Regulation of the Smart Contract in (Romanian) Civil Law

**Raluca ONUFREICIUC¹,
Lorena-Elena STĂNESCU²**

¹ Lawyer, Ph.D. candidate at the University of Bucharest and the University of A Coruna, Bucharest/A Coruna, Romania/Spain, teaching assistant Faculty of Law and Administrative Sciences, "Ștefan cel Mare" the University of Suceava
raluca.onufreiciuc@fdsa.usv.ro
onufreiciuc.raluca@drept.unibuc.ro
raluca.onufreiciuc@udc.es

² Ph.D. Lawyer
lorena@stanescuandpartners.com

Abstract: The research aims to organize, examine, and analyze the provisions on smart contracts available in Romanian civil law. "Smart contracts" are not smart, and are not necessarily contracts, although they can be.

As self-executing computer programs, smart contracts are operational on the blockchain and unlike traditional legal contracts, once the agreement has been concluded and the smart contract is set in motion, no party can intervene and it will be executed without interruption, modification, or breach. The crucial question in the final contract law topic is what happens when the smart contract's outcomes deviate from those required by law. To answer this issue, we must first understand that whether a smart contract becomes legally enforceable is determined by several circumstances, together with the unique use case, the type of smart contract employed, and the existing legislation.

The paper addresses the subject of determining and regulating smart contracts under Romanian current laws. Particular emphasis is placed on two ambiguous definitions of smart contracts: as computer code and as a civil-law contract. The authors conclude that the concept of smart contracts requires more legal regulation, particularly in terms of managing their meaning and comprehension.

Keywords: *smart contract, civil law, traditional contracts, blockchain, Ethereum*

How to cite: Onufreiciuc, R., & Stănescu, L.-E. (2021). Regulation of the Smart Contract in (Romanian) Civil Law. *European Journal of Law and Public Administration*, 8(2), 95-111. <https://doi.org/10.18662/eljpa/8.2/164>

1. Introduction

The growing popularity of decentralized cryptocurrencies has aroused the interest of both the business sector and academia. Bitcoin and Ethereum are two of the most popular cryptocurrencies, with a combined market valuation of 1 trillion 421 billion 904 million, as of May 2021 (CoinMarket Cap, 2020). The basic technology for issuing and trading cryptocurrencies is blockchain, used as a distributed consensus tool. Users of Bitcoin and Ethereum are allowed to develop rules or scripts to execute transactions. Unlike Bitcoin, Ethereum is a more advanced blockchain technology with a fully-embedded Turing programming language that anyone can use to create “smart contracts” (Buterin, 2013). As described in the Whitepaper, Ethereum aims to provide “a blockchain with a built-in fully fledged Turing-complete programming language” that may be used to produce “contracts” that can be used to encode arbitrary state transition functions, allowing users to build any of the systems mentioned above, as well as many others we have yet to envisage, by simply writing up the logic in a few lines of code.

Considered to be a new model of practice, smart contracts represent also a change in the global financial model (cryptocurrency transactions), corporate good practices (DAOs), or commercial rules. A smart contract can begin with a set of clauses drafted by contract law’s legal standards, but when these clauses transit via a blockchain platform and become self-enforcing, the “voice” of the doctrine becomes weaker and the applicable law is increasingly difficult to identify. The execution of a smart contract which “automatically and securely executes obligations without reliance on a centralized enforcement authority” (Levy, 2017) presents several problems concerning its legal impact on civil contract law, and the answers are not easy to come by. For example, does the development and running of a smart contract mean that a legally binding contract has been formed? To respond to this question, we must recognize that whether a smart contract becomes legally binding is determined by several circumstances including the use case, the type of smart contract employed, and the existing law.

