

Electronic Expertise Using Distributed Ledger Technology

Oleg Iakushkin(✉), Alexey Pismerov, Vladislav Popov, Anastasia Berlina, and Vladimir Korkhov

Saint-Petersburg University,
7/9 Universitetskaya nab., St. Petersburg 199034, Russia
o.yakushkin@spbu.ru

Abstract. This paper examines how to implement electronic expertise based on the Corda platform. The paper aims to implement such a system, taking as an example the organization of the online contest. A draft solution is created for this task, which demonstrates the necessary tools to solve it and key capabilities that will be required from the platform for the implementation of the electronic expertise system. It justifies the choice of the Corda platform and describes the solution with its use. The main advantages and disadvantages of this approach to the implementation of electronic expertise when using Blockchain are highlighted.

Keywords: Distributed ledger · Corda · Expertise · Grading system

1 Introduction

Systems using peer review have application in many areas of activity, but so far a significant part of them is somehow not computerized and difficult to control. The reasons for this are both attempts at external interference in the rating process (hacker attacks [4], the substitution of evaluated works) and the presence of compromised system participants (for example, the opinion of one or more members of the expert commission cannot be considered independent). There is a growing need for an electronic system to neutralize these potentially dangerous factors. The project, which is the object of this research study, aims to introduce transparency and manageability into the examination process by using technologies based on distributed registries, such as Blockchain.

A blockchain is essentially a distributed record base or publicly accessible transaction ledger [3]. Each transaction in the ledger is confirmed by the consensus [5] of most of the participants in the system, and besides the added information cannot be deleted. The blockchain contains a defined and confirmed record for each transaction ever completed. The essence of the blockchain is that it creates a system of distributed consensus in the digital world. By creating an irrefutable ledger record, it allows participating actors to know for sure that some digital event has occurred, thereby establishing a reliable communication and interaction system.

© Springer Nature Switzerland AG 2019
S. Misra et al. (Eds.): ICCSA 2019, LNCS 11620, pp. 567–576, 2019.
https://doi.org/10.1007/978-3-030-24296-1_45

After having studied the requirements put forward for the peer review system and the possibilities offered by the blockchain, it can be assumed that blockchain or distributed ledger technologies may serve as a suitable basis for the technological stack for development of an electronic examination system.

2 Motivation

We are presented with the challenge of implementing an electronic examination system. As an example of the usage of this system, it is considered the checking of contest tasks performed by students of educational institutions. In this way, it must have interfaces for downloading tasks, solving them, evaluating and viewing results. The assessment will be carried out by averaging the marks obtained from three unrelated auditors. Further, it is necessary to have an integrated authentication and authorization service to ensure secure access to the system and regulate the rights of various users. Among other things, the purpose of the work is to confirm the suitability of applying Blockchain to the provided task as well as uncover the advantages of this technology.

3 Related Work

As of now, expertise systems are attracting considerable interest and thus there are several studies covering different approaches to creating such. Each one of them is implementing distributed ledger technology to achieve its goals. Blockchain [2] is widely used in the design of secure and distributed education systems, as shown in [6] and [7]. Their core principles are akin to ones our system follows: independent and reliable participants of the network safely exchange data, albeit [7] relies more on conventional data-storing such as cloud database instead of distributing portions of data to each party. The similar concept is employed in [8], where Blockchain is used for conducting and evaluating tests in a peer-to-peer manner and creating a self-sustained education ecosystem. Some of the works pay great attention to encrypting the transactions and ensuring the security of the overall communication process.

One of the differences of this work is that here the Corda framework [1] is used to manage low-level Blockchain interactions as opposed to some papers' approach where the basic network structures are implemented manually, which, in turn, may lead to unexpected errors. And, in contrast to more specific software in the field, this system is meant to function as a template to be configured for solving varied tasks.

