IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received March 14, 2022, accepted March 29, 2022, date of publication April 4, 2022, date of current version April 15, 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3164444

Smart Contract Application for Managing Land Administration System Transactions

MIROSLAV STEFANOVIĆ, (Graduate Student Member, IEEE), ĐORĐE PRŽULJ, SONJA RISTIĆ, (Member, IEEE), DARKO STEFANOVIĆ, (Member, IEEE), AND DANILO NIKOLIĆ, (Graduate Student Member, IEEE)
Faculty of Technical Sciences, University of Novi Sad, 21000 Novi Sad, Serbia

Corresponding author: Sonja Ristić (sdristic@uns.ac.rs)

ABSTRACT Land administration systems are of great importance for a large number of stakeholders. One of the key problems related to land administration systems is the problem of the correctness of their state, meaning that data stored in land administration systems are not in concordance with the actual legal, spatial and topographic situation. The main causes of land administration systems' incorrect state are data collection and compilation, data processing, and data misuse. In this paper, we discuss the problems of data tampering, the long time needed for registering land administration system's transactions, and the possibility of double spending, which all can add incorrectness in a land administration system. Our research is based on the hypothesis that these problems may be addressed by means of distributed ledger technology, or to be more precise, by means of blockchain technology. The solution is presented in a form of a smart contract written in Solidity programing language that can cover even those more specific use cases in land administration systems such as sharing of ownership, transferring part of ownership, splitting or merging of real estate, and limiting the possibility of trading a real estate. The proposed smart contract represents an implementation of a programming interface that was created based on both ERC-20 and ERC-721 token standards, to satisfy the specific needs of land administration systems.

INDEX TERMS Blockchain, Ethereum, land administration, real estate, smart contract.

I. INTRODUCTION

The land is a limited resource and land administration significantly impacts the economy, country development, and civil society. Land administration is done through Land administration systems (LAS) that hold records about location, ownership, and use of the real estate, but also about physical, spatial, and topographic data. In most cases, data about location, ownership, and use are stored in land registers, while physical, spatial, and topographic data are stored in cadastre. Data stored in land register and data stored in cadastre together form one LAS [1]. Although LAS fulfills many different roles, the primary role is to manage data about rights, restrictions, and responsibilities a party has on real estate and to provide that data upon request to interested parties.

In Europe, there are several different families of legal systems for land administration. The Code Napoleon system is in use in France, Italy, Spain, and Portugal. The Nordic system is in use in Sweden, Denmark, and Finland, the common law system is used in the UK, the German system is being used in Germany, Switzerland, and Austria, Easter European system is used in Czechia, Slovakia, and Hungary [2]. According to a study conducted in 25 European countries, 10 have LAS based on civil court registration, while the other 15 have systems based on local rights and are managed by other state agencies [3]. Back in 1998, the expectation was that by 2014, LAS would be highly privatized, but as of 2020, this is yet to be done [4]. The fact that, in most cases, LASs are managed by courts or other state agencies places services provided by contemporary LAS in the domain of e-government.

Different legal systems for land administration in different countries still share similar problems related to maintaining the correctness of data stored in LAS. Term correctness is here used to represent a concordance of data stored in LAS with actual legal, spatial, and topographic situations in reality [5]. Sadly, LASs are often not in the correct state. Incorrectnesses could be found both in the land register and cadastre. Examples of incorrectnesses in the land register

The associate editor coordinating the review of this manuscript and approving it for publication was Laxmisha Rai.

could be related to data about the owner, share of ownership, mortgage, land use, real estate size, and other data. Similarly, incorrectnesses in cadastre could be related to data about location (spatial, horizontal, and vertical), size, boundaries, themes, and other data. Data collection and compilation, data processing, and data misuse are identified as sources that lead to cadastral data not being in the correct state [6]. The same might be said for data stored in the land register.

Regarding data collection and compilation, it is important to acknowledge that before the process of digitization of LAS data, all data were kept in ''paper form'', and it was rather hard to enforce any integrity constraints during processes of creating and updating of records. Term integrity constraint is here used to represent formal statements, definitions or qualifications that are being used to describe requirements for logical data consistency. Logical consistency is one of six quantitative data quality elements defined in ISO 19157:2013 Geographic information — Data quality standard [7]. It refers to the degree of adherence to logical rules of data structure, attribution, and relationships specified by means of integrity constraints.

During the data processing, in the course of LAS digitization, in most cases, incorrectness that existed in ''paper form'' is transferred to digital LAS. Unintentional human errors during the digitization process can add new incorrectnesses into LAS [5]. As such, the initial process of digitalization did not solve the problem of the correctness of data stored in LAS.

LAS data incorrectness also leads to a situation in which two LAS subsystems, land register, and cadastre, are often not in an internally consistent state. An internally consistent state is a state in which data stored in both subsystems, regarding the same real-life entity, for example, area of real estate, are equal. As an illustration, in [8] it is stated that only 5% of data stored in the land register and cadastre in Croatia are in an internally consistent state. It is pretty safe to claim that LAS that is not in an internally inconsistent state is undoubtedly in an incorrect state too [9] and that would mean that 95% of data stored in Croatian LAS are in an incorrect state. Even though lack of internal consistency indicates an incorrect state, if land register and cadastre are in a consistent state, they could still be in an incorrect state, since data still might not be in concordance with actual legal, spatial, and topographic situations in reality. This problem is rather common in countries wherein process of creation of contemporary LAS it was necessary to harmonize data from different sources and that even had different land administration systems in use in different parts of its territory. In those countries, it was necessary to define, by law, which source of data will be deemed correct in case of inconsistency. For example, in Serbia in the Law on State Survey and Cadaster, it is stated that in the case of inconsistency of data stored in land register and cadastre, data stored in the land register are presumed to be correct.

There is also a more dire reason for adding new incorrectnesses into LAS, which is especially common in underdeveloped and developing countries, and that is a problem of intentional ''human errors''. Data misuse, or data tampering, as a result of corruption and fraud is the reason for the appearance of new incorrectnesses that brings the entire LAS in an incorrect state [10], [11]. Term data tampering is here used to represent the act of deliberately editing, destroying, or manipulating data through unauthorized channels.

A less dire reason why LAS is not in the correct state is due to the fact that the process of registration of real estate transactions in LAS takes unusually long in most countries. For example, it can take up to six months to register a transaction in Sweden [12] and up to 24 months to register it in Serbia. During that period, the new owner of the real estate has limited rights on the property because he is still not recognized as *de jure* (lat.) owner of real estate. Since registration of transactions in LAS could take a lot of time, it opens the door for the possibility of ''double spending'' of real estate, and these cases are not unknown or unheard of.

''Double spending'' of real estate might sound strange since real estate represents a tangible asset, but transactions of real estate are, as previously mentioned, usually conducted through the process of registration in civil court or some other agency and not by simply taking possession of an asset. So, it is possible to have a case of ''double spending'' where the same real estate could be sold multiple times before the transaction is officially registered. It is expected that transaction that happened first, in reality, will eventually be registered in LAS. But often those cases end up in the court of law and take a long time to resolve. During that time, neither of the involved parties have their rights established on the specific real estate.

To resolve problems related to double spending in real estate transactions, data tampering, and the long duration of the real estate transactions' registration process, a novel system for registering transactions is needed, and a possible solution could be in the application of distributed ledger technology (DLT). DLT is a solution in which, instead of having a centralized ledger, a single ledger is distributed between multiple data nodes with decentralized control. These nodes record, share, and synchronize data across the network and keep the data secure by reaching a consensus on the content of the ledger [13]. The first implementation of DLT happened in 2009 with Bitcoin blockchain, but over the years, different DLT platforms were developed [14].

Blockchain represents a DLT that stores transactions in a chain of blocks. Blocks are added in chronological order in a way that makes it highly improbable that they can be tampered with and forged [15]. To achieve this, blockchain relies on cryptographic hash, asymmetric cryptography, and distributed consensus mechanism [16]. The main benefits of BCT are efficiency, security, resilience, and transparency. The fact that it is possible to easily track and manage complex data logs makes blockchain technology (BCT) efficient. The system is made secure by making falsifying data almost impossible by distributing data between many connected nodes. The distribution of data and the fact that there is no single point of failure make the system resilient, and the fact that usage of the system is public by default make it transparent for every interested party [17]. According to [16], blockchain also brings benefits of decentralization, anonymity, persistence, and audibility. Bitcoin blockchain, created in 2009, had a limited domain of application, but with the addition of smart contracts, a domain of possible application expanded, especially to government, science, IoT, and health [18], [19]. Smart contracts are computer programs that are coded and entered into the blockchain and are governed by the same rules that apply for all blockchain transactions.

In this paper, it is discussed how BCT could be used to solve problems of double spending in real estate transactions and data tampering. Also, a smart contract, written in Solidity programing language is presented that could serve the general purpose of storing and providing data about ownership, and for registering transactions on real estate. It can also handle those more specific cases in LAS, such as sharing of ownership, transferring part of ownership, splitting or merging of real estate, and limiting the possibility of trading real estate. The presented smart contract is based on a programming interface, that could be used as a blueprint for the creation of LAS specific smart contracts. This interface is also presented in the paper.

Apart from the Introduction and Conclusion, this paper is organized as follows. In section II, a short literature review on the subject of the Land Administration Domain Model and the possible application of BCT in e-government and BCT in LAS is given. The process of selection of blockchain platform for application in LAS is presented in section III. In section IV, a concrete solution for solving some of LAS problems through the application of BCT smart contract is given. In the same section, system infrastructure implementation is presented together with how the implementation of the proposed solution would change the process of registering transactions in LAS on the example of Serbia. Discussion of the proposed solution, together with some limitations is presented in section V.

II. LITERATURE REVIEW

International Organization for Standardization defines ISO/CD 19152:2021 Geographic information – Land Administration Domain Model (LADM) as a reference domain model for basic information-related components of land administration and provides basics for country profiles. According to [20], this standard has been applied for the creation of various country profiles all over the world [21]–[24] and has grown over the years in recognition and influence.

ISO/CD 19152:2021 Geographic information – Land Administration Domain Model (LADM) defines four core classes of LADM that should represent a foundation of LAS and they are presented in Fig. 1.

LADM core classes are defined so that their instance would represent:

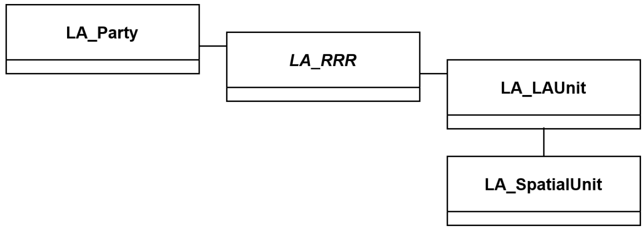

FIGURE 1. Core classes of LADM [25].

1. instances of LA_Party represent a party that could be a person, organization or group of persons or/and organizations),
2. instances of subclasses of LA_RRR represent rights, restrictions, and responsibilities,
3. instances of LA_LAUnit represent administrative information regarding spatial units
4. instances of LA_SpatialUnit is a class whose instances represent spatial information about real estate [25].

LADM is a good starting point for the development of any LAS solution, regardless of the manner of its implementation. For this reason, the solution proposed in this paper will also be based on LADM.

Since the focus of this research was on the application of DLT in LAS, or to be more precise on the application of BCT in LAS, in the remaining part of this section, a short literature review on the application of BCT in e-government and BCT in LAS is given. At the end of the sections, the main differences between this and other researches are stated.

For the first several years BCT has mainly been used for the creation of cryptocurrencies and therefore research done and papers published in those first years were mostly in the field of financial technology. A literature review conducted in 2017 detected only 67 papers that were on a subject on blockchain other than cryptocurrency [26]. Out of those 67 papers, 28 were in the discipline of computer science, 18 in information systems, 9 in law, 6 in finances, 5 in political sciences, and 3 were not categorized. A literature review published in 2019, for the period 2014-2018, has shown an increase in the number of papers published about BCT, other than cryptocurrency, totaling 260 papers. Those 260 papers were classified into fields of business and industry (58 papers), governance (32 papers), Internet of Things (32 papers), data management (26 papers), health (25 papers), privacy and security (24 papers), integrity verification (16 paper), finance (15 papers), education (8 papers), and with remaining 24 papers falling into the miscellaneous category. According to another research conducted in 2017, the application of BCT in e-government was at its beginnings at that time, since only 21 research papers were published on that subject [27].

The importance of BCT for e-government is expressed in [28] where together with artificial intelligence and big data, blockchain is identified as one of the disruptive information-communication technologies (ICTs) that together with established ICTs will be a foundation of government 3.0. Reasons stated for the application of BCT are the need for decentralization, trustworthiness, and security. Enabling smarter government through BCT, due to openness, security, and distributed nature is proposed in [29]. In [30] BCT is proposed for providing a highly secure and privacy preserving decentralized system for e-government that would resolve problems that bear upon centralized systems such as single point of failure, cyber-attacks, denial of service, and distributed denial of service. According to [31] e-government represents a field in which BCT could bring significant improvements due to transparency, transaction security, and trust. The application of blockchain is proposed for e-government services in China [32] due to the benefits that this technology provides, mainly improvements of provided services, transparency, accessibility, and information sharing. To address the problem of centralization and reliability on human control, the application of BCT in e-government is proposed in [33].

