https://doi.org/10.31891/csit-2025-1-17
UDC 004.89: 004.3

Viacheslav ASKEROV, Bohdan TOMCHYSHEN
Khmelnytskyi National University
Houda El BOUHISSI
University of Bejaia

USE OF SMART CONTRACTS ON THE TON BLOCKCHAIN FOR INNOVATIVE EDUCATIONAL SOLUTIONS DEVELOPMENT

In the modern world, blockchain technologies are gaining popularity due to their ability to ensure security, transparency and decentralization of data. One of the most promising platforms is The Open Network (TON), which provides unique opportunities for the development of smart contracts. This article discusses the main features of the TON blockchain and its advantages in the context of educational process automation. Smart contracts implemented on the TON platform can serve as a tool for optimizing educational systems. They allow to automate processes related to knowledge validation, grade management, and even finance in educational institutions. For example, smart contracts can provide automatic scholarships based on students' grades, as well as control over the implementation of curricula. The paper also analyzes the benefits of using smart contracts in the educational process, such as reducing administrative costs, increasing transparency, and reducing fraud risks. In addition, blockchain technologies provide an opportunity to create decentralized platforms for storing and sharing knowledge, which makes learning more accessible and effective. Particular attention is paid to the mathematical aspects that ensure the functioning of TON, as well as sharing mechanisms that allow the platform to process thousands of transactions per second. These technologies can be used to create educational applications requiring high bandwidth and data processing speed. The paper contains formulas that illustrate the technical characteristics of the TON blockchain and provides a detailed analysis of its architecture. The study shows that smart contracts on the TON platform have the potential to revolutionize educational processes by providing new tools for data management and security.

Keywords: blockchain; smart contracts; decentralization; TON.

В’ячеслав АСКЕРОВ, Богдан ТОМЧИШЕН
Хмельницький національний університет
Худа Ель БУХІССІ
Університет Беджая, Алжир

ВИКОРИСТАННЯ СМАРТ-КОНТРАКТІВ НА БЛОКЧЕЙНІ TON ДЛЯ РОЗРОБКИ ІННОВАЦІЙНИХ ОСВІТНІХ РІШЕНЬ

У сучасному світі технології блокчейн набирають популярності завдяки своїй здатності забезпечувати безпеку, прозорість і децентралізацію даних. Однією з найперспективніших платформ є The Open Network (TON), яка надає унікальні можливості для розробки смарт-контрактів. Ця стаття розглядає основні особливості блокчейну TON та його переваги в контексті автоматизації навчальних процесів. Смарт-контракти, що реалізуються на платформі TON, можуть слугувати інструментом для оптимізації освітніх систем. Вони дозволяють автоматизувати процеси, пов'язані з підтвердженням знань, управлінням оцінками та навіть фінансами в освітніх установах. Наприклад, смарт-контракти можуть забезпечити автоматичне нарахування стипендій на основі оцінок студентів, а також контроль за виконанням навчальних планів. У статті також аналізуються переваги використання смарт-контрактів у навчальному процесі, такі як зменшення адміністративних витрат, підвищення прозорості та зменшення ризиків шахрайства. Окрім того, блокчейн технології надають можливість створення децентралізованих платформ для зберігання та обміну знаннями, що робить навчання більш доступним і ефективним. Особлива увага приділяється математичним аспектам, що забезпечують функціонування TON, а також механізмам шардінгу, які дозволяють платформі обробляти тисячі транзакцій на секунду. Ці технології можуть бути застосовані для створення освітніх додатків, що потребують високої пропускної здатності та швидкості обробки даних. Стаття містить формули, які ілюструють технічні характеристики блокчейну TON, і надає детальний аналіз його архітектури. Дослідження показує, що смарт-контракти на платформі TON мають потенціал для революціонізації навчальних процесів, надаючи нові інструменти для управління даними і забезпечення їхньої безпеки.

Ключові слова: блокчейн; смарт-контракти; децентралізація; TON.

Introduction

A blockchain is a decentralized digital database that functions as a chain of blocks. Each block contains information about transactions and is protected by cryptography. Such a distributed database is the basis of the first cryptocurrency, bitcoin, for which blockchain technology was created in 2008. In essence, it is a kind of ledger for all transactions, which allows you to solve the issue of double spending without a central server or authority [1].

Protection against counterfeiting and modification is ensured by the fact that the hash of each block is included in the next one. Thus, changing one block requires changes to all subsequent blocks, which is usually extremely difficult or uneconomical.

