



TAX CONTROL AUTOMATION USING SMART CONTRACTS

Radovan Vladislavljević^{1 *}, Aleksandra Zlatić Tešić², Svetlana Marković²

¹ *University Business Academy in Novi Sad, Faculty of Economics and Engineering
Management in Novi Sad, Republic of Serbia*

² *University Business Academy in Novi Sad, The Faculty of Law for Commerce and Judiciary
in Novi Sad, Republic of Serbia*

Abstract: The aim of the work is to present a model of tax control automation using smart contracts, this is a relatively new application of blockchain technologies. The use of new technologies can greatly improve the operations of modern organizations that have digitized their operations. New technologies not only provide a high degree of automation but also provide a high degree of transparency. This leads to faster business with an increase in the level of trust of all participants in the business venture.

Keywords: Blockchain, smart contracts, business transparency, tax law.

1. INTRODUCTION

Digital technologies in the business sphere make new business models not only possible but also highly profitable. Modern business organizations rely more and more on information as primary strategic resources, one of the major advances in this domain is the emergence of a new way of storing data based on non-destructive methods. Classic databases either overwrite new values over old ones or created new records that have little relation to previous records. Blockchain technologies are based on the linking of records, the records are kept in blocks, and the blocks are linked via a unique hash code to each other.

This is just one of the characteristics of data storage in the blockchain, data security, i.e. protection against unauthorized data changes, is ensured by decentralization. Decentralization in the blockchain domain is achieved by allowing blockchains to be freely shared among users. When entering new blocks, the new data is first validated, then all copies of the chains are updated. An added value of this system is that all users can see changes to blocks and explore the history of block additions.

However, for the purposes of this paper, the most important feature of this system is the use of smart contracts. Smart contracts have little to do with "classic" contracts, i.e. smart

* Corresponding author: radovan.vladislavljevic@fimek.edu.rs

Radovan Vladislavljević, ORCID: 0000-0002-8502-8584

Aleksandra Zlatić Tešić, ORCID: 0000-0001-5993-3065

Svetlana Marković, ORCID: 0000-0003-2776-6519

contracts are not the same as legally defined contracts. Smart contracts are a piece of programming code that executes when conditions are met. In blockchain technologies, smart contracts are the basis for managing many actions and are the foundation of tokenization in the cryptocurrency world.

In the presented model, the purpose of this system is to create a transparent and secure business monitoring system with compliance with tax obligations prescribed by law. The transparency that this new technology brings to business provides an additional sense of security not only for participants but also for representatives of tax authorities. Virtually all interested parties can monitor financial flows with an emphasis on settling tax liabilities. "Other than all transactions being recorded on the blockchain, there is another side to transparency when it comes to smart contracts, which is the transparency of the program" (Arjuna Sky Kok, 2024, 46).

Many business ventures that started their business life relatively successfully end their life amid neglect of tax obligations. The presented model can be used to monitor the development of the business life of startup companies, which most often have no experience in the field of tax liability management. Of course, previously established companies can also benefit from our model. The financial segment related to taxes can be an extremely challenging field that, with transparency and external expertise, can be simplified and financial aspects improved. This is just another way to present companies to potential investors through new technologies and provide basic data needed for making investment decisions.

2. BLOCKCHAIN TECHNOLOGIES

Blockchain technologies are a relatively new term in the software industry. These technologies are most commonly associated with cryptocurrencies, however these technologies can exist independently of cryptocurrencies. On the other hand, cryptocurrencies are closely related to blockchain technologies and cannot exist independently of them. The plural is used intentionally because there is a whole range of technologies that support the operation of blockchain. Blockchain is not a single technology but a collection of different platforms, consensus algorithms and network models, all of which use the same basic principle of a distributed and cryptographically linked database.