4 Architecture

The system for conducting online contests consists of three types of participants. The first one forms the tasks that must be performed by the participants of the second type. Furthermore, a competent member of the system will check the

solution and rate it. To ensure the impartiality of the assessment, several experts must evaluate the work. The result is calculated as the arithmetic average of all grades. The differentiation of tasks for participants generates the separation of their access rights to different parts of the system. To divide the participants into groups, it is necessary to implement registration, with the further allocation of rights for users from the administrator, who can be the first registered member of the system. It can be seen that the first registered participant has the role of the administrator, but maintaining its existence is unfeasible after the task has been loaded and the roles of the other users are defined. The superuser in the system constitutes its vulnerable part since anyone having access to the administrator account will be able to influence other participants by altering their rights or deleting them from the system. In this way, it is necessary to implement an opportunity to get rid of the user with such broad rights. Each contest naturally has its own deadline. Therefore, one of the necessary parts of the project should be the implementation of time periods during which users will have an opportunity to solve tasks. The same restrictions can be set for work verification. Further, it must be provided an ability to view the results of the contest after its completion. Figures 1, 2, 3, 4, 5, 6 and 7 schematically show the organization of the electronic review system.

4.1 First Block

First, we consider Fig. 1, which shows the scheme of interaction between the administrator and users with the first node of the system.

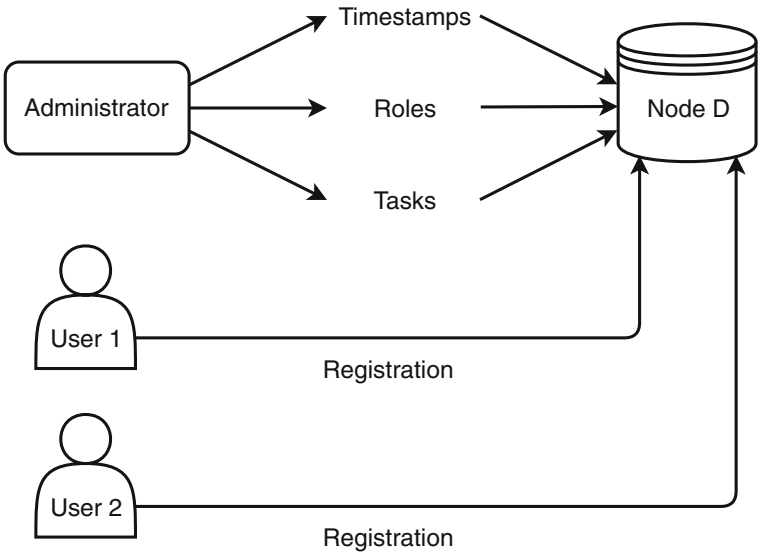


Fig. 1. Administrator and user interaction with the authentication node.

The first block is a node of the blockchain platform for the administrator and users who can register in the system. At this step user interaction with the site is limited. The administrator has a wider range of features; he is given the opportunity to give each registered user the necessary rights. In addition, the administrator is responsible for uploading tasks and determining the time intervals during which they will be available for downloading and performing, and their solutions will be available for checking. First, we consider Fig. 1, which shows the scheme of interaction between the administrator and users with the first node of the system.

4.2 Second Block

In the second block interaction of the contest participant with the task is provided. To confirm the access rights to the assignments, it is necessary to pass authentication, after which participants are given an opportunity to download tasks, write a solution and send it for verification. Only users defined as participants of the contest have access to this part of the system.

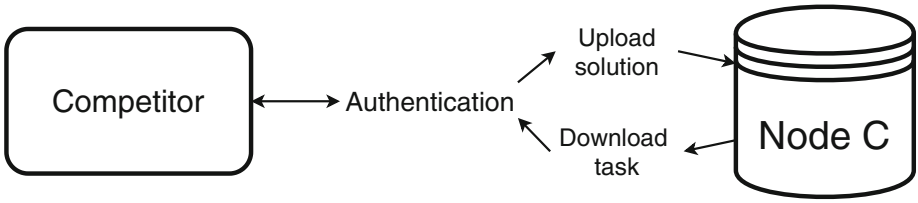


Fig. 2. User interaction with the solution node.

4.3 Third Block

The third blocks' structure is similar to the second, but they possess different functions. The third block is intended for users who, upon authentication, have rights to verify the solutions of other participants of the contest. To perform the functions for which the inspectors are registered, they are provided with an interface that allows loading the solution of tasks along with their conditions.

