Balázs Arató – Tamás Sajben

The connection points between smart contracts and artificial intelligence – Opportunities and challenges

Abstract

This study explores the opportunities and challenges presented by smart contracts and their technological foundation, the blockchain. It details the history and mechanics of blockchain technology, emphasizing its decentralized, immutable, and tamper-proof nature. The paper delves into the concept of smart contracts, tracing their origins to Nick Szabo and their evolution through the Ethereum blockchain. It highlights the role of oracles in enabling smart contracts to interact with the physical world and addresses various practical issues such as immutability, code errors, and the legal implications of smart contracts. The study also examines the integration of artificial intelligence (AI) in smart contracts, discussing how AI can function as an oracle to provide reliable information and support the contractual process. By examining different types of blockchains and smart contracts, the study provides insights into their potential applications and the inherent limitations and risks associated with their use, particularly in terms of legal enforcement and jurisdiction.

Keywords: smart contracts, blockchain technology, distributed ledger, immutability, oracles, AI oracles, Ethereum, legal implications, legal nature of smart contracts, code errors

I. Technological foundation of smart contracts – the blockchain technology

a) Brief history of blockchain

In 2008, Satoshi Nakamoto issued the so-called Bitcoin Whitepaper (URL1), a technology and business plan describing in detail how the first and one of the most important blockchain networks, the Bitcoin blockchain works.

b) Definition and technological side of blockchain

A blockchain is a decentralised, distributed electronic database (”ledger”), which is structured as a chain of consecutive – as a rule – tamper-proof and unalterable blocks.

The obvious question arises as to what is the basis on which the transactions on the blockchain are carried out. The answer to this is the protocol governing the blockchain, which is accepted by the participants

when from the outset, and which is able to ensure the operation of the system in a quasi-self-regulating way.

The participants are called *"nodes"*, which are in the physical sense computers, but would not be meaningful in themselves without related human decision-making, so nodes can also be determined as the people operating them. In the case of so-called public blockchains (see later), a node can be anyone who downloads and installs the blockchain software and starts running it on their computer. The blockchain requires a high amount of electronic storage space, which leads to distinguishing two main types of nodes. Full nodes store the full version of the blockchain starting from the first, so-called genesis block, while lightweight nodes store only a simplified version of the blockchain (block headers) on their computers, saving significant space on the storage device (URL2).

The third category of nodes are the so-called "miners", which, in addition to storing the entire blockchain, are also actively involved in its expansion. Put simply, miners validate subsequent blocks with the transactions in them (a block usually contains several transactions), continuously extend the chain by generating new blocks (URL3). While transactions are awaiting validation, miners compete to be the first to find the solution to a mathematical problem in order to create the next block and to submit it to and through the nodes of the system, thereby extending the blockchain. This is the so-called proof-of-work consensus mechanism used by the Bitcoin blockchain, whereby one party (the miner) has to perform a complex calculation, while sacrificing resources to receive something in return, and the other parties (the nodes) can easily and quickly verify that the reward has been earned (URL4). The "winning" miner is rewarded with newly minted bitcoins,[1] plus a transaction fee paid by the parties to the transaction (URL5).

How does the blockchain look like? As mentioned above, the first block is the so-called genesis block. Subsequent blocks, which can be extended indefinitely, contain, among other things, the details of the transaction and a hash, which can be defined as the unique fingerprint of the block. The hash is a security function, which is used to verify whether the content of the block has been tampered with or changed in any possible way. In addition to its own hash, a block also contains the hash of the block immediately preceding it (URL6).

The immutability and tamper-proofness of blocks and the transactions they contain are made possible by a combination of the above technological solutions. Any attempt to change a block would require not only the specific

[1] From April 2024, the reward of the miners is 3,125 BTC per block.

block to be changed, but also the contents of the blocks that follow it in the chain, which are often large in number. Changing an existing block would also change the hash associated with that block. When checking the validity of a given block, it would be immediately apparent that the hash is not the one related to the data stored in the block. As the hash of each block is used to generate the hash of all subsequent blocks, changing a block will also change the hash of subsequent blocks.

c) Different kinds of blockchain

In terms of participation in the blockchain network, we distinguish between three main categories, based on the right to read the content of blocks and the right to add and control them.

This classification distinguishes between public, private and consortium blockchains. A typical example of a public blockchain is the above-mentioned Bitcoin blockchain, in which anyone can participate as a node and even as a miner once the software has been installed. In the case of private and consortium blockchains the principle of decentralisation is inadvertently undermined, as the granting of participation and the necessary rights presupposes central actor(s) with additional rights. In the private blockchain, a closed circle, typically members of an organisation or a profession, participates. A consortium blockchain differs in that the participants are generally composed of members of several organisations (URL7).