2. Human language vs. programming language

Legal (traditional) contracts arise from the use of human language - even if it requires specialized legal language - whether the formulation is oral, written, or digital text. This traditional legal language is referenced as the “wet code” (Szabo, 2017). The law is flexible, involves human judgment, and can be (easily) enforced, but also breached while the software is more rigid and transparent. Therefore, it is easy to see that the provisions of traditional legal contracts are the product of human logic in opposition to intelligent contracts, which are based on bits, data, and Boolean logic.

Consequently, the rules and conditions of smart contracts’ execution are assessed by software code and the effective execution of the contracts is validated and conducted by sensors and software code (Drummer & Neumann, 2020). According to Szabo (2017), this process is called “dry code”.

The best-known smart contracts are developed on the Ethereum platform and generally refer to small programs that link a transaction to a statement, such as “if Team L wins, send X Ether to LGH”. These smart contracts and other programs can be written in a variety of programming languages, depending on the developer's preference, but Solidity, which is similar to the JavaScript and C programming languages is the most popular Ethereum language. Serpent, Vyper, LLL, Mutant, and JULIA are some of the other Ethereum programming languages (Çağlıyangil et al., 2020).

3. Definitions and requirements for a valid contract

According to art. 1166 of the Romanian Civil Code (Romanian Parliament, 2011) the contract “is an agreement based on the free will of two or more parties to it to establish, change or cease a legal relationship”. As the doctrine states, the definition of the contract “includes the essential elements for specifying the notion of contract. What is decisive in this definition is the agreement of will, bi or multilateral, which gives rise to, modifies, or ceases rights and obligations. According to the above-mentioned definition, we can easily conclude that “this is a legal act, namely a legal act of bilateral formation (plurilateral, multilateral)” (Statescu & Birsan, 2008).

Regarding the definition of the smart contract, we remind one of the first definitions attributed to Nick Szabo who compared the smart contract with vending machines (Bambara & Allen, 2018) that after collecting a penny, releases the product without the need for human participation in the distribution process (Szabo, 1997).

According to Nick Szabo (1997), a “smart contract is a set of promises, specified in digital form, including protocols within which the parties perform on the other promises “. As Szabo stated, smart contracts indicate a major transition in the world away from paper and toward digital institutions, such as computer-driven banking and digital databases, by lowering the costs of mediation, self-enforcement, and arbitration.

Another renowned definition belongs to Vitalik Buterin, the founder of Ethereum, and is based on a type of causation (“if x, then y”): a smart contract is computer software that manages some digital assets directly (Chamber of Digital Commerce, 2016). We note that both definitions emphasize the nature of the smart contract to represent a computer program in the form of a database in the blockchain registry which supports blockchain transactions. Smart contracts can store and transfer blockchain-managed digital assets as well as activate other blockchain-based smart contracts. Unlike the legal contract where the core is represented by the agreement of wills, the intelligent contract is characterized, in particular, by the causality of the algorithms (“if x, then y”) and, secondary, on the users’ intentions.

In the field of civil law, to produce legal effects and subsequently be enforced, a contract must meet certain conditions, which differ from one jurisdiction to another. According to art. 1179 of the Romanian Civil Code (Romanian Parliament, 2011), the essential conditions for the validity of the contract are the ability to contract, the consent of the parties, a determined and lawful object, and a lawful and moral cause. In addition, depending on the object of the contract, the provisions of the Civil Code or the specific legislation may provide other conditions that must be met for the validity of the contract. The smart contract, in the sense of a computer program, is valid and can run regardless of, for example, the age of the person who wrote the code, the legality of the object, or the morality of the cause.

It is important to remember that to be executed, the smart contract does not have to be signed, within the meaning of digital signature. Unlike the electronic contract - which to be valid in Europe, must meet the eIDAS standards- digital signatures on a blockchain must be verified by a TSP. A smart contract that seeks to produce legal effects requires such digital signatures to determine whether the signature is valid, refers to the correct person, and whether that person has the authority to sign.