Overview of listed advantages of application of BCT from listed related work is presented in Table 1.

TABLE 1. Overview of advantages of application of BCT.

Paper	Advantage
[28], [29], [30], [33]	Decentralization
[28], [31]	Trust
[28], [29], [31]	Security
[29], [31], [32]	Transparency
[32]	Accessibility

More specifically, the application of BCT in LAS is proposed in [34] together with the field of vehicle registration. In [35], BCT is recognized as a solution for long lasting problems that exist in LAS. Blockchain is identified as an innovative technology that will provide significant possibilities for application in a wide field of application within e-governments, such as identity management, document exchange, academic certificates, and land administration [36]. In [37] blockchain is identified as a technology that could make significant changes in the way land ownership is recorded and how transfers of ownership are managed. In [38], [39], decentralization, immutability, transparency, and smart contracts are identified as benefits that BCT could bring to LAS. Using blockchain architecture in a variety of domains, such as land administration, but also supply chain management, and intellectual rights management is proposed in [40] as a way to remove intermediators and third parties of trust. Even those LASs that are based on contemporary ICTs and do not suffer from a lack of trust, can benefit from the use of BCT. The usage of BCT in LASs is recognized as beneficial to shorten the duration of administrative operations [41]. Applying blockchain for archiving deeds and titles, for transaction process, or for validating and issuing titles are recognized as a possibility in [42]. Solving problems related to documenting proof of ownership of real estate through the use of blockchain is proposed in [43]. In [44] "trustless" transactions, durability, transparency, and immutability are identified as three major benefits of implementing BCT in LAS.

Apart from before mentioned, more detailed possibilities of application of BCT in LAS are discussed in [30], where Ethereum and Delegated Proof-of-Stake consensus mechanism, for running smart contracts that would represent simulations of a real contract, such as those used in the land registry, is proposed. In [45] blockchain is proposed as a recordkeeping database that will be connected to an external database containing legal and spatial data. Blockchain recordkeeping would preserve hashes of documents from existing databases and in that way create an audit trail of all changes made in original documents. Using BCT to register transactions is proposed as a solution for problems that exist in land administration in India [46]. All activities from interested parties such as buyer, seller, bank, registrar office, and revenue office, related to the transfer of ownership of real estate are registered in the blockchain. Those activities include not only the transfer of ownership between buyer and seller but also the application for mortgage and registration of transfer of ownership in the revenue office. Idea is to store data regarding ownership in the blockchain together with hash values of that data. Similarly, in [47] using BCT is proposed in the process of registering transactions to increase transparency and reduce corruption that happens in up to 20% of cases of land administration services, according to the Transparency International report. It is proposed that transfers will be registered on the blockchain through the use of smart contracts and that all necessary data about real estate will be stored together with transactions, including all related documents. Managing the transaction of land ownership in Ghana through blockchain is proposed in [48]. More specifically, acquiring confirmation from all necessary parties in the process of acquisition of "skin land" (tribal land) before transfer itself is registered in the blockchain.

Preserving anonymity of participating parties through the use of blockchain and smart contracts is proposed in [49] as a way to exchange private information about real estate location and to ensure delivery of that information upon meeting all required conditions. One of the ideas is also to use so-called colored tokens that can be created on top of the Bitcoin blockchain to represent the real estate and to benefit from all advantages offered by blockchain is proposed in [50]. In [51] the use of BCT in land administration is proposed for the following tasks: validation of land title information, validation of land title transactions, notarizing land registration process, and sharing fingerprints of information stored in LAS. Regarding storing spatial data and BCT, one proposition was made in [52] where it is suggested to divide the surface of Earth to 3m-by-3m squares with unique IDs that will represent a unit of transactions. In [53] it is concluded that it is possible to store land administration data in parallel blockchains. This is interesting because BCT is not considered suitable for storing a high volume of data, such as spatial data.

Somewhat specific application of smart contracts in land administration is proposed for managing transactions in [54]. In this case, smart contracts are used to validate participants in the process and data shared between them, while confirmation about the validity of shared data is done by additional participants, the land inspector. In Turkey, blockchain is proposed as a solution for managing real estate transactions and satisfying the needs of a large number of stakeholders [55]. It is not only that transfer of real estate ownership is conducted through blockchain, but smart contracts are also used to transfer money from buyer/bank to seller. Using BCT for preserving the validity of data through resolving issues of problematic boundaries by requiring interested parties to consent to change is proposed for solving this common problem in Turkey [56]. In the case of Serbia, managing transactions of real estate on the blockchain is proposed in [57]. Authors propose using permissioned public blockchain for managing transactions and storing all relevant documents on the blockchain network, and transactions that are happening on the blockchain network can also be used as an initial source for other tasks that should be performed after transactions are reregistered, such as taxation. During the process of digitization, transferring information into blockchain is proposed for Bangladesh [58]. It is proposed that new and old transactions should be initially registered in a public blockchain, by government officials, only to move to a full hybrid blockchain in later stages of the development.

Overview of specific benefits to LASs that could come from the application of BCT as listed in related work is presented in Table 2.

TABLE 2. Overview of specific benefits to LASs that could come from the application of BCT.

Paper	Field of application
[45], [46], [51]	Storing hash value of LAS documents
[47], [53], [57]	Storing full copies of LAS documents
[46], [47], [48], [51], [54], [55], [57], [58]	Registering transactions
[49], [51], [56]	Validating information
[50]	Using tokens for representation of the real estate
[52]	Storing spatial data
[55]	Managing payments

Apart from propositions previously mentioned in peer review papers, there are several ongoing pilot projects related to the application of Blockchain in LASs, most notably partnership between Lantmäteriet (Swedish land administration agency) and several private companies. The goals of these projects were to eliminate the need of archiving large numbers of physical documents, to improve resilience, and remove the redundancy of data stored in LAS, to increase the security of the system, to decrease the time needed for registering transactions, and to make them transparent, eliminate the possibility of double spending of property, and stealing of property. Due to a large number of actors in the most common process of transfer of real estate (seller, buyer, real estate agent, Lantmäteriet, bank, mortgage deed registry), there are quite a few steps that needed to be taken in this process (34 to be precise) and this process takes a lot of time (up to 6 months), but by applying BCT, the time taken to finish the whole process is reduced by 4 months. In this solution, they are using BCT to verify the validity of documents, sign documents, manage transactions, and store hash values of signed documents [12].

One of the companies involved with the pilot project in Sweden, Chromaway, was also involved in a similar project, but this time in India [59]. A pilot project between Ubitquity and the Brazilian State of Rio Grande do Sul started in 2017. In this project, color coins were used to add another layer to Bitcoin Blockchain so they would represent a property. In this way transferring ownership of the color coin Bitcoin would represent a transfer of ownership of the real estate. For storing documents related to transfers, BitTorrent or Inter Planetary File System are proposed [37]. Another pilot project was proposed in Honduras. The partnership between the local government and Factom was supposed to solve problems that plagued land administration in Honduras. Idea was to prevent any kind of tampering of records at any steps related to submission of data, acquisition, and recording process [60].

In the majority of presented papers, only general ''compatibility'' between problems that exist in contemporary LAS and possibilities offered by BCT are presented. Overview of these papers is given in Table 1. More specific possible applications are presented in Table 2, but still, in presented papers, those proposals were on a theoretical level and more importantly did not cover all possible cases that exist in the process of transferring of real estate, such as previously mentioned sharing of ownership, transferring part of ownership, splitting or merging of real estate, and limiting the possibility of trading a real estate.

III. SELECTING BLOCKCHAIN PLATFORM FOR APPLICATION IN LAS

Although several papers about the application of BCT in las are presented in section ii, the applicability of this technology in LAS could be tested by following the ten steps decision path proposed in [61], to determine if blockchain is a good fit for LAS and what kind of blockchain should be used. The decision path is shown in Fig. 2.

For LAS, ''Yes'' is the answer to all ten questions, and the decision path justifies the use of BCT, and proposes permissionless public blockchain for the implementation of the solution. Permissionless public blockchain refers to one more classification of blockchains: permissionless public, consortium/hybrid, and private blockchains [62]. in a permissionless public blockchain, anyone can join the network and participate in the registration of transactions. in consortium/hybrid

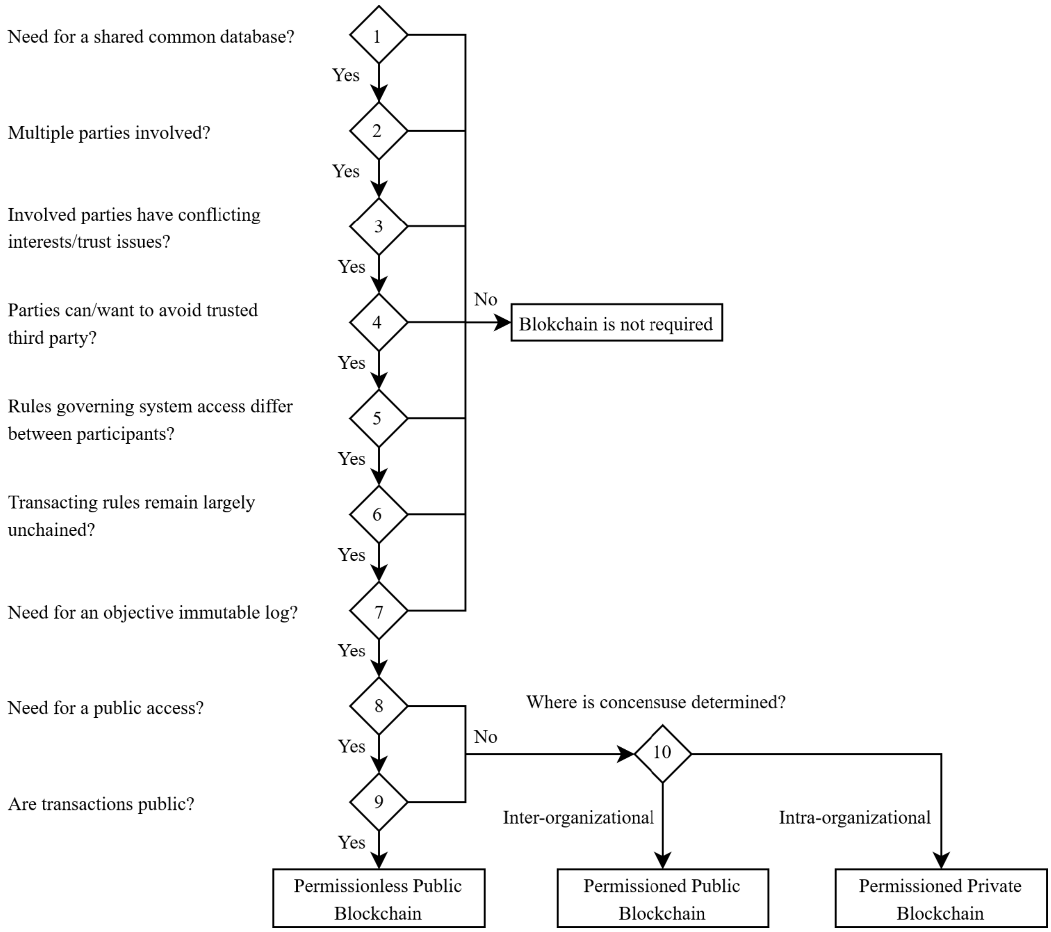

FIGURE 2. Overview of the blockchain decision path [61].

blockchains, transactions are usually confirmed by participants that are pre-registered, whereas everyone can see what is happening in the blockchain. in a private blockchain, only pre-registered parties can take part in confirming transactions and see what is happening in the blockchain.

The solution proposed in this paper will rely on the application of smart contracts. The Idea of smart contracts was introduced in 1996 and they were defined as a digital set of promises and protocols that parties perform based on those promises. One of the possible applications of smart contracts identified at that point in time is property rights [63]. In the realm of blockchain, smart contracts represent a system that could be used to automatically transfer assets based on some predefined rules [64]. Smart contracts verify and execute terms stated in them, in a case when predetermined events happen [65]. In LAS, smart contracts could be used to hold information about ownership and to manage transactions of real estate.

For using a permissionless public blockchain, there are two possibilities. The first one is the creation of a new permissionless blockchain network that could be based on existing solutions, and the other one is selecting one of the existing public blockchains. In the case of creating a new permissionless public blockchain network, the question of the sustainability of such a system could arise. Although there is a large interest from stakeholders to participate in such a blockchain, it is questionable how the incentive system would be successfully implemented. There is also a problem of the low hashing power of such a system and a possibility that dishonest nodes could take part in such a system and "easily" acquire enough hash power to perform a 51% attack. With this in mind using existing permissionless public blockchain might be a better idea.