The TON blockchain uses a proof-of-stake consensus algorithm where validators are responsible for verifying transactions and creating new blocks, and are selected based on the amount of TON they own and are willing to invest. This method supports high transaction bandwidth while being energy efficient. The platform also supports advanced smart contracts, allowing developers to create decentralized applications based on the TON

blockchain. TON is the native cryptocurrency of the TON blockchain, which is used to pay for transactions, staking, and as a medium of exchange in the TON ecosystem [2].

Another feature of TON's architecture is what they call 'dynamic partitioning,' or dividing the blockchain state 'into smaller pieces so that each set of nodes only needs to store and verify a subset of the data.' TON claims that this feature theoretically allows it to scale to millions of transactions per second, and effectively means that TON is not a single blockchain, but a blockchain of blockchains.

Further improvement of blockchain technologies is an urgent task, as they are actively changing the approach to processing, storing and transmitting information. Blockchain has long been used for financial transactions, but over time, the requirements for its scalability and speed have increased. One of the most important problems is the low bandwidth and high energy costs of traditional blockchains, which prevents their effective implementation in large digital systems. The papers emphasizes that technologies such as The Open Network (TON) can solve these problems by combining high speed and scalability with low energy costs.

Domain analysis

In the course of the study, we analyzed recent Ukrainian and foreign publications have been analyzed.

The paper [3] links a blockchain-based educational system to cryptographically secure Internet devices and the Indian New Education Policy (NEP-2020) with the current research. Some colleges utilize machine learning to predict student interests. There is a lack of credible literature written on the impact of BCT in education. The study [4] aims to identify the variables that drive educational transformation using BCT and to study the impact of BCT in the education industry, as confirmed by research studies conducted by European Universities (Grech et al., 2017). The results of the study show that strategy, learner needs, technology, regulation, business processes, skills and competencies affect the BCT implementation in the education domain (Ali Alammary et al., 2019). Blockchain governs inter-organizational business processes and enables decentralized autonomous organizations (DAO) with governance capabilities via smart contracts (SC). Due to the programmer's lack of prior knowledge of the contract domain, SCs are ambiguous and error-prone. Several works, i.e., SPESC, Symboleo, and SmaCoNat, exist to support the legally-binding SCs. The aforementioned SCLs present intriguing approaches to building legally-binding SCs but either lack domain completeness, or are intended for non-collaborative business processes. In accordance with this, [5] formally implements the SLCML and proposes evaluation approaches, such as running case and lab experiments, to demonstrate the SLCML's generality and applicability for developing legally-binding SCs. Overall, the results of this work ascertain the applicability, usefulness, and usability of the proposed SLCML for establishing legally-binding SCs for WES. The paper [6] develops a comprehensive classification taxonomy of smart contract threat mitigation solutions within five orthogonal dimensions: defense modality, core method, targeted contracts, input-output data mapping, and threat model. In [7] the application of blockchain technology in remote labs is proposed as a promising solution for future online learning as it combines a new pedagogical approach with various state-of-the-art technologies in an era that embraces Education 4.0 as the education norm. The paper [8] further develops a method for assessing the reputation of a medical institution (as a blockchain miner), which allows you to calculate and evaluate the reputation of a medical institution. Experiments were carried out on the management of medical data based on blockchain technologies using the developed method for assessing the reputation of a medical institution (as a blockchain miner) – experiments on assessing the reputation of a medical institution (for example, two different outpatient clinics of family medicine) – as a blockchain miner. The study [9] reviews existing Smart Contract Languages (SCL) and identifies properties that are critical to any future SCL for drafting legally binding contracts. This is achieved by conducting a Systematic Literature Review (SLR) of white- and grey literature published between 2015 and 2019. Using the SLR methodology, 45 Selected and 28 Supporting Studies detailing 45 state-of-the-art SCLs are selected. Finally, 10 SCL properties that enable legally compliant DAOs are discovered, and specifications for developing SCLs are explored.

Recent researches have provided examples of successful use of the TON platform, which provides thousands of transactions per second due to its unique architecture. An important aspect is its ability to scale horizontally through the use of sharding, which allows for efficient processing of parallel transactions. The article also discusses cryptographic algorithms that guarantee network security.

The purpose of this publication is to provide an in-depth study of blockchain technology, in particular the TON platform, as well as to analyse its advantages in terms of scalability, security and speed of transactions and how it can be used in educational processes.

Analysis of existing solutions based on Blockchain technology

According to public information in online sources [2], as of 2024, there are more than a thousand different blockchains. They can be of different types, public, private, hybrid, etc. The most popular ones besides the TON blockchain are Ethereum, Bitcoin, and Solana. These are cutting-edge, powerful solutions used by millions of users and developers around the world.

