, as the code is already deployed on a network that no single party can fully control; 3) the entire process is auditable and accessible to all, since every interaction is cryptographically signed; 4) if the smart contract is programmed to account for all possible outcomes, the possibility of legal dispute is eliminated – parties cannot contest the result of a process whose outcome was verifiably and publicly documented before the contract was entered into.
Smart contracts operate on the blockchain as autonomous agents, and their execution is entirely predictable. Accordingly, they are suitable for performing any blockchain-based transaction that follows "if–then" logic, provided that the necessary data is available on-chain.

The following section will present two examples: one as an instance of existing practice and the other, as a potential use case of smart contracts.

3.1. Insurance Contracts

Insurance contracts constitute one of the most highly regulated and detailed types of agreements under civil law. Their characteristics include standardization, proportionality, and the fundamental expectation that the insurer's performance is contingent upon the occurrence of a future, uncertain event. In this contractual context, the application of smart contracts may initially appear to pose technical challenges; however, in specific areas – particularly where risk events are objectively measurable – they offer a particularly viable alternative.

Blockchain technology and its smart contract mechanisms are especially well-suited for the automated handling of simpler insurance risks. In practice, this applies to types of insurance where the fulfillment of service conditions – that is, the occurrence of an insured event – can be verified digitally, from an authentic source, without the involvement of a third party. One such example is agricultural insurance, where weather data – such as rainfall measured in a specific region – can be queried directly from a public data source through an oracle. If the pre-defined condition is met – for instance, rainfall does not reach a certain threshold (x mm) within a given month – the smart contract executes the payout automatically.

A similar logic can be applied to microinsurance contracts covering travel delays or baggage losses, where digital systems that track flight delays or baggage arrival times provide the data necessary to execute the transactional protocol. In such cases, the insurance contract is not merely a formal expression of risk assumption, but also an executable code that performs automatically once the data source confirms the occurrence of the insured event.

The advantage of this type of smart contract lies not only in the speed and automation of execution but also in increased transparency, fraud prevention, and cost-efficiency. At the same time, however, traditional principles of insurance law – particularly the temporal scope of risk coverage, proportionality, insurable interest, and the protection of insurance secrecy –require reinterpretation in a system where automation precludes post hoc discretion or legal dispute.

In my view, the application of smart contracts as insurance contracts is particularly justified in cases where the relevant conditions can be modeled deterministically, the legal relationship does not require individual adjudication, and trust between the parties can be partially or fully substituted by technology. Nevertheless, for more complex insurance arrangements (e.g. life insurance, liability insurance), the applicability of this technology remains limited at present, and its integration will require further technological development as well as detailed legislative intervention[52].

3.2. Real Estate Registration System

Another potential application of smart contracts is in real estate registration systems[53]. It is important to emphasize, however, that the technological solution discussed here – namely, the transfer of ownership via a smart contract – is currently only a theoretical possibility and is not applicable under the Hungarian legal system.

[52]In 2017, an insurance company introduced a product that partially utilized blockchain-based solutions. The structure allowed customers to purchase flight delay insurance online through a traditional web interface. Once the insurance contract was concluded between the insurer and the customer, a self-executing smart contract automatically enforced the insurance terms. Customers did not need to manually claim compensation in the event of a delay – the smart contract automatically checked whether the flight met the delay criteria set out in the policy and, if so, initiated the payout process. For example, in the case of a two-hour delay, the smart contract automatically triggered the payment of the amount specified in the contract. The contract operated on a simple "if–then" logic: "if the flight is delayed by more than two hours, then initiate automatic payment."

[53]Oparah, 2016, n.p.

Under the current legal framework, entries in the real estate registry may only be made upon request and in compliance with specific formal requirements for documentation. In practice, sellers usually provide the consent required for registration only after the full purchase price has been paid, meaning payment precedes the official registration of ownership. The following example, therefore, does not reflect current domestic practice but aims to illustrate a potential direction for future digital transformation.