In [66] following blockchains were identified as representative platforms for running smart contracts:

1. Ethereum – Turing complete, general application, permissionless;
2. Fabric – Turing complete, general application, private;
3. Corda – Turing incomplete, application in digital currency, private;
4. Stellar – Turing incomplete, application in digital currency, consortium/hybrid;
5. Rootstock – Turing complete, application in digital currency, permissionless; and
6. EOS – Turing complete, general application, permissionless.

Out of those 6 representative platforms, only 3 are permissionless and at the same time Turing complete, and two of those, Ethereum and EOS have general applications.

A decision support system for the blockchain platform selection problem is presented in [67]. the proposed decision support system is based on 71 Boolean and 4 non-Boolean blockchain features that were detected during initial research. Following the proposed decision support system with public, permissionless, smart-contract, and turing complete features set as must-have, 5 blockchain platforms were proposed. The main differences between proposed blockchain platforms were the consensus mechanisms. Proposed blockchain platforms were:

1. Ethereum – General application, proof-of-work;
2. NEO – general application, delegated byzantine fault tolerance;
3. QTUM – general application, proof-of-stake;
4. Cosmos Network – interoperability platform, multiple consensus mechanisms;
5. Wanchain – interoperability platform, proof-of-stake.

Having in mind that the Ethereum blockchain is the only platform that is proposed both in [66] and [67] and that Ethereum:

- is the first blockchain that implemented smart contracts,
- was created with the idea of creating distributed applications based on smart contracts,
- supports solidity, a high-level object-oriented programming language for writing smart contracts, and
- is the second largest blockchain network with a hash rate of around 800.000 GH/s mid-September 2021 [68],

the decision is made to implement proposed solution on the Ethereum blockchain network.

On the ethereum blockchain network, Solidity is a native programming language for creating smart contracts that will, when compiled, run on Ethereum Virtual Machine (EVM). Solidity programming language is loosely based on ECMAScript (European Computer Manufacturers Association Script), but is statically-typed, and supports inheritance, libraries, and complex user-defined types, called structs. Since solidity was introduced in 2015, language has been under active development and as of the beginning of 2022, eight major versions were released. Solidity has concepts that are available in most contemporary programming languages and consequently does not have a steep learning curve for those familiar with these languages. Since 2015, solidity has gained wide community support.

IV. BCT BASED PROPOSAL FOR SOLVING LAS PROBLEMS

In the introduction of this paper, three main problems were identified in LASs that could be potentially solved by the implementation of BCT:

1. double spending in real estate transactions,
2. data tampering, and
3. long duration of the real estate transactions' registration process, which leads to LASs incorrectness during that period.

Regarding the double spending problem, blockchain is developed with the direct intention to eliminate any possibility of double spending in electronic transitions. Blockchain solves this problem by using a peer-to-peer distributed timestamp server that is based on a consensus mechanism and in that way creates a chronological list of all transactions that happen in the system. The most widely used consensus mechanism is proof-of-work.

The consensus mechanism serves multiple roles, firstly it eliminates the possibility of double spending by having nodes check all transactions that are being added to the blockchain, and secondly, it is there to set a standard elapsed time for adding a new block to the chain.

Another problem in current LASs is the possibility of data tampering. On a blockchain, it is practically impossible to "go back in time" and make a change in the block of transactions that were previously added to the blockchain. Changing even a single transaction will result in a new hash value for that block, and dishonest nodes would need not only to, for example in Bitcoin and Ethereum networks, do proof-of-work for that specific block, but also for all other blocks that were added to the blockchain after the fraudulent one. That would result in the creation of a new chain, a so-called fork in a blockchain. During that time, honest nodes would continue to work taking the hash value of the last correct block as their input value and adding new blocks. The result of this process is that the correct chain will be longer than the fraudulent one, and by implementation, the longest chain is always considered to be correct. This process should eliminate any possibility of tampering with data that was already stored in the blockchain.

The proposed solution for the third problem, the time needed for registering transactions, will be in a form of a smart contract. In the following section more specific cases of ownership, real estate transactions, and merging and splitting real estate will be presented. Programing code that could support each of those cases will be presented and discussed. The section will be concluded by a discussion on how the proposed solution would influence the time needed for registering transactions in LAS.

A. PROPOSAL OF SMART CONTRACT FOR LAS

On the Ethereum network, it is possible to create a smart contract that will be in charge of managing some new tokens. These tokens can be used to represent any real-life goods, including real estate. Tokens can be transferred from one owner to another, separately from other transactions, but still running on the same public Ethereum network.

When the application of tokens is proposed as solutions for representing specific real estate and smart contracts for managing transactions of those tokens, usually only "happy path" cases are covered, as mentioned in [50]. Those cases

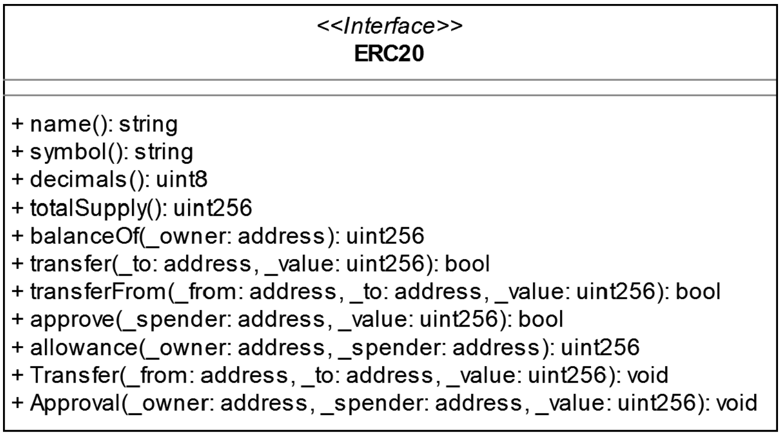

FIGURE 3. UML class diagram representing ÅRC-20 interface.

represent situations where one party (person or organization) with 100% ownership of real estate is selling the entire 100% of ownership to another party. Although this is indeed the most common type of transfer, other cases also exist and it is necessary to propose a solution that will be able to cover these cases too.

Proposed solutions will enable transfers of ownership in the following use cases:

A. Multiple entities could have a share in the ownership of the same real estate.

It is quite common for multiple entities to participate in ownership on real estate, married or civil partners can own a real estate together with equal or not equal shares. In cases of inheritance, there are often more than one inheritor and all inheritors will own a share in the ownership of real estate that they have inherited.

B. Owners can transfer less than 100% of ownership.

It is possible that the owner could transfer less than 100% of the ownership to another party, this could again be the case between married or civil partners, relatives, or any other entities.

C. Transferring ownership of real estate could be forbidden.

The transfer of ownership of real estate can be temporarily or permanently forbidden according to state legislation. For example, in Serbia, the Law on the seizure of property that is a result of criminal activity and the Law on restriction on the disposal of property in order to prevent terrorism and the spread of weapons of mass destruction restrict the transfer of the ownership for some owners.

D. Ownership could be transferred by a party other than the owner.

In cases of inheritance, transfer of ownership is not conducted by the owner anymore, it is usually conducted based on rulings made by either a court or a notary. Also, there could be a case in which a real estate transaction could be declared void by the court of law, so there must be a possibility for a party other than the owner, to perform a transaction that will practically revert the previous transaction that was declared void.

E. Real estate could be split into new real estate.

When real estate represents some land, it is common to only sell a part of the land. In that case, usually, the owner is not selling part of his ownership, since that would create the previously mentioned situation of owning an ideal part of real estate. Instead, the land is topologically divided into two or more new real estates and then ownership of those real estates is transferred.

F. Real estate could be merged into new real estate.

If one party owns two real estates that are topologically placed one to the other, the owner can request for those real estates to be merged and to create a new real estate.

For creating new tokens on the Ethereum network some standards are adopted. Most notably, ERC-20 (ERC – Ethereum Request for Comments) Token standard and ERC-721 Non-Fungible Token (NFT) standard. These two standards are defined in the shape of interfaces written in Solidity programing language. Both ERC-20 and ERC-721 are defining a standard set of APIs that will make it possible for tokens to be used by other applications, such as wallets or decentralized exchanges. By implementing either of these standards in a smart contract, basic functionality for transferring tokens by either owner or another approved party will exist [69], [70]. The main difference between ERC-20 and ERC-721 is that smart contract implementing ERC-20 is intended to be used for the creation and manipulation of fungible tokens, while in ERC-721, those tokens are non-fungible. Fungible tokens could, for example, be used for the creation of new cryptocurrencies or to represent a fiat currency, while NFTs are intended to be used to represent unique items, such as physical property, virtual collectibles, or "negative value" assets as loans or burdens. Another important difference is that ERC-20 tokens can be split into smaller parts, unlike ERC-721 tokens. Class diagrams representing ERC-20 and ERC-721 are presented in Fig. 3 and Fig. 4 respectively.

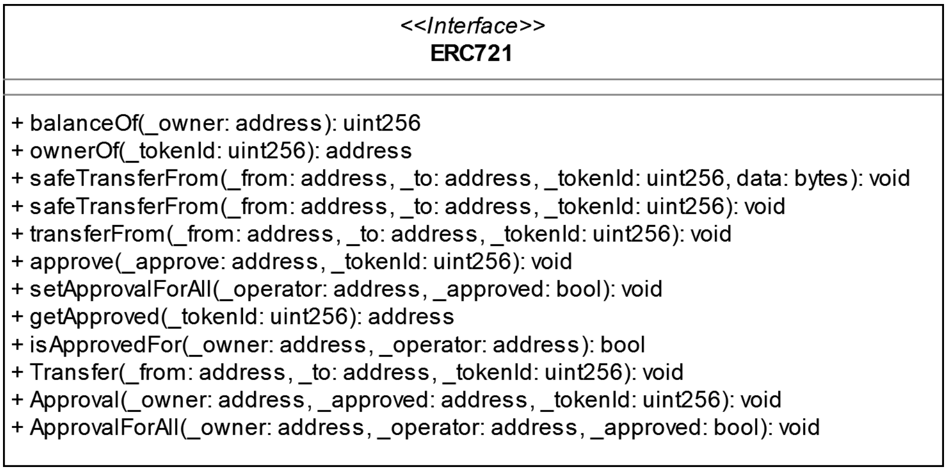

FIGURE 4. **UML class diagram representing ÅRC-721.**

Application of fungible tokens would not be a good solution for LAS because there is no way to distinguish one token from another and that is what is needed if tokens are to be used to represent the unique real estate. A possible solution for this problem could come from the application of the ERC-721 NFT standard, but according to this standard, NFT cannot be divided and they can be transferred from one account to another only as a whole, so multiple parties could not own a share in the same real estate. EVM running the Ethereum blockchain has no support for decimal values, but transferring less than one token is still possible in ERC-20. This is achieved by using larger integer values to represent one token, for example, 1000 could represent a single fungible token. So, for real estate, a combination of these two standards is needed. NFT from ERC-721 and the possibility of splitting a token from ERC-20. It is safe to make these kinds of adjustments because, in proposed solutions, it is not expected that tokens representing real estate will be exchanged for any other kind of tokens on some token exchange system as it is expected for ERC-20 and ERC-721 tokens.

The proposed solution is presented in the shape of the UML class diagram (Fig. 5) and smart contract code written in Solidity programing language. Interface *LandAdministrationSystemInterface*, presented in Fig. 5 is based on the ERC-721 standard and the corresponding Solidity code is presented in Listing 1. An example of implementation of class *LandAdministrationSystem*, from the same figure, is presented separately in Listings 2 through 4 and Listings 8 through 10, to make it easier to discuss the code. Three dots in those listings indicate code from the *LandAdministrationSystem* contract that is not relevant for the problem that is presented and discussed in that specific listing.

The proposed interface presents a set of functions that must be implemented by a smart contract that is implementing this interface. Apart from functions, the interface declares two events that are abstractions of the Ethereum logging protocol. In this and all other examples in this paper, the following parameter naming convention will be used:

- *_tokenId* – a parameter representing real estate token,
- *_from* or *_owner* – a parameter representing current real estate shareowner,
- *_to* – a parameter representing new real estate shareowner,
- *_share* – a parameter representing a share of real estate ownership,
- *_documentHash* – a parameter representing the hash value of documents that real estate transaction is based on and that will be further discussed in section V,
- *_transferable* – a parameter representing Boolean value if real estate token is transferable or not.

In Listing 1. The following functions are declared:

- *ownersOf* – Original function *ownerOf* function from ERC-721 is replaced with a new declaration of function *ownersOf*. This declaration requires that instead of returning just one, all owners of a real estate token are returned as a result.
- *shareOf* – In ERC-721 there is no corresponding function for *shareOf*. Declaration of function *shareOf* provides a possibility to get information about the share of ownership a party has over real estate.
- *transferFrom* – in *transferFrom* function declaration, apart from *_from*, *_to* and, *_tokenId* parameter, that could be found in the ERC-721 declaration of function with the same name, there are two more additional parameters, *_share* and *_documentHash*.
- *isTransferable* and *setTransferable* – Functions *isTransferable* and *setTransferable* represent functions that will make it possible to flag specific real estate as being allowed to trade on or not based on some specific reasons.