Each blockchain has some parameters, one of which is the transaction bandwidth (TPS - Transactions Per Second), which is an important indicator of scalability, comparing the main blockchains, we can highlight the following indicators:

- Bitcoin: Uses Proof of Work (PoW), with a throughput of approximately 7 transactions per second (TPS).
- Ethereum: Prior to Ethereum 2.0, it operated at 15-30 TPS, but with the introduction of Proof of Stake (PoS), it can potentially reach up to 100,000 TPS.
- TON: Thanks to the sharding mechanism, it is capable of processing thousands of transactions per second.

In general, this can be expressed using the formula (1):

$$TPS = N / T \tag{1}$$

where N is the number of transactions processed, T is the time in seconds. The sharding mechanism in TON, which allows parallel processing of transactions, provides much higher throughput than in Bitcoin or the basic version of Ethereum.

Another and no less important indicator is energy consumption, which differs depending on the consensus mechanism. Proof of Work (PoW), used in Bitcoin and earlier in Ethereum, is known for its high-power consumption.

Energy consumption in the Bitcoin blockchain (PoW) is displayed as a function below (2):

$$E_{BTC} = P_{hash} * T_{block} * nE \tag{2}$$

where Phash — energy consumed by a single hash, Tblock — is the average block generation time (10 minutes for Bitcoin), and n is the number of miners. This leads to significant energy consumption.

Considering the TON (PoS) and Ethereum 2.0 (PoS) blockchains, which are much more efficient because they rely on validators rather than miners. The power consumption for PoS can be simplified by the following equation (3):

$$E_{PoS} = P_{validator} * N_{validators} \tag{3}$$

where Pvalidator — energy consumed by one validator, Nvalidators — number of validators.

The TON blockchain contains sequential block numbers. Each block B in the blockchain can be referred to by its sequence number BLK-SEQNO(B), which starts at zero for the very first block and increases by one when moving to the next block.

Therefore, the following equations follows (4),

$$BLK - SEQO(B) = BLK - SEQNO(BLK - PREV(B)) + 1 \tag{4}$$

However, the sequence number does not uniquely identify the block if there are branches.

Account IDs. The main account identifiers used in the TON blockchain, or at least its masterchain and Workchain Zero, are 256-bit integers that are considered public keys for 256-bit Elliptic Curve Cryptography (ECC) for a particular elliptic curve. Thus, the dependence (5) of the account identifier is given by,

$$BLK - SEQO(B) = BLK - SEQNO(BLK - PREV(B)) + 1 \tag{5}$$

Here Account is the type of account, and account_id : Account is a special variable of type Account.

Other chains may use other account ID formats, 256-bit or otherwise. For example, you can use Bitcoin-style account IDs that are equal to the sha256 of the ECC public key.

However, the bit length l of the account ID must be xed during chain creation (in the master chain), and it must be at least 64, because the first 64 bits of account_id are used for sharding and message routing.

Sharding is a mature concept originating from database design. It involves dividing and distributing a single logical data set across multiple databases that have nothing in common and can be deployed on multiple servers. Simply put, sharding provides horizontal scalability - breaking data into separate, independent parts that can be processed in parallel. This is a key concept in the global shift from data to big data. When data sets become too large to be processed by traditional means, there is no other way to scale than to split them into smaller parts.

The sharding mechanism in the TON blockchain allows you to process a large number of transactions simultaneously. TON's architecture consists of a single masterchain and up to 232 workchains. Each working chain is a separate blockchain link with its own rules. In addition, each workchain can be divided into 260 shardchains, or subchains, each of which contains a part of the working chain's state. Currently, there is only one working chain in TON - the Basechain. The main idea behind sharding in TON is that if account A sends a message to account B and account C sends a message to account D, these operations can be performed asynchronously.

The Masterchain is the main chain that stores the network configuration and the final state of all work chains.

The Masterchain itself is divided into separate chains called Workchains. Workchains are individual blockchains tailored to specific transactions or use cases that run in parallel on the TON network. An example of chain communication is shown in Figure 1.

