Software Based on Blockchain Technology for Consolidation the Medical Data about the Patients Examination

Andriy Pukas, Vitalii Smal, Vadym Zabchuk

Department of Computer Science, Ternopil National Economic University, UKRAINE, Ternopil, 8 Chekhova str.,
email: apu@tneu.edu.ua, vitalii.smal@gmail.com, vadzab5@gmail.com

Abstract: **Software architecture based on blockchain technology for efficient medical records exchanging in safe mode, data confidentiality protecting, and giving to patients more control over their medical information is described in this paper. New software system architecture was proposed to consolidate medical data on patient screening based on blockchain technology and decentralization principles, which increased the level of data access security in heterogeneous medical systems.**

Keywords: **software, data consolidation, blockchain, patient examination.**

I. Introduction

Hospital-based medical information systems are becoming increasingly popular worldwide. Paper medical cards are gradually losing their relevance, being replaced by hospital-based medical information systems. Generally doctors are used them and already know how to use them properly. Such systems have many advantages. First, the electronic card will never get lost, and the patient will not be able to take it home. Thus, the information is always kept in medical facility. The second advantage of an e-card is that there is no need to look for it to be later handed over to an appropriate specialist by the receptionist. All data is always available on a doctor's computer. Another advantage is that it eliminates the need for a permanent patching of additional sheets, advisory opinions, analyses and research results. All such information is recorded into the specified fields of the program, which gives the necessary information at doctor's first request. It also enables several specialists of the clinic to simultaneously get access to patient's electronic card contents. It makes possible for doctors not only to simultaneously read the patient's card, but also to fill it. This feature significantly optimizes the medical staff activities.

Despite the advantages, they also have drawbacks. They are very fragmented and decentralized. Another problem is that such systems are not unified and different medical institutions use different means of electronic data storage. This significantly complicates the transfer of such data between healthcare facilities. Consider a case when a patient needs to show their disease history to a doctor of another medical establishment. To do this, the patient has to make a disease history request and wait for it to be processed. It can take a long time, and in some situations, even a few hours can significantly affect the person's health or even life.

Production engineers and healthcare professionals [1, 2] worldwide see blockchain technologies [3] as a way to efficiently exchange medical records in safe mode, protect the data privacy from hackers, and give patients more control over their information.

The development of a blockchain-based system that would consolidate patient medical data from different providers should address such issues as fragmentary and slow access to patient medical data, incompatibility of medical information systems, and improve the quality and quantity of data for medical research.

II. Task Statement

The aim of research is to consolidate medical data on patient surveys from various medical institutions and increase the level of access security to such data, based on creation a prototype of software system built on principles of decentralization and blockchain architecture.

Research objectives:
- analysis of modern electronic medical systems;
- design a secure and compatible software system for the medical data consolidation;
- implementation a prototype of software system based on blockchain technology.

It is easy to assume that when a doctor examines a patient or gives them a new prescription, the patient will agree to add a reference or "index" to blockchain - a decentralized electronic system similar to the bitcoin [4]. But instead of making payments, this block chain will write medical information in a cryptographic database supported by a network of computers that is accessible to anyone who works with it. Each index that the physician will add to the journal will become part of the patient's registry, regardless of which electronic system the doctor used. Therefore, any other doctor will be able to use it without worrying about the issues of incompatibility.

III. Analysis of Medical Information Systems in Ukraine

Electronic medicine is a development synthesis of medical and information technologies. This trend consists of many areas: from the creation of open digital registries of patients to their remote treatment.

The following are major factors in the field of electronic medicine development:

· introduction of automated informational sectoral systems, which, in particular, will enable the transition to electronic processing of medical documentation;
· development of telemedicine;
· improvement of the national health monitoring system;
· creation and implementation of new computer technologies of disease prevention, diagnostics, medical process support;
· creation of publicly available electronic medical resources;
· development of self-diagnostic methods and the construction of a personal health paradigm by e-medicine methods.

