Ethereum Smart Contract Deployment for a Real Estate Management System (REMS) Implemented in Blockchain

Admirim Aliti [1], Marika Apostolova [1], Artan Luma [1], Azir Aliu [1], Majlinda Fetaji [1], Halil Snopce [1]

[1]*South East European University, Tetovo, North Macedonia*

***Abstract* – The real industry is crucial sector of the world wide financial system, with significant economic, social and environmental implications. However, real estate transactions are often slow, complex and costly and can be prone to fraud errors, which can lead to significant financial losses and legal disputes. This paper proposes the adoption of blockchain technology in real estate cadastral systems as a solution to the challenges encountered in managing property ownership and transactions. We have described the implementation of our prototype called REMS (Real Estate Management System) and we have presented the deployment of smart contracts in Ethereum platform. Based on measurement, benchmarks and other observation of the system, we have evaluated the server usage of the blockchain network and decided whether Blockchain-as-a-Service (BaaS) should be involved or not in our system. The study results demonstrate the successful implementation of a real estate management system (REMS) using blockchain technology and Ethereum's smart contracts. This study is important because it confirms that similar solutions can be implemented in other areas of public administration, where the structure of the work is similar, i.e. where we deal with issuing documents to citizens.**

***Keywords* – Blockchain, smart contracts, Ethereum, real estate property.**

DOI: 10.18421/TEM123-18
https://doi.org/10.18421/TEM123-18

Corresponding author: Admirim Aliti,
South East European University, Tetovo, North Macedonia
Email: aa03511@seeu.edu.mk

Received: 22 March 2023.
Revised: 07 July 2023.
Accepted: 19 July 2023.
Published: 28 August 2023.

1. Introduction

Real estate assets, including land, buildings, and infrastructure, frequently make up the majority of individual and institutional portfolios [1]. Governmental cadastre systems, which are used to manage and track ownership of land and property, are essential to the procedure of purchasing, selling or transferring the property ownership. Many parties are involved in those processes, including buyers, sellers, real estate agents, lenders, attorneys, as well as governmental organizations in charge of overseeing property transactions and keeping land records [2]. Thus, the transactions should be fast, secure, immutable and transparent.

Blockchain technology has been explored in various industries, but its potential impact on the real estate industry and cadastre system is significant. Using this technology in sector of real estate and specifically in cadastre can create a decentralized system that is transparent and resistant to fraud. It can also help reduce the need for intermediaries, improve the accuracy and efficiency of transactions, reduce costs and enhance trust and security [3], [6].

In our previous paper [7], we have proposed a framework for an administration system, which is based in blockchain and implemented in Ethereum. Furthermore IPFS (Interplanetary File System) network is generally used when there is a case to store large files, scanned documents, images, or photos of a property. Additionally, in this paper, we will continue our research with smart contract deployment in Solidity language. The focus is to develop a first prototype that is ready to be used by cadastre agency and to conduct user-testing for prototyping and smart contracts.

Deliverables of our system implementation would be: domain hosting the application, verifier web app, issuer web-app, IPFS network, blockchain network (with smart contract deployed), benchmarking scripts, blockchain gateways and IPFS gateways.

The tests we will provide include: (1) blockchain network, (2) verifier interface, (3) certificate issuer, (4) IPFS network, and (5) cost of transaction.

In Section 2 related work regarding using blockchain in Real Estate property certificates management are presented. The key aspects of blockchain, consensus algorithms and platform we have chosen are analyzed in Section 3. Section 4 describes all main components of deployed smart contracts. The benchmarks and testing are evaluated in Section 5. Final section concludes the paper and discuss future work.

2. Related Works

Blockchain is a distributed ledger technology that is characterized by its decentralized and immutable nature. It has several key features that make it appealing for various applications, such as the ability to ensure data integrity, security, transparency and privacy [8], [9]. Several research studies and pilot projects have explored the potential of blockchain technology in real estate and cadaster systems.

A thorough investigation into the application of blockchain technology in the real estate sector was carried out by several authors [10]. The authors looked at the potential applications in a range of fields, including land management, real estate transactions, investments, leasing, and maintenance.

A method for increasing fairness in real estate buying and selling that goes beyond financial factors is presented in study [11]. The plan includes the concept of a "smart house," which could use IoT (Internet of Things) technology to identify the current owner and, upon verification, grant access to all services or, if verification is not possible, report the owner to authorities.