A second classification distinguishes between permissionless and permissioned blockchains. The former is often associated with the public blockchain and the latter with the private and consortium networks, but this is not necessarily true in all cases, although the pairing most often occurs in such constellations. In practice, public, private and consortium blockchains can all be created in a permissioned or permissionless form. In the case of a permissioned blockchain, the ability to control and/or add transactions to the blockchain is restricted for certain nodes or a certain set of nodes, whereas in the case of a permissionless blockchain, there are no such restrictions.

We can see that in the case of private and consortium blockchains, not all of the elements/principles indicated above are fully implemented, especially the principle of decentralisation, it is therefore questionable whether the blocks in them are truly immutable. However, this does not and should not diminish the practical benefits of these systems.

In addition, it should be borne in mind that the use of blockchain – which is also true for smart contracts – is not always the most optimal solution.

Often, the novelty and modernity of the technology is an excellent marketing tool ("hype"), emphasising the innovative approach of the economic operator concerned, but it is always necessary to consider whether the objective can be achieved more effectively with classical centralised solutions.

II. Basics of smart contracts

a) The beginning of smart contracts and definition

Smart contracts have become widespread thanks to the second largest blockchain, the Ethereum. Like the Bitcoin, this term does not only cover a kind of blockchain, but also currently the second most significant cryptocurrency, the Ether (URL8). In order to further exploit the potential of the blockchain technology, it became necessary to create an additional "layer" within the system. Thanks to this layer, in Ethereum, it is possible to create smart contracts (URL9).

However, the inventor of smart contracts was not Buterin, but a Hungarian-born American computer scientist-cryptographer, Nick Szabo, who first created the concept of smart contracts in the mid-1990s (URL10). According to Szabo, smart contracts are computer codes that fulfil contractual conditions. He envisaged that this would reduce the need to rely on trusted third parties, and that automatic performance will reduce the risk of abuse and will result in lower costs of performance.

Similarly to blockchain, there is no officially recognised definition of smart contracts, which can also be defined by their properties, mostly based on Nick Szabo's concept presented above. In other words, a smart contract is a computer code that is created between two or more parties, and when the event(s) specified in the code occur, the contractual obligations will be automatically fulfilled. However, a significant difference with Szabo's definition is that smart contracts in the 21st century are running on the blockchain, which ensures that they are immutable and tamper-proof. While immutability is certainly to be welcomed, since it is in the interest of the contracting parties that their agreement should not be falsified, it virtually precludes the possibility of amending the contract in the event of a change in the circumstances and the parties' relations, which is far from being positive.

b) Oracles – Communication of smart contracts with the outside world

Although an increasing part of our lives takes place in cyberspace, the physical world still remains the most relevant area which cannot be ruled out.

To this end, it is essential to ensure that smart contracts are able communicate effectively and securely with the physical space.

The means of doing this are the so-called oracles which/who provide proof that the conditions programmed into the smart contract have been fulfilled (URL11). The oracles can be natural persons, for instance notaries, who, by virtue of their official nature and obligations of neutrality and impartiality, fit perfectly into the oracle role. Besides, software, websites, databases and registries can also function as oracles, just as hardware (e.g. sensors) that are able to communicate with each other and with the specific smart contracts thanks to the Internet of Things (IoT) phenomenon (URL12).

The essential characteristics of oracles are the ability to credibly attest the fulfilment of conditions off-chain (i.e. outside the blockchain) and to connect to the smart contract on the blockchain. In case of use of centralised oracles, there is a risk of failure if they get hacked, manipulated, or shut down, leading to the circumvention of the contractual conditions by any of the parties. In order to tacke this issue it is possible to programme several oracles (decentralised oracles) at once into the smart contract and also to specify a "yes/no ratio" for conflicting information to be accepted (URL13).

In addition to all this, it is important to underline that the – often voiced – presumption of authenticity of the data registered in blockchain and smart contracts is in itself wrong, and any recognition of this in law or in practice could lead to significant harm. Technology alone does not make data authentic. In order for this data to be considered authentic, it is essential that oracles are involved, whereby data from the physical world is passed through a reliable "filter" into the on-chain space.

In the case of smart contracts, the role of the oracle can even be fulfilled by artificial intelligence. Generally, artificial intelligence as an oracle can be extremely useful in many aspects throughout the entire contractual process. Primarily, the "if-then function" is notable, which can assist the parties in cases where a legal consequence is tied to the occurrence of an objective condition (e.g., weather parameters, exchange rate changes, price developments, inflation, election or sports results, etc.) under the contract. In such cases, the artificial intelligence oracle provides reliable information about the occurrence of these objective conditions to the blockchain platform, thus triggering the legal consequence tied to the condition in the contract.