Information technology today can make medicine more affordable and effective. Many electronic systems exist on the Ukrainian market: Helsi, MIS EMSIMED, Doctor Eleks, MEDSTAR, MEDICS, Queue-Free Clinic etc [5].

All of the above systems are good solutions that are already implemented and work successfully in health care facilities. However, lack of properly established connection between them appears to be their main disadvantage. Each health care facility has its own system, so when a patient is treated in two or more different medical facilities, that person's analyses, examinations, and histories of illness are scattered across different medical systems. The patient does not have a single safe place, protected from hackers, to store their medical data, consolidated from different sources.

Therefore, having analyzed the Ukrainian medical systems market, we found the need for a software system that could consolidate medical data from various electronic medical systems in Ukraine and meet requirements such as reliability, security, and intuitive interface.

IV. Architecture of Software System

A software system for consolidating medical data is based on Ethereum [6], a platform for creating decentralized online blockchain services based on intelligent contracts [7]. Intelligent contracts are scripts that simplify, verify, ensure negotiations, execution of a contract, or check unwanted clauses of the agreement. Intelligent contracts, as a rule, also have a user interface and often follow the logic of contractual provisions. Thus, intelligent contracts allow for more complex blockchain transactions. Ethereum consists of a system of nodes (personal computers, clusters, virtual machines) in a decentralized network. Smart contracts are not a substitute for contracts in the traditional sense, but act as agreements on the implementation of certain actions or code. In this case, these contracts can be used to encode a set of indexes to medical data placements.

Lets consider the structure of smart contracts. The proposed system does not store electronic medical cards directly on Ethereum, but instead uses a relational set of smart contracts to encode indexes that can be used to locate and authenticate to the medical cards storage point. The system identifies three main types of contracts owned by patients, suppliers and other consumers. Namely:
· Registration contract;
· Patient-provider contract;
· Final contract.

The registration contract reflects the accordance of the participants' identifiers (patients, providers) with the Ethereum address (equivalent to the public key). The regulation of new identities can be encoded in the contract, ensuring that only certified institutions can add new information to the blockchain. In turn, new information about the patient (for example, about new relationships) is added only with the approval of this patient. Each identification string is located at its blockchain address, where it is referred to by the final contract.

A contract on relationship between the patient and the provider is concluded between the two nodes of the system, where one node stores and manages medical records for another. Although we consider the case of a patient getting medical care at the healthcare institution, this concept applies to any pairwise interaction with data support. The patient-provider relationship defines the range of data indexes and associated access permissions that identify the records stored by the care provider. Each index consists of a query string, which, when executed in the supplier database, returns a subset of the patient data. The request string/line is embedded in the hash of this data subset to ensure that the data has not been changed in the source. Additional information shows where one can get access to the provider's database in the network, for example, the host name and port in the standard network topology. Data requests and related information are developed by the care provider and modified when new records are added. To allow patients to share records with other users, the dictionary implementation (hash table) displays the addresses of users of the list of additional request lines. Each line may indicate the portion of the patient's data to which the third-party user has access permission.

A prototype created demonstrates this design with SQL data queries. In a simple situation, the supplier refers to the patient's data by a simple SELECT request, based on the patient's address. Patients can use a tool that allows them to check the fields they want to share through the developed graphical interface. The system formulates the corresponding SQL queries and downloads them to the patient-provider relationship contract in a particular block. It's worth noting that, using common lines, the system can closely interact with any database implementation. Consequently, the prototype can be conveniently integrated with the existing infrastructure for data storage of the provider. At the same time, patients engage their micro level control of access to their medical records, thus choosing any part they want to share.

The final contract is comparable to the bread crumble trail, where each participant can find a summary of their relationship with any other participant. The final contract encodes the list of links to contracts on relations between patients and suppliers, providing for both current and previous interactions with other nodes of the system. Each relationship also stores the "status" variable indicating when the relationship was established and whether it was approved by the patient. Acceptance, rejection or removal of the relationship is controlled by the patient, giving full control over all the records in their history that they want to acknowledge. This function of the system is the key to satisfying its convenience criterion: an index to fragmented records is made in a single dedicated location.