Blockchain technology, according to the authors of study [12], makes real estate transactions faster, safer, and more low-cost. They are largely focused on investigating and synthesizing blockchain-based methods utilized in Russia and elsewhere, as well as outlining the main routes and alternatives for its further development and improvement as a highly effective and optimistic financial and economic mechanism for real estate transaction participants. They forecast that blockchain technology will be widely employed in six to seven years.

One of the blockchain's successful pilot projects, Sweden's Land Registry [13], is being used to test the viability of using the technology for real estate transactions. The project used a private blockchain to track and record real estate transactions to decrease fraud and boost transparency.

The Dubai Land Department has launched a blockchain-based platform called the Real Estate Self Transactions (REST) that allows property buyers and sellers to complete transactions online including time verification, escrow, and registration [14].

Few cases have been successful, and many have been marked with failure, as is seen in the [15], which mentions the cases of Ukraine and Honduras. According to reports, the Honduran blockchain-based land record project was targeted by political opposition and failed in its early phases, whereas in Ukraine, environmental factors such as Russian intervention rendered the project less effective.

Additional research and development are required to overcome blockchain technology's challenges and limitations and ensure successful applications in real-world settings. Institutional difficulties, which are also substantial impediments, may hinder or accelerate the adoption of disruptive technologies such as blockchain.

3. Blockchain

During the development of this project a lot has changed in the blockchain technology, some of these changes are noted below. This part will focus mainly on the technical decisions made, why they were made and as well as the development and deployment environments, how past and previous changes in the blockchain ecosystem will affect this project, and how we might adjust a few deployment settings based on feedback and data from PoC.

3.1. Blockchain Switching From PoW to PoS Consensus Algorithm

Previously, Ethereum operated on a Proof-of-Work (PoW) consensus algorithm [16], which determined the inclusion of blocks in the network based on solving a cryptographic problem. Each node compiled a list of transactions, combined them with a random number, and attempted to fulfill a cryptographic hash. This computationally expensive process accepted the block of the first node to meet the requirement. However, in 2022, Ethereum transitioned to its Proof-of-Stake (PoS) mechanism, which offers improved energy efficiency, security, and scalability for the implementation of new solutions.

PoS is another consensus algorithm on which transactions and block are confirmed and sealed [17]. Unlike PoW, where each node has to compete in solving a computationally expensive problem, PoS randomly selects a miner based on his stake and the time of the stake on the network.

The condition that needs to be satisfied is:

$$U \leq \frac{\text{bal}_\text{i} * t_\text{i}}{D} \leq 1 \quad (1)$$

U represents the hash operation, while **D** specifies the target difficulty, $\boldsymbol{bal_i}$ the balance of the selected node and $\boldsymbol{t_i}$ the time since the node was last selected. This means the more time a miner is a staker, the more chances he has of confirming a block [18]. Should be noted that this time resets to 0 once a miner wins and confirms a block. This makes PoS: faster, less computationally expensive, less prone to confirmation failures, open to more peers, more decentralized.

Although such a move might not have changed a lot in terms of transaction fees, it has certainly reduced block times by at least 20%. This is seen clearly in the official Ethereum resources, where average block confirmation time went from 15 seconds to 12. Apart from this, the above-mentioned advantages still remain, making Ethereum a more viable choice in the future in case of a hybrid blockchain selection.

However, as of now, these changes do not prove to be sufficient to switch to a fully public network. This, along with the fact that private blockchain provides finer and greater control, makes Proof-of-Authority blockchain consensus our choice for PoC.

3.2. Hybrid vs. Private Implementation

Blockchain can be categorized as public, private, consortium or hybrid implementation. Unfortunately, smart contracts used on the present public blockchain may disclose private information [5], [4]. This refers to the way in which the network is connected. In a public network we connect to the main network of Ethereum, where all worldwide transactions happen. While this is very secure in terms of attacks, it is the most expensive type of blockchain available, and offers little to no benefits compared to other solutions. While a private blockchain network depends only on predefined nodes, which are owned by predefined parties. In a private implementation, the blockchain enforces strict membership and has a system in place to control who is allowed to sign up. Every node is therefore authorized and authenticated, and other nodes are aware of its identity [19], [20].