Finally, two events are declared, *Transfer* and *Transferable* that will be used to publish information about transfers and transferability of real estate to all interested stakeholders.

Land Administration Domain Model (LADM) that is presented in section II could be modeled in the following way:

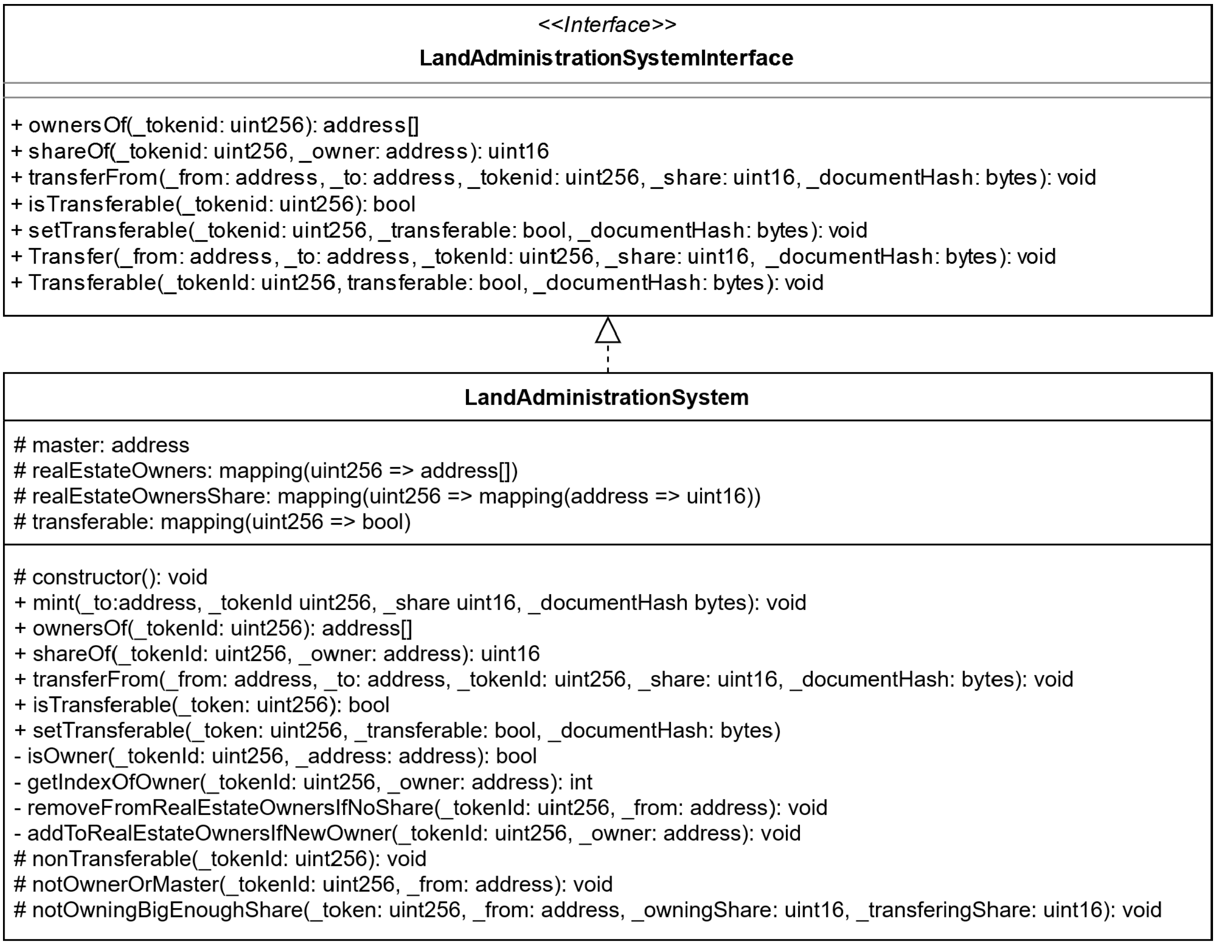

FIGURE 5. UML class diagram representing LandAdministrationSystemInterface interface and LandAdministrationSystem class.

```
Pragma solidity ^0.8.4;
interface LandAdministrationSystemInterface {
   function ownersOf(uint256 _tokenId) external view returns (address[] memory);
   function shareOf(uint256 _tokenId, address _owner) external view returns (uint16);
   function transferFrom(address _from, address _to, uint256 _tokenId, uint16 _share,
                         bytes memory _documentHash) external payable;
   function isTransferable(uint256 _tokenId) external view returns (bool);
   function setTransferable(uint256 _tokenId, bool _transferable,
                            bytes memory _documentHash) external payable;
   event Transfer(address indexed _from, address indexed _to,
                  uint256 indexed _tokenId, uint16 _share, bytes _documentHash);
   event Transferable(uint256 indexed _tokenId, bytes _documentHash);
}
```

LISTING 1. ***LandAdministrationSystemInterface.***

- Instances of LA_Party could be modeled by *address*, the 20-byte elementary Solidity types.
- Instances of LA_RRR rights (ownership share) and restrictions could be modeled by *uint16* and *bool* elementary types respectively.
- Instances of subclasses of LA_LAUnit could be represented by an NFT, that is with *uint256* elementary type.
- Instances of LA_SpatialUnit are not required for ownership transfer and will not be represented in the smart contract, but it will be stored within

```
Contract LandAdministrationSystem is LandAdministrationSystemInterface {
  address master;
  mapping(uint256 => address[]) realEstateOwners;
  mapping(uint256 => mapping(address => uint16)) realEstateOwnersShare;
  mapping(uint256 => bool) transferable;

  constructor() {
         master = msg.sender;
  }
  …
}
```

LISTING 2. Modeling LADM core classes in smart contract.

```
Contract LandAdministrationSystem is LandAdministrationSystemInterface {
  …
  function ownersOf(uint256 _tokenId) override external view
                  returns (address[] memory) {
         return realEstateOwners[_tokenId];
  }
  …
}
```

LISTING 3. An example of the implementation of *ownersOf* function.

a system whose architecture will be presented in section IV-B.

An example of smart contract modeling these core classes is given in Listing 2.

In Listing 2. The following properties are declared:

- *master* property of type *address* represents an address that is used to deploy smart contract and this property is being set through function *constructor* at the time of deployment of smart contract,
- mapping *realEstateOwners* maps relation between real estate and real estate owners by mapping *uint256* value, representing real estate token, into an array of *addresses*, representing real estate owners,
- mapping *realEstateOwnersShare* maps relation between *uint256* value, representing real estate token, into a mapping of *address*, representing owner, into *uint16* value, representing a share of ownership, the value of "full" share should be big enough integer to allow for all smaller possible parts of share to be denoted as whole numbers,
- mapping *transferable* maps *uint256* value, representing real estate token, into *bool* type that will have the value *true* or *false* based on transferability of specific real estate.

Apart from properties in Listing 2, the constructor function is declared and implemented. In smart contracts, constructor functions are called only once, at the time of deployment/creation of the smart contract, and in this specific case, implementation of the constructor is such that the transaction that is creating a smart contract is queried for the sender address of transaction by calling *msg.sender* and that value will be stored in address *master* property.

An example of the implementation of function *ownersOf* is shown in Listing 3.

In Listing 3, function *ownersOf* is implemented as a get function where function parameter *_tokenId* is used to query *realEstateOwners* property and get an array list of *address* type values representing all owners of a property.

By declaring properties is shown in Listing 2 and by implementing *ownersOf* function shown in Listing 3, a solution for use case A, where multiple entities could share ownership of the real estate, is achieved.

An example of the implementation of *shareOf* and *transferFrom* functions is given in Listing 4.

Implementation of *shareOf* function is a get call on property *realEstateOwnersShare* where firstly *_tokenId* and then *_owner* parameters from function call are used as key values to retrieve data from *realEstateOwnersShare* property representing a share of ownership.

In function *transferFrom*, three conditions are checked. First, it is checked if the transfer of real estate is allowed by checking the value of the transferable property based on

```
Contract LandAdministrationSystem is LandAdministrationSystemInterface {
  …
  function shareOf(uint256 _tokenId, address _owner) override public view
                 returns (uint16) {
        return realEstateOwnersShare[_tokenId][_owner];
  }

  function transferFrom(address _from, address _to, uint256 _tokenId, uint16 _share,
                   bytes memory _documentHash) override external payable {
        if(!transferable[_tokenId]) {
              revert nonTransferable({
                    _tokenId: _tokenId
              });
        }
        if (!(msg.sender == master || isOwner(_tokenId, _from))) {
              revert notOwnerOrMaster({
                    _tokenId: _tokenId,
                    _from: _from
              });
        }
        if (shareOf(_tokenId, _from) < _share) {
              revert notOwningBigEnoughShare({
                    _tokenId: _tokenId,
                    _from: _from,
                    _owningShare: shareOf(_tokenId, _from),
                    _transferingShare: _share
              });
        }
        realEstateOwnersShare[_tokenId][_from] -= _share;
        realEstateOwnersShare[_tokenId][_to] += _share;
        addToRealEstateOwnersIfNewOwner(_tokenId, _to);
        removeFromRealEstateOwnersIfNoShare(_tokenId, _from);
        emit Transfer(_from, _to, _tokenId, _share, _documentHash);
  }

  function isOwner(uint256 _tokenId, address _address) private view returns (bool) {
        address[] memory allOwners = realEstateOwners[_tokenId];
        for (uint i=0; I < allOwners.length; i++) {
              if (allOwners[i] == _address ) {
                    return true;
              }
        }
        return false;
  }

  function getIndexOfOwner(uint256 _tokenId, address _owner) private view
                     returns (int){
        for(uint I = 0; i< realEstateOwners[_tokenId].length; i++){
              if(_owner == realEstateOwners[_tokenId][i])
                    return int(i);
              }
        return -1;
  }

  function removeFromRealEstateOwnersIfNoShare(uint256 _tokenId, address _from)
                                      private {
        if (shareOf(_tokenId, _from) == 0) {
              int i = getIndexOfOwner(_tokenId, _from);
              if (i != -1) {
                    delete realEstateOwners[_tokenId][uint(i)];
              }
        }
  }

  function addToRealEstateOwnersIfNewOwner(uint256 _tokenId, address _owner)
                                 private {
        if (!isOwner(_tokenId, _owner)) {
              realEstateOwners[_tokenId].push(_owner);
        }
  }
  …
}
```

LISTING 4. An example of the implementation of ***shareOf*** and ***transferFrom*** functions.

```
Transact to LandAdministrationSystem.transferFrom errored: VM error: revert.
       revert
The transaction has been reverted to the initial state.
       Error provided by the contract:
nonTransferable
Parameters:
{
""_tokenId"": {
       ""value"":""12""
       }
}
```

LISTING 5. **An example of error data if *_tokenId* is not transferable.**

```
transact to LandAdministrationSystem.transferFrom errored: VM error: revert.
       revert
The transaction has been reverted to the initial state.
       Error provided by the contract:
notOwnerOrMaster
Parameters:
{
""_tokenId"": {
       ""value"":""12""
       },
""_from"": {
       ""value"":""0x5B38Da6a701c568545dCfcB03FcB875f56beddC""
       }
}
```

LISTING 6. **An example of error data if address calling a function is not master address or *_tokenId* owner.**

_tokenId. If it is false, the revert statement is called with a *nonTransferable* error. As the result, error data that are shown in Listing 5 is passed back to the function caller. Then it is checked if the address used to call *transferFrom* function is the master or token owner's address, based on the master property, and by calling isOwner function with *_from* parameter. If both are false, then the revert statement is called with a *notOwnerOrMaster* error. As the result, error data that are shown in Listing 6 is passed back to the function caller. Finally, it is checked if *_share* parameter is larger than real estate owners share in which case the revert statements is called with a *notOwningBigEnoughShare* error. As the result, error data that are shown in Listing 7 is passed back to the function caller. Declarations of errors are shown in Listing 8.

If real estate is transferable and if the address used to call function is the master address or real estate owner address and if the share is equal or lower of real estate owners share, transfer of share or real estate ownership is registered by:

- decreasing ownership by *_share* for *_from* on *_tokenId* in *realEstateOwnersShare* property,
- increasing ownership by *_share* for *_to* on *_tokenId* also in *realEstateOwnersShare* property,
- calling *addToRealEstateOwnersIfNewOwner* function to add *_to* for *_tokenId* into *realEstateOwner* property if it was not previously owner of share of *_tokenId*,
- calling *removeFromRealEstateOwnersIfNoShare* function that removes *_from* for *_tokenId* from *realEstateOwners* property if it does not own a share of *_tokenId* anymore, and
- finally, calling *Transfer* event and emitting data from the function call.