The basis of these technologies is related to cryptographic techniques of binding one block to another. A "hash" function is a cryptographic technique that takes a variable input and produces an output in the form of a "random" sequence of numbers and letters. Deliberately, the word is randomly placed in quotation marks, because a "hash" code is obtained based on a precisely determined mathematical function of the input. Even the slightest change to the input values affects the final cipher, this technique belongs to one-way encryption. Once obtained, the "hash" code can no longer be returned to its original form, for the purposes of the blockchain it is not even necessary. The power of the "hash" function is that it can take a large amount of data as an input. In the blockchain domain this is significant because the entire block enters the "hash" function where the output in the form of a code becomes an integral part of the next block. In this way, a chain of blocks is formed and they are conditioned by the content of the previous block. Of course the first block, which is called the "genesis" block, does not have two predecessors. In this way, the contents of the blocks are secured against possible changes, whether intentional or unintentional. "Transactions are the basic units of activity documented on the blockchain, which represent actions or operations performed within the blockchain network" (Ramasamy & Khan, 2024).

Another advantage of blockchain is decentralization without a single system of governance or oversight. "Decentralization is an important characteristic that can tell

blockchain systems from traditional centralized systems" (Wu, 2025, 77). The blockchain itself is transparent and can be searched by anyone, also anyone who is interested can download the entire transaction chain to their computer or server. The insertion of new blocks into the chain is done through a pre-agreed rule that ensures that there will be no unnecessary duplication of data or illegal entries.

Table 1. Types of blockchain technologies (based on Types of Blockchain: PoW, PoS, Private, and DLT, (n.d), Understanding the Different Blockchain Types - Paxos | Blog, (2024), GeeksforGeeks, (2022))

Blockchain technology	Network type	Consensus algorithm	Smart contracts	Main purpose
Bitcoin	Public	Proof of Work (PoW)	No (limited)	Digital money, transfer of value
Ethereum	Public	Proof of Stake (PoS)	Yes	Smart contracts and decentralized applications
Solana	Public	Proof of History + PoS	Yes	High-throughput decentralized applications
Cardano	Public	Ouroboros (PoS)	Yes	Secure and scalable blockchain applications
Polkadot	Public	Nominated Proof of Stake	Yes	Connecting different blockchain networks
Hyperledger Fabric	Private / Consortium	Practical Byzantine Fault Tolerance (PBFT)	Yes	Business and industrial applications
Corda	Consortium	Notary consensus	Limited	Financial institutions and the banking sector
Quorum	Private	Istanbul BFT / Raft	Yes	Corporate blockchain systems

The previous table shows some of the most important technologies that are currently used in practice. As can be seen not all blockchain technologies are open to smart contracts, the best example is the Bitcoin blockchain which is closely coupled with digital currency. Some like the Ethereum blockchain are far more flexible and can be easily adjusted for non-cryptocurrency needs.

3. THE DIFFERENCE BETWEEN DATABASES AND BLOCKCHAIN TECHNOLOGIES

Traditional databases are based on the mutable data model, where existing records can be changed or overwritten with new values. In contrast, blockchain systems use a model of immutable data storage, where new transactions are added to the block chain without the possibility of changing previously recorded data. "The shared ledger is immutable because it records all transactions from the beginning until the end. The data recording of transactions in a shared ledger is done only once" (Setyowati et al., 2020).

Table 2. Differences between relational data and blockchain

Feature	Relational databases	Blockchain
Data modification	allowed	not allowed
Model	mutable	immutable
Record type	overwrite or append	append only
Integrity	system control	cryptographic verification
History	not required	there is always

Databases can be updated directly, that is, new data can be overwritten over old data. This is a destructive or mutable mode of working with data.

UPDATE racun SET stanje = stanje - 200 WHERE id = 1;
--

Figure 1. An example of a SQL query where a record change is made directly

In this way, the trace of the old data is lost, while only the new updated data is visible. In most cases, this is a valid way of handling data. However, in the financial and banking sector, it is very important to have a complete picture of changes in client accounts. And for this reason, it is possible to work in the append mode, that is, the mode of adding new records in the database.

