Jurnal Teknik Informatika (JUTIF)
Vol. 3, No. 5, Oktober 2022, hlm. 1373-1378

DOI: https://doi.org/10.20884/1.jutif.2022.3.5.473
p-ISSN: 2723-3863
e-ISSN: 2723-3871

WALLET-BASED AUTHENTICATION ON COLLEGE INFORMATION SYSTEM

Rickard Elsen*[1], Muhammad Rikza Nashrulloh*[2], Ade Sutedi*[3]

[1,3]Teknik Informatika, Fakultas Ilmu Komputer, Institut Teknologi Garut, Indonesia
[2]Sistem Informasi, Fakultas Ilmu Komputer, Institut Teknologi Garut, Indonesia
Email: [1]rickardelsen@itg.ac.id, [2]rikza@itg.ac.id, [3]adesutedi@itg.ac.id

(Naskah masuk: 15 Juli 2022, Revisi: 4 September 2022, diterbitkan: 24 Oktober 2022)

Abstract

Since the widespread use of cryptocurrency, blockchain technology start to be adapted in various applications. Some businesses are already adopting blockchain technology because of its advantages such as data integrity and privacy. One of them is Web 3.0. Web 3.0 puts forward data decentralization so that users can choose what data will be sent to the server. User data is provided locally with the help of a crypto wallet and the server just receives wallet info. With this mechanism, user privacy can be maintained directly by the user himself. All data will be processed at the users' end first before being sent to the server. With the new mechanism of web 3.0 and the advantages of blockchain, we build an application to authenticate students' login activities and grant roles to them based on their wallets. In this paper, we use the prototyping model as the method to build the application. We managed to utilize students' wallet addresses as credentials. And with the help of Web3 module, we managed to decentralize the authentication process. And as a result of the successful authentication process, students can access their data based on their roles.

Keywords: *Authentication, Blockchain, Role-Based Access Control, Wallet.*

1. INTRODUCTION

Blockchain is a technology that allows data to be distributed in a decentralized manner[1]. It uses a chain of blocks to store data, linked by a pointer to establish a connection between blocks. New data will be stored and linked to the last data. Adding new data means extending the chain of blocks. Added data cannot be modified to keep the integrity of the chain. This mechanism will guarantee the trust of data without the need for a trusted third party[1]–[3].

Web 3.0 is driven by blockchain technology to make users' activity on the internet can be decentralized[3]–[5]. One of a problem with web 2.0 is privacy[1], [6]. An application provider can gain users' data and store it as their big data to do some research like ad targeting. It is possible because users' data is stored at their end. A case by Facebook related to a data breach[7] is an example of how users' data is being utilized by application providers without users' consent.

With all benefits offered by the blockchain, many organizations and companies are trying to apply it to their business[1]. Blockchain offers immutable data[1], [3] because data stored on blocks is nearly impossible to be modified even by the owner of the system. It can be used by companies to store data permanently without worrying about data changes. This mechanism is very suitable for company activities related to recording data continuously like healthcare, logistic, and smartcity[8]–[11].

Blockchain is also possible to be applied to the education system[2], [6], [12]–[15]. The benefit of applying blockchain in the education system is to improve the management of students' records. It also maintains student privacy to access their sensitive data like personal information, score, and payment status. College can expose all data but only student who owns the data can retrieve and read it. It also can be utilized to create digital graduation certificates by adding blockchain data to the certificate as verification of ownership[16], [17].

2. METHODOLOGY

In this study, the methodology used to apply blockchain to college information systems is Prototyping Model. The phases in this method are requirement gathering and analysis, quick design, build a prototype, initial user evaluation, refining prototype, and implement product and maintain, as seen in Figure 1.

- Requirement gathering and analysis. The purpose of this phase is to gather user needs and analyze them. Users will define all features needed in the application in detail. This phase will show users' expectations related to software development. All user's needs will be analyzed to determine how to apply that requirement to software.
- Quick design. The purpose of this phase is to draw the analyzed requirement to show the working steps of a software. The design of the results of this phase will be shown to the user for verification. After the user agrees with the

design, it will be delivered to the developer as their guide to developing software.