By declaring properties as shown in Listing 2 and by implementing *shareOf* and *transferFrom* functions shown in Listing 4, a solution for use case B, where real estate owners can transfer less than 100% of ownership, is achieved. By emitting *Transfer* event within *transferFrom* function, all interested stakeholders could be informed about the change in ownership of the real estate. Transfer event emits the same data that was used in *transferFrom* function call. Also, it is important to mention that all transactions stored on

```
transact to LandAdministrationSystem.transferFrom errored: VM error: revert.
       revert
The transaction has been reverted to the initial state.
       Error provided by the contract:
notOwningBigEnoughShare
Parameters:
{
""_tokenId"": {
       ""valu"":""12""
       },
""_from"": {
       ""value"":""0x5B38Da6a701c568545dCfcB03FcB875f56beddC""
       },
""_owningShare"": {
       ""value"":""""
       },
""_transferingShare"": {
       ""value"":""10""
```

LISTING 7. **An example of error data if *_share* is larger than owner's share of *_from* on *_tokenId*.**

```
contract LandAdministrationSystem is LandAdministrationSystemInterface {
…
  error nonTransferable(uint256 _tokenId);
  error notOwnerOrMaster(uint256 _tokenId, address _from);
  error notOwningBigEnoughShare(uint256 _tokenId, address _from, uint16 _owningShare,
                                uint16 _transferingShare);

}
```

LISTING 8. **An example of error declarations.**

blockchain are stored permanently, thus making it possible to have a complete history of all transactions regarding any specific token representing the real estate.

An example of the implementation of functions *isTransferable* and *setTransferable* is shown in Listing 9.

In Listing 9. Implementation of *isTransferable* function is a get call on transferable property with *_tokenId* as key value, whereas *setTransferable* is a set call on the same property with *_tokenId* and *_transferable* bool type value that is being set. For the *setTransferable* function, *_documentHash* parameter is also added to the function call. Calling *setTransferable* function is limited to the master address and emits a *Transferable* event with all forwarded parameters.

By declaring properties as shown in Listing 2 and by implementing *isTransferable* and *setTransferable* functions shown in Listing 9, solution for use case C, transferring of ownership of real estate could be forbidden, is achieved.

Use case D requires that ownership could be transferred by a party other than the owner, and it is achieved by previously declared properties in Listing 2 and by implementing *transferFrom* in Listing 4. Specifically, it is tied to adding master address property in the smart contract. This address could be managed by LAS and used upon instruction from some state institution, such as a court that can make rulings requesting transfer of ownership of the real estate.

Use cases E and F assume that real estate could be split or merged into new real estate. Splitting real estates is modeled by deletion of *_tokenId* representing that real estate and creation of new *_tokenIds* representing new real estates. Merging of real estates is modeled by deletion of *_tokenIds* representing real estates that are being merged and creation of new *_tokenId* representing the newly created real estate. For the destruction of tokens, there is no need for any additional code apart from code already presented in Listing 2 and Listing 4, because the destruction of tokens is achieved by sending *_tokenId* to 'zero address'. For the creation of a token, a new mint function is necessary and an example of implementation of that function is given in Listing 10. This function will also

```
contract LandAdministrationSystem is LandAdministrationSystemInterface {
…      …
    function isTransferable(uint256 _tokenId) override external view returns (bool){
            return transferable[_tokenId];
    }

    function setTransferable(uint256 _tokenId, bool _transferable,
                            bytes memory _documentHash) override external payable {
            require(msg.sender == master);
            transferable[_tokenId] = _transferable;
            emit Transferable(_tokenId, transferable[_tokenId], _documentHash);
    }
…
}
```

LISTING 9. An example of the implementation of *setTransferable* and *isTransferable* functions.

be used for the initial creation of tokens when the system is implemented for the first time.

According to Listing 10, creating a token could only be done if the function is called by master address at which time *_tokenId* is added to transferable property with a *bool* type value of true, *_to* is added to *realEstateOwners* property for key *_tokenId* and *_share* is added to *realEstateOwnersShare* for *_to* on *_tokenId*, and event *Transferable* is broadcasted with all function parameters.

In real life implementation, the mint function would need to check for the total possible value of a share of ownership, also, adding a property that would map address to token would probably be beneficial, as well as deleting owners from *realEstateOwners* since setting 0 at empty index and still occupies that space.

ERC-721 also defines a few more functions, namely, approve, *setApprovalForAll*, *getApproved*, that could be used to delegate the right to transfer ownership to another party, for example like giving a power of attorney, that could be used in LAS smart contract in their native form, if necessary.

By implementing the transaction process in this manner significant savings in time are achieved. During 2020, according to [71], the longest time needed for a block to be added to the Ethereum network was 0.229 minutes and the shortest 0.217 minutes. So, on the Ethereum network, a new block is added to the blockchain, on average, every 13s, and the average number of transactions per second is around 14 [72]. For example, In Serbia, a country with an estimated population of just under 7 million people, there were under 200 000 notarized real estate sale contracts in 2020, so one transaction every 3 minutes. According to [73] in France, a county with an estimated population of just over 67 million people, there were under 1.600.000 notarized real estate transactions in 2020, so one transaction every 20 seconds. Furthermore, this is in the case when all transactions are managed by a single authority, while in reality, LAS are usually organized in smaller administrative units.

To verify performance, the proposed smart contract was deployed on Ropsten Ethereum Testnet. Ropsten Ethereum Testnet is a public blockchain that is running on the same protocol as the Ethereum mainnet and with the same consensus mechanism, making it the most look-a-like network for testing smart contracts running on EVM.

Smart contract is deployed at address 0xDc7919cbd85ea93370c642d937c9E87DecFa7674 (https://ropsten.etherscan.io/address/0xDc7919cbd85ea93370c642d937c9E87DecFa7674) and for testing purposes 1000 simulated real estate transactions were performed. Times between:

- making the request for transfer by the frontend and
- receiving confirmation on the frontend that transaction was mined and added to the blockchain

were measured with the following results:

- average time elapsed – 27.34 seconds,
- median value – 20 seconds,
- standard deviation (σ) – 25.14 seconds,
- percentage of transactions within -1σ and $+1\sigma$ (2.20s – 52.48s) – 86.60%,
- 99% of transactions were completed within 114 seconds.

An overview of the number of transactions per elapsed time in seconds is shown in Fig. 6.

Based on this data it is clear that Ethereum Blockchain has the potential to manage the required number of real estate transactions.

Overview of system architecture for the proposed solution is given in section IV-B.

B. SYSTEM INFRASTRUCTURE IMPLEMENTATION

The proposed solution represents a combination of the blockchain network and traditional information systems. Transactions are registered on the blockchain, and queries are

```
contract LandAdministrationSystem is LandAdministrationSystemInterface {
  …
  function mint(address _to, uint256 _tokenId, uint16 _share,
                bytes memory _documentHash) public {
         require(msg.sender == master);
         transferable[_tokenId] = true;
         realEstateOwners[_tokenId].push(_to);
         realEstateOwnersShare[_tokenId][_to] = _share;
         emit Transfer(master, _to, _tokenId, _share, _documentHash);
  }
  …
}
```

LISTING 10. **Example of implementation of mint functions.**

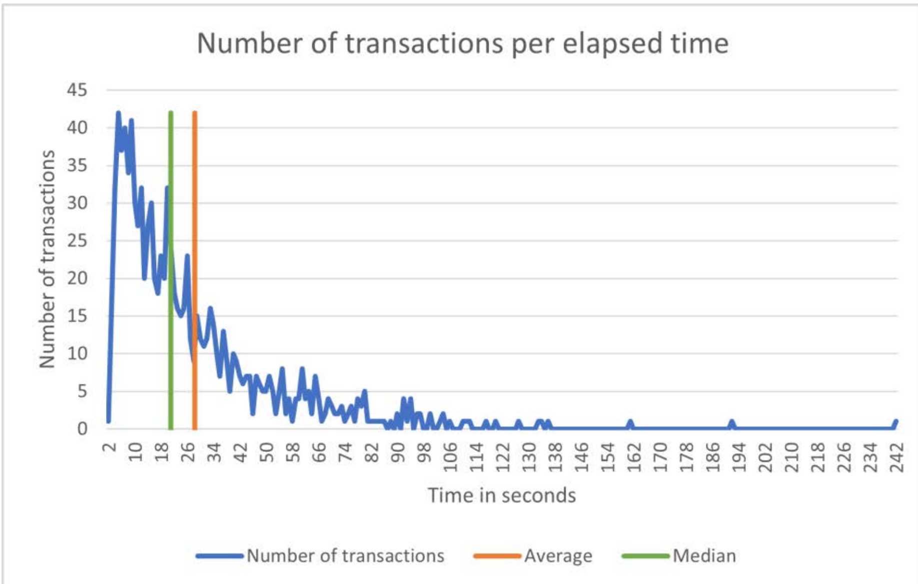

FIGURE 6. **An overview of the number of transactions per elapsed time in seconds.**

made to the blockchain network requesting information about ownership of the real estate, whereas full data, both legal and spatial, are stored in traditional databases. For example, addresses used to represent LA_Party in blockchain could be a key-value or mapped to an ID in a traditional information system holding all necessary data about the owner.

Multilayered architecture could be used and the application frontend is communicating both with blockchain network and traditional database as a source of information. For communication with the Ethereum network, any technology that can interface with Web3 API for Ethereum could be used. Web3 API represents a collection of libraries that makes it possible to interact with the Ethereum network and nodes on that network. This infrastructure is presented in Fig. 7.

As an illustration two use cases will be described, one for registering transactions and one for retrieving data about real estate ownership. For registering transactions, the frontend (upon successfully passing all previously explained steps) will communicate with the blockchain network and perform a real estate transaction. Upon receiving confirmation that a transaction is added to block, additional data, mainly documents will be stored in a traditional database through a backend application.

For retrieving data about real estate ownership, the frontend application will query the blockchain network and get the address of the owner of a specific token, and then query a traditional database about full ownership information based on blockchain address.

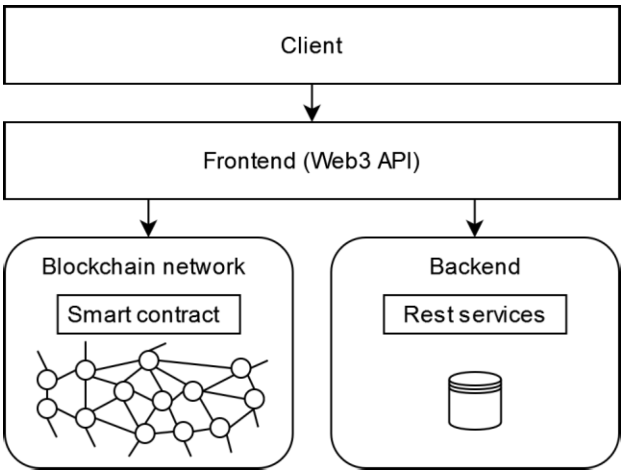

FIGURE 7. Proposed system infrastructure.

How the implementation of BCT based solution for solving the LAS problem would influence the process of registering real life transactions is illustrated on an example of Serbia and presented in section IV-C.

C. CHANGES IN PROCESS OF REGISTERING TRANSACTION IN LAS ON EXAMPLE OF SERBIA

The current process of registering transactions in land administration in Serbia, in its simplest form is as follows. The seller that owns the real estate, decides to sell the real estate. Either by advertising the sale on his own or through a real estate agent, information is made public. Interested buyers contact seller/real estate agent and get basic information about real estate and have a possibility to visit the real estate in question. The process of negotiation is usually what follows. It is important to notice that at this point relations between seller and buyer are based on good faith. Good faith on a part of the buyer is related to the issue that even though it is possible to check online what party is an owner of a property, or to request a document from LAS confirming ownership, the data stored in LAS could still not be up to date, especially when real estate was not subject of a recorded transfer for a long time or when a transfer is registered by the notary, but not yet at LAS. On a part of a seller, unless it is a case of selling a property through a mortgage, there are no requirements to provide bank guarantees that in fact, a buyer had enough funds to pay for the real estate. So, at this point, the buyer believes that in fact, the seller is the owner of the property, that, for example, there is no mortgage on that property and on the other side the seller believes that the buyer has enough funds to pay for the real estate. Once everything is initially agreed upon, involved parties are required to have their contract notarized by a notary.

Notary offices have direct access to LASs and can validate ownership at any point in time. A real estate transfer contract is usually drafted either by a notary and these contracts are in most cases standardized. The real estate transfer contact is required to have the following sections:

- Title – stating that it is a real estate transaction contract,
- Introduction – stating the parties involved in the transaction,
- Articles defining:
  - Real estate that is the subject of transaction – information about LAS where data about real estate is held, address of real estate and size,
  - The price of the real estate,
  - Payment arrangements,
  - Date when the buyer will take possession of the real estate,
  - *Clausula intabulandi* (lat.) – an explicit statement that buyer can register his/her rights on real estate,
  - Transitional and final provisions – which law will be applied for issues not defined by the contract and what court will have jurisdiction in case of any disputes,
- Signatures of parties involved in transactions.

