Blockchain-based Healthcare DApp with MetaMask Integration

Barnabás Kuglics
John von Neumann Faculty of Informatics
Obuda University
Budapest, Hungary
kubarni@stud.uni-obuda.hu

Kata Egres
John von Neumann Faculty of Informatics
Obuda University
Budapest, Hungary
egres.kata@nik.uni-obuda.hu

Miklós Sipos
John von Neumann Faculty of Informatics
Obuda University
Budapest, Hungary
sipos.miklos@nik.uni-obuda.hu

2024 IEEE 22nd Jubilee International Symposium on Intelligent Systems and Informatics (SISY) | 979-8-3503-8560-1/24/$31.00 ©2024 IEEE | DOI: 10.1109/SISY62279.2024.10737534

***Abstract*—The aim of this research is to develop a system that addresses the shortcomings and limitations of the current, widely used Hungarian Electronic Health Service Space. The engine of the system, presented below, is the Ethereum blockchain, which provides one of the most secure data storage options available today. The blockchain ensures data storage while protecting sensitive patient data from tampering and manipulation. Throughout the research, numerous systems were examined with similar functionalities and delved deeply into the world of blockchain during literature review. Although all planned components were developed during the implementation, it was not necessarily in the planned manner due to complications. The development was followed by thorough testing, and in the concluding section, a summary is provided, along with some possibilities for further development mentioned at the end of the research.**

***Index Terms*—blockchain, ethereum, healthcare, solidity, metamask, ganache**

I. Introduction

Bitcoin emerged in early 2009 [1], marking the beginning of a new economic and IT factor. With the birth of Bitcoin came the first cryptocurrency and the first blockchain, the latter being the focus of the project. Bitcoin is open source and its creator is unknown, although many articles attribute the creation of Bitcoin to Satoshi Nakamoto, but this is far from certain. Bitcoin is based on a peer-to-peer network, without the control or insight of any government or financial company.

After more than a decade, many blockchain-based digital currencies have been created. They have entered the public consciousness and people have started to engage with them. Today, cryptocurrencies can be traded by ordinary people in the same way that they are traded on the stock exchange. But the research does not approach this new technology from the cryptocurrency side. The task will be based on the technical possibilities of blockchain. Blockchain is a very interesting, new and complex technology that holds a lot of potential for us. It is far from being fully developed. Many of us believe, that there is huge potential in the world of blockchain.

Health systems are supposed to store patient data. 10 years ago, this was still done in a very outdated paper format. Patients' data was stored in cardboard boxes by GPs. Paper-based data storage harbours many dangers and inconveniences. The potential for data loss and leakage is huge and the transmission of data is time-consuming. Just think how easy it is to lose a piece of paper or how difficult it is to transfer it to another healthcare institution.

This system has been replaced by a software where the data is still stored locally on a hard drive (in many cases, actually only on a thumb drive). Data transfer was simplified thanks to electronic storage, but there was a risk of total loss of data if the hard disk failed.

Then, a few years ago, the Electronic Health Service Space (”Elektronikus Egészségügyi Szolgáltatási Tér”, in the followings: EESZT) [2] was created, a cloud-based service that allows data to be stored and shared. However, it still has some drawbacks. The whole technology is based on centralised data storage. Centralised data storage is far from secure, and in the event of a server failure, EESZT will go down. If someone hacks the server, patient and doctor data could fall into the wrong hands. The COVID19 pandemic caused so much problems in the world [3], and this platform (as many others [4], [5]) was also affected by the huge online traffic.

We would like to find solutions to the problems listed above. Blockchain is probably one of the most secure forms of data storage. Because blockchain is not run on a central server, but by a number of nodes, it is less susceptible to minor failures. The research's goal is to create a decentralised blockchain-based health data storage system that addresses the previous problems.

The preceding research [6], which was significantly influenced by another study in the same domain [7], on blockchain simulation environments provided a comprehensive foundation for understanding and reviewing the essential elements of a blockchain model. This foundational knowledge was

instrumental in the development of a practical application. The future integration of a transaction visualizer, as outlined in prior research [6], could further enhance this system. By providing a visual representation of transaction flows, the simulation allows users to comprehend the real-time processing and validation of votes within the blockchain network. This additional layer of transparency and traceability is crucial for fostering trust and understanding among participants.