An example worth mentioning is when the parties are negotiating off-chain, and the AI oracle supports the formation of the smart contract by translating natural language into computer language. The AI oracle can also function to oversee contractual performance, establishing, for instance, non-

performance by the obligor and, within certain limits, the reason for it. If, for example, the subject of the contract is not delivered, the AI oracle examines whether the non-performance occurred within the obligor's sphere of interest or, for instance, was due to force majeure, and then forwards the result of the examination to the blockchain platform. This information either triggers the contractual consequence of non-performance (e.g., a penalty) recorded in the smart contract or prevents it from taking effect, considering the unavoidable external cause. Finally, the AI oracle can also serve to modify the smart contract in cases where unforeseen changes in circumstances occur that necessitate or justify modifying the contract. In such cases, the AI oracle sends this information to the blockchain platform, enabling the smart contract to be modified before self-execution. Of course, other uses of the AI oracle are conceivable; only the most typical ones have been mentioned above.

We can differentiate between software and hardware oracles. While the former can relay any online accessible information to the blockchain system, the latter are connected to and gather information from the physically perceptible world. The literature also distinguishes between outbound and inbound oracles, depending on whether the oracle relays information from an external source to the blockchain platform or vice versa, from the blockchain platform to the physically perceptible world. Lastly, it is worth mentioning the distinction based on whether the oracle merely relays information from the physically perceptible world or also performs calculations and transmits their results to the blockchain platform. The former are referred to in the literature as data carrier or automated oracles, while the latter are known as computation oracles. A good example of a calculation-performing oracle is when someone's creditworthiness must be assessed based on various criteria, and if the individual is deemed creditworthy, this automatically triggers the loan disbursement.

Generally, regardless of how artificial intelligence is used, the so-called black box effect arises, meaning that in some cases, the decision-making processes or operational mechanisms of AI systems are not transparent or comprehensible to human users, and often not even to developers. If artificial intelligence serves as an oracle in a smart contract, this phenomenon poses considerable risks, as the oracle may provide inaccurate or false information to the blockchain platform, unjustifiably triggering or preventing a contractual consequence. This phenomenon, referred to in the literature as the oracle problem, can be addressed by so-called consensus oracles. This refers to a decentralised network of oracles that work together to achieve the same goal, meaning they reach a result on the same issue, but not the individual results, for example, their average, is the information relayed to the blockchain platform

and triggers a contractual consequence in the smart contract. However, it is emphasised that the use of consensus oracles does not eliminate the problem described above but only reduces the likelihood of severe functional disorders. Examples of this include the unjustified fulfilment (self-execution or self-performance) of the smart contract or its non-fulfilment due to AI error (the failure of self-execution or self-performance), as well as the formation or non-formation of the smart contract if the contract is created by AI. An example of an AI-created contract is the AI's assessment of creditworthiness, depending on the result of which the loan or credit agreement is either formed or not.

Thus, the improper functioning of the artificial intelligence oracle can cause numerous problems and raises the issue of legal liability from multiple perspectives. The literature has proposed various views and approaches regarding who bears the risk of damage and who should be liable if artificial intelligence acts as an oracle in a smart contract and a malfunction occurs, resulting in harm to one of the parties. Without delving into the possible forms of legal liability in detail here, it is merely noted that analogies based on the liability of parents and animal keepers have even appeared in connection with the AI service provider's or AI owner's liability. (Papadouli and Papakonstantinou: 2023) If a single guiding principle were to be established concerning the complex liability and risk-bearing issues associated with artificial intelligence oracles, it would likely be the emerging principle that the party fundamentally responsible for any disruptions endangering the proper execution of a smart contract is the one who requested the involvement of the artificial intelligence as an oracle, or the one obligated to ensure the correct and flawless operation of the artificial intelligence. In summary, the liability issues related to smart contracts are particularly unique because the legal relationships involve entities that, although not parties to the contracts themselves, can significantly influence them through their actions. These entities either provide the platform for the execution (performance) of the contract or facilitate or hinder its completion by transmitting information. Therefore, legal science cannot approach these liability issues from traditional perspectives; instead, innovative thinking and highly multifaceted analysis are required to develop clear and fair solutions for these entirely new legal disputes. It is also possible that artificial intelligence will play a key role in mapping out all possible variations and potential scenarios requiring resolution.

c) Is a smart contract really a contract?

The subtitle above raises the following two important questions: 1. Are smart contracts really smart? 2. Are they contracts at all?

As determined above, a smart contract is nothing more than computer code written by humans that performs predetermined actions when predetermined conditions are met. Both the conditions and the actions are therefore determined by the human programmer, not by the contract/code. The smart contract therefore has no intelligence or creativity to create for itself the obligations to be fulfilled and the related conditions, nor does it shape the code that has already been programmed.