Generally, both the buyer and the seller wish to finalize the agreement and sign the necessary documents as quickly as possible. However, the traditional real estate market is not known for its speed or simplicity. Real estate transactions have always been complex and cumbersome, and government regulations around the world further slow the process by imposing restrictions or additional costs on property transfers. Due to a range of such restrictions, the sale of property can proceed at a snail's pace, even when both parties are eager to close the deal.

At present, most buyers and sellers rely on *escrow*[54] service providers as a form of security, ensuring that both parties comply with the terms of the transaction and thus reducing the risk of fraud.

In the near future, a smart contract could serve as the legal instrument for property transfer, integrated into a blockchain-based payment system. E.g., the smart contract could be connected to the land registry's database. The purchase price of the property would be locked in the buyer's bank account until the land registry confirms the buyer's ownership registration. Upon this confirmation, the funds would be transferred automatically.

Blockchain technology offers the possibility of eliminating intermediaries – in this case, the escrow service provider. If the authenticity of the land registry can be proven through a distributed database, the rightful owner could transfer the asset immediately and lawfully without the need for a third-party guarantor.

The development of the internet and digital tools has made it significantly easier to produce forged documents and publish fraudulent real estate listings. One of the most serious problems is the forgery of ownership documents that falsely claim someone owns a property when in fact they do not. The internet has only worsened this issue: replicating notary seals and transfer deeds is now easier than ever.

According to Morgan Brennan[55], one of the most common forms of fraud is "rental scam" fraud[56], where the perpetrator copies data and photos from legitimate listings and reposts them on another website as their own. The fraudster then requests an advance payment from interested renters – labelled as a "deposit[57]" or "service fee" – or asks for a transfer to a third party as proof of solvency. In most cases, the deceived renter never sees their money again.

Blockchain offers an immutable and auditable data structure in which all financial transfers are logged with sender and recipient information. In such a system, properties could be linked to digital ownership certificates. These certificates would be virtually impossible to forge and would be directly associated with specific properties, making it nearly impossible to advertise or sell an asset without having legitimate ownership. As a result, forged ownership documents and fraudulent listings could be eradicated.

The vast majority of real estate transactions are not cash-based, and mortgage approvals often involve lengthy administrative processes and eligibility checks. Although various online sources offer advice on how to speed up these processes, blockchain could provide a truly novel and practical solution.

[54]Escrow refers to a custodial service provided by a neutral third party, ensuring that funds or documents related to a transaction are only released once the conditions specified in the contract have been fulfilled. Arner et al., 2017, pp. 1–14

[55]Brennan, 2013, n.p.

[56]Ptk., § 6:331 [Lease contract]; Rptk., art. 1777 [Contract of lease – Definition].

[57]Ptk., § 6:360 [Deposit contract]; Rptk., art. 2103 [Contract of deposit – Definition].

With blockchain, it would be possible to create a digital identifier for the property and digital identities for both the buyer and the seller. This would allow for an integrated, streamlined, and rapid process for both mortgage approval and ownership transfer[58].
For buyers, credit histories and income information could be verified instantly, eliminating the need for time-consuming in-person procedures with banks, lawyers, or real estate agents. Property owners could digitally prove their ownership, and the ownership chain of a given property could be transparently recorded, along with a documented list of past repairs and renovations, and even forecasts of future operating costs.

Conclusions

The study has demonstrated that smart contracts, as blockchain-based, self-executing codes, can – in specific legal and technical contexts – constitute legally binding contracts under civil law. Their legal classification depends on how they are integrated into the contractual process: as purely technical scripts, as tools for executing an off-chain agreement, as declarations of intent, or as hybrid instruments combining on-chain and off-chain elements. Where validity requirements such as *offer and acceptance* are met, smart contracts may function as enforceable agreements.
The technical operation of smart contracts is deterministic, transparent, and auditable. However, only contingent conditions – i.e. those independent of human will – are currently suitable for codification, due to the lack of reliable, verifiable data on potestative or mixed conditions.
As for practical application, smart contracts are particularly well-suited to insurance contracts involving objectively verifiable events (e.g. flight delays, weather events), where automation enhances transparency and efficiency. Their integration into real estate registration systems, while currently theoretical under Hungarian law, represents a potential future direction – provided that formal legal and technological conditions (e.g. interoperability with official registries) are met.
In summary, smart contracts hold transformative potential in selected legal areas where automated, data-driven execution can complement – or in some cases replace – traditional contracting mechanisms. Their use, however, requires careful legal qualification, robust technical design, and often legislative clarification to ensure enforceability and compliance with fundamental civil law principles.