II. Related work

To ground our current research within the existing body of knowledge, numerous studies related to our topic have been comprehensively analyzed. In this section, several key works that have significantly contributed to the field and influenced our approach are examined in detail.

A. A blockchain-based preserving and sharing system for medical data [8]

1) General overview: The system aims to solve three problems by combining the Internet of Thing (IoT) and blockchain. The first and perhaps most important problem is protecting patient data. Rather than storing data locally or in the cloud, the system uses a secure Hyperledger Fabric-based blockchain that is difficult to forge. Patients have little or no access to their data in traditional systems. The system offers a solution to this problem, allowing patients to easily track the status of their check-ups and their medical history. A proxy re-erncypiton algorithm is used to transmit data securely. A lot of useful information can be lost during surgery. The system is trying to address this problem by using IoT tools that eliminate the need to manually record all data. These tools automatically collect and record data throughout the entire surgery. And with this information, even post-operative recovery can be accelerated.

2) Participants: The participants in the scheme can be divided into four main parts. In order to meet the requirements of traditional health systems, it must include participants with administrator privileges. Both patients and physicians can be part of the data-producing group. Doctors make the diagnosis during the examination, while patients can upload it to the blockchain. The cloud server group includes everything related to data storage. Encrypted indicators are stored on the blockchain, while cloud servers store patient data. The last group is the group of data consumers. This includes primarily patients and doctors, but also health insurers, medical device companies and research institutions.

3) Security: Everyone in the system is authenticated and TLS certified. For security, data is transmitted via TLS. In addition to patient data, the system also encrypts the data needed to identify the patient, making it almost impossible to access sensitive information without the proper permissions. In addition, patient data is stored in the cloud, which is guaranteed by the cloud provider. And once downloaded from the cloud, we have the possibility to validate the data to make sure that it has not been manipulated in the meantime.

B. Medrec [9]

1) General overview: MedRec is an Ethereum-based private peer-to-peer network similar to our implementation, but the system does not store all data on the blockchain. MedRec is designed in such a way that only authentication and authorization is done on the blockchain, everything else is handled by a synchronization algorithm, so to speak ”off-chain”. This synchronisation algorithm is responsible for exchanging data between the patient’s database and the provider’s database. To accomplish this task, smart contracts are used to log patient-provider interactions, certain status transitions, viewing permissions. Since the actual patient data is stored in several local databases, the system needs to be somehow protected against data tampering. To solve this problem, a cryptographic extract of each record is stored on the blockchain. The provider can add new records to a patient’s medical history and the patient has the possibility to authorise them or set their visibility. The system has a notification function that informs both providers and patients when they receive a new record. Identification is done using a widely used DNA-like implementation, which requires the patient’s key details.

2) Smart contracts: The system identifies three types of contract. The first is the registration contract, the purpose of which is to manage patients new to the system. Only official institutions are entitled to register new patients. During the process, a unique identifiable string is stored in the Ethereum network, which works in a similar way to a public key. A patient-care contract is created between two nodes, where one node stores the data associated with the other node. The contract generates a pointer to the stored data, consisting of characters that point to the stored data. In order for the patient to have access to their health data, authorised providers must also be stored. The system database can be communicated with via SQL queries. The user can set the desired visibility levels through a graphical interface in an easy-to-understand way, and the system will generate the appropriate SQL command in the background. The aggregation contract is responsible for providing the history of the data. It contains more information about the relationships between patients and providers. Thanks to the summary contract, it is possible to store the data of patients even after they have been deleted, so that if they decide to return to the service later, their data will still be available. If a healthcare institution updates a patient’s details, the patient will be notified, as the aggregation contract will keep track of their status.

3) Summary: All in all, MedRec is a very well built system, which is only enhanced by the underlying Ethereum blockchain. Its user-friendly interface is accessible on both computers and mobile phones. The design and operation of the system components is well thought out. However, the system itself is only in a pilot version so far, so there are no clear results from the research. The system does not store all health data on the blockchain, only their indicators. This has led to significant performance gains, but the poorly protected local storage of sensitive health data remains a concern.