On the other hand, there is a hybrid implementation, which utilizes both types of networks, in an attempt to gain the benefits of both, in exchange with increased complexity. Hybrid is among one of initially discussed and proposed solutions.

However, this would add a lot of complexity in return for almost no benefits, thus making this implementation inefficient in our case.

Such implementation would consist of storing large amounts of data in a private network, and an important part to the public blockchain. Even though some might refer to our solution as a hybrid network, because it utilizes a public IPFS network, this would not be the case, as the public network is only used for redundancy and edge storage. It would be the case if we only referred to IPFS as hybrid, as opposed to the whole blockchain network. This means that the PoC will be focused on private blockchain, with a private IPFS implementation while closely monitoring IPFS server in case a public integration shall be made.

3.3. Blockchain Being Used as a Service

Blockchain as a Service (BaaS) is another term used to describe services that offer fully managed blockchain solutions, public or private. These services make maintaining a network and spinning up new nodes much easier. The downside is that these services are more expensive than using a self-hosted or ad hoc solution. However, they offer greater security and redundancy. Or next step is to perform several tests among several BaaS platforms in order to come up with the best solution for our use case. The PoC will determine whether such services are needed for production, or the currently proposed in PoC suffice.

4. Smart Contract Deployment

We must create the Ethereum platform's smart contracts in order to implement the system. A smart contract is a piece of software that records negotiation rules, checks for compliance automatically, and then implements the terms of the agreement [21]. Smart contracts are enforced correctly without the need for a reliable authority. Solidity is the programming language we used to create our system's smart contracts [22].

In our prototype, the smart contract code is split in two files. The first file, called *'certs.sol'* is responsible for handling certificate operations and other certificate related services. The other file is called *'permissions.sol'*, which takes into account all operations related to authorization. It makes sure that no operation is done without the right permissions. Below are presented the majority of the functions used in the contract, from which the first one is from the first file, and the others are from *'Persmissions.sol'*. These files are then compiled into bytecode and then run on the EVM as a single smart contract.

```
//Function addCert
function addCert(bytes[] memory_noProperty, String[]
   memory _rawData, string [] memory_data, State[]
   memory _state, string[] memory _photoHash)
public {
require(isIssuer(msg.sender));
for(uint i=0; i< _noProperty.length; i++)
{
  if(certs[_noProperty[i]].created !=0)
        return;
  certs[_noProperty[i]]= certStruct(_rawData[i], _data
    [i], now, _state[i], _photoHash[i]);
}
      }
```

Figure 1. addCert() function implementation

We initially check if the address has the required permissions. After that, we get an array of deployed certificates and push it to the array that holds all the certificates that are currently issued. Array is being used in order to make the process parallel. In upcoming tests we will try to specify the maximum number of the certificates that can be deployed at a time. In case a certificate with the same ID is found, we stop the process.

The rest of the functions belong to the address roles. The most basic and important rule is to check the issuer of the certificate. We should make sure that the address is allowed to add certificates. Blockchain works in a way that every address can make transactions to the network and our smart contract, and in order to verify that this transaction has been made by an address we can use its private key. This allows us to make 100% sure that the address is allowed to make this transaction. This is also known as transaction signing.

Similar to the certificate we have the issuing entity which has the following fields: *Address: issuerAddress*, *String: issuerName*, *Bool: issuerState.*

In the case we have a constructor, which defines the master address of the contract, as well defines the contract sender as root. The above function enables adding new issuer address. Only the address is required and then the user is added to the list of issuers.

```
//Function addIssuer
function addIssuer(address _issuer, string memory
  _issuerName) public
              {
    if(msg.sender == root || msg.sender == master)
      {
issuers[issuerCount].issuerAddress = _issuer;
issuers[issuerCount].issuerName = _issuerName;
issuers[issuerCount].issuerState = true;
issuerCount++;
      }
}
```

Figure 2. addIssuer( ) fundtion implementation

The above function enables adding new issue address. Only the address is required and then this user is added to the list of issuers.

```
//Function switchIssuer
function switchIssuer(address _issuer) public
              {
    if(msg.sender == root || msg.sender == master)
    for(uint i = 0; I <= issuerCount; i++)
    {
        if(issuers[i].issuerAddress == _issuer)
          {
            issuers[i].issuerState ? issuers[i].
             issuerState = false : issuers[i].
             issuerState = true;
          }
              }
        }
}
```