References

'What are the 4 different types of blockchain technology' (2025) *TechTarget* [Online]. Available at: https://www.techtarget.com/searchcio/feature/What-are-the-4-different-types-of-blockchain-technology (Accessed: 25 June 2025).

Arner, D.W., Barberis, J.N. & Buckley, R.P. (2017) 'Fintech and regtech: Impact on regulators and banks', *Journal of Banking Regulation*, 19, pp. 1–14. https://doi.org/10.1057/s41261-017-0037-3

Brennan, M. (2013) '3 insidious real estate scams and how to avoid them'. *Forbes* [Online]. Available at: https://www.forbes.com/sites/morganbrennan/2013/07/16/3-insidious-real-estate-scams-and-how-to-avoid-them/ (Accessed: 27 September 2025).

Brennan, M. (2013) '3 Insidious Real Estate Scams and How to Avoid Them'. *Forbes* [Online]. Available at: https://www.forbes.com/sites/morganbrennan/2013/07/16/3-insidious-real-estate-scams-and-how-to-avoid-them/ (Accessed: 31 May 2025).

Cachin, C., Schubert, S. & Vukolić, M. (2016) 'Non-determinism in byzantine fault-tolerant replication', *arXiv preprint*, arXiv:1603.07351. Available at: https://arxiv.org/abs/1603.07351 (Accessed: 27 September 2025).

[58]Oparah, 2016, n.p.

Conti, M., Kumar, S., Lal, C. & Ruj, S. (2018) 'A Survey on Security and Privacy Issues of Bitcoin', *IEEE Communications Surveys & Tutorials*, 20(4), pp. 3416–3452. https://doi.org/10.1109/COMST.2018.2842460

Corrales, M., Fenwick, M. & Haapio, H. (eds.) (2019) *Legal Tech, Smart Contracts and Blockchain*. Singapore: Springer. https://doi.org/10.1007/978-981-13-6086-2

Csitei, B. (2019) 'Okos szerződések', *Opuscula Civilia*, 2019(6), p. 6.

Douceur, J.R. (2002) 'The Sybil attack', in *Peer-to-Peer Systems (Lecture Notes in Computer Science, vol. 2429)*. Berlin: Springer, pp. 251–260. https://doi.org/10.1007/3-540-45748-8_24

ELI – European Law Institute (2022) *Principles on Blockchain Technology, Smart Contracts and Consumer Protection*. Brussels: EU Justice Programme, p. 25.

Eris Industries (n.d.) *Documentation – Blockchains; Documentation – Smart Contracts*. Available at: https://docs.erisindustries.com (Accessed: 27 September 2025).

European Law Institute (2022) *Principles on Blockchain Technology, Smart Contracts and Consumer Protection*. Brussels: EU Justice Programme [Online]. Available at: https://www.europeanlawinstitute.eu/projects-publications/publications/eli-principles-on-blockchain-technology-smart-contracts-and-consumer (Accessed: 27 September 2025).

Greenspan, G. (2015a) 'Avoiding the Pointless Blockchain Project'. [Online]. Available at: https://www.multichain.com/blog/2015/11/avoiding-pointless-blockchain-project (Accessed: 27 September 2025).

Greenspan, G. (2015b) 'Ending the Bitcoin vs Blockchain Debate'. [Online]. Available at: https://www.multichain.com/blog/2015/05/bitcoin-vs-blockchain-debate (Accessed: 27 September 2025).

Juhász, Á. (2020) 'Online szerződéskötés, digitális tartalom és szolgáltatás, intelligens szerződések – a szerződési jog új korszaka?', *Infokommunikáció és Jog*, 2020/2 (75. e-különszám), p. 2.