C. Lightweight Blockchain Simulation with Transaction Graph Visualizer [6]

The preceding study introduced a lightweight blockchain simulation and transaction graph visualization application, which has proven to be exceptionally advantageous for the ongoing research. Utilizing sophisticated data visualization techniques, this application offers a comprehensive and user-friendly platform for exploring and scrutinizing intricate transaction data within the blockchain. By integrating principles from graph theory, this tool enables the identification of crucial nodes within the blockchain, elucidating key participants, addresses, or entities involved in various transactions. This dual functionality, encompassing both blockchain simulation in C# and visualization in JavaScript, holds significant potential for augmenting the analysis of blockchain networks. It is poised to provide valuable insights for researchers, cryptocurrency enthusiasts, and cybersecurity professionals, thereby aiding in their efforts to understand and fortify blockchain ecosystems.

The insights and methodologies derived from the preceding research have established a robust foundation for comprehending and analyzing blockchain models. This knowledge has played a pivotal role in the development of a practical application in the current study, facilitating the implementation and testing of advanced blockchain concepts in real-world scenarios.

D. Legislations

Security is an extraordinary concern in this research, as there are strict regulations data, particularly healthcare data, that must be adhered to. The system must comply with the following laws, listed below. Furthermore, European Union members are subject to General Data Protection Regulation (GDPR) No. 2016/679.

- Act CXIII of 2011 on the Right of Informational Self-Determination [10]
- Act LXVI of 1992 on the Protection of Personal Data [11]
- Act XLVII of 1997 on the Registration of Addresses [12]
- Act L of 2013 on Electronic Information Security [13]

E. Possibilities for the Utilization of an Automatized, Electronic Blockchain-based, Students' Attendance Register, using a Universities' Modern Security Cameras [14]

1) General overview: One such study that has been examined and utilized in our research introduces an AI-based security camera system designed to automate the composition of a students' attendance register. This system alleviates administrative burdens by reducing paperwork, a solution not yet implemented at any University. To ensure the secure storage of data, blockchain technology is implemented, encrypting data and distributing it across various nodes, thereby enhancing security over traditional cloud-based storage systems.

2) Data storage on the blockchain: The use of blockchain for data storage in this study has been noted for its ability to provide greater security and data protection. The decentralization inherent in blockchain technology makes data less susceptible to corruption and unauthorized access, as files are fragmented and stored across multiple nodes globally, with no single node having access to the complete file. This study's approach to utilizing a proprietary university-based blockchain, the Universities Data Storage Chain (UDSC), for securely storing video recordings and photos for student identification and attendance tracking, has informed and influenced our own research methodology.

III. Methodology

Regarding the methodology, numerous aspects of the previously visited researches has been taken into consideration.

A. Planning

The first thing that greets the user when the program is started is the MVC application that handles authentication. This application checks the correctness of the login credentials provided by the user. It is important that the login credentials are always available, and since we are talking about highly sensitive data, even if encrypted, security is also a very important factor. For these reasons, we decided to entrust the storage of this data to Microsoft Azure. To do this it requires an SQL server running the SQL database.

The web application will essentially be the user interface. Doctors will be redirected to the main Web Application page by the MVC application after successful login. However, since the doctor will be constantly modifying the blockchain within transactions during the order, he/she will need to log into Metamask [15] first. Through Metamask, the doctors has the option to pay the various transaction fees. In effect, the system uses two-step identification, further increasing security.

The system is based on the Ethereum blockchain, which can be modified in the way described above thanks to smart contracts. A framework is needed to write, test and upload the appropriate smart contracts. In this case, this will be Node.js [16], which is a server-side JavaScript environment. It also makes it easy to use the Truffle framework [17], the Ganache developer network and the Web3.js library [18]. The full system design is shown on Figure 1.