Requested numbers of identical contracts are then signed by seller and buyer, once their identities have been confirmed by the notary. A signed contract is then digitized by the notary and a digital copy is signed by the notary with his digital signature. Within 24 hours a notary is required to submit digitized data online to LAS. As soon as documents are submitted by the notary to LAS, a note is made in the system stating that real estate has been sold. After receiving documents, land administration officials check all documents submitted by the notary and upon verifying them, register the transfer in LAS. As mentioned before, registering transactions in LAS can take months and, in some cases, even more than a year. This process is illustrated in Fig. 8.

Apart from the time needed for finalizing registration in LAS, several problems could be identified here. First of all, good faith between buyer and seller in the initial stage of real estate transfer. There is no simple and cheap way of determining if the seller is the owner of the property and if there are some restrictions related to the transfer of specific property. Buyer, seller, and notary keep hard copies of contract documents, whereas land administration is dealing with both hard copies and electronic documents signed by notaries. Neither seller nor buyer has access to the document that was submitted on their behalf. Allowing 24 hours for registering transactions by a notary still leaves a possibility of double spending. Of course, there is still the possibility of tampering with data once it is stored in LAS as well as antedating documents, for example.

After the implementation of smart contracts in the LAS, the process of transaction of real estate could be executed as follows:

1. Notary drafts the contract, or verifies that the contract that seller and buyer are about to sign is a valid legal document.
2. Notary validates that seller and buyer are present at that time and that they are participating in this transaction in their own free will.

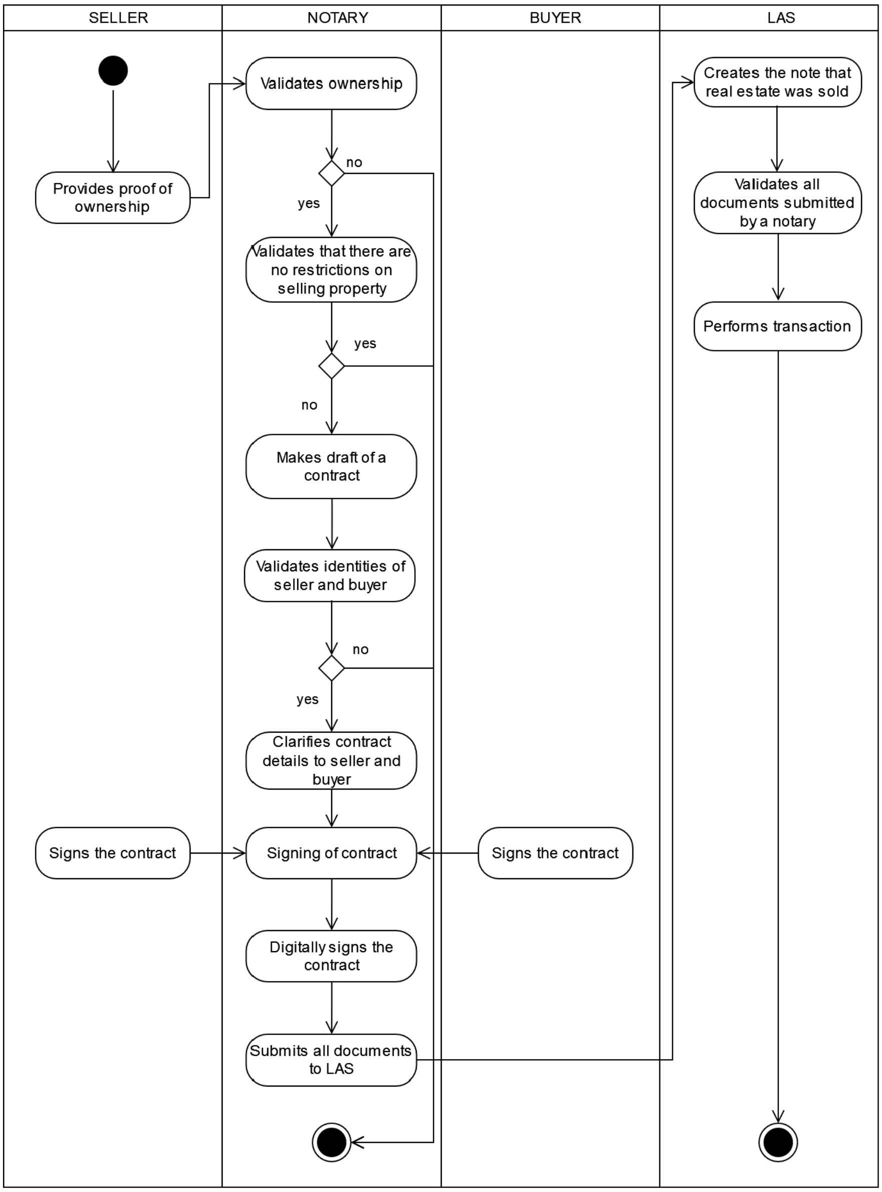

FIGURE 8. **The current process of real estate transactions in Serbia.**

3. Notary advises participants about the main points of the contract.
4. Seller and buyer will sign a digital version of the contract with their digital signatures.
5. By providing his digital signature, the notary will sign the contract and validate the identities of participating parties and that they are informed about the content of the contract they are about to sign.
6. Transaction of a token representing real estate will be submitted to the transaction pool and verified by the blockchain network.
7. LAS stores electronic versions of submitted documents. This flow is represented in Fig. 9.

V. DISCUSSION AND LIMITATIONS

It is important to notice that even if a transaction is added to the block, it does not mean that transaction is confirmed and permanently added to the record of transactions. This contradicts how confirmed transactions are regarded compared to centralized systems. Regarding the proof-of-work consensus mechanism, it is considered to be a probabilistic-finality consensus protocol [74]. There is always a theoretical possibility that longer chains could be created and that specific transactions will remain in shorter chains and therefore eventually be removed from blockchain. This could happen in the case of a 51% attack, but it can also happen in the regular working of a blockchain. Not all nodes are instantly updated when a new block is mined, there is a possibility that one parent block

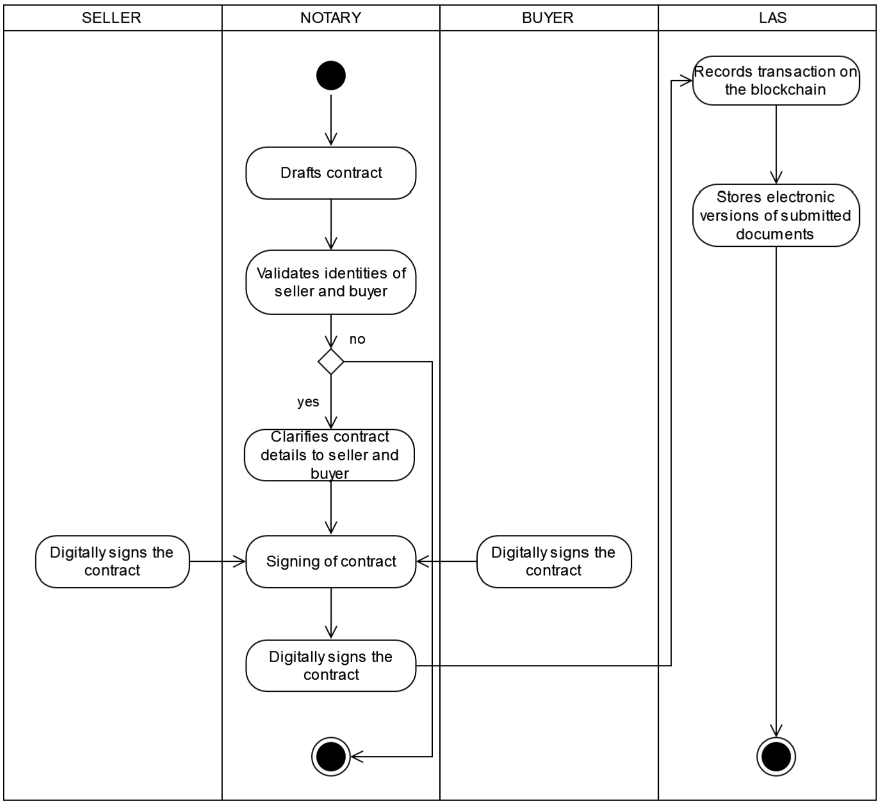

FIGURE 9. Simplified process of real estate transactions.

could have more than one child block at a specific moment in time. This happens when you have two blocks that are mined closed together. Both blocks get submitted to the network for validation, but only one block ends up being validated. Blocks that are not validated are referred to as orphan blocks in the Bitcoin blockchain and uncle blocks in the Ethereum blockchain. So, transactions could end up being in one of the orphaned or uncle blocks and therefore not being registered in the blockchain. For this reason, it is custom that sellers of real-life goods usually wait for several additional blocks to be added to the blockchain before accepting payment made as final and delivering purchased goods. Even with this in mind, registering transactions in the blockchain is still a much faster process than registering transactions in LAS.

Also, the possibility of double spending and tampering with data stored in LAS is eliminated. The proposed system represents a concrete solution of how to use NFT tokens to represent the real estate and how to support those not so common cases of transfers, but also does not represent a full transition from a traditional to a blockchain-based system. In this solution, two positions of a trusted third party are still necessary, one is a notary and the other is the handler of the master address.

Although one of the initial premises of blockchain was that there is no more need for trusted third parties, and in the proposed solution transfer is possible without trusted third parties. But, for real estate transactions, the existence of notaries (or similar officers) will probably always be necessary. The reason for this is rather simple, for most people real estate represents the most valuable tangible assets they will ever own, and implementing a system where transfer of such an asset could be achieved by "simply" digitally signing a digital document could open a door to different possibilities of misuse such as misleading, identify theft or coercion. The role of a notary is reduced to verifying the identities of participating parties, drafting or validating contract participants who are about to sign, and confirming all of this with his digital signature. Therefore, the role of the notary is to eliminate those possibilities since it is something that cannot be achieved through the application of BCT.

The problem of trusted third-party institutions that will handle usage of the master address could be questionable, but with the implementation of BCT, any usage of the master address will be recorded together with all necessary data. Furthermore, access to the master address could be limited and managed for example with the use of multiple signatures. Master address could be managed by the office of the LAS, but only based on some court ruling, for example as in a case of inheritance or for example if a party loses his private key. On the Bitcoin and Ethereum blockchain network, if the owner loses his private key, there is no way to recover his cryptocurrency. An analogous situation would be that if a party loses his private key, they would lose a possibility of transferring their real estate and that is why a master address is needed to eliminate such a case.

Apart from managing the master address, the role of the office of LAS should be extended. Firstly, the job of the initial creation of tokens and assigning ownership of those tokens to the current owner is something that should be done by the office of LAS. The issue of assigning public/private key pairs to different parties could also be done through the office of LAS, although in some countries different parties already can get their public/private key pair assigned by the state.

By implementing blockchain supported LAS, another improvement is that all events, in these cases transfers of real estate ownership are broadcasted to the network and any interested stakeholder can act upon those events. For example, after registering transactions in LAS, it is usually necessary to pay taxes (in some cases this is done before the transaction is registered), to change the identity of the party that electricity bill and utility bills will be billed to. Currently, that process requires new owner/owners to inform Electricity and Utility companies about this change, but by implementing blockchain supported LAS, those companies, as well as any other stakeholders can be informed about change as soon as it happens.

One of the most commonly mentioned limitations of BCT is that blockchain is not suitable for storing large quantities of data. As it was suggested in [45], [46], and [51], and in the proposed solution also, only hash values of accompanying documents are stored in blockchain. This is because running smart contracts on the Ethereum network requires gas paid in Ether, the cryptocurrency of Ethereum blockchain, gas is also needed for storing data on the blockchain, and it is rather expensive to store a high volume of data on Ethereum blockchain. For example, to store 1MB of data, 32,768 ''words'' of data are necessary. A word represents 256 bits on the Ethereum network. The Cost of storing one nonzero 256-bit word on the Ethereum blockchain is 20,000 gas. Gas prices are not fixed and in mid-September 2021 it was at around 100 gwei (1 ether = 109 gwei), so at the price of Ether and mid-September 2021, storing 1MB of data on the blockchain would cost just over US$750.000. Other possible solutions such as Inter Planetary File System (IPFS), Ethereum Swarm, and Torrent network have also their shortcomings, mainly lack of incentive for storing data. On IPFS, only hash values of documents are exchanged between neighboring nodes, documents are stored on multiple nodes if they are accessed by them. So, it is possible to remove a file from IPFS as long as it is hosted on only one node and there is no guarantee that in fact the file will be shared on multiple nodes. The combination of Filecoin and IPFS is also mentioned in some cases as it creates a combination of blockchain and IPFS [50], but even in this case, data are not stored on a distributed network without further effort from participants and there is also no guarantee that data will be preserved. The same can be said about the Torrent network, unless a file is downloaded by multiple nodes, if the only node that seeds a file stops the seed, the file will be removed from the network. Regarding Ethereum swarm, there should be a possibility of safely storing files over a distributed network, but at this point in time Swarm is still in its beta and according to creators, until an incentive mechanism is implemented it cannot be used as permanent storage for files. It is worth mentioning that with storing hash values of related documents, the possibility of tampering with those documents is removed, but storing is still done off the blockchain network.