Figure 3. switchIssuer( ) function implementation

This part disables or enables users when needed. Only the master is allowed to execute this function.

```
//Function isIssuer
function isIssuer(address _issuer) public view returns
  (bool)
              {
    for(uint i = 0; i <= issuerCount; i++)
    {
        //NOTE: ISSUERSs state should also be true
        if(issuers[i].issuerAddress == _issuer &&
         issuers[i].issuerState == true)
    {
              return true;
    }
       else if (issuerCount == i)
    {
      return false;
    }
        }
}
```

Figure 4. IsIssuer ( ) function implementation

The changes in the blockchain ecosystem will affect this project, and how we might tweak a few deployment settings based on feedback and data from PoC.

5. Hosting Web Apps and Benchmarking

The web app is built using ReactJS Framework. There are many ways of hosting such web applications but considering this framework compiles to a simplified HTML/CSS/JS bootstrap, a specified or dedicated server is not needed. The chosen hosting solution has been CloudFlare Pages. CloudFlare Pages is a service for hosting such applications, and its pricing is based on builds/deployments per month. As of now, it is reasonable solution at no costs.

Hosting the web app is as follows: (1) Domain is linked to CloudFlare DNS; (2) Web App source code is hosted in GitHUB; (3) GitHub is connected to CloudFlare; (4) Web App is pulled from CloudFlare, compiled and stored; (5) Compiled app is served form CloudFlare Pages. Verifier Web App is presented in figure below.

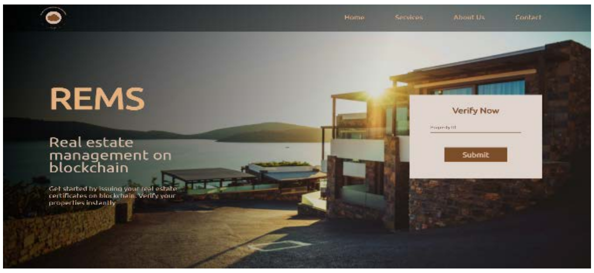

Figure 5. Verifier Web App

User initially fills the property ID and submits the form. If the property exits, it will be prepared for decryption and once the decryption key has been entered, the user will be presented with the details of the property.

5.1. Resource Usage

In this section are presented a set of graphs regarding resource usages of the server hosting specific services in relation to the system are presented. Graphs are represented withing a 7 day timeframe, each dedicated to specific services and purposes. Figure 6 sets a baseline for our benchmarks.

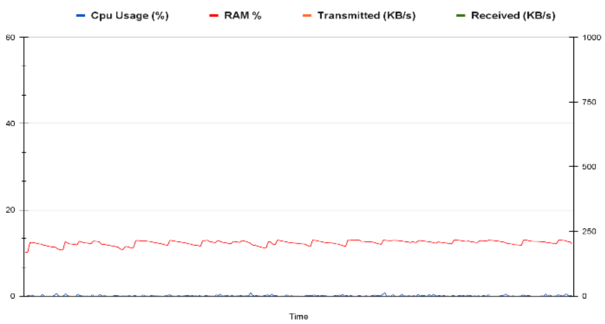

Figure 6. Baseline for our benchmark

The above figure represents the load of the server under no active services. It means that it is the minimum resource usage we can get on our server. As expected, network usage is non-existent, this is because the server is not communicating with any other significant servers. This is not entirely true, because servers usually ping and receive data regarding software updates and default system apps.

The RAM usage represents the minimum required of the operating system and its default apps and packages.

CPU, similar to traffic/network usage, is almost at 0% usage. Values close to 0% are from pinging software servers and checking for updates, default packages and operating system operations. Meanwhile, Figure 7, represents resource usage from running the IPFS node only.