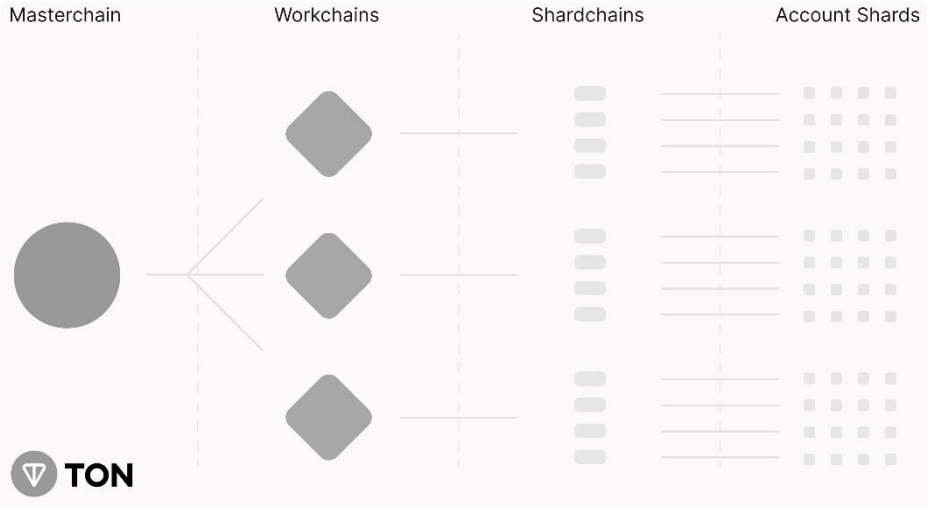

Fig.1. TON blockchain sharding [2]

Analysis of Smart contracts in TON blockchain

Smart contracts are typically used to automate the execution of a transaction so that all parties can be sure of the outcome immediately, without the need for intermediaries and without wasting time. They can also automate workflow by triggering the next action when predefined conditions are met.

Smart contracts are executed instantly when a condition is met. Since they are digital and automated, there is no need to process paper documents, nor to spend time correcting errors that often occur with manual processing. The absence of third parties and encrypted transaction records that are available to all participants eliminate the need to question whether data has been altered for personal gain.

Transaction records in the blockchain are encrypted, making them difficult to hack. In addition, since each record is linked to the previous and subsequent ones in the distributed ledger, hackers need to change the entire chain to alter one record. They also eliminate the need for intermediaries to carry out transactions, thereby reducing the delays and fees associated with their involvement.

The TON blockchain uses the FunC programming language to write contracts. FunC is a domain-specific C-like language with static typing. Here is a simple example of a method for transferring money written in FunC pseudocode:

```
() send_money(slice address, int amount) impure inline {
   var msg = begin_cell()
      .store_uint(0x10, 6) ;; nobounce
      .store_slice(address)
      .store_coins(amount)
      .end_cell();

   send_raw_message(msg, 64);
}
```

Smart contracts find their application in many areas where process automation and efficiency are required. One of the main areas of use is financial transactions and asset management. Smart contracts can automate payment processes and ensure the secure exchange of assets without the need for intermediaries. This greatly simplifies loan agreements, as the contract can automatically make loan payments or activate sanctions in case of breach of contract. In addition, they can be used to manage investments, where contracts will independently distribute profits or pay dividends in accordance with the established conditions.

Various games using smart contracts are also gaining popularity. Since TON is a product of the company that created one of the most popular messengers Telegram, their close interaction allows both developers and users to get the most comfortable conditions for using the ecosystem.

The creation of various applications within Telegram opens up many prospects for creating crypto wallets, games, quests, etc.

Smart contracts and TON infrastructure can also be used for educational institutions and their internal processes. For example, scientific articles can be stored in the blockchain, which would allow for a transparent approach to their storage and editing by co-authors.

In the TON blockchain, in order to create a smart contract, you will need to pay a fee to deposit it into the network, and then pay a fee for its existence and use. Today, most blockchains strive to make transactions as fast as possible for the lowest possible fee for the use.

All computation costs are denominated in gas units and are fixed in a certain amount of gas.

The price of gas units is determined by the chain configuration and can only be changed with the consent of the validators. Unlike other systems, users cannot set their own gas price, and there is no commission market.

The current settings in the base chain are as follows: 1 unit of gas costs 400 nanotons. This is reflected in dependence (6).

$$1\,gas = 26214400\ / 2^{16}\,nanotons = 0{,}0000004\ TON \tag{6}$$

The current settings in the masterchain are as follows: 1 unit of gas costs 10,000 nanotons. This is reflected in dependence (7).

$$1\,gas = 655360000\ / 2^{16}\,nanotons = 0{,}00001\,TON \tag{7}$$

Even if the price of TON increases 100 times, transactions will remain ultra-cheap - less than $0.01. Moreover, validators can lower this value if they see that fees have become expensive. That is why this blockchain is very promising as it has a number of advantages over other, older blockchains such as Bitcoin and Ethereum, where fees can reach up to $20 per transaction. Table 1 shows examples of fees in several of the most popular blockchains.