INSERT INTO transactions (account_id, type, amount) VALUES (101, 'payment', 1000);	INSERT INTO transactions (account_id, type, amount) VALUES (101, 'payment', 200);
--	---

Figure 2. Example of SQL query where new values are entered

Table 3. Appearance of the transaction in the database

id	account	type	amount	date
1	101	uplata	1000	2026-03-30
2	101	isplata	200	2026-03-30

This mode of operation is most similar to blockchain, especially when paired with an audit log system. In this operating mode, balance tables, transactions and audit logs are separated. Status monitoring and access control become possible by following certain procedures. However, even in this operating system, someone with administrative rights can make data changes.

4. SMART CONTRACTS

Basically smart contracts are a piece of code that executes within the blockchain, even though they have the word contract in the name, smart contracts have nothing to do with legal contracts. The purpose of smart contracts is to automate certain transactions within the blockchain. However, smart contracts are outgrowing their purpose and today they are intensively used in other areas as well.

An entire branch of cryptocurrencies is based on the use of smart contracts. Some cryptocurrencies and most meme coins are regulated by smart contracts that define basic parameters. This just shows the importance of smart contracts as well as the flexibility of this technology.

Smart contracts are programming code that runs on the blockchain network and automatically enforces predefined rules and conditions of transactions between participants. The basic idea of smart contracts is to automate the execution of contractual obligations without the need for intermediaries. When the defined conditions are met, the smart contract

automatically executes the appropriate action. This approach enables greater transparency, security and efficiency in the implementation of various business processes. "Traditional systems, which often use centralized architectures, face several vulnerabilities and limitations that make them susceptible to various cyber threats and cyber attacks" (El Madhoun, 2024, 19).

When developing a smart contract, it is first necessary to define the business model or scenario that the contract should implement. This implies the identification of the participants in the process, the definition of the events that trigger the execution of the contract, as well as the determination of the conditions that must be met in order for a certain action to be carried out. In this context, the smart contract works according to the "if - then" logic, i.e. if a certain condition is met, the corresponding operation defined by the contract code is executed.

One of the basic elements of a smart contract is represented by the participants of the system. Participants are identified through blockchain addresses and represent the parties participating in the transaction or process that the contract regulates. Depending on the specific application, the participants can be a buyer, a seller, a system administrator or an external data source (oracle).

Another important element is the state of the contract. Smart contracts often go through different stages or states during their life cycle. For example, a contract can have a creation state, a funds payment state, a transaction execution state, and a closing state. Changing the state enables control of the process flow and prevents the execution of unauthorized operations.

The third element refers to the rules and conditions of execution. These rules define who has the right to perform a certain function and under what circumstances a certain operation can be initiated. For example, only a specific participant can confirm receipt of goods or initiate a refund.

In addition, smart contracts contain functions that represent operations that participants can run. Functions can include various activities such as payment of funds, confirmation of service performance, and transfer of funds or closing of contracts. These functions change the state of the contract and update the data stored on the blockchain.

Data storage is also an important element. The smart contract must store relevant information about the transaction, such as the identity of the participants, the amount of the transaction, the status of the process, and time parameters. These data enable the transparency and verifiability of the performed operations.

Additionally, smart contracts often generate events that allow tracking of activity within the blockchain network. Events represent a record of actions performed and allow external systems to react to changes in contract state.

A very important aspect of the development of smart contracts is security. Since smart contracts are executed on decentralized networks and their logic often manages financial assets, it is necessary to implement appropriate access control mechanisms, condition checks and protection against potential vulnerabilities.

The development of smart contracts requires the use of specific technologies that enable the implementation and execution of program code on blockchain networks. These technologies include programming languages for writing smart contracts, development environments, testing tools, and platforms for executing smart contracts. The largest number of modern smart contracts are developed for blockchain platforms that support the execution of decentralized applications, among which Ethereum stands out in particular.

Blockchain platforms that enable the development of smart contracts typically contain a virtual machine that executes the contract's programming code. In the case of the Ethereum platform, it is the Ethereum Virtual Machine (EVM), which is a decentralized environment for executing smart contract code. The virtual machine allows smart contracts to be executed on a

large number of nodes in the network, thus ensuring the decentralization and reliability of the system.