B. Authentication

The first idea, as described in the Planning section, was to authenticate users using an MVC application. Based on our current knowledge at the time, this seemed like the best solution, as we knew we could properly and securely build a project that would perform this task. This solution was implemented during development, and after authentication, the DApp main page opened in a new browser page. However, we found more and more problems with this solution. On the one hand, they are two completely separate projects, which are difficult to reconcile during local operation, but would not have been workable later on. While the MVC project solved the authentication, it was easily circumvented by trying to access DApp via another link. Alignment of the two projects would have been necessary, but there was no adequate solution to the problem, so this solution was discarded.

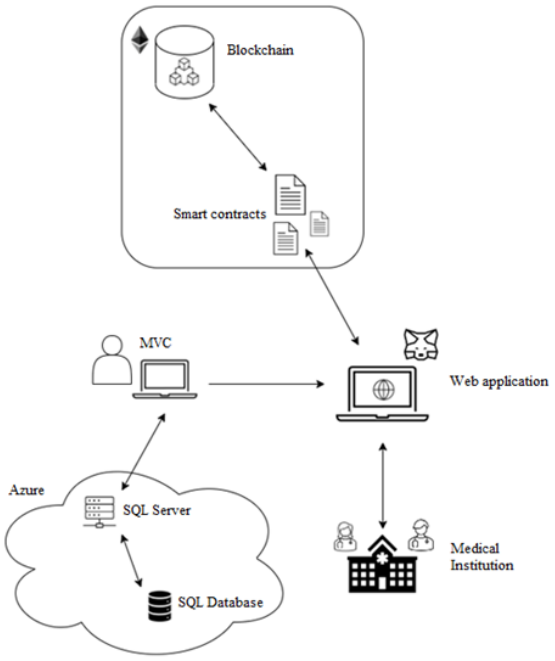

Fig. 1. *The high-level system design.*

After the ”failure” described above, we tried to implement this function without using other tools and projects. That's how this operation was included in one of the smart contracts. The current authentication is done in such a way that both doctors and pharmacists' public addresses and their corresponding passwords are stored in a data structure. The data structure was desgined to be searchable in a similar way to a hash table, so that on login we have the ability to retrieve the password of a given user with a runtime of ordo 1. The DApp works by extracting the user's public address from the MetaMask. So before you can enter the system, you have to enter MetaMask first. MetaMask also stores the user's public address in a similar way, so we only have to use the password we entered during the MetaMask registration [19]. Another advantage is that MetaMask can handle multiple wallets at the same time, so it's easy to switch users.

Obviously for security reasons, the DApp store passwords hashed within the blockchain. In order to avoid transmitting users' passwords as plain text, they are encrypted them on the client side. During the erncypiton process, a jsSHA object is created and parameterize it as shown in Figure 5. Thus, the password is encrypted using SHA-512 hashing, which produces a fixed-length hexadecimal value from a text input. The numRounds parameter is used to specify the number of hash iterations. Since the password is hashed and stored in the database, there is nothing to do but compare the client-side encrypted value with the stored value in the smart contract.

C. Sessions

After successful authentication in the smart contract on the client side, the DApp store the corresponding values in the browser local storage. If the authentication was unsuccessful a red error message appears on the page that either the password or the public address was incorrect during login. Key-value pairs can be stored in local and session storage. For optimal performance, it is not recommended to use the smart contract to identify the user every time, because it takes significantly more time than checking the key-value pairs stored on the client side. In implementing this task, both local and session storage was tried, testing the advantages and disadvantages of both.

Thanks to the Local Storage, users don't have to log in again if they accidentally close the page in the browser where the DApp is running. Thus, the management of the DApp session is the same as MetaMask. MetaMask also stores the user's data after login for the lifetime of the browser.

D. Distinction between doctors and pharmacists

The DApp also use Local Storage to distinguish whether the logged in user is a doctor or a pharmacist. In the smart contract, a function has implemented for doctors and a function for pharmacists, which checks their public address and the password they have entered. Depending on the return value of the operations, the corresponding key-value pair is set in Local Storage. While doctors can use all the features available in the app, pharmacists can only use some of the features they need to do their job. Accordingly, pharmacists are able to list patients on the main page of DApp and open all the tests for patients, so they can dispense the medicines prescribed by the doctor. What is distinctively different, however, is that, unlike doctors, pharmacists are not able to add new patients, nor are they able to record new test results for patients. If they try to perform such actions, the site redirects them back to the main page.