It is shown in Fig. 1 the possible connections between different contracts and between customers and suppliers. It is

worth noting that the variable of the status of a particular contract may have different values depending on the permissions that it permits. Contracts are also used only for indexes: database requests that return records are processed off-line.

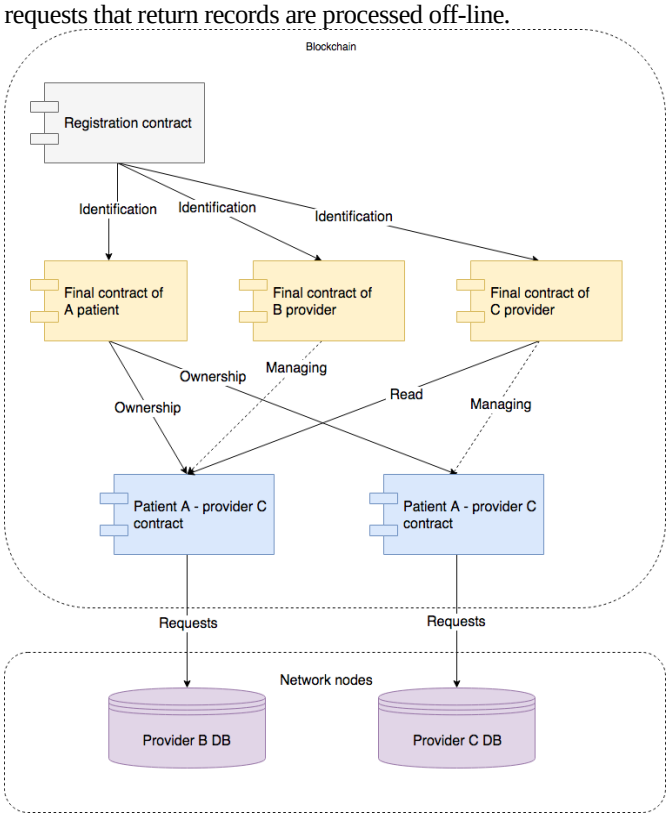

Fig. 1. Structure of patient-provider contract

Lets consider data processing in the system. A patient requests access to certain medical records by sending a request to the data provider that is part of the infrastructure outside the network of the system being developed. The data provider implements the interface of access to the local database of the patient node, which is governed by the rights that are stored in a flowchart. It runs a server that listens to requests that are cryptographically signed by the issuer from clients in the network. The cryptographic signature allows the gatekeeper to confirm the identity, and then checks blocking contracts to check if the requesting address is allowed to access the request. If the address is valid, it executes the request in the local database of the node and returns the result to the client.

It is assumed that many nodes of providers, especially those associated with service providers, already store data on networked servers with a high degree of security. The system also defines a modular protocol of interoperability that can interact with any application and user interface. Patient nodes also contain a local database, albeit more "light", which functions as a cache of patient data. The patient's node is a "light" node that can be executed on a PC or mobile phone.

Blockchain technologies introduce a number of confidentiality restrictions, some of which are alleviated by the use of the permitted read access structure and a private flowchart. The key issue is that even without a direct disclosure of the patient's name, the conclusion about who the particular patient is, can be made based on the metadata of one Ehereum address with several others. Even with a private blockchain, there is a strong need to consider retrieving nodes that handle these sensitive metadata. One of the solutions is to require full permission from all the retrievers, and to require medical researchers working at retrieving centers only to provide secure systems.

The proposed solution to the problem of confidentiality is to use a system of "delegated contracts", where each provider creates separate Ethereum identifiers for each new relationship between patient providers. This means that instead of a single provider's address, from which the relations with particular patients can be easily obtained, the provider's identifier is distributed throughout the network. However, in order for the relationship to be safely established, the provider should not be able to add a new block containing this new address (since it would be easy to track each of these delegate addresses to the original). Therefore, when creating a new delegate account, the provider performs an arbitrary transaction with another verified provider by giving them details of the new delegate account that they can add as confirmed account information to the blockchain.