The validity and effectiveness of the smart contract are provided by the ecosystem of participants in the transactions on the blockchain platform, which verifies and authorizes the transactions (miners). The usage of smart contracts which are based on blockchain technology is the mechanism for implementing automatic controls into these platforms' transactions. The incorporation of the code in the blockchain allows the automatic execution of transactions in response to the fulfillment of certain conditions, which leads to an assured execution (Akyuz & Gursoy, 2020). As a result, smart contracts allow for the storage of programmable and configurable logic in their content and they represent algorithmic, self-executing arrangements that "have the ability" to work automatically. They may store the rules agreed by the interested parties by the prior terms of the agreement between them and the performance of the contract may be verified automatically. As a result, automated procedures contain consensus-based protocols and contracts, and the agreed-upon terms and conditions of business are recorded in the transaction database (Akyuz & Gursoy, 2020).

Although it is premature to say that smart contracts can be used to replace legal contracts, they can sometimes be used to automate or monitor the performance of obligations under these legal contracts. They can also define a procedure for interacting with diverse parties, such as a collaboration process between businesses (Xu et al., 2019).

3.1. Complexity of the contract

The complexity of the traditional contract is superior to the smart contract in that the parties can provide for exceptional and various situations (the well-known "exception from the exception, which cannot be included in code terms, for example: "death of the user"). Some situations are difficult to code or cannot be carried out automatically as a simple operation: "if this,

then that”. In this sense, force majeure (Tjong Tjin Tai, 2018) is a suitable example. Also, subjective judgment and good faith are notions found in contracts. At present, these notions cannot be easily translated into logical assertions.

A smart contract can only be executed if its results can be read by the software and determined algorithmically. In other words, “discretion, reasonableness or judgment” has no place in a smart contract taking into consideration that conceivable outcomes will be binary in the form (Sanitt, 2019).

3.2. Applicable law

Currently, although there are various legislative initiatives (Adcock, 2020) on the regulation of smart contracts, however, there is no international or national legal framework specifically designed for smart contracts. The lack of formal legislation does not, however, imply that smart contracts are wholly unregulated. While the traditional legal contract is subject to national jurisdiction, the law applicable to smart contracts is currently determined by general principles (Woebbeking, 2019). Consequently, we are in the presence of a divergent assessment of smart contracts in different jurisdictions.

3.3. Form of the contract

In the first stage of a contractual arrangement, intelligent and traditional contracts do not vary much. This is because before any agreement can function, two parties must agree to a set of conditions that describes their expectations entering into the new relationship (Raskin, 2017).

According to the Romanian legislation, to constitute an offer to contract, the proposal must have sufficient components for the formation of the contract (art. 1188 of the Romanian Civil Code) (Romanian Parliament, 2011). The offer to contract, according to doctrine, must be an expression of real, undivided will, concretized in a definite, comprehensive, and solid proposition (Statescu & Birsan, 2008). We observe that the laws governing offer and acceptance will not, in theory, prevent smart contracts from being recognized as legally binding. To begin, participants committing resources to a blockchain-based smart contract by giving their cryptographic private keys

offers evidence of a commitment. Since one party must publish his (on-chain smart) 'contract' on the blockchain on platforms (for example, Ethereum) and the other party approved by the cryptographic key, such communication (the posting of the on-chain smart 'contract' on the blockchain) will likely be held as an offer (Durovic, 2019). In other words, smart contracts may, in principle, meet the requirements for contract creation and that present difficulties are not insurmountable. The traditional contract is characterized by the principle of consensuality, as provided in art. 1178 of the Romanian Civil Code (Romanian Parliament, 2011), a contract is executed by the mere consensus of the parties (solo consensus), should the law provide no particular formalities for the valid execution thereof (such as written form, authentication of the document). In opposition to the principle of freedom of form, smart contracts are conditioned by the written form and online environment and, as a rule, are developed only on blockchain platforms.

4. Modification of the contract

In general, the parties may change their mind on their initial intentions and agreement and may rethink its provisions, by art. 1243 of the Romanian Civil Code (Romanian Parliament, 2011): Modification of the contract: "If the law does not provide otherwise, any modification of the contract is subject to the formal conditions required by the law for its conclusion."