E. Developer network

In the search for a developer network, we finally decided to use Ganache [20], a private blockchain specifically for development. It provides a secure and deterministic environment throughout the development cycle, from development through deployment and testing. Ganache can be used by developers/testers in CMD thanks to a CLI version, but if one prefers the UI, a desktop application is also available. In the end, the desktop application was selected because all the features of the CLI version are available, but we think it is much more visual and easier to use. The only downside is that the CLI version has better performance, the UI puts a significant load on the developer/tester's computer.

Ganache is an RPC server, which in our case is running at 127.0.0.1:7545. The DApp can reach it through this address and install the contracts on the blockchain through it. The server uses JSON-RPC protocol, which allows communication between the application and the server in JSON format.

According to Ganache's documentation, it is a "one click blockchain", a claim that has been proven to be correct during development. After downloading and installing the desktop application, you have two options on the main screen. "Quick start" offers us a completely new, untouched blockchain each time. This has the advantage, for example, that every time you start, the maximum amount of money will be in the wallet of the 10 Ethereum accounts provided by Ganache, but after every start you will have to migrate to install your DApp. We made our own Workspace the - local-blockchain (ethereum) - at the beginning of development and worked in it, because this avoided a permanent pre-development migration. Ganache's view can be seen on Figure 2 and on Figure 3, showing the transactions and the blocks.

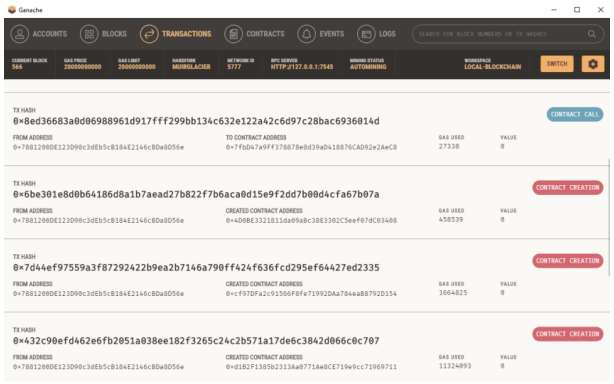

Fig. 2. Transactions visible in Ganache.

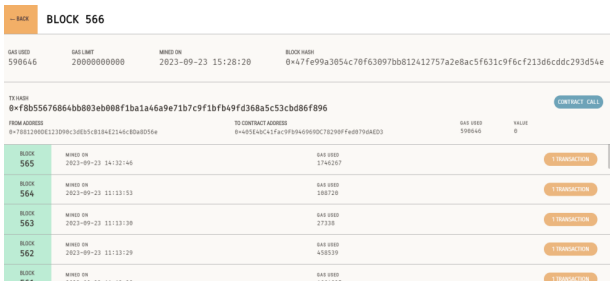

Fig. 3. Blocks visible in Ganache.

F. Smart contract deployment cost

During the development, four smart contracts were created, whose names are shown in Table I. In the table, the contracts are arranged in order of complexity, from the simplest to the most complex. Since the cost of deploying the contract is directly proportional to the complexity, it can be observed that while Migration.sol costs only about 5000 HUF, my most complex contract reaches up to 290,000 HUF. The increasing cost, due to increasing complexity, can easily be exponential, as Figure 4 shows. The Migration.sol contract merely describes the logic of migration. It is a mandatory element of every DApp, responsible for managing migration during deployment. The Find.sol contract defines the structure of a medical examination/report. It stores the patient's complaints, the disease diagnosed by the doctor, the patient's complaints, and the creation date. Additionally, methods necessary for querying and setting are defined in the contract. The Patient.sol is intended for storing the necessary data of the patient and the examinations taken. Currently, the patient's name, date of birth, mother's name, social security number, gender, residence, and tax number can be stored. In addition to storing and querying this data, I implemented the necessary functions in this contract. In the final and largest contract, Healthsystem.sol, three data structures can be found: patients, doctors, and pharmacists. In addition to their extensibility, this contract implements querying of patient data and querying of associated test results. Furthermore, the logic for authenticating doctors and pharmacists is also implemented in this smart contract. Calculations were created at May of 2024, using Coinbase converter. [21]

TABLE I
DEPLOYMENT COSTS REGARDING EACH SMART CONTRACT.