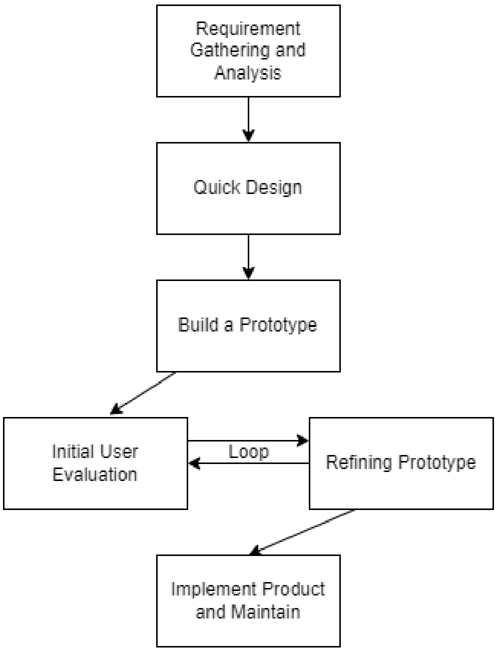


Figure 1. Prototyping model phases

- Build a prototype. The purpose of this phase is to build preliminary software that can represent the user’s requirements. as much as possible. In this phase, the software delivered to the user is not a completely finished software but is still a prototype to give an idea of how the feature will work on the software.
- Initial user evaluation. The purpose of this phase is to get user feedback about the prototype. Users will see on the prototype how features will be applied to the software. In this phase, users are also involved in keeping the defined requirements actually implemented in the software.
- Refining prototype. The purpose of this phase is to fix the prototype based on user feedback and make sure all defined requirements are applied to the software. This phase is not done just once, but continuously with the initial user evaluation phase until the prototype met the users’ expectations and defined requirements. The number of repetitions in this phase cannot be determined because it needs the user’s agreement to stop the repetition.
- Implement product and maintain. The purpose of this phase is to deliver software to users and make sure users can use the software in their environment. This phase also provides to maintain software to ensure that this software can be used for a long time

3. RESULT AND DISCUSSION

At this stage, the prototyping model’s phases are used in building software are up to refining prototype, since the implementation of software cannot be done because the system is not ready to implement blockchain-based authentication.

3.1. *Requirement Gathering and Analysis*

Based on users’ requirements and analysis results, we can conclude the functional requirements will be applied to software as shown in Table 1.

Table 1. Functional requirement

Code	Requirement
FR01	Students can log in to the system
FR02	Students can link their wallets to the system
FR03	The system can recognize students by their wallet address
FR04	Student can access their data based on their role
FR05	The system can show student information by their wallet address

We also conclude one non-functional requirement which is data integrity. It not only requires data to be consistent and accurate but also can integrate with existing data in the existing system.

3.2. *Quick Design*

Based on the analysis result, we create a use case diagram to determine how students will interact with the system.

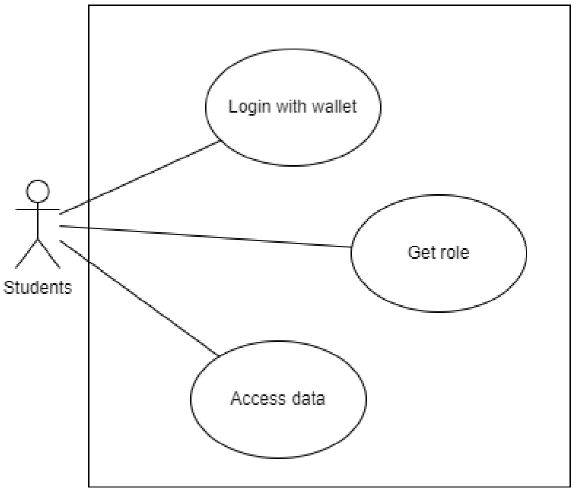


Figure 2. Use case diagram

As shown in the use case in Figure 2, students have three main activities served by the system.

- Students must be able to log in to the system using their wallets. The mechanism is different compared to the existing system since user data is not stored on the system. The wallet will be an important point to make sure student can gain their role.
- Student can get their role. After students log in to the system with their wallets, the system will examine the wallet address and match it with existing data on the system. After the system recognizes the student, the role will be given to the student.

- Student can access their data. After student got their role, the student can access their data based on the role given by the system.

3.3. *Building a Prototype*

3.3.1. *Choosing Components*

In building the prototype, we determine the components needed to develop software.