Table 1

Average blockchain fees for the period July 2024 - September 2024

№	Blockchain name	Fee, native coin	Fee, $
1	TON	0.0055 TON	0.003 $
2	Bitcoin	0.000012 BTC	0.79 $
3	Ethereum	0.0016 ETH	4.35 $

As in any blockchain, fees are difficult to calculate in advance, as their size depends on the transaction execution time, account status, message content and size, blockchain network settings, and a number of other variables that cannot be calculated until the transaction is sent.

There are certain formulas for calculating the commission, in the TON blockchain, the dependence (8) for calculating the commission looks like this:

$$transaction\_fee = storage\_fees + in\_fwd\_fees + computation\_fees + actoin\_fees + \\ + out\_fwd\_fees \tag{8}$$

where: *storage_fees* - is the amount you pay for storing a smart contract on the blockchain. In fact, you pay for every second of storing a smart contract on the blockchain.

in_fwd_fees - is a fee for importing messages only from outside the blockchain, for example, external messages. Every time you make a transaction, it must be delivered to validators who will process it. For normal contract-to-contract messages, this fee does not apply.

computation_fees - is the amount you pay to execute code on a virtual machine. The larger the code, the more you need to pay.

action_fees - fees for sending outgoing smart contract messages, updating smart contract code, updating libraries, etc.

out_fwd_fees - a fee for sending messages outside the TON Blockchain to interact with off-network services (e.g., logs) and external blockchains.

In fact, this is a general formula for calculating fees in the TON blockchain, but it consists of several complex elements. If we take a closer look at the storage_fees part, as mentioned earlier, it is responsible for paying for the storage of the smart contract in the middle of the blockchain. The storage fee is charged from the smart contract balance at the stage of storing any transaction through payments for storing the account state (including the smart contract code and data, if any) up to the present time. As a result, the smart contract may be frozen.

The storage fee depends on the size of the contract: the number of cells and the sum of the number of bits from these cells. Only unique hash cells are taken into account when calculating storage and transfer fees, i.e. 3 identical hash cells are considered as one. This means that you will have to pay for storing a TON of wallet (even if it is very, very small).

Using the formula (9) below, you can approximately calculate the fee for storing smart contracts:

$$storage\_fee = (cells\_count * cell\_price + bits * bit\_price) * time\_delta / 2^{16} \quad (9)$$

Where:
storage_fee - price for storage during time_delta seconds
cells_count - the number of cells used by the smart contract
bits_count - the number of bits used by the smart contract
cell_price - the price of one cell
bit_price - the price of one bit

Both cell_price and bit_price can be obtained from the TON Network Config, currently these values are as follows:

Workchain:
bit_price_ps: 1
cell_price_ps: 500
Masterchain:
mc_bit_price_ps: 1000
mc_cell_price_ps: 500000

In order to calculate the approximate cost, we first need to determine the size that should be represented in bits, let the weight of our smart contract be 1 Mb, then we have dependence (10):

$$size = 1024 * 1024 * 8 = 8388608 \quad (10)$$

And let's say that the duration of our smart contract is 1 year, so we have a formula for calculating the duration (11):

$$duration = 60 * 60 * 24 * 365 = 31536000 \quad (11)$$

We take into account the values for the workchain, namely the price of a bit (12) and the price of a cell (13):

$$bit\_prise\_ps = 1 \quad (12)$$

$$cell\_price\_ps = 500 \quad (13)$$

The next step is to calculate the price per second, which uses the size of the smart contract, the price per bit set in the chain, and the price per cell in the TON blockchain, so formula (14) for this calculation is as follows:

$$price\_per\_sececo = size * bit\_price\_ps + (size/1023) * cell\_price\_ps = \\ = 8388608 * 1 + (8388608/1023) * 500 = 1?24886 * 10^7 \quad (14)$$

Since we already have many of the necessary parameters for calculating the commission, namely the price per second and the number of seconds equal to one year, formula (15) for calculating the commission in TON will look like this:

$$fee = (price\_per\_decond * duration / 2^{16}) * 10^{-9} = \\ = (1/24886 * 10^7 * 31536000 / 2^{16}) * 10^{-9} = 6.00977 TON \quad (15)$$

Therefore, due to this calculation, we realised that the cost of storing a 1 megabyte smart contract for one year will cost about 6 TON, as of September 2024, the price of a TON on the exchanges was trading at about $5.5, so the cost of storage for a year will be about $33.