Name of SC	Cost of deployment (ETH)	Cost of deployment (HUF)
Migration.sol	0,00342	4 740,25
Find.sol	0,00825	11 434,82
Patient.sol	0,02996	41 525,73
Healthsystem.sol	0,20376	282 419,32

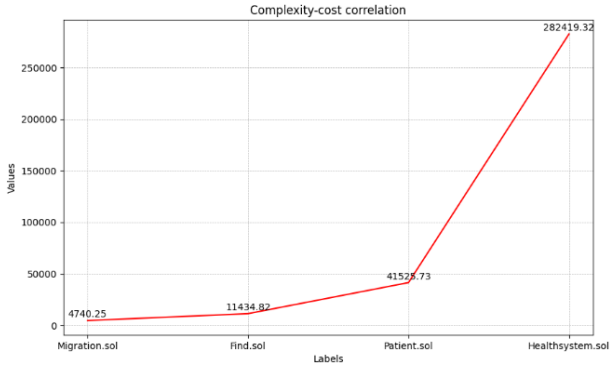

Fig. 4. Deployment cost plot, displaying HUF values.

IV. Evaluation

A. Application-wise evaluation

Regarding the initial objectives, the results show that all functions were implemented as planned. As Figure 5 and Figure 6 shows, the patients findings can be listed, as well as creating new patients which will be stored securely on the blockchain.

B. Performance-wise evaluation

When evaluating the performance of the completed system, we encountered two main difficulties. Firstly, since the cost of deploying the project exceeds the amount we would have spent to upload the DApp to the main blockchain for testing purposes only, we cannot test the performance on it. We have looked for other options where it is cheaper to deploy, but these

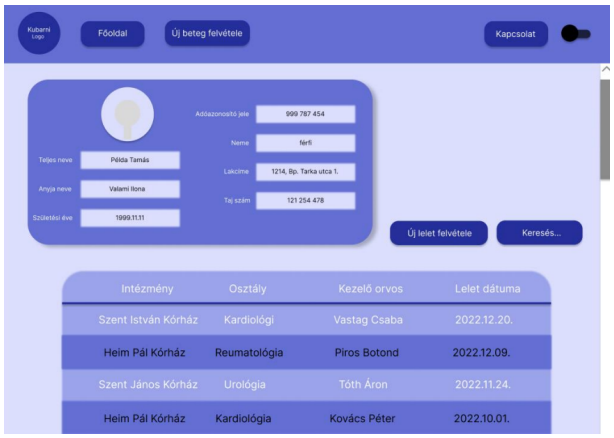

Fig. 5. The listing view of patient findings.

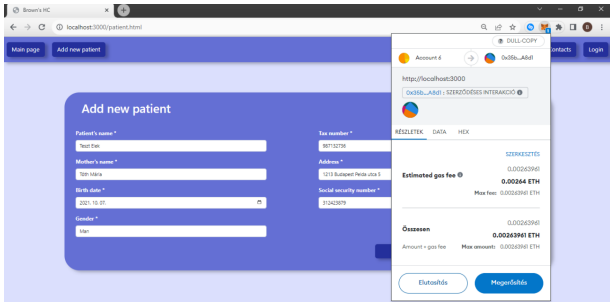

Fig. 6. Creating a new patient, confirming the blockchain transaction using Metamask.

test networks, although similar in operation to the main, public Ethereum blockchain, are not comparable in performance.

The other obstacle is that it is not possible to measure the performance of DApps in the same way as traditional systems, as their operation is also significantly different from that of an application running in the cloud, for example. For these reasons, in this chapter, rather than examining the performance of our system, the focus will be on the performance of EVM in general, the network, and for comparison, some of the larger DApps deployed on the Ethereum network.

Basically, the EVM is a virtual machine that runs the Ethereum protocol, allowing us to deploy smart contracts on the blockchain, but also has the additional task of processing transactions, for example. The EVM is not just a computer, but a pool of thousands of machines hosted in the cloud worldwide.