It is much more difficult to give a clear answer to the second question. In the case of smart contracts, we are dealing with a global phenomenon which did not emerge in a particular country or its legal system and become cross-border in nature, but which developed in a cross-border environment from the outset. With smart contracts, individual states and communities of states (including the European Union) have been given a ready-made product that is an integral part of everyday reality and which they have had to understand and define how it fits into their legal system. Nevertheless, with rare exceptions, individual states have not been able to develop a specific regulatory environment quickly and effectively. As a consequence, it is necessary to assess whether smart contracts constitute a contract in a given country on the basis of the current rules of contract law and, if so, whether they fulfil the legal requirements of a written form which is a condition of validity for several transactions.

As smart contracts are relatively new and still evolving developments in law and technology, there is no consensus regarding their legal nature. Some opinions hold that smart contracts cannot be considered contracts in the legal sense but are merely computer protocols that facilitate the execution (performance) of already established agreements. In contrast, another viewpoint asserts that smart contracts are indeed contracts capable of entirely replacing traditional agreements. There is also an opinion that questions the contractual nature and character of smart contracts primarily based on whether the coding and execution of the smart contract accurately and completely reflect the parties' intentions, which is an indispensable conceptual element of contracts as per the Civil Code. (Werbach and Cornell: 2017) This issue may particularly arise for contracting parties who are unfamiliar with the technology used in smart contracts or do not understand the programming language involved.

d) Self-enforcement

In addition, it is crucial to clarify how enforcement relates to smart contracts, mainly for terminological reasons. Smart contracts, by virtue of their

automated nature, their immutability and unforgeability, conceptually exclude the possibility of the breach of contract. Enforcement can take place in case of breach of contract according to Hungarian (and many other national) legal terminology. Nevertheless, in several cases we hear/read about the automatic enforcement or self-enforcement of smart contracts. The root of the problem lies in the English terminology commonly used to describe the characteristics of smart contracts. In English, often the term "self-enforcing or self-executing contract" is used. In reality, however, we are talking about self-performing agreements. Therefore, in the English terminology, it would be more appropriate to use the term "self-performing contract" or "self-fulfilling contract" which could lead to the avoidance of misunderstandings.

III. Key practical issues related to smart contracts

It should be noted at the outset that the questions that arise cannot always be answered in a clear-cut way. With the exception of a few countries where regulation has already taken concrete shape, the answers are mostly theories developed by academics and researchers, which may however serve as perfect basis for future regulation.

It may be worth comparing the smart contract with the concept of a contract under the Hungarian Civil Code (Ptk.) and the recognised methods of contract formation in the Civil Code to get closer to answering the question of whether a smart contract is a contract under Hungarian law. According to Section 6:58 of the Civil Code, a contract is a mutual and concurrent declaration of will by the parties, creating an obligation to perform a service and a right to claim the service. A contract is formed through the mutual and concurrent expression of the parties' will, either orally, in writing, or through conclusive conduct. For certain types of contracts, the Civil Code requires written formality and validity (e.g., contracts affecting the ownership rights of real estate). Implicitly, the Civil Code also defines the requirement of written form by clarifying that a declaration must be considered in writing if it is made in a form suitable for recalling the content of the declaration unchanged, identifying the person making the declaration, and the time of making the declaration. Consistent judicial practice considers electronic signatures that meet the legal requirements as such, while simple email exchanges are regarded merely as conclusive conduct. (In summary, it can be stated that an email does not meet the written requirements necessary for unequivocally determining the identity of the declarant, and thus a legal statement made via electronic mail cannot be considered as being in written

form. This is because, instead of verifying the identity of the sender, it can at most be established from whose electronic mailbox the email originated. This remains true even if the email includes a so-called embedded signature. The relevant judicial practice is now considered uniform: contracts are formed through conclusive conduct as a result of an offer and acceptance made via email. Compared to other forms of behaviour reflecting the intent to enter into a contract, the sole advantage of conclusive conduct conveyed by electronic mail is that the content of the agreement can be more easily proven later.) In line with the Civil Code's spirit and technological neutrality, in principle, the smart contract can meet the requirement of written form, meaning it is conceivable that smart contracts will play a role in simpler real estate transactions in the future, initially in ensuring the execution of contractual content. However, as long as it is necessary to submit paper-based copies of contracts to the land registry authority, smart contracts may only play a supplementary supporting role in real estate transactions, but this could undoubtedly increase security.

a) Possible areas of application of smart contracts

The first question to be asked is which transactions can be covered by smart contracts at all. To answer this question in concrete terms, we must examine the contract law of the country in question, both in respect of material and formal validity requirements. In case if the content of a smart contract is in conformity with the law of a given country, it can be a valid contract (URL14). However, numerous states provide for formal validity rules for specific transactions.