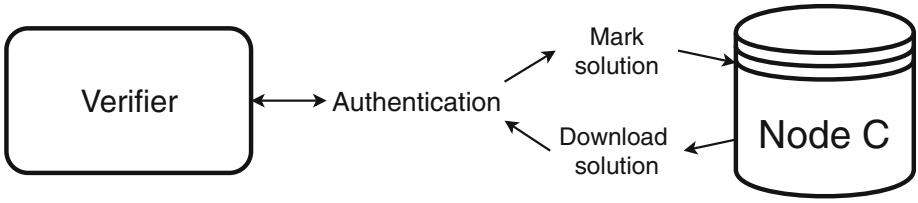


Fig. 3. User interaction with the marking node.

After evaluating the work, the reviewer has an ability to rate it on a 100 point scale and send it further along with the blockchain.

4.4 Fourth Block

The final block of the system is designed to derive the results, which, as mentioned above, are obtained by averaging the estimates received from different inspectors. Viewing of the results is available only to authorized users. At the moment, it remains unclear how nodes interact with each other and what functions they should perform. Further, we consider the pairwise interaction of nodes and what tasks they solve.

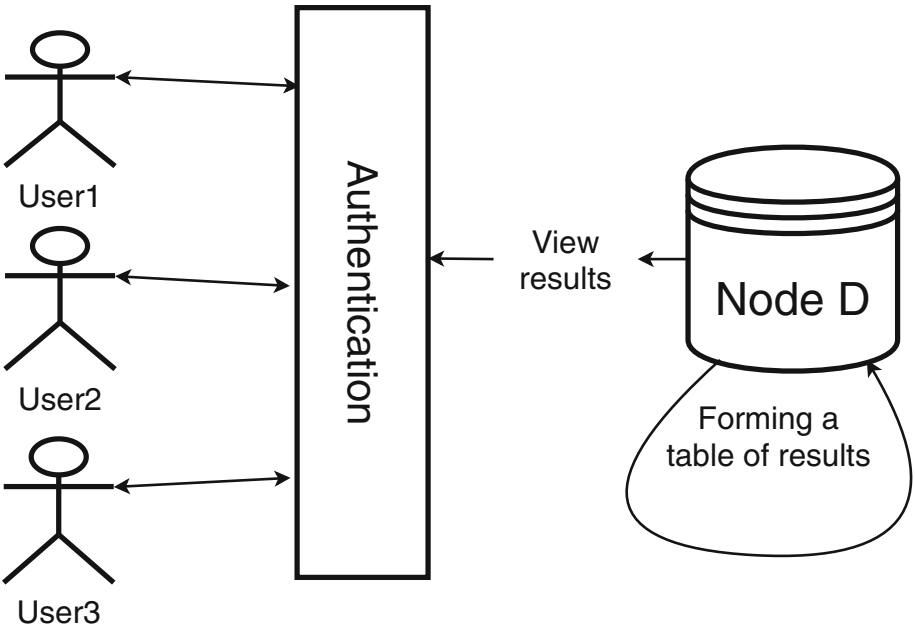


Fig. 4. User interaction with the end result node.

4.5 Task Transfer

Figure 5 shows the administration node and the node for users to interact with. In this case, each participant must have copies of the task loaded by the administrator and solutions should be provided in a predetermined time window. To solve this problem, the node A, up to a certain point, gives only the headers of the tasks without their contents. To control the second time limit, a restriction is implemented on sending a solution to the node B and receiving it at the node C. Also node B provides the ability to download the task to your computer, fill in the fields with the solution and send it to the verifier.

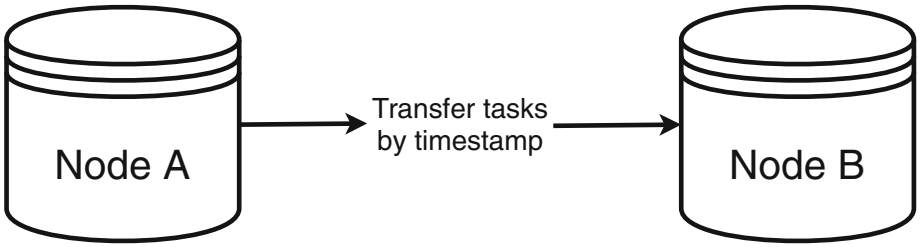


Fig. 5. Interaction between task upload node and solution node.