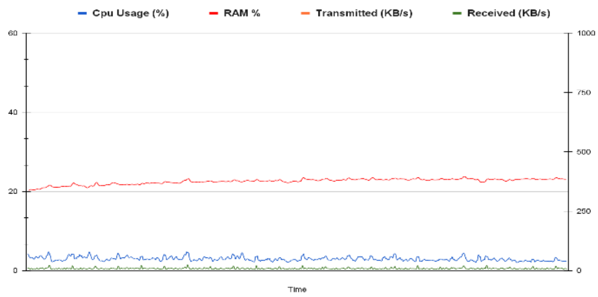

Figure 7. Running the IPFS node only

As can be noticed in the graph, an increase of 10% in RAM usage and 5% in CPU usage is evaluated. Despite the apparent rise in network traffic, it primarily stems from internal network data transfers, amounting to approximately 90 bits per second. Furthermore, Figure 8 presents the blockchain node requirements.

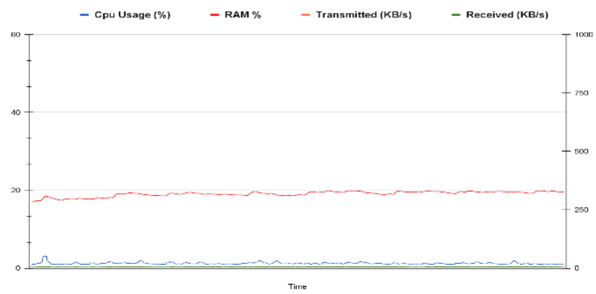

Figure 8. Blockchain node requirements

In the graph we can see an increase of 6.5% in RAM usage, while the CPU and network usage remains low, under 1%.

5.2. Discussing the Benchmarks and Evaluations

From the above metrics, we can determine the minimum server requirements for hosting our services. From the IPFS network we can conclude that: a minimum of one core is needed and a minimum RAM of 512 MB (2048 MB * 0.23 = 471 MB), with 512 MB being the closest RAM available. For the blockchain network we can determine that a minimum of one core is needed, with at least 389 MB of RAM.

As concluded above, with 512 MB being the closest package. Both services can be hosted under one server, with a minimum of one core and 604 MB of RAM, with the closest RAM available being 1 GB. This is determined by knowing that the baseline of running the OS and default apps is 256 MB of RAM, 113 MB for the blockchain node and 215 MB for the IPFS node.

Should be noted that these are derived from the system under no load, and actual production resource usages may vary, but this being the case on later steps of the development. Also, these results might be contradictory to previous measures, but this is because these measures span across a larger time frame.

6. Conclusion

With the given information we have an idea how to build the prototype based on measurements, benchmarks and other observations. As stated, the PoC will consist of a private PoA based authority blockchain with a private IPFS network.

The main focus of the blockchain network will be the server usage, and whether BaaS should be involved or not. This is one of the most significant conclusions to be made after the PoC has concluded, and perhaps at a much later stage, due to the main cost contributor as of now.

While having the IPFS network up and running, observations will be made to determine whether a public network should be included or not, and what scaling methods will be most suited for our use case.

We believe that blockchain can improve the real estate industry in multiple ways, including: (1) increased liquidity; (2) a more manageable and easily tracked supply chain; (3) more control over investments; (4) more exposure to real estate assets; (5) increased efficiency for the operational side of the real estate industry; (6) more transparency amongst global financial networks; (7) reduction in costs due to the removal of manual tasks and human involvement.

The benefits of adoption of blockchain technology in public administration include increased transparency, reduced fraud risks, improved transaction efficiency, and enhanced trust and security. By eliminating intermediaries and streamlining processes, blockchain can potentially lower costs and simplify property transactions. However, challenges such as scalability, regulatory frameworks, and institutional barriers need to be addressed for widespread implementation. Future research should focus on exploring scalable blockchain solutions, addressing privacy concerns, and developing interoperability standards to enable seamless integration with existing systems.

References:

[1]. Saari, A., Vimpari, J., & Junnila, S. (2022). Blockchain in Real Estate: Recent developments and empirical applications. *Land Use Policy*, *121*, 106334-106345.

[2]. Seigneur, J. M., Pusterla, S., & Socquet-Clerc X. (2022). Blockchain real estate relational value survey. *In Proceedings of the 35th Annual ACM Symposium on Applied Computing (SAC '20)*, New York, USA, 279-285,

[3]. Saldic, G., Milosavljevic, B., Nikolic, S., & Sladic, D., Radulovic, A. (2021). A Blockchain solution for securing real property transactions: A case study for Serbia. *ISPR International Journal of Geo-Information*, *10*(1), 35-57.