If we move away from the national legal requirements that may stand in the way of the validity of smart contracts, we need to take into account the essential differences between human language and code language. Human language is characterised by its flexibility, which allows us to express our ideas in a nuanced way. In the context of contract law, this means that we can put almost any provision into a contract which is in conformity with the applicable law. The wording of our contracts often allows for multiple interpretations, precisely because of the diversity of human language. In contrast, computer code is a rigid, exact "language" where interpretation has no role, or if it does have some, it is extremely limited.

Due to the rigidity of the code language, smart contracts are best suited to agreements that are highly constrained even when written in human language, and where the parties have little or no desire to leave room for different interpretations. This is one of the reasons why smart contracts first appeared in financial transactions, which traditionally contain rigid provisions.

One of the main limitations of smart contracts is their inability to handle abstractions such as good faith, which often require human interpretation, context-dependent decision-making, and flexible judgement that code-based systems currently cannot provide. Smart contracts, functioning as program code, automatically execute based on pre-defined conditions and events. Consequently, these contracts rigidly follow the code and cannot consider factors such as the parties' intentions, the context of the situation, or the behavioural requirements posed at a fundamental level by civil law.

b) Jurisdiction and applicable law

As mentioned above, both the blockchain technology and smart contracts are a global phenomenon, with "dispersed", in most cases anonymous, nodes around the world, and with the possibility that the contracting parties are not located in the same country. The legislators are therefore faced with the task of classifying and/or regulating a phenomenon which, due to its inherently cross-border nature, is not subject to purely domestic experience.

One of the first steps in a judicial procedure is to establish whether we are facing a cross-border agreement at all and, if so, to determine the jurisdiction and the applicable law. It is important to stress, however, that the mere fact that a given smart contract runs on the computers of nodes all over the world does not in itself mean that the given smart contract is a cross-border agreement [ELI Principles on Blockchain Technology 4 c)]. In this context, it can be useful for the contracting parties to consider choosing the applicable law if there is no mandatory legal provision which excludes this option (e.g. for real estate transactions). The choice of law can help avoiding difficulties that may arise in the absence of choice, especially when the parties choose the laws of a legal system having introduced specific and clear rules for smart contracts.

c) The immutability of smart contracts

As explained above, smart contracts are based on blockchain technology, which conceptually precludes the subsequent reversal or modification of blocks and transactions. On the one hand, this is positive, as immutability also leads to tamper-proofness, contributing to the security of transactions through technological means.

However, one also has to consider the negative side to his kind immutability. Relations between the parties are in numerous cases dynamic, just as the parties to a transaction and the circumstances. In longer-term relationships, the need to modify the contract is common, but may not materialise if the

parties have agreed in a smart contract. For foreseeable events, parties can program changes into their smart contracts from a fixed date which will be carried out automatically, but in unforeseeable situations requiring quick actions, smart contracts are not the most convenient choice. The solution to this issue may be the use of such underlying blockchain technology which allows the subsequent modification of the blocks or the implementation of technically feasible reverse transactions (ELI ELI Principles on Blockchain Technology, Principle 10.).

Immutability can also lead to conflicts with consumer protection rules. Just to highlight one of these rules: in the case of distance contracts, the 14-day right of withdrawal granted to the consumer by Article 9 of the above-mentioned Consumer Rights Directive (Directive 2011/83/EU of the European Parliament and of the Council of 25 October 2011 on consumer rights, amending Council Directive 93/13/EEC and Directive 1999/44/EC of the European Parliament and of the Council and repealing Council Directive 85/577/EEC and Directive 97/7/EC of the European Parliament and of the Council Text with EEA relevance). It is precisely the immutability of smart contracts that could prevent the consumer's right of withdrawal from being exercised, thus lead to the situation when the smart contract does not comply with the relevant legislation on consumer protection.

d) Code errors

Smart contracts may be affected by code errors or bugs. A code bug can result in significant financial losses for the parties to the contract and raises serious liability issues. One of the most serious code bugs in the past concerned the Ethereum system, more precisely the so-called The DAO (Decentralised autonomous organisation, details of which will be described later). In the case of The DAO, the smart contract contained a code bug which was exploited by some participants to cause a loss of 3.6 million Ethereum (USD 50 million at the time) to the community. Crucially, there was no consensus among the h participants in The DAO on how to proceed in such a case. The majority of those who adhered to the principles of smart contracts believed that the status quo should not be touched, as "the code is law". The majority, however, was of the opinion that the damage had to be remedied, and in order to do so, breached the principle of immutability by consensus (URL15). This led to the so-called hard fork of the Ethereum blockchain, resulting in the genesis of the Ethereum classic and Ethereum. The former is effectively a renaming of the cryptocurrency still used in the previous system, while Ethereum is used by the majority who accept the changed state (which includes the refunded damage) (URL16).