For the development of smart contracts, special programming languages have been developed that enable the definition of contract logic and interaction with the blockchain infrastructure. The most famous and widely used programming language for the development of smart contracts is Solidity, which was developed within the ecosystem of the Ethereum platform. Solidity is a high-level object-oriented programming language, which is syntactically similar to programming languages such as JavaScript, C++, and Python. "Solidity is by far the most popular language for EVM ("Ethereum Virtual Machine")" (Modi, 2022, 61).

The Solidity programming language allows defining the structure of a smart contract through the declaration of variables, functions, events and control structures. Using this language, it is possible to implement complex business processes and automate the execution of various types of transactions. Solidity supports the concept of contracts as the basic unit of a program, where each contract can contain data and functions that define the rules of interaction between participants.

One of the important features of the Solidity language is the ability to define modifiers and access controls. In this way, it is possible to limit the execution of certain functions only to certain system participants, thus ensuring the security and integrity of smart contracts. In addition, Solidity allows the generation of events that are recorded on the blockchain and that allow monitoring of activities within the system.

In addition to the programming language, the development of smart contracts also requires the use of specialized development tools. Among the most well-known tools for developing smart contracts are development environments and frameworks such as Truffle, Hardhat, and Remix. These tools allow developers to write, test and implement smart contracts on the blockchain network. The Remix tool, which represents an integrated development environment intended for the development and testing of smart contracts written in the Solidity language, is particularly important.

In addition to the Ethereum platform, smart contracts can be developed on other blockchain platforms that support code execution. Platforms such as Solana, Cardano and Polkadot stand out among them, which use different programming languages and development models to implement smart contracts.

Nevertheless, thanks to a large community of developers, wide tool support and a developed ecosystem of decentralized applications, Solidity has become the dominant programming language for developing smart contracts. Its popularity has been further boosted by the development of token standards, such as ERC-20 and ERC-721, which allow the creation of cryptocurrencies and digital tokens on the blockchain network.

Technologies for the development of smart contracts include a set of programming languages, development tools and blockchain platforms that enable the implementation of automated digital contracts. Among these technologies, Solidity occupies a central place, because it represents the standard language for the development of smart contracts within the ecosystem of the Ethereum platform and the basis for the development of a large number of decentralized applications and blockchain projects.

5. DIGITALIZATION OF TAX SYSTEMS

Digitalization of tax systems represents one of the key directions of modernization of public administration and improvement of the fiscal efficiency of states. The development of information and communication technologies enabled the transformation of traditional tax administrations into digital systems that use automated processes, electronic data exchange and

advanced analytical tools for managing tax obligations and controlling taxpayers. The special importance of digitalization is reflected in the fight against tax evasion, which represents one of the biggest challenges of modern tax systems.

Tax evasion involves the illegal avoidance of paying tax obligations by concealing income, falsifying financial statements or using complex financial structures to reduce the tax burden. Such activities have a negative impact on public finances, reduce the state's fiscal revenues and violate the principle of tax fairness. In this context, the digitalization of tax systems is an important instrument for improving the transparency of financial flows and more effectively detecting irregularities in the operations of taxpayers.

One of the key elements of digitalization of tax systems is the introduction of electronic submission of tax returns and electronic exchange of data between taxpayers and tax administrations. Electronic systems enable automatic processing of a large amount of data and reduce the possibility of information manipulation. In addition, digital systems enable the integration of various databases, such as banking data, data on the circulation of goods and services, as well as data from customs and financial institutions.

Modern digital tax systems increasingly use advanced analytical methods, including Big Data analysis and artificial intelligence algorithms, to identify patterns of behavior that may indicate tax evasion. These systems enable automatic identification of suspicious transactions, analysis of financial flows and risk assessment of individual taxpayers. In this way, tax administrations can more efficiently focus their control activities on entities that represent the greatest risk for the fiscal system.