In terms of smart contracts, the rule is that they are recorded and programmed immutably in the distributed register and cannot be modified. Exceptionally, even in the absence of a central authority, the smart contract can be modified by using fork software. The latter is a mechanism by which the smart contract is copied and modified. The original draft code still exists but is now separate from the new one, which is taking a different direction. Depending on this new direction, we can distinguish between a hard fork and a soft fork.

Hard forks are software updates that are incompatible with previous versions. They usually occur when nodes add new rules in a way that can conflict with the rules of old nodes. The new nodes are only accessible to

people operating the new version. As a consequence, the blockchain splits into two networks, one with the new rules and the other with the old rules which run concurrently. Soft fork corresponds to an updated compatible form with previous versions, which means that the updated nodes can still communicate with those that are not updated. In the soft fork, the insertion of a new rule that does not clash with the earlier rules is common.

Therefore, it is difficult to cancel a smart contract affected, for example, by causes of illegality. We remind that such an event had already taken place on the Ethereum platform and the community has reacted with a technical response such as a hard fork (Siegel, 2016).

As a result, one of the most significant distinctions between traditional and smart contracts is human intervention: whereas a traditional contract can be modified and changed at any time with the parties' approval, a smart contract, once made, cannot be canceled or changed. As we have shown, by way of exception, there are options for developers to terminate or suspend the smart contract, but these situations are very rare because the main reason behind the formation of smart contracts is to avoid human intervention and trust in a predetermined system. In this respect, the immutability of smart contracts, which guarantees that they cannot be changed without the entire ecosystem acknowledging and validating the operation, is an advantage over the issues faced by traditional contracts - at least those in physical format - regarding the easy modification of their written provisions. Traditional contracts are designed to comply at all times with the common wills of the involved parties, which is why they can be modified by the intention of the signatories. Smart contracts, on the other hand, are self-executing codes that cannot be easily altered.

We also note that blockchain technology is making progress towards so-called "hybrid contracts" which include the benefits of the two types of contracts exemplified, namely, the flexibility of traditional contracts and the self-execution of smart contracts.

5. Execution of contracts and fulfillment of obligations

Execution of obligations is the main criterion that distinguishes smart contracts from legal contracts. Smart contracts, unlike traditional

contracts, are self-executing computer programs that are concluded according to current legal standards. Thus, in the case of smart contracts, certain obligations or even all contractual obligations can be performed automatically when a certain digitally verifiable event occurs. The traditional contract is recognized by law, even if its fulfillment does not exactly match the specific conditions laid down. This sort of leeway cannot be recognized by a computer program as it implies a result that the parties have not considered and stated (Raskin, 2017). For example, according to traditional contracts, the parties must pay the amounts due in time - personally, with extra organizational effort on their behalf. The remittance of due amounts is automated and carried out based on the criteria agreed upon and recorded in the specific code governing the parties' relationship. We notice how the execution of the contract is automated and is done by the computer, following its programmed instructions. In this regard, Max Raskin (2017) provides that smart contracts "solve the problem of gift-promises by giving both the promisor and promisee the ability to encode finality so that parties can organize their behaviors around a mechanical certainty or lack thereof". However, it can be said that the use of smart contracts does not solve or eliminate the problem of contract breaches, contractual liability, and the performance of the obligations assumed by the parties (The European Union Blockchain Observatory and Forum, 2018).

The problem that arises again is the lack of tools available to easily identify the actors in a blockchain network, which will require finding a solution to identify the parties involved in transactions. The issue of IT resources also emerges in the case of smart contracts. The blockchain is currently incapable of processing hundreds of transactions per second. We give scalability a lot of thought. Each node in the unique blockchain network must be aware of all global transactions, which can cause substantial network delays. The goal is to make all transactions more efficient while maintaining the network's decentralization and security (The European Union Blockchain Observatory and Forum, 2018). It is also necessary to ensure interoperability between different implementations of blockchain platforms so that there is communication between them.