Another possible limitation is related to legal constraints that need to be implemented. In [75] four legal challenges were identified for conveyancing LAS through blockchain and those are control of IDs, the legality of contracts, registration of co-ownership, and amendment of the ledger. Registration of co-ownership and amendment of the ledger is already discussed in this proposal, but control of IDs and legality of contracts is necessary to be done by the state. Those changes are probably the ones that will need the most time to be implemented and without the support of the state, LAS in blockchain as e-government services will never be implemented. In Italy, smart contracts are legally recognized as equal to traditional contracts since 2018 [76].

Running smart contracts on the Ethereum network requires Ether cryptocurrency. Whereas in current systems, registering transactions in the majority, if not all countries, requires some sort of payment, paying for registering transactions on the Ethereum network brings volatility to the price. To start with, prices of cryptocurrency are notoriously volatile, and fees for running smart contracts are volatile too. Since the beginning of August 2021, with the implementation of Ethereum Improvement Protocol 1559 (EIP-1559), the volatile nature of transaction fees should be reduced, but the volatile nature of cryptocurrency value is something that cannot be influenced easily. This could make it hard to predict for example taxes needed to be paid to LAS for registering a transaction.

Security is another important issue having in mind the importance of the system. For example, the possibility of a 51% attack has already been mentioned, but that is not the only security problem that exists in the blockchain. In [77] apart from 51% attack, selfish mining attack, Border Gateway Protocol attack, Eclipse attack, Liveness attack, and Balance attack are identified as possible problems, and as such, there is a need for them to be addressed in the development of future systems. Since the proposed solution is based on smart contracts, possible vulnerabilities in smart contracts are also something that needs to be paid attention to. The DAO (Decentralized Autonomous Organization) attack is one example of what can happen if purely designed smart contracts are being used. Issue such as readability and functional issues in process of creation, contract correctness, dynamic control flow in process of deployment, trustworthy oracle, transaction-ordering dependence, execution efficiency in process of execution, privacy and security, and scams in process of completion of smart contracts are identified in [66] as major challenges in the life cycle of a smart contract.

VI. CONCLUSION

In this paper, it is presented how some common problems in LAS could be solved using BCT. Apart from proposing BCT for solving problems of double spending and data tampering, a smart contract for reducing the time needed for registering transactions is presented. The proposed solution has the possibility of supporting those more specific cases in LAS, such as sharing of ownership, transferring part of ownership, splitting or merging of real estate, and limiting the possibility of trading a real estate, which was not addressed in previously published work. This is achieved by defining a programming interface that is based on two existing interfaces (ERC-20 and ERC-721), that on their own did not have the necessary support for managing those more specific LAS cases.

One direction for further research is looking for a solution for storing a high volume of data from LAS in some distributed system, instead of just storing a hash value of some of that data. Another question that could be addressed in the future is regarding specific platforms for implementation. As described in section III, a permissionless public blockchain is proposed for this solution, but a valid question could be raised if that is indeed the most acceptable solution for the point of view of the state. Due to the fact that in most countries LAS are government-run due to the importance of the system, it is highly unlikely that "handing over" control of this system to a public blockchain network would be feasible, even with having in mind that large blockchain networks are most resilient to 51% attack. So, it is open for discussion if a permissioned public blockchain network could be used, where only selected nodes could participate in changing the state of the network, whereas transactions will still be public. Also, understanding smart contracts written in Solidity would probably not be such an easy task for most of the stakeholders and that would again raise a question of trust, in this case in the party that created smart contract, but that issue could be resolved by developing domain-specific languages for the development of smart contract as proposed in [78]. Another question is tied to the general usage of public blockchains and is closely related to the problem of decentralization. Decentralization is represented as one of the strong points of blockchain networks, but due to high electricity costs related to proof of work consensus mechanism, in fact, the largest public blockchain networks are centralized in countries where electricity costs are low, so it raises the question about the level of decentralization of public blockchains.

REFERENCES

[1] J. Henssen and I. Williamson, "Land registration, cadastre and its interaction—A world perspective," in *Proc. XIX FIG Congr.*, Helsinki, Finland, 1990, pp. 14–43.

[2] C. Schmid, C. Hertel, and H. Wicke, "Real property law and procedure in the European Union," Eur. Univ. Inst. (EUI) Florence/European Private Law Forum Deutsches Notarinstitut (DnotI), Würzburg, Germany, Tech. Rep., May 2005, vol. 3.

[3] United Nations Economic Commission for Europe (UNECE) Working Party on Land Administration (WPLA), *Survey on Land Administration Systems*, United Nations, Geneva, Switzerland, 2014.

[4] J. Kaufmann and D. Steudler, "Cadastre 2014—A vision for a future cadastral system," Fédération Int. des Géomètres, Copenhagen, Denmark, Tech. Rep., 1998.

[5] S. Bittner and A. U. Frank, "A formal model of correctness in a cadastre," *Comput., Environ. Urban Syst.*, vol. 26, no. 5, pp. 465–482, Sep. 2002.

[6] G. J. Hunter and K. Beard, "Understanding error in spatial databases," *Austral. Surveyor*, vol. 37, no. 2, pp. 108–119, Jun. 1992.

[7] *Geographic Information—Data Quality*, Standard ISO 19157:2013, 2013.

[8] N. Vučić, M. Markovinović, and B. Mičević, "LADM in the republic of croatia-making and testing country profile," in *Proc. 5th LADM Workshop*, Kuala Lumpur, Malaysia, 2013, pp. 329–344.

[9] M. Stefanović, D. Pržulj, D. Stefanović, M. Vukmanović, and S. Ristić, "OCL specification of inter-register integrity constraints in land administration systems," in *Proc. 28th CECIIS*, Varaždin, Croatia, 2017, pp. 273–281.

[10] J. Vos, "Blockchain-based land registry: Panacea illusion or something in between?" presented at the IPRA/CINDER Congr., Dubai, UAE, 2017.

[11] V. L. Lemieux, "Trusting records: Is blockchain technology the answer?" *Rec. Manage. J.*, vol. 26, no. 2, pp. 110–139, Jul. 2016.

[12] M. Kempe, "The land registry in the blockchain-testbed," Kairos Future, Stockholm, Stockholm, Sweden, 2017.

[13] A. Shahaab, B. Lidgey, C. Hewage, and I. Khan, "Applicability and appropriateness of distributed ledgers consensus protocols in public and private sectors: A systematic review," *IEEE Access*, vol. 7, pp. 43622–43636, 2019.

[14] M. J. M. Chowdhury, M. S. Ferdous, K. Biswas, N. Chowdhury, A. S. M. Kayes, M. Alazab, and P. Watters, "A comparative analysis of distributed ledger technology platforms," *IEEE Access*, vol. 7, pp. 167930–167943, 2019.

[15] Y. Lu, "The blockchain: State-of-the-art and research challenges," *J. Ind. Inf. Integr.*, vol. 15, pp. 80–90, Sep. 2019.

[16] Z. Zheng, S. Xie, H.-N. Dai, X. Chen, and H. Wang, "Blockchain challenges and opportunities: A survey," *Int. J. Web Grid Services*, vol. 14, no. 4, pp. 352–375, 2018.

[17] A. H. Mohsin, A. Zaidan, B. Zaidan, O. Albahri, A. Albahri, M. Alsalem, and K. Mohammed, "Blockchain authentication of network applications: Taxonomy, classification, capabilities, open challenges, motivations, recommendations and future directions," *Comput. Standards Interfaces*, vol. 64, pp. 41–60, May 2019.

[18] J. L. Zhao, S. Fan, and J. Yan, "Overview of business innovations and research opportunities in blockchain and introduction to the special issue," *Financial Innov.*, vol. 2, no. 1, p. 28, Dec. 2016.

[19] F. Casino, T. K. Dasaklis, and C. Patsakis, "A systematic literature review of blockchain-based applications: Current status, classification and open issues," *Telematics Inform.*, vol. 36, pp. 55–81, Mar. 2019.

[20] E. Kalogianni, K. Janečka, M. Kalantari, E. Dimopoulou, J. Bydłosz, A. Radulović, N. Vučić, D. Sladić, M. Govedarica, C. Lemmen, and P. van Oosterom, "Methodology for the development of LADM country profiles," *Land Use Policy*, vol. 105, Jun. 2021, Art. no. 105380.

[21] A. Radulović, D. Sladić, M. Govedarica, A. Ristić, and D. Jovanović, "LADM based utility network cadastre in Serbia," *ISPRS Int. J. Geo-Information*, vol. 8, no. 5, p. 206, May 2019.

[22] J. Lisjak, M. Roić, H. Tomić, and S. M. Ivić, "Croatian LADM profile extension for state-owned agricultural land management," *Land*, vol. 10, no. 2, p. 222, Feb. 2021.

[23] M. Abdeslam Adad, S. El Hassane, M. El-ayachi, and F. Ibannain, "Supporting land data integration and standardization through the LADM standard: Case of Morocco's country profile MA-LADM," *Land Use Policy*, vol. 97, Sep. 2020, Art. no. 104762.

[24] B.-M. Lee, T.-J. Kim, B.-Y. Kwak, Y.-H. Lee, and J. Choi, "Improvement of the Korean LADM country profile to build a 3D cadastre model," *Land Use Policy*, vol. 49, pp. 660–667, Dec. 2015.

[25] *Geographic Information—Land Administration Domain Model (LADM)*, Standard ISO 19152:2012, 2012.

[26] M. Risius and K. Spohrer, "A blockchain research framework: What we (don't) know, where we go from here, and how we will get there," *Bus. Inf. Syst. Eng.*, vol. 6, no. 59, pp. 385–409, 2017.

[27] F. R. Batubara, J. Ubacht, and M. Janssen, "Challenges of blockchain technology adoption for e-government: A systematic literature review," in *Proc. 19th Annu. Int. Conf. Digit. Government Res.: Governance Data Age*, Delft, The Netherlands, 2018, pp. 1–9.