In the event of a code error, it is essential that it is fixed quickly to ensure the reliability of the system and to avoid user abandonment. The question is: who is responsible for these errors and the resulting damage: the programmer, the nodes, or the contracting parties? The programmers are certainly the ones who should bear the responsibility, but their actual liability and the obligation to pay damages is questionable because of their frequent anonymity and the problem of establishing jurisdiction and applicable law in the event of a breach of anonymity. This results in that in many cases the only means of remedying the problems is to take a decision similar to that in The DAO case.

Another important factor is that most of the software used to create smart contracts is open source, which means that – within the limits set by its creator – anyone is free to develop it further. In the course of such development, a code error in the program further complicates the assessment of liability.

e) A Linguistic Perspective on Smart Contracts, Some Specific Issues of Interpreting Smart Contracts

The examination of smart contracts from a linguistic perspective can be particularly enlightening, as their emergence introduces new contexts for the long-established civil law principles developed over centuries regarding the interpretation of contracts written in natural human language. According to the current Hungarian Civil Code, a legal statement, in case of a dispute, should be interpreted in a manner that considers the intended meaning that the recipient, given the declarant's presumed intent and the circumstances of the case, was expected to understand based on the generally accepted meaning of the words. Judicial practice has further refined this main rule by considering factors such as what an average educated consumer might have understood when concluding a particular contract or reading its text, or what meaning one party might have attributed to a particular expression depending on their own level of education.

One might think that such issues do not arise in connection with smart contracts since they are written not in human language but using programming codes. Nonetheless, there can be situations where interpretation becomes necessary, although smart contracts fundamentally aim to execute agreements between parties in an automatic and unambiguous manner. Smart contracts are written in code, which must be executed according to precisely defined rules. However, coding errors, misinterpretations, or a lack of proper understanding of business logic can cause problems.

Although smart contracts operate independently from a technical standpoint, they can still be part of traditional contracts and fall under local

legal regulations. Consequently, in case of a dispute, courts might need to interpret what the code actually meant to the parties. Did the parties understand what specific operation the code performed? When drafting smart contracts, it is not always clear what (business) intention the different parts of the code reflect. If a dispute arises over the meaning of certain parts of a smart contract, the documents related to the contract preparation, emails, and other communications can be relevant during interpretation.

Errors made during the drafting of a smart contract or unforeseen events occurring in the meantime can also raise interpretive issues, especially if it is not clearly defined what should happen in such cases. It is evident that a different dimension of interpretation opens up in the case of smart contracts, and this interpretation's starting point will not be the linguistic understanding of the contract text, but rather inferring the parties' intentions based on other specific aspects.

In conclusion, although the purpose of smart contracts is to minimize interpretive issues through automatic execution, in reality, situations can still arise where communication and intention between the parties play a crucial role in the precise interpretation of the contract. If linguistic interpretation based on the generally accepted meaning of words is necessary at all, it is limited to the statements and declarations of intent made during the negotiations preceding the contract conclusion.

IV. Categorisation of smart contracts

Generally, we can distinguish four main variants of smart contracts, however, this classification is not official and there may be many more situations and perspectives for the smart contract classification.

The first variant is the smart contract created purely as a code, without the use of human language. In this case, the question arises: is the code itself law?

In the case of the second type, the code is only a means of automatic fulfilment of a legal agreement off-chain and written in human language. In this case, the smart contract is not a contract, but a tool of performance alongside the human language contract.

In the third version, the contract written in human language is converted into a smart contract, i.e. a code. In this case, the parties aim to take advantage of blockchain technology and automatic fulfilment. The risk of this variant lies in the differences between the two (rigid code and flexible human) languages, as presented above. Not every nuance of the human language

can be clearly transformed into the code language, so there is a real risk that the code will contain something other than the parties' original intentions as expressed in human language. Theoretically, it is possible that the parties take the human-language contract as the prevailing one if this were to be discovered, but the unstoppable self-fulfilment of a smart contract makes it virtually impossible to stop the performance process.

The last variant is the hybrid smart contract, in which some provisions appear off-chain as a contract written in human language, and other provisions are on-chain in code, intended for self-fulfilment. This solution allows the parties to capture those concepts and provisions that leave open multiple interpretations and are thus difficult to convert into code (e.g. the force majeure is clearly one of them) in the off-chain world, and to place those that are suitable for adoption in self-fulfilling code in the on-chain environment.