The performance of the EVM, or network, can be broken down into two main parts. One is reading data from the blockchain. The performance of this is roughly comparable to that of, for example, a database running in the cloud. Reading is fast thanks to the redundantly stored data. The nodes store all or part of the blockchain, serving the user in case of a read task. The final performance can be affected by the speed of the internet and the data storage of the node (e.g. SSD, HDD).

The more complex part is writing the data to the blockchain. As there are countless DApps available on the Ethereum blockchain, the resources of the blockchain are shared between them. When a DApp writes to the blockchain, a new block is created, which is created by a miner. This is considered a resource-intensive task. Basically, the speed of the Internet can be a slowing factor here, but in general, the most strongly influencing factor is the Gas fee. The Gas fee was defined before, but in short, this fee is paid to the miners for the block created. The higher the Gas price we set for our block the faster a miner will complete the task. But obviously we don't want to set too high price either, to avoid our system being too expensive to run. So, it is important to find the right balance in setting the Gas tariff. Since the completed system is not deployed on a public network but only runs on a local test network, there was not possible to measure the actual time to create a block, since block mining on the Ganache test network is instantaneous. In general, however, it takes a few seconds on the public network, depending on the load.

To prove that the system would work on a public network, even in live operation, there are some examples of other DApps that handle heavy traffic [22].

KAI-CHING is currently the program of most interest, deployed on an Ethereum blockchain. It is a Chinese-developed DApp that has recently appeared on the market. The program allows you to pay at a discount in designated shops and to personalize your phone's lock screen. The DApp is used by roughly 664,000 UAW [1] per day, with around 683,000 transactions between wallets.

The second most popular app is a game called Alien Worlds, which is generating staggering numbers. It has only 130,000 active users per day, but they make approximately 6.34 million transactions in 24 hours.

The third on the list is also a game with 100,000 UAW and more than 1.36 million transactions per day. Due to the limited options, although we was not able to test the performance of the DApp, seeing the numbers that these applications produce when running smoothly suggests that the project would be able to run without problems. Unfortunately, an approximate number was not found of how many medical examinations take place in Hungary in a day, so it is not possible to predict how much traffic the system would handle in 24 hours, but given the population of Hungary, we highly doubt that it would even approach 1.36 million transactions per day.

V. Conclusion

A. General overview

Using the possibilities offered by blockchain, we have succeeded in creating a working, secure healthcare system that, unlike today's systems, works in a way that prevents tampering and modification. In this research the design and development processes were described in detail, highlighting the blockchain technology used and the benefits it brings. Many obstacles were encountered during the research, which were overcome with additional research and investigation.

[1] Unique Active Wallets

B. Design and Development

The basic concept of the task has not changed significantly from the one described in the Planning section. During the development we was introduced to the Solidity programming language, which was very challenging, and the Ganache development environment, which helped us a lot.

C. Testing

Once the development was completed, there was a strong focus on testing, as smart contracts are expensive to deploy. During testing, all the functionality was tried to be covered and achieve 100% code coverage. With the unit and integration tests, all major methods and operations were tested. Although we had intended to create automated UI tests during the design phase, later it was realized that the complexity of the functionality did not necessarily require this, so comprehensive manual tests were created instead.

References

[1] Krisztián Bálint et al. "Connecting Bitcoin Blockchain with Digital Learning Chain Structure in Education". In: *Acta Polytechnica Hungarica* 16 (Jan. 2019), pp. 2019–77. DOI: 10.12700/APH.16.1.2019.1.5.

[2] Belügyminisztérium Elektronikus Egészségügyi Szolgáltatási Tér. *EESZT lakossági portál*. https://www.eeszt.gov.hu/hu/nyito-oldal [Accessed: 2024-05-25].

[3] Mónika Garai-Fodor. "The Impact of the Coronavirus on Competence, from a Generation-Specific Perspective". In: *Acta Polytechnica Hungarica* 19 (Jan. 2022), pp. 111–125. DOI: 10.12700/APH.19.8.2022.8.7.