6. Contractual liability

Given the immutable nature of smart contracts, contractual liability for non-fulfillment or improper fulfillment of obligations under smart contracts is complex and problematic, both legally and technically.

First of all, as a rule, a smart contract cannot be broken by one of its parties (Savelyev, 2016). This stems from its self-enforceability and is a natural result of its “code is law” character. A party to it cannot violate a contract if circumstances change and a more profitable alternative to its execution becomes available. As a result, unless specifically stated in the code, all traditional remedies for breach of contracts, such as damages, fines (liquidated damages), and specific performance, are irrelevant for smart contracts.

However, if after the execution of a smart contract a party wants to cancel the transaction, there are some technical issues to assess. Also, the Romanian legal institution of “the restoration to the original condition”, governed by art. 1254 alin. (3) of the Civil Code, cannot be applicable (Romanian Parliament, 2011). For example, in the case of concluding an online commercial agreement (Woebbecking, 2019), the consumer has the right to cancel and return the order within 14 days, regardless of the reason and without any justification.

As we have previously mentioned, given that the list of public, chained blocks, without permits, as a basis for the smart contract, guarantees the immutability of the transaction, special attention must be paid to the technical implementation of the cancellation or withdrawal of an operation (Woebbecking, 2019). In general, so-called reverse transactions can be used to revert a blockchain to its previous state. This method, however, does not remove the transaction history, therefore the withdrawn transaction is still recorded in the blockchain. Contract law does not offer any obstacles, albeit disagreements over the intricacies of personal data protection may occur. Furthermore, because a reverse transaction may only be initiated by the owner of the private key, it can be difficult to complete.

One of the purposes of using blockchain technology is to eliminate third parties (such as banks, notaries, lawyers) from relations between participants. As a result, a smart contract's life cycle must never include

anyone other than the contracting parties (because, for example, payment can be made directly from one bitcoin wallet to another bitcoin wallet). A standard contract might take anywhere from one to several days to prepare and draft, depending on the quality of legal services and the availability of the contracting parties. If a predetermined contractual platform is used, such as Ethereum, Hyperledger Fabric, or others, this period can be shortened to a few minutes for smart contracts (Toshev & Boteva, 2019).

7. The cost of concluding a contract

Depending on the complexity of traditional legal contracts, they may create additional costs for the parties to negotiate, formulate, explain the consequences, and conclude them. However, it is good to know that smart contracts are not without costs either. The transaction payment system is known as 'gas' and, in general, the price is lower compared to the authentication fee of a contract, for example. Miners use their computers to run smart contracts in Ethereum. The designers and users of smart contracts pay a fixed amount to miners as remuneration for contributing their resources to the development and validation of activities on the blockchain platform (Chen et. al., 2020).

The amount paid to the miners is calculated as follows: gas cost x gas price. The cost of gas depends on the computing resource that the transaction will take, and the price of gas is provided by the creators of the transaction. The minimum gas price is Wei (1 ether = 10¹⁸ Wei). Miners have the right to choose which transaction can be executed and transmitted to the other nodes on the blockchain. Therefore, if the price of gas is too low, transactions may not be executed. To limit the cost of gas, when a user submits a transaction to invoke the code of a smart contract, there will be a limit (gas limit) that determines the maximum cost of gas. If the cost of gas exceeds the gas limit, execution ends with an exception often called “out of gas error”(Chen et al., 2020).

8. Security and data protection

Unlike a legal contract whose content is found in physical format (on paper) or digital format (Word, pdf), a smart contract can provide a

qualitative level of security by using a cryptographic protocol. The cryptographic protection provided by blockchain technology, on the other hand, allows for a high level of contract confidentiality when the contract is maintained in a private register rather than a public register. Thus, we recall that vulnerabilities have been noted regarding the public nature of the most well-known smart contracts: the public fields of the contract are visible to everyone, and the transactions reveal a lot of personal data. To address the issue of smart contract transparency across the blockchain ecosystem, various solutions have been proposed (Kosba et al., 2016) to protect the data of parties involved in cryptocurrency transactions.