The digitalization of tax systems also includes the development of electronic invoice systems and digital fiscal devices that enable monitoring of transactions in real time. Such systems enable the tax authorities to have a direct insight into the turnover of goods and services, thus reducing the scope for hiding income and manipulating financial statements. In many countries, electronic fiscalization systems have been introduced that automatically send data on each transaction to the tax administration.

New technologies such as blockchain and smart contracts have a special potential in the digitalization of tax systems. These technologies enable transparent and immutable recording of transactions, which can contribute to reducing the possibility of data manipulation. Smart contracts can automate certain tax processes, such as tax calculation and payment, thereby reducing the need for manual interventions and increasing system reliability.

In addition to technological advantages, the digitalization of tax systems also brings certain challenges. Among the most important challenges are data protection, security of information systems and the need for continuous improvement of the digital capacities of tax administrations. Also, successful implementation of digital tax systems requires adequate regulatory support, development of digital infrastructure and education of taxpayers.

Digitalization of tax systems is an important instrument for improving the efficiency of tax administration and reducing tax evasion. By using modern information technologies, it is possible to improve the transparency of financial flows, increase the level of tax discipline and ensure more stable public revenues. In this sense, the digital transformation of tax systems is one of the key elements of modern fiscal policies.

6. A MODEL OF AUTOMATED TAX CONTROL USING A SMART CONTRACT

The model presented in this paper is based on an Escrow account with the idea that during each transaction a part of the funds is automatically allocated as tax and forwarded to

the tax administration, while the rest is transferred to the seller. This model can reduce the possibility of tax evasion because tax is collected automatically at the time of transaction.

An escrow account is a special form of temporary financial account used for the safe execution of transactions between two or more parties. The essence of the escrow mechanism is based on the involvement of a neutral third party that temporarily holds financial resources or assets until the predefined transaction conditions are met. "An escrow account consists of funds held by a third party, who collects, holds, and disburses the funds pursuant to a contract or an obligation between two parties" (Mills, 1994, 204). "It is essentially a deposit of documents, securities, goods or money in a neutral and impartial party with specific instructions on how, in what way, and to whom the escrow holder hands over the documents, goods or money" (Handayani et al., 2021, 13293). When all conditions are met, the funds are released and transferred to the appropriate party in accordance with the contractual rules. This model is used in various business and financial transactions, especially in situations where there is a need to increase security and trust between participants who do not have a previously established business relationship. In modern digital systems, the concept of escrow accounts is increasingly implemented through smart contracts on blockchain platforms, such as Ethereum, where programming code automatically executes transaction rules without the need for traditional intermediaries. Such automation enables greater transparency, reduction of transaction costs and more efficient management of financial transactions.

At the very beginning of the code (uint public taxRate;) the tax rate is defined which can be a certain percentage of the amount, for example 10% or 20% sales tax.

The code segment (... createTransaction(address seller) ...) or rather the function defines a specific transaction. In this segment, the buyer sends funds to the contract, the contract automatically calculates the tax rate (... tax = amount * tax rate; to seller = amount - tax; ...). The function (...releasePayment() ...) of completing the transaction makes a payment to the tax authority and redirects the money to the seller's account.

```
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.0;

contract TaxEscrow {

    address public taxAuthority;
    uint public taxRate; // poreska stopa u procentima

    constructor(uint _taxRate) {
        taxAuthority = msg.sender;
        taxRate = _taxRate;
    }

    struct Transaction {
        address buyer;
        address seller;
        uint amount;
        uint tax;
        bool completed;
    }

    uint public transactionCount;
    mapping(uint => Transaction) public transactions;
    event TransactionCreated(uint id, address buyer, address seller, uint amount);
    event TransactionCompleted(uint id, uint taxPaid);

    modifier onlyTaxAuthority() {
        require(msg.sender == taxAuthority, "Only tax authority allowed");
        _;
    }

    function createTransaction(address _seller) public payable {
        require(msg.value > 0, "Payment required");
        uint taxAmount = (msg.value * taxRate) / 100;
        uint sellerAmount = msg.value - taxAmount;
        transactionCount++;
        transactions[transactionCount] = Transaction(
            msg.sender,
            _seller,
            sellerAmount,
            taxAmount,
            false
        );

        emit TransactionCreated(transactionCount, msg.sender, _seller, msg.value);
    }

    function releasePayment(uint _transactionId) public {
        Transaction storage txn = transactions[_transactionId];
        require(!txn.completed, "Transaction already completed");
        txn.completed = true;
        payable(txn.seller).transfer(txn.amount);
        payable(taxAuthority).transfer(txn.tax);
        emit TransactionCompleted(_transactionId, txn.tax);
    }

    function changeTaxRate(uint _newRate) public onlyTaxAuthority {
        taxRate = _newRate;
    }
}
```