[28] G. V. Pereira, Y. Charalabidis, C. Alexopoulos, F. Mureddu, P. Parycek, A. Ronzhyn, D. Sarantis, L. Flak, and M. A. Wimmer, "Scientific foundations training and entrepreneurship activities in the domain of ICT-enabled governance," in *Proc. 19th Annu. Int. Conf. Digit. Government Research: Governance Data Age*, Delft, The Netherlands, May 2018.
[29] S. Ølnes, "Beyond bitcoin enabling smart government using blockchain technology," in *Proc. 15th IFIP WG*, Guimarães, Portugal, 2016, pp. 253–264.
[30] N. Elisa, L. Yang, F. Chao, and Y. Cao, "A framework of blockchain-based secure and privacy-preserving E-government system," *Wireless Netw.*, pp. 1–11, doi: 10.1007/s11276-018-1883-0.
[31] D. Geneiatakis, Y. Soupionis, G. Steri, I. Kounelis, R. Neisse, and I. Nai-Fovino, "Blockchain performance analysis for supporting cross-border E-government services," *IEEE Trans. Eng. Manag.*, vol. 67, no. 4, pp. 1310–1322, Nov. 2020.
[32] H. Hou, "The application of blockchain technology in E-government in China," in *Proc. 26th Int. Conf. Comput. Commun. Netw. (ICCCN)*, Vancouver, BC, Canada, Jul. 2017, pp. 1–4.
[33] N. Diallo, W. Shi, L. Xu, Z. Gao, L. Chen, Y. Lu, N. Shah, L. Carranco, T.-C. Le, A. B. Surez, and G. Turner, "EGov-DAO: A better government using blockchain based decentralized autonomous organization," in *Proc. Int. Conf. eDemocracy eGovernment (ICEDEG)*, Ambato, Ecuador, Apr. 2018, pp. 166–171.
[34] S. Ølnes, J. Ubacht, and M. Janssen, "Blockchain in government: Benefits and implications of distributed ledger technology for information sharing," *Government Inf. Quart.*, vol. 34, pp. 355–364, Sep. 2017.
[35] J. Vos, "What should we (not) do with land administration data?" presented at the World Bank Conf. Land Poverty, Washington, DC, USA, 2018.
[36] C. Alexopoulos, Y. Charalabidis, A. Androutsopoulou, M. A. Loutsaris, and Z. Lachana, "Benefits and obstacles of blockchain applications in e-government," in *Proc. Annu. Hawaii Int. Conf. Syst. Sci.*, Honolulu, HI, USA, 2019, pp. 3377–3386.
[37] V. L. Lemieux, "Evaluating the use of blockchain in land transactions: An archival science perspective," *Eur. Property Law J.*, vol. 6, no. 3, pp. 392–440, Dec. 2017.
[38] M. Stefanović, D. Pržulj, S. Ristić, D. Stefanović, and M. Vukmanović, "Blockchain and land administration: Possible applications and limitations," in *Proc. EBM*, Kragujevac, Serbia, 2018, pp. 1–8.
[39] M. Stefanović, S. Ristić, D. Stefanović, M. Bojkić, and D. Pržulj, "Possible applications of smart contracts in land administration," in *Proc. 26th TELFOR*, Belgrade, Serbia, Nov. 2018, pp. 420–425.
[40] G. Miscione, R. Ziolkowski, L. Zavolokina, and G. Schwabe, "Tribal governance: The business of blockchain authentication," in *Proc. 51st Hawaii Int. Conf. Syst. Sci.*, Waikoloa Village, HI, USA, 2018, pp. 4484–4493.
[41] H. Müller and M. Seifert, "Blockchain, a feasible technology for land administration," in *Proc. FIG Working Week*, Hanoi, Vietnam, 2019, pp. 1–9.
[42] C. Lemmen, J. Vos, and B. Beentjes, "Ongoing development of land administration standards," *Eur. Property Law J.*, vol. 6, no. 3, pp. 478–502, Dec. 2017.
[43] M. Nofer, P. Gomber, O. Hinz, and D. Schiereck, "Blockchain," *Bus. Inf. Syst. Eng.*, vol. 59, no. 3, pp. 183–187, 2017.
[44] R. Thomas and C. Huang, "Blockchain, the Borg collective and digitalisation of land registries," *Conveyancer Property Lawyer*, vol. 81, no. 1, pp. 14–25, 2017.
[45] H. P. Wouda and R. Opdenakker, "Blockchain technology in commercial real estate transactions," *J. Property Investment Finance*, vol. 37, no. 6, pp. 570–579, Sep. 2019.
[46] V. Thakur, M. N. Doja, Y. K. Dwivedi, T. Ahmad, and G. Khadanga, "Land records on blockchain for implementation of land titling in India," *Int. J. Inf. Manage.*, vol. 52, Jun. 2020, Art. no. 101940.
[47] K. Veeramani and S. Jaganathan, "Land registration: Use-case of e-governance using blockchain technology," *KSII Trans. Internet Inf. Syst.*, vol. 14, no. 9, pp. 3693–3711, 2020.
[48] K. Mintah, K. T. Baako, G. Kavaarpuo, and G. K. Otchere, "Skin lands in Ghana and application of blockchain technology for acquisition and title registration," *J. Property, Planning Environ. Law*, vol. 12, no. 2, pp. 147–169, May 2020.
[49] S. Kumar, V. Talasila, and R. Pasumarthy, "A novel architecture to identify locations for real estate investment," *Int. J. Inf. Manage.*, vol. 56, Feb. 2021, Art. no. 102012.
[50] A. Annand, M. McKibbin, and F. Pichel, "Colored coins: Bitcoin, blockchain, and land administration," presented at the Annu. World Bank Conf. Land Poverty, Washington, DC, USA, 2017.
[51] C. Kombe, M. Manyilizu, and A. Mvuma, "Design of land administration and title registration model based on blockchain technology," *J. Inf. Eng. Appl.*, vol. 7, no. 1, pp. 8–15, 2017.
[52] D. van Bochove, L. de Bruin, and C. Lemmen, "From bitcoins to bitsquares–blockchain in land administration," *GIM Int.*, vol. 30, no. 9, pp. 36–37, 2016.
[53] F. Jahan, M. Mostafa, and S. Chowdhury, "SHA-256 in parallel blockchain technology: Storing land related documents," *Int. J. Comput. Appl.*, vol. 175, no. 35, pp. 33–38, Dec. 2020.
[54] A. Sahai and R. Pandey, "Smart contract definition for land registry in blockchain," in *Proc. IEEE 9th Int. Conf. Commun. Syst. Netw. Technol. (CSNT)*, Gwalior, India, Apr. 2020, pp. 230–235.
[55] A. F. Mendi, K. K. Sakakli, and A. Cabuk, "A blockchain based land registration system proposal for Turkey," in *Proc. 4th Int. Symp. Multidisciplinary Stud. Innov. Technol. (ISMSIT)*, Istanbul, Turkey, Oct. 2020, pp. 1–6.
[56] A. Torun, "Hierarchical blockchain architecture for a relaxed hegemony on cadastre data management and update: A case study for Turkey," in *Proc. UCTEA Int. Geographical Inf. Syst. Congr.*, Adana, Turkey, 2017, pp. 1–11.
[57] G. Sladić, B. Milosavljević, S. Nikolić, D. Sladić, and A. Radulović, "A blockchain solution for securing real property transactions: A case study for Serbia," *ISPRS Int. J. Geo-Inf.*, vol. 10, no. 1, p. 35, Jan. 2021.
[58] K. M. Alam, J. M. Ashfiqur Rahman, A. Tasnim, and A. Akther, "A blockchain-based land title management system for Bangladesh," *J. King Saud Univ.-Comput. Inf. Sci.*, pp. 1–15, doi: 10.1016/j.jksuci.2020.10.011.
[59] P. Singh, "Role of blockchain technology in digitization of land records in Indian scenario," in *Proc. ICECAE*, Tashkent, Uzbekistan, 2020, pp. 1–13.
[60] J. Graglia and C. Mellon, "Blockchain and property in 2018: At the end of the beginning," presented at the Annu. World Bank Conf. Land Poverty, Washington, DC, USA, 2018.
[61] A. B. Pedersen, N. Group, M. Risius, and R. Beck, "A ten-step decision path to determine when to use blockchain technologies," *MIS Quart. Executive*, vol. 18, no. 2, pp. 99–115, 2019.
[62] S. Wang, L. Ouyang, Y. Yuan, X. Ni, X. Han, and F.-Y. Wang, "Blockchain-enabled smart contracts: Architecture, applications, and future trends," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 49, no. 11, pp. 2266–2277, Nov. 2019.
[63] N. Szabo, "Smart contracts: Building blocks for digital markets," *Extropy*, vol. 16, pp. 50–53 and 61–63, 1996.
[64] V. Buterin. *A Next-Generation Smart Contract and Decentralized Application Platform*. Accessed: Dec. 9, 2021. [Online]. Available: https://people.cs.georgetown.edu/~clay/classes/fall2017/835/papers/Etherium.pdf
[65] M. Giancaspro, "Is a 'smart contract' really a smart idea? Insights from a legal perspective," *Comput. Law Secur. Rev.*, vol. 33, no. 6, pp. 825–835, 2017.
[66] Z. Zheng, S. Xie, H.-N. Dai, W. Chen, X. Chen, J. Weng, and M. Imran, "An overview on smart contracts: Challenges, advances and platforms," *Future Gener. Comput. Syst.*, vol. 105, pp. 475–491, 2020.
[67] S. Farshidi, S. Jansen, S. Espana, and J. Verkleij, "Decision support for blockchain platform selection: Three industry case studies," *IEEE Trans. Eng. Manag.*, vol. 67, no. 4, pp. 1109–1128, Nov. 2020.
[68] Etherscan. *Ethereum Network Hash Rate Chart | Etherscan*. Accessed: Sep. 15, 2021. [Online]. Available: https://etherscan.io/chart/hashrate
[69] F. Vogelsteller and V. Buterin. *EIP-20: EPC-20 Token Standard*. Accessed: Jan. 15, 2021. [Online]. Available: https://eips.ethereum.org/EIPS/eip-20
[70] W. Entriken, D. Shirley, J. Evans, and N. Sachs. *EIP-721: Non-Fungible Token Standard*. Accessed: Jan. 15, 2021. [Online]. Available: https://eips.ethereum.org/EIPS/eip-721
[71] Blockchair. *Bitcoin Transactions Per Second Chart–Blockchair*. Accessed: Jan. 15, 2021. [Online]. Available: https://blockchair.com/bitcoin/charts/transactions-per-second?interval=3m
[72] Blockchair. *Ethereum Transactions Per Second Chart–Blockchair*. Accessed: Jan. 15, 2021. [Online]. Available: https://blockchair.com/ethereum/charts/transactions-per-second
[73] Ministère de la Transition Écologique. *House Prices in France: Property Price Index, French Real Estate Market Trends in the Long Run*. Accessed: Oct. 21, 2021. [Online]. Available: http://www.cgedd.developpement-durable.gouv.fr/house-prices-in-france-property-price-index-french-a1117.html
[74] S. Zhang and J.-H. Lee, "Analysis of the main consensus protocols of blockchain," *ICT Exp.*, vol. 6, no. 2, pp. 93–97, Jun. 2020.

[75] R. M. Garcia-Teruel, ''Legal challenges and opportunities of blockchain technology in the real estate sector,'' *J. Property, Planning Environ. Law*, vol. 12, no. 2, pp. 129–145, Jan. 2020.
[76] R. Guido, G. Mirabelli, E. Palermo, and V. Solina, ''A framework for food traceability: Case study–Italian extra-virgin olive oil supply chain,'' *Int. J. Ind. Eng. Manag.*, vol. 11, no. 1, 2020, pp. 50–60.
[77] X. Li, P. Jiang, T. Chen, X. Luo, and Q. Wen, ''A survey on the security of blockchain systems,'' *Future Gener. Comput. Syst.*, vol. 107, pp. 841–853, Jun. 2020.
[78] M. Wohrer and U. Zdun, ''Domain specific language for smart contract development,'' in *Proc. IEEE Int. Conf. Blockchain Cryptocurrency (ICBC)*, May 2020, pp. 1–9.

MIROSLAV STEFANOVIĆ (Graduate Student Member, IEEE) was born in Loznica, Serbia, in 1975. He received the B.S. degree in information management and the M.S. degree in information systems engineering from the Faculty of Technical Sciences, University of Novi Sad, Novi Sad, Serbia, in 2014 and 2016, respectively, where he is currently pursuing the Ph.D. degree in industrial engineering and engineering management.

From 2015 to 2016, he was a Teaching Associate and since 2016, he has been a Teaching Assistant at the Department of Industrial Engineering and Engineering Management, Chair for Information and Communication Systems, Faculty of Technical Sciences, University of Novi Sad. His research interests include blockchain technologies, especially the implementation of blockchain technology in fields other than cryptocurrency, mainly e-government and land administration systems.

ĐORĐE PRŽULJ was born in Sarajevo, Bosnia and Herzegovina, in 1973. He received the B.S. degree in mechanical engineering and the Mr. and Ph.D. degrees in industrial engineering and engineering management from the Faculty of Technical Sciences, University of Novi Sad, Novi Sad, Serbia, in 1999, 2004, and 2013, respectively.

From 1999 to 2013, he was a Teaching Assistant and from 2013 to 2018, he was an Assistant Professor at the Department of Industrial Engineering and Engineering Management, Chair for Information and Communication Systems, Faculty of Technical Sciences, University of Novi Sad, where he has been an Associate Professor, since 2018. He has published in several international information systems journals. His research interests include land administration systems, service oriented architecture, microservices, ontologies, domain specific languages, and blockchain applications in information systems.

SONJA RISTIĆ (Member, IEEE) received the bachelor's degree (Hons.) in mathematics from the Faculty of Science, University of Novi Sad, Novi Sad, Serbia, in 1983, and the bachelor's degree (Hons.) in economics and the Mr. (two years) and Ph.D. degrees in informatics from the Faculty of Economics, University of Novi Sad, in 1989, 1994, and 2003, respectively.

From 1984 until 1990, she worked with Novi Sad Cable Company NOVKABEL–Factory of Electronic Computers. From 1990 till 2006, she was with the Novi Sad School of Business, and since 2006, she has been with the Faculty of Technical Sciences, University of Novi Sad. She currently works as a Full Professor at the Faculty of Technical Sciences, University of Novi Sad. Her research interests include database systems, software engineering, model-driven software engineering, and domain specific languages. She is the author or coauthor of over 100 papers and ten industry projects and software solutions in the area.

DARKO STEFANOVIĆ (Member, IEEE) was born in Novi Sad, Serbia, in 1972. He received the B.S. degree in mechanical engineering and the Mr. and Ph.D. degrees in industrial engineering and engineering management from the Faculty of Technical Sciences, University of Novi Sad, Novi Sad, in 1999, 2005, and 2012, respectively.

From 2001 to 2012, he was a Teaching Assistant and from 2012 to 2017, he was an Assistant Professor at the Department of Industrial Engineering and Engineering Management, Chair for Information and Communication Systems, Faculty of Technical Sciences, University of Novi Sad, where he has been an Associate Professor, since 2017. He is currently the Vice-Dean for the Science and International Cooperation, Faculty of Technical Sciences, University of Novi Sad, and the Head of the Chair of Information and Communication Systems. He has published in several international information systems journals. His research interests include ERP systems, e-learning systems, e-government systems, data mining, and business process mining in production planning.

DANILO NIKOLIĆ (Graduate Student Member, IEEE) was born in Vršac, Serbia, in 1996. He received the B.S. and M.S. degrees in information systems engineering from the Faculty of Technical Sciences, University of Novi Sad, Novi Sad, Serbia, in 2019 and 2020, respectively, where he is currently pursuing the Ph.D. degree in information systems engineering.

From 2019 to 2020, he was a Teaching Associate and since 2020, he has been a Teaching Assistant at the Department of Industrial Engineering and Engineering Management, Chair for Information and Communication Systems, Faculty of Technical Sciences, University of Novi Sad. His research interests include static code analysis and tools for static code analysis, especially the implementation of static code analysis on domain-specific languages.