Kar, I. (2016) 'Estonian Citizens Will Soon Have the World's Most Hack-Proof Health-Care Records'. *Quartz*. [Online]. Available at: https://qz.com/748566/estonian-citizens-will-soon-have-the-worlds-most-hack-proof-health-care-records (Accessed: 27 September 2025).

Kelly, J. & Williams, A. (2016) 'Forty Big Banks Test Blockchain-Based Bond Trading System'. *Financial Times*. [Online]. Available at: https://www.ft.com (Accessed: 27 September 2025).

Lacey, S. (2016) 'The Energy Blockchain: How Bitcoin Could be a Catalyst for the Distributed Grid'. *Greentech Media*. [Online]. Available at: https://www.greentechmedia.com (Accessed: 27 September 2025).

Lyons, T., Courcelas, L. & Timsit, K. (2019) *Legal and Regulatory Framework of Blockchains and Smart Contracts – A Thematic Report*. European Union Blockchain Observatory and Forum, 27 September, p. 22.

Narayanan, A., Bonneau, J., Felten, E., Miller, A. & Goldfeder, S. (2016) *Bitcoin and Cryptocurrency Technologies*. Princeton: Princeton University Press, pp. 131–138.

Oparah, D. (2016) '3 Ways That the Blockchain Will Change the Real Estate Market'. *TechBullion*. [Online]. Available at: https://techbullion.com/3-ways-that-the-blockchain-will-change-the-real-estate-market (Accessed: 27 September 2025).

Oparah, D. (2016) '3 Ways That the Blockchain Will Change the Real Estate Market'. *TechCrunch* [Online]. Available at: https://techcrunch.com/2016/02/06/3-ways-that-blockchain-will-change-the-real-estate-market/ (Accessed: 27 September 2025).

Rohr, J. & Wright, A. (2017) 'Blockchain-Based Token Sales, Initial Coin Offerings, and the Democratization of Public Capital Markets', *Cardozo Legal Studies Research Paper No. 527*. https://doi.org/10.2139/ssrn.2956720

Szabo, N. (1994) 'Smart Contracts' [Online]. Available at: https://www.fon.hum.uva.nl/rob/Courses/InformationInSpeech/CDROM/Literature/LOTwinterschool2006/szabo.best.vwh.net/smart_contracts.html (Accessed: 27 September 2025).

Szabo, N. (1994) 'Smart Contracts'. [Online]. Available at: https://www.fon.hum.uva.nl/rob/Courses/InformationInSpeech/CDROM/Literature/LOTwinterschool2006/szabo.best.vwh.net/smart_contracts_2.html (Accessed: 27 September 2025).

Szabo, N. (1997) 'The Idea of Smart Contracts'. [Online]. Available at: https://nakamotoinstitute.org/the-idea-of-smart-contracts (Accessed: 27 September 2025).

Tangem (n.d.) *Script – Glossary* [Online]. Available at: https://tangem.com/en/glossary/script/ (Accessed: 27 September 2025).

Turányi, N. (2018) '„Okos szerződések" avagy okos életünk következő lépcsőfoka'. *Jogtudományi Közlöny*, 2018.

Walport, M. (2016) 'Distributed ledger technology: Beyond block chain', *UK Government Office for Science*, London, Tech. Rep., January. Available at: https://www.gov.uk/government/publications/distributed-ledger-technology-beyond-block-chain (Accessed: 27 September 2025).

Wang, S., Ouyang, L., Yuan, Y., Ni, X., Han, X. & Wang, F.Y. (2019) 'Blockchain-enabled smart contracts: Architecture, applications, and future trends', *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 49(11), pp. 2266–2277. https://doi.org/10.1109/TSMC.2019.2895123

Welmann, Gy. (2014) *A szerződések általános szabályai az új Ptk.-ban – II. rész*. Budapest: Orac

Yaga, D., Mell, P., Roby, N. & Scarfone, K. (2018) *Blockchain Technology Overview*. Gaithersburg, MD: National Institute of Standards and Technology (NIST), NISTIR 8202. https://doi.org/10.6028/NIST.IR.8202