[4]. Martyn, A (2018). The concept of Land Plot as a Combination of Smart Contracts: A Vision for creating blockchain cadastre. *Baltic surveying. International scientific journal*, *8*(1), 68-73.

[5]. Karamitsos, I., Papadaki, M., & Al Barghuthi, N. B (2018). Design of the blockchain smart contract: A use case for real estate. *Journal of Information Security*, *9*(3), 177-190.

[6]. Muller, H., & Markus, S. (2019). Blockchain, a feasible technology for land administration. In *Proceedings FIG Working Week 2019, Geospatial Information for a Smarter Life and Environmental Resilience,* Hanoi, Vietnam, 22-26.

[7]. Aliti, A., Apostolova, M., & Luma, A. (2022). Design and development of smart contract in public administration focused on real estate certificates' management by using Ethereum and IPFS. *International Conference on Engineering Technologies (ICENTE)*, Konya, Turkey, 117-122.

[8]. Leka, E., & Selimi, B. (2020). BCert – A decentralized academic certificate system distribution using blockchain technology. *International Journal on Information Technologies & Security (IJITS)*, *12*(4), 103-118.

[9]. Yli-Huumo, J., Ko, D., Choi, S., Park, S., & Smolander, K. (2016). Where Is Current Research on Blockchain Technology?—A Systematic Review. *PLoS ONE*, *11*(6).

[10]. Saari, A., Junnila, S., & Vimpari, J. (2022). Blockchain's grand promise for the real estate sector: A systematic review. *Applied Sciences*, *12*(23), 11940-11964.

[11]. Gutierrez, O. C. U., & Xu, G. (2023). Blockchain and smart contracts to secure property transactions in smart cities. *In Applied Sciences*, *13*(1), 66-86.

[12]. Paankratov, E., Grigoryev, V., & Pankratov, O. (2020). The Blockchain technology in real estate sector: experience and prospects. *IOP Conf. Series: Materials Science and Engineering 869*, DOI: 10.1088/1757-899X/869/6/062010.

[13]. Graglia, J. M., & Mellon, C. (2018). Blockchain and property in 2018: At the end of the beginning. *Innovations: Technology, Governance, Globalization 12(2)*, 90-116.

[14]. Propy.(2019) *What is Blockchain Deed Registration in Real Estate?* Retrieved from: https://propy.com/browse/what-is-blockchain-deed-registration-in-real-estate/ [Accessed: 18 January 2023].
[15]. Ansah, B. O., Voss, W., Asiama, K. O., & (2023). Wuni, I. Y. A systematic review of the intuitional success factor for blockchain-based land administration. *Land Use Policy*, *125*, 106473.
[16]. Baliga, A. (2017). Understanding blockchain consensus models. *Persistent*, *4*(1), 14.
[17]. Bains, P. (2022). *Blockchain consensus mechanisms: A primer for supervisors*. International Monetary Fund.
[18]. Ciao, B., Zhenghui, Zh., Duquan, F., Shengli, Zh., Lei, Zh, Munge, & P. Yun. L. (2020). Performance analysis and comparison of PoW, PoS and DAG based blockchains. *In Digital Communications and Networks*, *6*(4), 480-485.
[19]. Ghag, O., & Chitre, V. (2019). Blockchain: Comparative analysis of public and private networks. *Journal of Merging Technologies and Innovative Research (JEITR)*, *6*(6), 13-20.
[20]. Bhanupriya, P., Gauni, S., Kalimuthu, K., & Menimegalai, C. T. (2021). A modified hybrid blockchain framework for secured data transaction. *Journal of Physics: Conference Series (ICACSE 2022)*, *1964*(4).
[21]. Matzut, R., Pennekamp, J., Buchholz, E., & Wehrle, K. (2022). Utilizing public blockchain for the sybil-resistant bootstrapping of distributed anonymity services". In *Proceedings of the 15th ACM ASIA Conference on Computer and Communication Security (ASIACC '20)*, Taipei, Taiwan, 531-542. Doi: 10.1145/3320269.3384729.
[22]. Zhou, H., Fard, A. M., & Makanju, A. (2022). The state of Ethereum smart contracts security: vulnerabilities, countermeasures, and tool support. *Journal of Cybersecurity and Privacy*, *2*(2), 358-378.