Security. The decentralized nature of the block-based systems gives the system the advantage of reliability both in authorization data support (stored in each node of the network) and in the repositories themselves (stored both by the patient and the corresponding provider node). With many organizations involved in the system, consensus mechanisms are also present to avoid separate points of failure. Since medical data and a global authorization log are distributed, there is no central goal for attacks or failures, and the network is intervention safe (since the modified node will conflict with other, unchanged nodes, thus making consensus impossible).

This system does not attempt to resolve security problems at the provider's database level (which must be duly managed by an IT service administrator), nor does it solve the security of the endpoint (a patient's compromised computer may potentially allow data theft).

Scalability is a constant concern in the Ethereum community and has not yet been resolved. One of the key issues is that any event stored at any time in a flowchart will appear in each subsequent block. Although this is also a feature of Bitcoin blockchain, since Ethereum provides both data storage and more complex operations, the effects of this growth are a big issue.

To integrate with the existing infrastructure of the electronic healthcare management system and records, it is necessary to design components of system nodes. Assume that many nodes and service providers in particular, are already trustingly managing databases with patient data stored on servers with network connections. The proposed system consists of four software components: a server unit library, an Ethereum client, a database gatekeeper and a medical records manager. They can be executed on servers, being united to create a consistent, distributed system. A prototype for implementing these components, which are integrated with the SQLite database and managed through a designed web-based user interface, is also offered. It should be noted that any implementation of the firewall and the user interface can participate in the system, using the module interaction protocol, defined through block diagrams.

Patient nodes in the proposed system contain the same basic components as suppliers. Their implementation can be done on a local computer or mobile phone. Their local database can be one of many light database implementations. Databases can function simply as a cache of patient medical data. Missing data can be obtained online at any time, following the final agreement of the center.

It is proposed to create server part library containing several utilities to facilitate the operation of the system. The library represents a connection to the blockchain and exports the API function. Management recording programs and their user interfaces, thus, can avoid interference with their direct work with the blockchain. One such obstacle is checking that each transaction sent is accepted with high trust from the network. The developed library automatically processes the indeterminacy when the transactions are retrieved and examines the cases when they are rejected. The backend library interacts with the Ethereum client to implement low-level formatting and analysis of the Ethereum protocol.

By using the blockchain registration contract, the patient's identifier first turns into the corresponding Ethereum address and the corresponding final contract is located. Then the provider downloads a new patient-provider relationship flowchart, indicating how they control the data belonging to the patient's Ethereum address. Then the provider's node creates a request for a link to this data and accordingly updates the patient-provider contract. Finally, the node sends a transaction that connects a new patient-provider contract with a patient's final contract, allowing the patient's nodes to later find it in a specific block.

Ethereum client implements the full functionality required for joining and participating in the Ethereum network. The client processes a wide range of tasks, such as: peer-to-peer network connection, encoding and sending transactions, and maintaining a verified local copy of the block template. The client has to be changed so that it enables the mapping of identity and addresses. Then the service is implement to find the final contract of the node, by means of a register address search of the contract with the recorder. This service is constantly working within the client for monitoring real-time changes to the final contract. In the case of an update, the service signals that the medical data manager issues a user's message and, if necessary, synchronizes the local database. Modified Ethereum client constantly monitors its final contract. When a new block is retrieved from a newly-contracted patient-provider, the client makes a signal that results in a user's message. Then the user can confirm or refuse to communicate with the provider, accordingly updating the General Contract. If the message is accepted, then the implementation of the prototype automatically issues a request for new medical data. The client uses the information in a new patient-provider contract to locate the provider on the network and connect to its server's database gatekeeper.

The database gatekeeper implements an off-network access interface for the local node database, which is managed by the rights stored in blockchain. Gatekeeper launches a server that listens to requests from customers in the network. The request contains a request line/string, as well as a reference to the patient-provider contract, which requires permissions to run it. Request is cryptographically signed by the issuer, which allows the gatekeeper to confirm its identity. Upon confirmation of the issuer's signature, the gatekeeper checks the contracts to determine whether access to the request is allowed on the requesting address. If the address is available, it executes the request in the local database of the node and returns the result to the client.