[4] Klaudia Ivancova, Martin Sarnovsky, and Viera Krešňáková. "Fake news detection in Slovak language using deep learning techniques". In: Jan. 2021, pp. 000255–000260. DOI: 10.1109/SAMI50585.2021.9378650.

[5] Vladimir Zah et al. "Paying for Digital Health Interventions – What Evidence is Needed?" In: *Acta Polytechnica Hungarica* 19 (Nov. 2022). DOI: 10.12700/APH.19.9.2022.9.10.

[6] Miklós Sipos and Sándor Szénási. "Lightweight Blockchain Simulation with Transaction Graph Visualizer". In: *2023 IEEE 23rd International Symposium on Computational Intelligence and Informatics (CINTI)*. 2023, pp. 31–36. DOI: 10.1109/CINTI59972.2023.10382093.

[7] Carlos Faria and Miguel Correia. "BlockSim: Blockchain Simulator". In: *2019 IEEE International Conference on Blockchain (Blockchain)*. 2019, pp. 439–446. DOI: 10.1109/Blockchain.2019.00067.

[8] Zeng Chen et al. "A blockchain-based preserving and sharing system for medical data privacy". In: *Future Generation Computer Systems* 124 (2021), pp. 338–350. ISSN: 0167-739X. DOI: https://doi.org/10.1016/j.future.2021.05.023. URL: https://www.sciencedirect.com/science/article/pii/S0167739X21001734.

[9] Asaph Azaria et al. "MedRec: Using Blockchain for Medical Data Access and Permission Management". In: *2016 2nd International Conference on Open and Big Data (OBD)*. 2016, pp. 25–30. DOI: 10.1109/OBD.2016.11.

[10] Nemzeti Adatvédelmi és Információszabadság Hatóság. *2011. évi CXIII. törvény információs önrendelkezés joga*. https://naih.hu/files/Infotv-2015_08.pdf/ [Accessed: 2024-05-27].

[11] Jogtár. *1992. évi LXVI. törvény személyes adatok védelme*. https://net.jogtar.hu/jogszabaly?docid=99200066.tv/ [Accessed: 2024-05-27].

[12] Jogtár. *1997. évi XLVII. törvény lakcímek nyilvántartása*. https://net.jogtar.hu/jogszabaly?docid=99700047.tv/ [Accessed: 2024-05-27].

[13] Jogtár. *2013. évi L. törvény elektronikus információbiztonság*. https://net.jogtar.hu/jogszabaly?docid=a1300050.tv/ [Accessed: 2024-05-27].

[14] Krisztián Bálint. "Possibilities for the Utilization of an Automatized, Electronic Blockchain-based, Students' Attendance Register, using a Universities' Modern Security Cameras". In: *Acta Polytechnica Hungarica* 18 (Jan. 2021), pp. 127–142. DOI: 10.12700/APH.18.2.2021.2.7.

[15] Wei-Meng Lee. "Using the MetaMask Chrome Extension". In: Sept. 2019, pp. 93–126. ISBN: 978-1-4842-5085-3. DOI: 10.1007/978-1-4842-5086-0_5.

[16] NodeJS. *Node.js documentation*. https://nodejs.org/en/docs/ [Accessed: 2023-03-07].

[17] Truffle. *Truffle Suite documentation*. https://archive.trufflesuite.com/docs/ [Accessed: 2023-07-10].

[18] Nethereum Documentation. *Web3*. https://docs.nethereum.com/en/latest/introduction/web3/ [Accessed: 2024-05-27].

[19] Wei-Meng Lee. "Using the MetaMask Crypto-Wallet". In: Apr. 2023, pp. 111–144. ISBN: 978-1-4842-9270-9. DOI: 10.1007/978-1-4842-9271-6_5.

[20] Ganache. *Ganache documentation*. https://trufflesuite.com/docs/ganache/ [Accessed: 2023-10-22].

[21] Coinbase. *Coinbase ETH/HUF converter*. https://www.coinbase.com/converter/eth/huf [Accessed: 2024-05-25].

[22] DappRadar. *Top Blockchain Dapps*. https://dappradar.com/rankings/ [Accessed: 2024-04-11].