In Ethereum, fees also depend on the amount of data and complexity of transactions, but are calculated using the gas mechanism. In Ethereum, users pay for gas, which measures the computing resources required to complete transactions. Formula (16) for calculating the commission in Ethereum looks like this:

$$fee = gas\_used * gas\_price \quad (16)$$

where gas_used is the amount of gas required to complete the transaction, and gas_price is the price of gas in monetary terms.

The main reason for the differences in fees between TON and Ethereum is the architectural features. TON uses a sharding mechanism that allows you to distribute the load between several shard chains, which significantly reduces the overall cost of transaction processing. At the same time, Ethereum, especially before the transition to Proof of Stake, faces bandwidth limitations, which leads to high fees during busy periods.

As a result, the TON blockchain provides lower fees and higher efficiency, which makes it more attractive to users and developers, especially in the context of educational processes and the use of smart contracts.

If we consider the use of a smart contract in the context of preserving scientific articles, the blockchain itself does not provide a direct implementation for saving, for example, pdf files, but this is not entirely appropriate since they are usually not very small, which is why storing such files would require significant financial resources. Therefore, the IPFS protocol can be used for this purpose.

IPFS (InterPlanetary File System) is a protocol and system for distributing data according to its content, which is based on identifying data using a hash, i.e. a unique identifier that is determined based on the internal content of the data. This approach differs significantly from the current addressing system, where objects are assigned artificial names that do not reflect either the structure or content of the data. IPFS uses a peer-to-peer model to store and share hypermedia data in a distributed file system. The system was created by Juan Benet and is an open source project with community support.

IPFS is a powerful technology for decentralised file storage and distribution that provides efficient operation through the use of a peer-to-peer network. The main difference between IPFS and traditional file storage methods is that the system addresses data by its content rather than location. Each file or block of data stored in IPFS is assigned a unique hash identifier based on its contents. This makes it easy to verify the authenticity and integrity of data, as any changes to a file automatically change its hash.

To set up IPFS, you first need to download and install the IPFS software on your local computer or server. After that, the network node becomes part of the global peer-to-peer system. Users can add files to the system by obtaining hashes that make the files available to any user on the IPFS network. To do this, a file is uploaded to the local node, and the system generates a hash for it. This hash acts as a reference to the file on the network, allowing other users to download the file directly from the various nodes that store copies of it.

Applications of IPFS in modern technology include decentralised data storage for blockchains, distributed websites and applications that can run without a central server. For example, instead of storing large files on the blockchain, as is the case with scientific articles or digital media files, IPFS allows you to store only a hash of the file in a smart contract, while the data itself is stored in an external storage. This greatly reduces storage costs and improves system efficiency by ensuring file reliability and availability without the need for a centralised server.

In fact, in order to make such a project happen, operations need to be implemented: Adding a co-author to an article, deleting it, adding the article itself, and reviewing the article.

The main model is the scientific article itself, and if we start from storing articles using the IPFS protocol, we will need to have a data structure that will be stored in the blockchain:

```
Article {
   ipfs_hash;  ;; PDF file hash (IPFS)
   title;      ;; Article name
   authors;    ;; List of authors
   status;     ;; Article status
}
```

The smart contract itself must necessarily contain the recv_internal method, as this is a function that is called when messages are received from other contracts in the TON blockchain. In this method, you can place any important logic that should occur at the very beginning of interaction with the contract. In fact, this function accepts input data and then processes it directly, depending on what operation the initiator wants to perform and what data he has provided to it. This is where all the necessary data validation should take place first, followed by the execution of the required operation.

The most important method that is necessary to implement the idea is to add a new article, respectively, for this you need to store the received hash of the article, its title, author and status in the blockchain. The status should be either NEW or DRAFT, because the article needs to be checked by someone. And later, when it is checked, people who have the appropriate role within the blockchain will be able to change the status and make the article published. In such a scheme, the question arises as to how these checkers will appear. To do this, you need to implement the appropriate functionality, a certain role structure so that you can subsequently appoint the appropriate people to check. You can also implement some automatic mechanisms for becoming a reviewer, for example, if your collection of published articles reaches 20, you can automatically get the rights or something like that. Almost any idea can be implemented with the help of smart contracts and they will be decentralised, which is a modern approach to solving problems.

Before launching smart contracts on the blockchain, it is very important to make sure that they work correctly and do not contain any vulnerabilities. Since smart contracts cannot be changed once deployed, any mistakes can lead to serious consequences, such as loss of funds or unauthorised access to data.