Traditional contract archiving involves time, space, administration, and oversight. In the case of smart contracts, the archiving procedure occurs automatically and securely in storage, memory, or call data, with no time or other resources wasted.

9. Good faith vs. trust

The problem of trust between people has perhaps always existed. Perhaps since Eve gave Adam a taste of the forbidden apple, and thereby deviated from the divine order. However, we must acknowledge that to progress, a society or a group of people must cooperate. When people cooperate, they can do more collectively than they could individually (Harari, 2018). However, in doing so, if one person does not respect the trust received from the others, the trust in the whole community is damaged. To overcome this problem, communities have created rituals, enacted laws, and even constructed governing structures. All these elaborated techniques are in place to solve the problem of trust. To better explain the issue of trust, we will refer to a common example found in the literature advocating for the use of blockchain technology. A person frequently uses his credit card to pay for goods and services. One enters a store, chooses an item, goes home, and pays by credit card. The purchase is allowed because, by dragging or inserting the bank card in the chip reader, the store checks with the bank if there is financial availability of payment from the buyer and, in some cases, can confirm the person to whom the card was issued is the buyer. The bank will collect the amount from the buyer later, at the time of invoicing

(Bambara & Allen, 2018). Thus, there is confidence in the whole chain of participants. The store has faith in the bank and the buyer, the bank has faith in the store and the buyer, and the buyer has faith in the store and the bank. Instead of money, the buyer can carry a little piece of plastic. He can replace the card in a few days if he loses it. If he had to carry the cash equivalent, he might lose it or be robbed and never receive it. Based on this trustworthy item, it is easier to purchase items. As barriers to payment decrease, more trade occurs. This returns us to the beginning, where cooperation requires confidence, and cooperation leads to advancement (Bambara & Allen, 2018).

Using a trusted third party to tackle the issue of trust is one option. The bank and credit card scenario mentioned above exemplifies this perfectly. A bank enters and records transactions between a customer, a merchant, and a buyer (credit card issuer). The bank enables the transaction and can also intervene and resolve a dispute if a customer discovers an unauthorized transaction on his monthly account statement. What we've just outlined is a centralized approach to dealing with the issue of trust.

There is another way to solve the problem of trust, respectively, by using a distributed register (or shared register) combined with a consensus methodology. So instead of recording transactions by a single third party, a copy of each transaction is sent to all parties in the network. All participants at the network keep a continuous record of all transactions. Therefore, each user has the same set of transactions (Sklaroff, 2017). At any given time, everyone knows how much the buyer owes the trader. If a dispute arises, the majority of the consensus (51%) in the register holders' network decides what the debit is. Blockchain technology uses this method of distributed registry and consensus.

In terms of computer programming, these Aristotelian syllogisms are implemented using an algorithmic construction: “if ... then ... otherwise ...”. A smart contract combines one or more of these conditions with the capacity to automatically carry out the obligations. The distributed registry system stores all of the data and, theoretically, capabilities required for a smart contract to execute independently. The smart contract specifies the necessary circumstances, and if they are met, the contractual obligations are fulfilled automatically (Sklaroff, 2017). However, there is no need for courts

or mediation in this scenario because the facts are available to the contract, and a bad decision cannot be made.

10. Final remarks

The idea of performing legal transactions in the digital sphere was probably born with the Internet. But the concept of a computerized protocol to execute the contractual provisions is being formed, understood, and implemented. There will be examples of hybrids developed with different components: on the blockchain and outside the blockchain, as the smart contract grows. Finally, smart contracts will become more widespread as blockchain technology improves and becomes more broadly recognized.