4.6 Solved Task Distribution

After the competitor has written the solution of the assignment and sent it further along the chain, it is received at the nodes of the inspectors. These nodes are used for checking the time window, grading, downloading the task itself and its solutions for checking and sending the score.

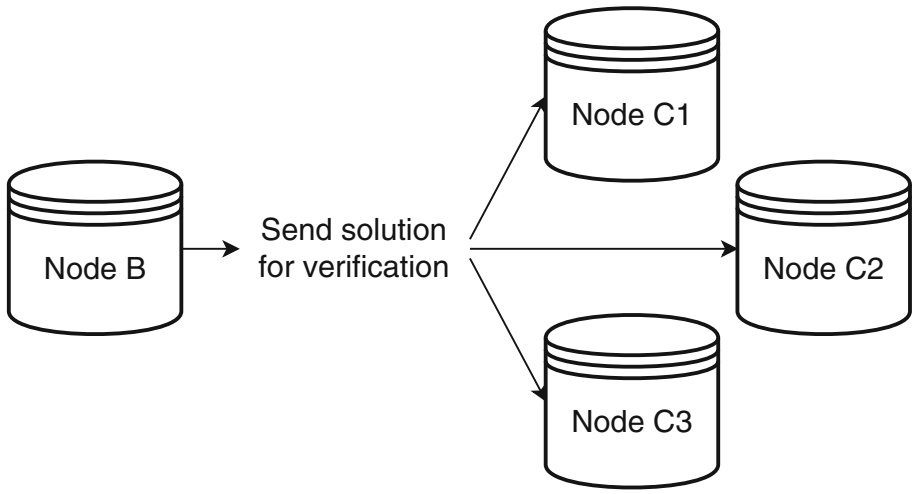


Fig. 6. Interaction between solution node and marking nodes.

4.7 Aggregation of Marks

Afterward, the marks are collected at the node D, and when all the experts submit the scores for the provided solution, the result is aggregated and displayed in the table. To access all the functions provided by the nodes, the user must authenticate at each node. Node A functions as the authentication server, handling all the requests for login and checking user rights and identities.

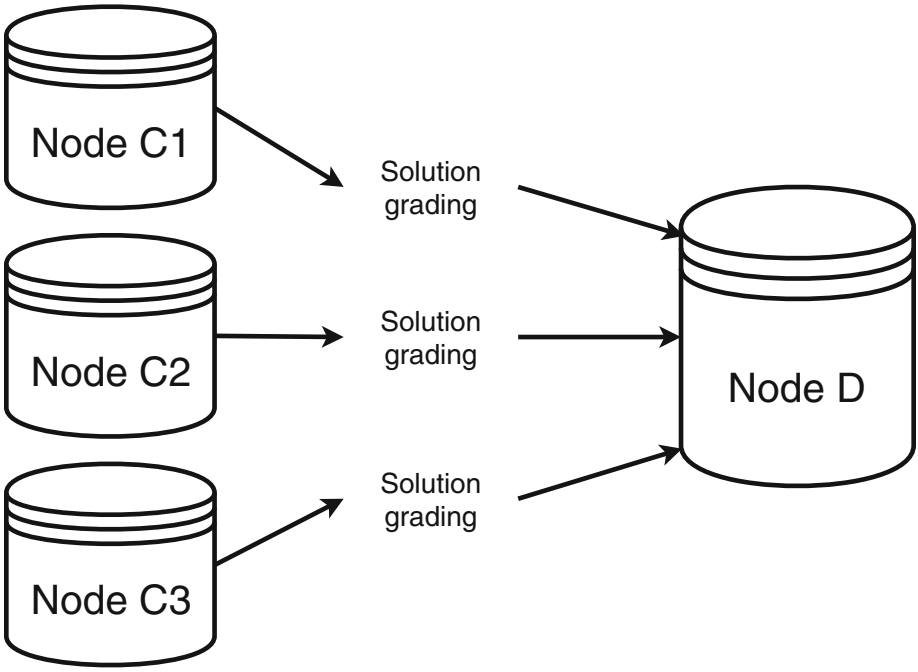


Fig. 7. Interaction between marking nodes and the end result node.