To avoid this, unit tests are first conducted. They check individual parts of the smart contract, its functions, and structure by recreating various interaction scenarios. This allows you to make sure that the logic is correct and identify potential problems.

The next stage is the simulation of real-world use. Here, the scenarios of using the contract in the network are modelled, including testing transactions, checking access rights, changing the contract state, and handling errors.

For an additional level of assurance, it is advisable to conduct a code audit. This is an independent check by companies that specialise in blockchain security. They can find vulnerabilities, such as resource overruns or possible attacks on the contract's security system.

Finally, formal verification can be performed, which uses mathematical methods to confirm the correctness of the contract. This ensures that the contract works according to the intended logic and does not contain critical errors.

Testing of a smart contract

Testing is an important part of smart contract development. Smart contracts often deal with money, and it is therefore very important that no user loses money because there is a bug in the smart contract. That is why each smart contract developer should share an automated test suite with their FunC implementation. Any user who wants to make sure that the contract is working properly can run the test suite and see for themselves.

Since testing is of great importance when developing smart contracts, there are a large number of tools and infrastructure dedicated to this topic in the TON ecosystem. Today, some of the most popular testing tools are:

The deployment of the contract in the testnet (testnet) is a live alternative instance of the entire TON blockchain, where the TON coin is not real and can be obtained for free. Obviously, this copy is not as secure as the main network, but it offers an interesting environment in which to do a lot of testing without using real funds. Therefore, any testing of the functionality of the final code should start with the test network

Local Blockchain with MyLocalTon is a Java-based desktop executable that runs a personal, local instance of the TON blockchain on a computer in which contracts can be deployed and interacted with. Another way to run a local TON private network is to use Kubernetes with the ton-k8s library. Undoubtedly, this method of interaction with the blockchain is much more complicated than the previous method, since it is necessary to have deeper knowledge in the field of both the blockchain and the configuration of various systems. However, this method is better in terms of interaction with the network itself, since there will be no problems in the connection itself. The fact is that when public networks are used, they are provided by different providers, sometimes they can be completely free, and sometimes in order to receive certain resources or a larger number of requests, you need to pay for various subscriptions. But the public test network is slow and the transaction execution time can be very, very long, which is unacceptable during the testing period of your product. Therefore, the deployment of a local blockchain provides speed in testing for conducting transactions, deploying smart contracts, etc.

Writing tests in the FunC language is an important stage in the development of smart contracts, which allows you to test their functionality before deployment in a real network. To do this, we use the toncli tool, a command-line utility written in Python that runs on the local machine and supports debugging and unit tests for FunC contracts. Interestingly, the tests themselves are also written in FunC, which simplifies the process of testing the logic of contracts. In general, toncli simplifies the process of developing and testing FunC contracts, allowing for pre-validation of logic and vulnerabilities, which is critical for blockchain applications.

Deploying beta versions of smart contracts to the mainnet is an approach known as "testing in production," where instead of using a testnet, beta versions of smart contracts are deployed directly on the mainnet. In this process, real, not free, TON tokens are used, which provides an opportunity to test the operation of the contract in a real environment with all its features and nuances. This approach should be used not instead of the test network, but after it, as the final stage in testing the finished smart contract. Although testing on the mainnet is risky and expensive, as you have to pay a fee each time for each patched contract deployment, it provides a unique opportunity to test a contract in real-world conditions, which is often impossible to achieve in a test environment.

Conclusions

This work considered the possibilities of using smart contracts in the TON blockchain for automating educational processes and managing scientific data. Blockchain technologies, in particular TON, open new horizons in education systems, providing transparency, security and decentralization, which helps reduce costs and reduce the impact of the human factor.

Special emphasis is placed on sharding technology in the TON blockchain, which provides high bandwidth and scalability of the network, allowing to process thousands of transactions simultaneously. This is a significant advantage over traditional blockchains such as Ethereum and Bitcoin, which face speed and energy efficiency limitations. The Proof of Stake consensus model used by TON makes it more energy efficient compared to the Proof of Work used in Bitcoin and the older version of Ethereum.

The integration of IPFS as a mechanism for storing large files such as scientific articles allows combining the benefits of decentralized storage with the efficiency of smart contracts on the TON blockchain. This allows you

to create transparent information storage systems with the possibility of confirming authorship, which is especially useful for scientific and educational processes.