As an example, here are some of the situations of use and recognition of the legal effects of smart contracts:

- smart contract code, for example, Ethereum Solidity which is stored, verified, and executed on a blockchain, used to perform specific operations on distributed registers;
- smart legal contracts written as a specification for the use of the smart contract code in the completion or complete replacement of legal contracts;
- a hybrid contract – a traditional and intelligent agreement. Some contracts may include self-enforcing provisions as well as conditions that are not part of the program code. This is because, especially in complicated contracts, not all contract judgments can be reduced to logical assertions.

Taking into consideration the legal consequences of implementing intelligent contracts, the transition to contracts written in computer code language can create new inefficiencies (Sklaroff, 2017), arising from: automation, which requires each agreement to consist of fully defined terms; decentralization, which conditions the performance of verification by all users in the ecosystem; and anonymity, which discourages the use of contracts in a commercial context. In a volatile climate or where there is a level of ambiguity about the parties' agreement, it is consequently exceedingly costly to conclude smart contracts.

The important question in the final contract law matter is what occurs when the consequences of the smart contract differ from the

outcomes required by law. Smart contracts are probably not “the mature end of the evolution of electronic agreements” (Werbach & Cornell, 2017); however, they represent a new stage in contracts. Existing contract law can withstand the challenges of smart contracts. Aside from the places where smart contracts and contract law meet, there are aspects of smart contracts that are not covered by existing rules, and there are legal provisions that are difficult to comply with inside smart contract language.

References

- Adcock, C. (2020). An Update on State Smart Contract Legislation. *The National Law Review*, 10(106). <https://www.natlawreview.com/article/update-state-smart-contract-legislation>
- Akyuz, G. A., & Gursoy, G. (2020). Transformation of Supply Chain Activities in Blockchain Environment. In U. Hacıoglu (Ed.), *Digital Business Strategies in Blockchain Ecosystems. Transformational Design and Future of Global Business* (pp. 153-177). Springer Nature Switzerland AG.
- Bambara, J., & Allen P. (2010). *Blockchain A Practical Guide to Developing Business, Law, and Technology Solutions*. McGraw-Hill Education.
- Chamber of Digital Commerce (2016, November 8). *Panel 1: Law 2.0 Understanding Smart Contracts* [Video file]. YouTube. Chamber of Digital Commerce. https://youtu.be/ZuHZOryZ_f0
- Buterin, V. (2013). *Ethereum Whitepaper*. Ethereum. <https://ethereum.org/en/whitepaper/>
- Çağlıyangil, M., Erdem, S., & Özdağoglu, G. (2020). A Blockchain-Based Framework for Blood Distribution. In U. Hacıoglu (Ed.), *Digital Business Strategies in Blockchain Ecosystems. Transformational Design and Future of Global Business* (pp. 63-83). Springer Nature Switzerland AG.
- Chen, J., Xia, X., Lo, D., Grundy, J., & Luo, X., Chen, T. (2020). Defining Smart Contract Defects on Ethereum. In *IEEE Transactions on Software Engineering*, 1-17. <https://doi.org/10.1109/TSE.2020.2989002>
- CoinMarket Cap (2020). *Today's Cryptocurrency Prices by Market Cap*. CoinmarketCap <https://coinmarketcap.com/>
- Drummer, D., & Neumann, D. (2020). Is code law? Current legal and technical adoption issues and remedies for blockchain-enabled smart contracts. *Journal of Information Technology*, 35(4), 337-360. <https://doi.org/10.1177/0268396220924669>
- Durovic, M. (2019). The Formation of Smart Contracts and Beyond: Shaking the Fundamentals of Contract Law? In L. DiMatteo, M. Cannarsa & C.