Another but in many respects similar approach distinguishes three current forms of smart contracts. The first group includes smart contracts that serve to execute the performance of agreements established through other means, effectively fulfilling a security function. The second category encompasses hybrid contracts, where the contract is partially formed as a smart contract. The third form is the standalone smart contract, where all elements of the parties' legal relationship are contained within it. (Papadouli–Papakonstantinou 2023)

V. Some examples of smart contracts from the real world

a) Non-fungible tokens

Non-fungible tokens (or NFTs) are assets tokenized within the blockchain through a process called minting. Tokens are unique identification codes, stored on the blockchain with the corresponding assets elsewhere. NFTs many times take the form of pieces of art but their use is much broader than the field of art (e.g. also real estate can be tokenized). By having private keys to the given NFT, the rights joint to it can be exercised. The NFTs have a unique identifier associated with a blockchain address (URL17).

As shown by their name, NFTs are not fungible, irreplaceable, but are definitely tradeable on the blockchain through special marketplaces like the OpenSea (URL18). Often, the question arises, what the sense of an NFT artwork is in the era where digital files can be easily copied. The value of the NFTs is largely attributed to the originality, i.e. the related private key implies that its owner is the owner of the original digital file. This is not different in the physical world either: for instance, a famous painting of Rembrandt can

also be easily copied but the copy – even if of perfect quality – will not give any value close to that of the original one. However, compared to the physical world, NFTs include a digital and blockchain-based certificate attesting their originality, and as we saw it above, the blockchain technology provides the advantage of tamper-proofness.

The process of minting several times entails smart contracts which assign ownership and make possible the secure transfer of NFTs. There exist several smart contract blockchains with NFT creation tools (e.g. TRON, Tezos). The advantages of smart contracts for minting and trading NFTs are multifold. Smart contracts can automatically check the time and place of creation of NFTs and they can help prove ownership and its change through the use of the above-mentioned digital certificate (URL19). In some cases, smart contracts can be useful to guarantee the collection of resale royalties (within the field of copyright, resale rights entitle artists to a share of the sale price when their artwork is resold) after the subsequent sales of their NFT's (URL20).

b) Initial Coin Offering

In the financial-investment area, smart contracts are a common manifestation of so-called Initial Coin Offerings (ICOs). In an ICO, the initiating company raises a fixed amount of cryptocurrency from investors in order to launch its own token platform. This platform is set up to achieve a specific goal, just like a classical company.

Tokens can be divided into several types. Utility tokens can be used to access an online service and/or product developed or designed by the initiators of ICO projects. In addition, tokens can also be used as an investment and the profits generated can be distributed as a quasi-dividend to the token holders (security tokens).

The first ICO took place in 2013 and was announced by the then Mastercoin (URL21). However, in their current form, they have spread through the Ethereum blockchain, thanks in part to the system's ability to create smart contracts. A smart contract collects the required amount of cryptocurrency, automatically issues the corresponding number of tokens in exchange, later "manages" them, and is able to pay commissions and dividends.

It should be stressed that most ICO projects do not have a real product or service behind them, just an idea. Their advantages are speed, avoiding the administrative and bureaucratic burdens of the stock exchange and the transparency provided by blockchain technology. As Ethereum, for example, is a public blockchain, investors can track the use of the cryptocurrency they collect.

c) Decentralised autonomous organisations

Another practical example of smart contracts worth mentioning is the DAO, or decentralised autonomous organisation. The DAO scandal, which was the result of a code error and led to a hard fork of the Ethereum blockchain, was mentioned above. But what is the DAO as a phenomenon and a legal institution?

A DAO is a complex network of smart framework contracts and smart contracts into which the structure and operation of a virtual organisation is programmed. The "life" of a pure DAO takes place entirely in the on-chain world, without, for example, a physical seat or registration in national company/legal entity registers although their State registration is – depending on the fulfilment of legal conditions, like the form – possible under some jurisdictions (e.g. Wyoming, Gibraltar, Switzerland) (URL22). By joining, members receive tokens which grant them voting rights. The decisions are automatically implemented by the smart contract according to the rules it sets out.

In practice, transparency is a positive feature of DAOs, but the lack of hierarchy can lead to slow decision-making processes due to the large number of actors involved. The main risk, however, is legal uncertainty due to the lack of legal regulation in most of the countries globally in respect of the legal form, legal personality and liability.

References

ELI Principles on Blockchain Technology, Smart Contracts and Consumer Protection, Report of the European Law Institute, Principle 4 c).

ELI Principles on Blockchain Technology, Smart Contracts and Consumer Protection, Report of the European Law Institute, Principle 10.

Kevin Werbach – Nicolas Cornell 2017. Contracts Ex Machina, *Duke Law Journal,* Volume 67, pp. 313–82.