It is also worth noting the possibility of using the TON blockchain to create decentralized platforms for sharing knowledge and educational materials. Such platforms can ensure the safety and authenticity of scientific publications, as well as provide access to knowledge without intermediaries. In addition, the development of smart contract technologies will allow automating the processes of financing educational programs and managing grants, ensuring transparency and control over costs.

Future updates to the TON blockchain, including improvements in sharding and scalability mechanisms, can further reduce transaction costs and increase data processing speed, which will facilitate its widespread adoption in educational and scientific systems. This opens up new opportunities for the development of personalized educational programs and decentralized educational initiatives.

Thus, the TON blockchain shows great potential for implementation in educational systems, providing effective tools for data storage, management and exchange. Further developments in this direction can contribute not only to technological progress, but also to the development of educational and scientific processes throughout the world.

References

1. Blockchain. URL: https://en.wikipedia.org/wiki/Blockchain (Last accessed 23.12.2024)
2. What is The Open Network? A beginner's guide to the TON blockchain. URL: https://www.theblock.co/learn/298587/what-is-the-open-network-a-beginners-guide-to-ton?modal=newsletter (Last accessed 27.12.2024)
3. Nehru, R. S. S., Cuong, T. Q., Prakash, A. R., & Huong, B. T. T. (2023). Higher Education: AI Applications for Blockchain-Based IoT Technology and Networks. In AI Models for Blockchain-Based Intelligent Networks in IoT Systems: Concepts, Methodologies, Tools, and Applications (pp. 261-283). Cham: Springer International Publishing.
4. Iyer, S., Seetharaman, A., & Maddulety, K. (2021). Block chain technology and its impact on education sector. International Journal of Innovation in Education, 7(1), 1-16.
5. Dwivedi, V. K., Iqbal, M., Norta, A., & Matulevičius, R. (2023). Evaluation of a legally binding smart-contract language for blockchain applications. Journal of Universal Computer Science, 29(7), 691-717.
6. Ivanov, N., Li, C., Yan, Q., Sun, Z., Cao, Z., & Luo, X. (2023). Security threat mitigation for smart contracts: A comprehensive survey. ACM Computing Surveys, 55(14s), 1-37.
7. Al-Zoubi, A., Dmour, M., & Aldmour, R. (2022). Blockchain as a learning management system for laboratories 4.0. iJOE, 18(12), 17.
8. Hovorushchenko, T., & Osiadlyi, V. . (2022). A METHOD FOR EVALUATING THE REPUTATION OF A MEDICAL INSTITUTION (AS A BLOCKCHAIN MINER). Computer Systems and Information Technologies, (3), 56–60. https://doi.org/10.31891/csit-2022-3-7
9. Dwivedi, V., Pattanaik, V., Deval, V., Dixit, A., Norta, A., & Draheim, D. (2021). Legally enforceable smart-contract languages: A systematic literature review. ACM Computing Surveys (CSUR), 54(5), 1-34.
10. Shards. URL: https://docs.ton.org/develop/blockchain/shards (Last accessed 29.12.2024)
11. Smart contracts. URL: https://www.ibm.com/topics/smart-contracts (Last accessed 22.12.2024)
12. Transaction Fees. URL: https://docs.ton.org/develop/smart-contracts/fees (Last accessed 22.12.2024)
13. IPFS. URL: https://en.wikipedia.org/wiki/InterPlanetary_File_System (Last accessed 23.12.2024)
14. Testing. URL: https://tonhelloworld.com/04-testing/ (Last accessed 23.12.2024)

В'ячеслав Аскеров **Viacheslav Askerov**	Postgraduate student of Computer Engineering & Information Systems Department, Khmelnytskyi National University, Khmelnytskyi, Ukraine, e-mail: vyacheslav@askerov.com https://orcid.org/0009-0009-1176-9812	аспірант кафедри комп'ютерної інженерії та інформаційних систем, Хмельницький національний університет, Хмельницький, Україна.
Богдан Томчишен **Bohdan Tomchyshen**	Master degree student of Computer Engineering & Information Systems Department, Khmelnytskyi National University, Khmelnytskyi, Ukraine, e-mail: btomchishen@icloud.com	магістрант спеціальності «Комп'ютерна інженерія», Хмельницький національний університет, Хмельницький, Україна.
Худа Ель Бухісі **Houda El Bouhissi**	PhD, Associated professor of LIMED Laboratory, Faculty of Exact Sciences,University of Bejaia, Algeria e-mail: houda.elbouhissi@gmail.com https://orcid.org/0000-0003-3239-8255	Доктор філософії, доцент лабораторії LIMED, факультет точних наук, Університет Беджая, Алжир.