- Binance Smart Chain (BSC). This component's purpose is to create a unique token. This token cannot be mined so we can determine how much token is needed by the system and no one can change the number and ownership of the tokens except the owner. BSC provides free token creation. We can use it but will be delisted so our token will be invisible to the public. We use BSC Evolution Proposal 20 (BEP20) as a token standard.
- Metamask. This component's purpose is as a crypto wallet so the student can log in using their crypto address. Metamask will be integrated with created token in BSC to use the smart contract so students can have a crypto address.
- Web3 module. This component's purpose is as a module to support the utilization of metamask. It is a collection of libraries that allow the software to interact with a local or remote node using HTTP, IPC, or WebSocket.
- React. This component's purpose is as a library to build user interfaces. Since we use a decentralized scheme, software must be built as Single Page Application (SPA) to make software will run on the client side and interact directly with Metamask.

3.3.2. *Authentication Activities*

The prerequisite for the students to log in to the system using their wallet is as follows:

- Students must have a Metamask account.
- Students must connect their Metamask to Binance Smart Chain.
- Students must install the Metamask extension to their web browser.

Student must register their wallet to the system.

As seen in Figure 3, students can log in to the system using Metamask by connecting their wallets to the system. The system will ask Metamask about students' wallet info. Since it uses decentralized authentication, Metamask will check user credentials locally on students' computers. After students successfully log in to Metamask, it will send wallet info to the system as a callback. The system will use this info to grant students roles to give access to their data.

As seen in Figure 4, we add Wallet and Network tables as requirements for the system to apply decentralized authentication. Students are able to have more than one wallet as long as it is registered to a different network. The system will check wallet data based on wallet info sent by Metamask and match wallet addresses with students' IDs. After this phase, the system will grant the role based on students' role data stored on the database.

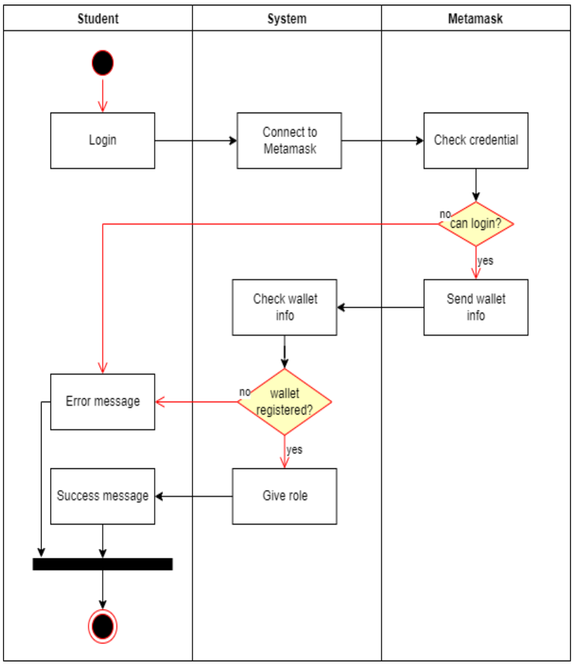


Figure 3. Activity diagram

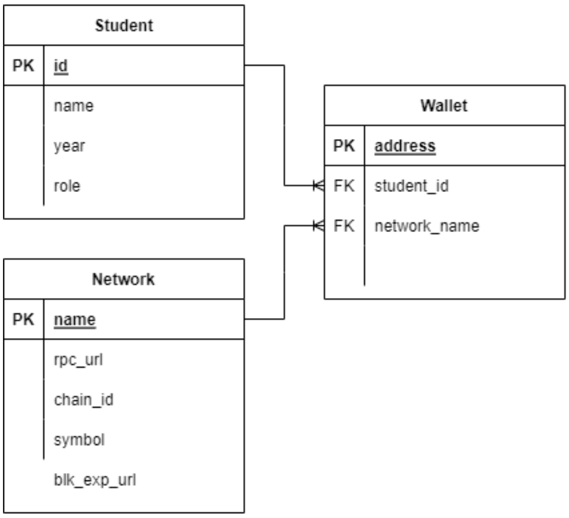


Figure 4. Entity relationship diagram

3.3.3. *Building Application*

In building application, we utilize Web3 module called Usedapp. This module is based on the Ethereum network. Since we use BSC in this application, and in fact BSC is based on the Ethereum network, we used to modify the configuration by changing the chain ID and URL in the config file so the module can connect to BSC.