5 Implementation

To accomplish this task, we apply the technology related to the blockchain and in many aspects identical with it - the distributed ledger. As in the blockchain, the distributed ledger contains entries that are not stored or confirmed by a single central site. But in fact, the one who organizes the ledger can manage the structure of the distributed network, its purpose and the form of the transmitted data. Thus, the reliability and security that the blockchain provides are combined with flexibility and the ability to relatively fine-tune the system.

In relation to this task, the use of distributed registries allows increasing the number of users without a threat of system failure by adding new nodes, as well as altering the network structure as the task conditions change. The distribution also means high reliability of the system as a whole - if one of the nodes fails, all other users will be able to continue their activities. This project was implemented using the Corda framework, which provides the most extensive possibilities for working with distributed registries. Its another advantage is the possibility of programming both in Java and in Kotlin [9]. Kotlin was chosen for this work because of the better readability of the code and the large base of programs and examples using Corda.

5.1 Project Structure

The system consists of two layers - a network of nodes, which store and exchange information with each other; and web interfaces that provide user interaction with the corresponding nodes.

To implement the network structure, the following types of nodes are required:

- The notary who oversees transactions
- Node for uploading tasks to the vault
- Node for storing tasks and sending solutions
- Node for grading solutions
- Node for storing results
- Authentication node controlling access rights to other nodes.

The Corda platform operates on several key concepts:

- Flows are sequences of steps used by the node to update the status of the ledger. They determine the behavior of nodes when they receive or send a transaction that changes data on the node. For the download node, there exists only the transaction sending flow, for the view node there is only the flow that accepts the transaction. The remaining nodes can either accept transactions from the previous one in the chain or update the data on the next one.
- Data is presented in the form of states - immutable objects corresponding to facts that are known to one or several nodes. Their immutability is an important feature of the system. States describing the available data consist of: UploadState - contains the hash of the attachment file, SendState - in addition to the hash, contains the name of the student who sends task and its solution, MarkState - the state that stores the name of the completed task, its hash and its mark.
- Also, contracts are used to verify the correctness of transactions - they are classes that contain transaction verification functions.

Web interfaces are implemented for interacting with all nodes except the notary since its existence must remain hidden to users. Each interface provides a user login form to prevent illegal actions against the system. To distribute the application to individual machines, nodes and the corresponding interfaces were assembled into portable executable files using Portable Java. Every application has the form of a built-in browser displaying the web interface, which was implemented using the Electron [10] service.

6 Benchmark

When the system is running, there are additional costs for connecting to the nodes, transferring messages and files from the user's computer to the node

as well as between nodes, and the interaction of the nodes among themselves requires the intervention of a notary who checks every transaction and makes entries in the ledger. In order to find out how much these architectural features of the interaction of the nodes between themselves and the user influence the execution time of the typical 6 types of operations, a speed test of the pipeline constructed from nodes connected in the chain was carried out. A pipeline is a sequence of nodes where a file of a certain size is first loaded, and then the file is transferred along the chain to the last node. For testing, we used files of various sizes from two hundred bytes to ten megabytes ([200 bytes, 128 KB, 512 KB, 1–10 MB]). A total of 32 nodes was used, the time was measured using 2-step increments in node count.

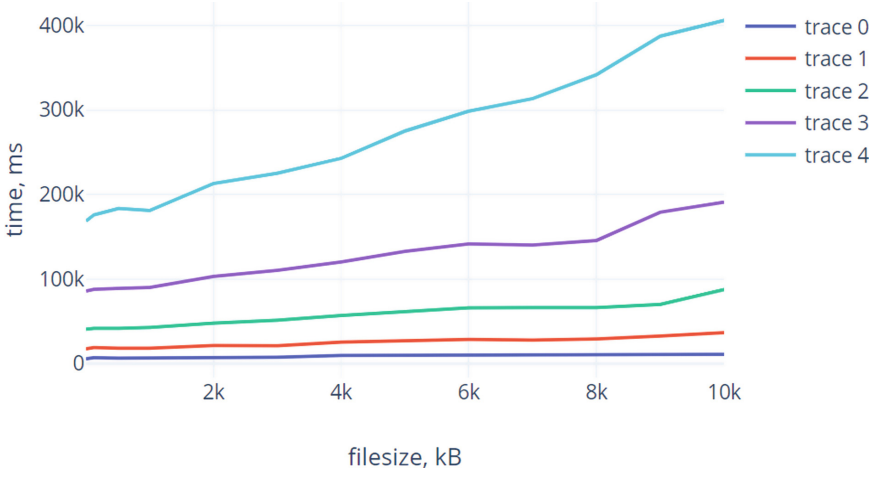


Fig. 8. Chart showing relation of the program run time to the transmitted file size.