It should be noted that the created components in the same way support the receipt of patient data by third-party: the patient chooses the data to be sent and updates the corresponding patient-provider interaction contract with a third-party address and request line. If necessary, the patient's node may allow a third-party address using a registration contract. Then, the patient node connects an existing patient-provider contract with a third-party modifying provider. The third party is automatically notified of new permissions and can follow the link to find all the information needed. The gatekeeper of the provider's database will allow access to such a request, confirming that it was issued by the patient in a patient-provider shared user contract.

The medical data management system combines all of the software components mentioned above and the user interface. The program provides data from local SQLite databases (intended for interchange with other database software) to view and provides updates to users, as well as sharing and receiving data. The created user interface offers an intuitive, clear and informative design. The developed software system is conveniently accessible through a web interface built using JavaScript and AngularJS framework. Its compatibility with mobile devices is of particular note, since modern users expect easy and high-quality access from anywhere.

"Retrievers/(miners)" are encouraged to participate in the network and provide their computational resources to achieve a credible and gradual chain promotion. A model is proposed that embraces the medical community in the area of network management - the system developed involves health researchers and health stakeholders in their network. In turn, providers and patients give access to aggregated, anonymous medical data as a reward for retrieving/(mining). This idea is investigated in the developed prototype by introducing a special function in the patient-provider relationship contract. This requires providers to attach a request to any transaction they send by updating the patient-provider contract.

For example, this remuneration request can be arranged to return the average iron levels in the blood test done by the provider to all patients in the previous week. When the block containing the record-update operation is retrieved, the retrieving function automatically adds the block retriever as the request owner. The retriever can then collect it by simply sending a request for this reward to the provider's database gatekeeper. Since the unit is signed by the provider as part of the transaction, the remuneration request is safe from harmful changes. This "remuneration request" or retrieving/mining reward allows medical researchers to access data on medical treatment and health care outcomes at the community level. It is anticipated that future upgrades of the retrieving/mining model, where retrievers/miners can indicate the benefits for demographic groups and the peculiarities of the data which they seek, in order to provide accurate medicine and targeted research (while maintaining the confidentiality of patients).

V. Software System Implementation

To implement the server part, the C# programming language and the .NET Framework were used together with connection of Ethereum platform modules. To implement the web client, it is used JavaScript programming language and AngularJS framework. The web site interface is adapted to mobile devices.

To begin, the user logs in to the system by sending photos of identification code and the first pages of the passport for verification. After verifying the data, the user creates a password, where the identification number acts as the password (Fig. 2).

By logging in with the ID number and password, the user is directed to the home page of the website (Fig. 3).

Here the history of all user's medical records, consolidated from various medical institutions, is presented.