As seen in Figure 5, we managed to create a frontend application to call the Metamask plugin when the "Login with Metamask" button clicked. Since Metamask only can be accessed locally, we implement Single Page Application (SPA) so the application can run in the client environment. In this phase, students must fill their credentials to Metamask so the application can read their wallet address and send information to the backend to give roles to students.

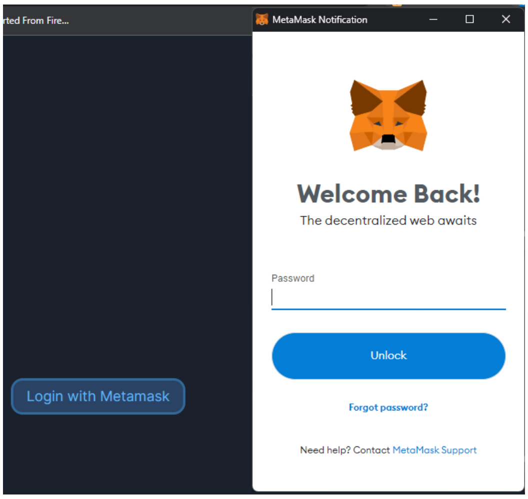
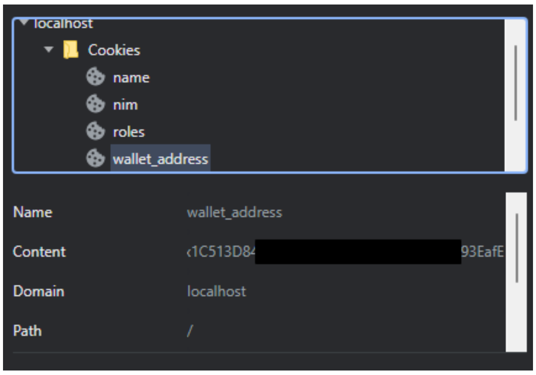


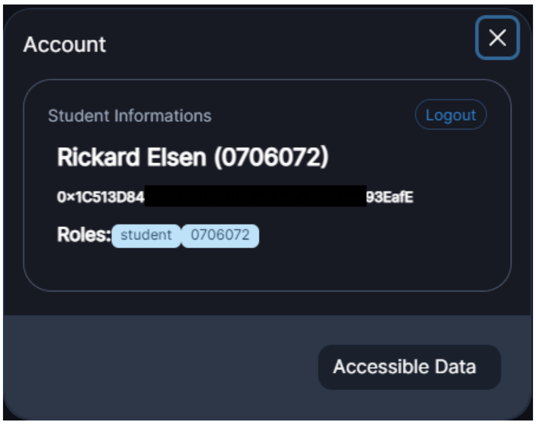
Figure 5. Login with Metamask

In the backend, wallet information gathered from the frontend is processed to gather students' information from the database. Students' information contains ID, name, wallet address, and roles. All this information are stored in cookies as a session to manage sustain credential. An example of stored cookies can be seen in Figure 6.



Name wallet_address
Content ‹1C513D8[redacted]93EafE
Domain localhost
Path /

Figure 6. Cookies example



Student Informations

Rickard Elsen (0706072)

0x1C513D84[redacted]93EafE

Roles: student 0706072

Figure 7. Student's Information

After the backend can determine student information and save the session to cookies, the backend will send a response to the frontend with status code 200 means all process has been succeded. Backend return no data since needed information about students is already stored in cookies. Frontend will show student information as seen in Figure 7.

After students successfully logged in to the application, they can access their data based on the role given by the backend. A student can be granted more than one role. All data based on students' roles are listed and sorted by their roles as seen in Figure 8.

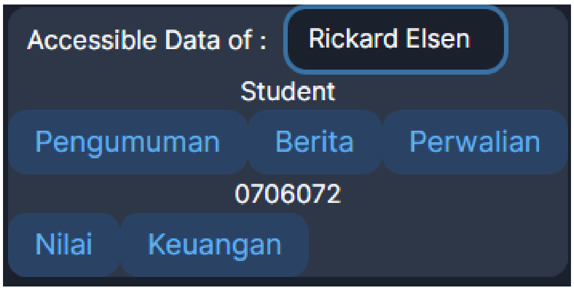


Figure 8. Accessible Data Based on Students' Roles

As a result of this phase, we managed to create software to log in using The software can interact with Metamask by request for wallet information and receiving wallet information as a callback. The software can process this info by matching the wallet address to stored data in the database and granting roles to students

3.4. *Initial User Evaluation*

In this phase, we show the software behavior to stakeholders by simulating the student login process and receiving 2 feedback.