Figure 3. Code of the tax liability automation model on the example of an Escrow account

7. CONCLUSION

The presented model clearly shows the importance of smart contracts in automating the control of tax liabilities. Blockchain technologies provide great flexibility and transparency, so that all stakeholders including the tax administration can monitor business transactions. Tax evasion is reduced to a minimum. The model shown in this paper is only a prototype that shows a business venture that includes the Escrow mechanism. Of course a different implementation is possible without using the mentioned mechanism.

The implementation of this model is possible in different scenarios, E-commerce platforms as well as digital fiscal systems or blockchain payment systems are ideal candidates for the implementation of an automatic tax control system. The payment of value added tax in the supply chain is also a typical candidate for the implementation of this or a similar model of tax control automation.

The tax administration, like everyone in business ventures, has a clear picture of the settlement of tax obligations. Of course it is possible to define a whole series of other parameters that can be monitored in real time with a smart contract. This leads to increased confidence in business as well as better planning of future business ventures.

Of course, this technology also has its drawbacks, which are reflected primarily in the relative rigidity of the data storage method itself. It is possible to enter data incorrectly due to human error. The correction of this error must be foreseen in advance or the system may be blocked. Also, a smart contract is not legally defined, nor is it the basis of legal protection, so in addition to smart contracts, legally defined contracts must be formed. Also, this technology must be adopted by state authorities, primarily the tax administration. On the other hand, we are witnessing drastic changes in society, state administration, as well as in the business sphere, so it is to be expected that new technologies will be accepted more quickly in the future.

REFERENCES

- Arjuna Sky Kok. (2024). Hands-On Blockchain for Python Developers - Second Edition. GeeksforGeeks. (2022, April 27). Types of Blockchain. GeeksforGeeks. <https://www.geeksforgeeks.org/ethical-hacking/types-of-blockchain/>
- Handayani, O., Sulistiyono, A., & Muryanto, Y. T. (2021). Mandatory escrow account as a manifestation of good faith in peer-to-peer lending.
- Mills, E. S. (1994). The functioning and regulation of escrow accounts. *Housing Policy Debate*, 5(2), 203–218. <https://doi.org/10.1080/10511482.1994.9521160>
- Modi, R. (2022). Solidity Programming Essentials: Second Edition, Packt.
- El Madhoun, N., Dionysiou, I., & Bertin, E. (Eds.). (2024). *Building Cybersecurity Applications with Blockchain and Smart Contracts*. Springer. <https://doi.org/10.1007/978-3-031-50733-5>
- Ramasamy, K., L., Khan, F. (2024). Blockchain for Global Education, Springer. <https://doi.org/10.1007/978-3-031-52123-2>
- Setyowati, M. S., Utami, N. D., Saragih, A. H., & Hendrawan, A. (2020). Blockchain technology application for value-added tax systems. *Journal of open innovation: technology, market, and complexity*, 6(4), 156. <https://doi.org/10.3390/joitmc6040156>
- Types of Blockchain: PoW, PoS, Private, and DLT. (n.d.). Gemini. <https://www.gemini.com/cryptopedia/blockchain-types-pow-pos-private>
- Understanding the Different Blockchain Types – Paxos. Blog. (2024). Paxos.com. <https://www.paxos.com/blog/understanding-the-different-blockchain-types>
- Wu, J. (2025). Blockchain Transaction Data Analytics. Springer Nature.