Vasiliki Papadouli – Vagelis Papakonstantinou 2003. A preliminary study on artificial intelligence oracles and smart contracts: A legal approach to the interaction of two novel technological breakthroughs. *Computer Law and Security Review,* Volume 51.

Internet Resources

URL1. Nakamoto, S. (2008) Bitcoin: A Peer-to-Peer Electronic Cash System. https://bitcoin.org/bitcoin.pdf (Downloaded: 1 April 2024)

URL2. Dr. Ravi Chamria, What are lightweight nodes in blockchain? Advantages and Disadvantages, https://www.zeeve.io/blog/what-are-lightweight-nodes-in-blockchain/ (Downloaded: 1 April 2024)

URL3. What are the Types of Nodes in Blockchain? https://utimaco.com/service/knowledge-base/blockchain/what-are-the-types-of-nodes-in-blockchain (Downloded: 1 April 2024)
URL4. Scott Nevil, What Is Proof of Work (PoW) in Blockchain?, https://www.investopedia.com/terms/p/proof-work.asp (Downloaded: 1 April 2024)
URL5. Bitoin.com, What are Bitcoin network fees? https://www.bitcoin.com/get-started/what-are-bitcoin-network-fees/ (Downloaded: 1 April 2024)
URL6. Bitpanda, What is a hash function in a blockchain transaction? https://www.bitpanda.com/academy/en/lessons/what-is-a-hash-function-in-a-blockchain-transaction/ (Downloaded: 1 April 2024)
URL7. Shobhit Seth, Public, Private, Permissioned Blockchains Compared, https://www.investopedia.com/news/public-private-permissioned-blockchains-compared/ (Downloaded: 1 April 2024)
URL8. https://ethereum.org/en/whitepaper/ (Downloaded: 1 April 2024)
URL9. The most popular programming languages for the creation of smart contracts are: Solidity and Vyper, https://chain.link/education-hub/smart-contract-programming-languages (Downloaded: 5 April 2024)
URL10. Nick Szabo "Smart Contracts" (1994) and Smart Contracts: Building Blocks for Digital Markets (1996)
URL11. https://ethereum.org/en/developers/docs/oracles/ (Downloaded: 5 April 2024)
URL12. Smart contracts and IoT, https://www.geeksforgeeks.org/smart-contracts-and-iot/ (Downloaded: 5 April 2024)
URL13. Cem Dilmegani, Guide to Oracles in 2024: What Are They, Types & Use Cases, https://research.aimultiple.com/blockchain-oracle/ (Downloaded: 5 April 2024)
URL14. Example of Liechtenstein. https://www.mondaq.com/technology/935298/blockchain-comparative-guide (Downloaded: 5 April 2024)
URL15. Samuel Falcon, The Story of the DAO – Its History and Consequences, https://medium.com/swlh/the-story-of-the-dao-its-history-and-consequences-71e6a8a551ee (Downloaded: 8 April 2024)
URL16. Ethereum Classic and the Ethereum Hard Fork, https://help.coinbase.com/en/coinbase/getting-started/crypto-education/eth-hard-fork (Downloaded: 8 April 2024)
URL17. Rakesh Sharma, Non-Fungible Token (NFT): What It Means and How It Works, https://www.investopedia.com/non-fungible-tokens-nft-5115211 (Downloaded: 8 April 2024)
URL18. https://opensea.io/
URL19. All You Need to Know About NFT Smart Contracts, Binance Blog, https://www.binance.com/en/blog/nft/all-you-need-to-know-about-nft-smart-contracts-5687454135877030 85 (Downloaded: 8 April 2024)
URL20. Crypto Council for Innovation, How NFT Royalties Work – and Sometimes Don't, How NFT Royalties Work – and Sometimes Don't, https://

cryptoforinnovation.org/how-nft-royalties-work-and-sometimes-dont/ (Downloaded: 8 April 2024)
URL21. Ruben Merre, ICO 101 — History of Initial Coin Offerings (ICOs), https://medium.com/hackernoon/ico-101-history-of-initial-coin-offerings-icos-part-1-from-mastercoin-to-ethereum-4689b7c2326b (Downloaded: 8 April 2024)
URL22. Nestor Dubnevich, The Best Entities and Countries for DAO Registration in 2024, https://legalnodes.com/article/choose-a-crypto-friendly-country-for-dao (Downloaded: 8 April 2024)

Balázs Arató
Habilitated Associate Professor
Károli Gáspár University of the Reformed Church in Hungary, Faculty of Law, Department of Commercial and Financial Law/Department of Civil Law
E-mail: arato.balazs@kre.hu
https://orcid.org/0000-0001-5854-8018

Tamás Sajben
Head of the Brussels Representation at Hungarian National Chamber of Notaries
E-mail: sajben.tamas@kamara.mokk.hu