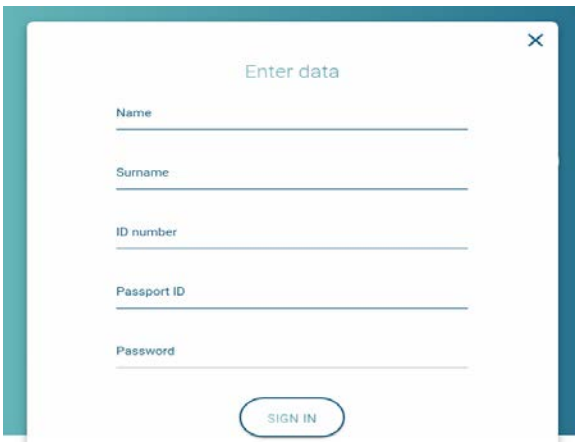

Fig. 2. New patient registration page

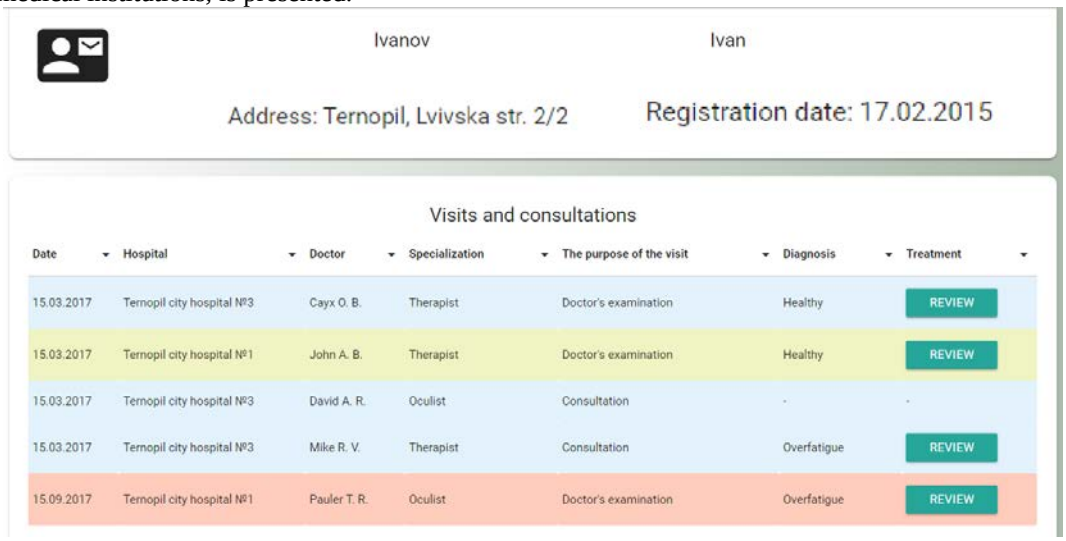

Fig. 3. Patient home page

When a medical institution wants to make a new record in history, notification is sent to the user, who can view the changes.

VI. Conclusion

The prototype of a safe compatible system of medical data management based on Blockchain technology, which aggregates patient medical data from different providers, medical institutions' information systems, is developed. The system solves problems such as fragmented and slow access to patient medical data, incompatibility of medical information systems and improves the quality and quantity of data for medical research. Using intelligent Ethereum contracts to organize a content access system on separate sites for storage and provision of services, the authentication log determines access to medical records by providing patients with a comprehensive overview and data exchange. An innovative approach to integration with existing provider systems is demonstrated, identifying the priority of the open APIs and the transparency of the network structure.

References

[1] Xia, Q.a, Sifah, E.B.b, Asamoah, K.O.b, Gao, J.c, Du, X.d, Guizani, M.e. Article "MeDShare: Trust-Less Medical Data Sharing among Cloud Service Providers via Blockchain" *IEEE*, 22 July 2017, pp. 14757-14767.

[2] Roehrs, A., da Costa, C.A. Author, da Rosa Righi, R. "OmniPHR: A distributed architecture model to integrate personal health records", *Journal of Biomedical Informatics*, July 2017, pp. 70-81.

[3]. Zyskind, Guy, and Oz Nathan. "Decentralizing privacy: Using blockchain to protect personal data", *Security and Privacy Workshops (SPW)*, 2015, IEEE, pp. 180-184.

[4] Nakamoto, Satoshi. "Bitcoin: A peer-to-peer electronic cash system", 2008.

[5]. Kachmar V.O., "Medical Information systems – the state of development in Ukraine", *Ukrainian journal of telemedicine and medical telematics*, 2010, Vol.8, №1, p.12-17.

[6]. Wood, Gavin. "Ethereum: A secure decentralised generalised transaction ledger", *Ethereum Project Yellow Paper*, 2014.

[7]. Croman, Kyle, Christian Decker, Ittay Eyal, Adem Efe Gencer, Ari Juels, Ahmed Kosba, Andrew Miller, Prateek Saxena, Elaine Shi, and Emin Gün. "On scaling decentralized blockchains", *Financial Cryptography and Data Security. FC 2016*, pp. 106-125, 2016.