- Poncibo (Eds.), *Smart Contracts and Blockchain Technology: Role of Contract Law* (pp. 1-27). Kings College London.
- Harari, Y. N. (2018). *Homo deus. Scurtă istorie a viitorului* [Homo deus. A brief history of the future]. Polirom.
- Kosba, A., Miller, A., Shi, E., Wen, Z., & Papamanthou, C. (2016). Hawk: The Blockchain Model of Cryptography and Privacy-Preserving Smart Contracts. *2016 IEEE Symposium on Security and Privacy*, 839-858. <https://doi.org/10.1109/SP.2016.55>
- Levy, K. (2017). Book-Smart, Not Street-Smart: Blockchain-Based Smart Contracts and The Social Workings of Law. *Engaging Science, Technology, and Society*, 3, 1-15. <https://doi.org/10.17351/ests2017.107>
- Raskin, M. (2017). The Law and Legality of Smart Contracts. *Georgetown Law Technology Review*, (304), 305-326. <http://dx.doi.org/10.2139/ssrn.2842258>
- Romanian Parliament. (2011). Legea nr. 287 din 24 iulie 2009 privind Codul Civil [Law no. 287/2009 regarding the Civil Code]. *Monitorul Oficial, partea I*, 511, 17 Iulie 2009 [Official Gazette of Romania, Part I, no. 511 of July 17, 2009], republished in the *Monitorul Oficial, partea I*, 409, 10 Iunie 2011 [Official Gazette of Romania, Part I, no. 409 of June 10, 2011]. <http://legislatie.just.ro/Public/DetaliiDocument/175630>
- Sanitt, A. (2019). *Smart Contracts*. Norton Rose Fulbright. <https://www.nortonrosefulbright.com/en/knowledge/publications/1bcd200/smart-contracts>
- Savelyev, A. (2016). Contract Law 2.0: «Smart» Contracts As the Beginning of the End of Classic Contract Law. *National Research University Higher School of Economics*, (71), 1-24. <http://dx.doi.org/10.2139/ssrn.2885241>
- Siegel, D. (2016). *Understanding The DAO Attack*. <https://pullnews.medium.com/understanding-the-dao-hack-for-journalists-2312dd43e993>
- Sklaroff, J. M. (2017). *Smart Contracts and the Cost of Inflexibility*. *University of Pennsylvania Law Review*, 263-303. https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3008899
- Stătescu, C., & Bîrsan, C. (2008). *Drept civil. Teoria generală a obligațiilor* [Civil law. General theory of obligations]. Hamangiu.
- Szabo, N. (1997). Formalizing and Securing Relationships on Public Networks. *First Monday*, 2(9). <https://doi.org/10.5210/fm.v2i9.548>
- Szabo, N. (2017). *Winning Strategies for Smart Contracts*. Blockchain Research Institute. <https://www.blockchainresearchinstitute.org/project/smart-contracts/>
- The European Union Blockchain Observatory and Forum. (2018). *Blockchain for Government and Public Services. Thematic Report*.

- https://www.eublockchainforum.eu/sites/default/files/reports/eu_observatory_blockchain_in_government_services_v1_2018-12-07.pdf
- Tjong Tjin Tai, E. (2018). Force Majeure and Excuses in Smart Contracts. *European Review of Private Law*, 26(6), 787-804. <https://doi.org/10.2139/ssrn.3183637>
- Toshev & Boteva. (2019). *Blockchain technologies in healthcare – opportunities, and perspectives*. Toshevboteva
<https://www.toshevboteva.com/news/blockchain-technologies-in-healthcare-opportunities-and-perspectives>
- Werbach, K., Cornell, N. (2017). Contracts Ex Machina. *Duke Law Journal*, 67(313), 314-381. https://papers.ssrn.com/sol3/papers.cfm?abstract_id=2936294
- Woebbeking, M. K. (2019). The Impact of Smart Contracts on Traditional Concepts of Contract Law. *JIPITEC*, 10(106).
<https://www.jipitec.eu/issues/jipitec-10-1-2019/4880>
- Xu, X., Weber, I. & Staples, M. (2019). *Architecture for Blockchain Applications* (1st ed.). Springer.