In Fig. 8 the steady increase in time delay can be seen, and it is especially noticeable when there are at least eight nodes in the network. Other conclusions that can be made from the graph are that there is a linear relation between running time and node count for example, it takes twice the time for system with 32 nodes to run as opposed to the one with 16 nodes, as well as better than linear relation between running time and filesize it takes about twice the time to transmit 10 MB file through the chain compared to transmission of the 1 MB file. Tests show that the costs of distributed registries for actions as initiating a connection, checking transactions and writing to the ledger are initially large, but the more messages we send, the lower the percentage of work time is spent on expenses caused by the distributed ledger architecture. Even with a relatively large message size, we still experience a slowdown due to the use of distributed registries. Thus, given the fact that in a specific project, nodes can perform secondary functions that will introduce additional delays, it can be concluded that optimizing the system based on distributed registries is one of the key areas

of work in developing such a system. The key to optimization can be reducing the number of connections to various nodes initiated by the user, getting rid of redundant checks by contracts and minimizing the number of participants in the transaction, therefore avoiding long waiting for signatures by other participants.

7 Conclusion

The Corda platform allows implementing an electronic expertise system which solves the task at hand and satisfies the specified requirements - the ability to easily change the network structure, to increase the number of users without system failure, and work with node failures. Among the advantages of the used technology is the ability to tune the type of the data received at each node - each participant knows only the information that he must have in order to make a decision. These characteristics were achieved because of the usage of distributed ledger technology, and although it has disadvantages such as low data processing speed and it requires a large amount of memory to store local information, the opportunities provided by it greatly simplify the designing of complex systems.

References

1. Brown, C., Grigg, H.: Corda: an introduction (2016). http://r3cev.com/s/corda-introductory-whitepaper-final.pdf
2. Yli-Huumo, J., Ko, D., Choi, S., Park, S., Smolander, K.: Where is current research on blockchain technology?—A systematic review. PLoS ONE **11**(10), e0163477 (2016). https://doi.org/10.1371/journal.pone.0163477
3. Mike, H.: Corda: A distributed ledger (2016). https://block.academy/researches/corda-technical-whitepaper.pdf
4. Das, D.: Hacking into the Indian Education System. https://deedy.quora.com/Hacking-into-the-Indian-Education-System
5. Watanabe, H., Fujimura, S., Nakadaira, A., Miyazaki, Y., Akutsu, A., Kishigami, J.J.: Blockchain contract: a complete consensus using blockchain. In: 2015 IEEE 4th Global Conference on Consumer Electronics (GCCE), pp. 577–578 (2015). https://doi.org/10.1109/GCCE.2015.7398721
6. Turkanovic, M., Holbl, M., Kosic, K., Hericko, M., Kamisalic, A.: EduCTX: a blockchain-based higher education credit platform. IEEE Access Multidiscip. Open Access J. **6**, 5112–5127 (2018). https://doi.org/10.1109/ACCESS.2018.2789929
7. Islama, I., Md. Kaderb, F., Shina, S.Y.: BSSSQS: a blockchain based smart and secured scheme for question sharing in the smart education system. Preprint submitted to Journal of Parallel and Distributed Computing arXiv:1812.03917v1 (2018)
8. Acharya, R., Binu, S.: Blockchain based examination system for effective evaluation and maintenance of examination records. Int. J. Eng. Technol. **7**(2.6), 269–274 (2018). https://doi.org/10.14419/ijet.v7i2.6.10781
9. "kotlin-stdlib". kotlinlang.org. JetBrains. https://github.com/JetBrains/kotlin/releases/latest
10. "From native to JavaScript in Electron — Electron Blog". electronjs.org. https://github.com/electron/electron/releases/latest