- Student may forget their wallet credential so we must apply a mechanism to reset student wallets and revoke their authority to access their data.
- Software must be able to reduce stored data since needed data is mostly provided by users. We must reanalyze data from the existing system and the student's end.

3.5. *Refining Prototype*

In this phase, we apply feedback from stakeholders to the software.

- Software support student's wallet reset. To reset their wallet, students must report to management. After approval, management will enter the student's new wallet address into the software. The software will reset the student's wallet address and revoke data access to the new address. This mechanism cannot be done automatically to reduce the risk of the account being taken over by unauthorized and irresponsible parties.
- Software now exclude students' detail on transaction data. All data related to students was

replaced with their wallet addresses. With this mechanism, it reduces almost half of the stored data. But in exchange, query activities are increased but not very influential to the system.

4. CONCLUSIONS

In the conclusion, we managed to build a web 3.0 based system to authenticate students' login and give them their roles to access their data. We use external components i.e. BSC, Metamask, Web3 module, and React to build this system. Students can log in using their wallet and use their wallet address as the credential. This mechanism can be done by integrating software with Metamask. It shows that blockchain technology can be used as an authentication system by using a wallet and utilizing it as a user credential to determine user authority based on roles. But what we built is still only a prototype. It is not used yet in the system since we must change the whole college system if we want to implement it. We also only can test the data with a copy of existing data so we cannot monitor the dynamics of the transaction by applying blockchain. In the future, we hope it can continue to bigger and more integrated system so we can implement it in the whole college system. And also we hope we can utilize BSC more than this by using their network to create a new coin and use this coin for all transactions in college.

ACKNOWLEDGEMENT

We thank Institut Teknologi Garut by support us in funds and development environment.

REFERENCES

[1] U. Bodkhe *et al.*, "Blockchain for Industry 4.0: A Comprehensive Review," *IEEE Access*, vol. 8, pp. 79764–79800, 2020, doi: 10.1109/ACCESS.2020.2988579.

[2] S. Rouhani and R. Deters, "Blockchain based access control systems: State of the art and challenges," in *IEEE/WIC/ACM International Conference on Web Intelligence*, Oct. 2019, pp. 423–428. doi: 10.1145/3350546.3352561.

[3] G. Korpal and D. Scott, "Decentralization and web3 technologies," 2022.

[4] S. Aghaei, "Evolution of the World Wide Web : From Web 1.0 to Web 4.0," *International journal of Web & Semantic Technology*, vol. 3, no. 1, pp. 1–10, Jan. 2012, doi: 10.5121/ijwest.2012.3101.

[5] F. A. Alabdulwahhab, "Web 3.0: The Decentralized Web Blockchain networks and Protocol Innovation," in *2018 1st International Conference on Computer Applications & Information Security (ICCAIS)*, Apr. 2018, pp. 1–4. doi: 10.1109/CAIS.2018.8441990.

[6] D. Berdik, S. Otoum, N. Schmidt, D. Porter, and Y. Jararweh, "A Survey on Blockchain for Information Systems Management and Security," *Information Processing & Management*, vol. 58, no. 1, p. 102397, Jan. 2021, doi: 10.1016/j.ipm.2020.102397.

[7] J. Isaak and M. J. Hanna, "User Data Privacy: Facebook, Cambridge Analytica, and Privacy Protection," *Computer (Long Beach Calif)*, vol. 51, no. 8, pp. 56–59, Aug. 2018, doi: 10.1109/MC.2018.3191268.

[8] P. K. Sharma and J. H. Park, "Blockchain based hybrid network architecture for the smart city," *Future Generation Computer Systems*, vol. 86, pp. 650–655, Sep. 2018, doi: 10.1016/j.future.2018.04.060.

[9] E. Tijan, S. Aksentijević, K. Ivanić, and M. Jardas, "Blockchain Technology Implementation in Logistics," *Sustainability*, vol. 11, no. 4, p. 1185, Feb. 2019, doi: 10.3390/su11041185.

[10] V. Filimonau and E. Naumova, "The blockchain technology and the scope of its application in hospitality operations," *International Journal of Hospitality Management*, vol. 87, p. 102383, May 2020, doi: 10.1016/j.ijhm.2019.102383.

[11] A. A. Monrat, O. Schelen, and K. Andersson, "A Survey of Blockchain From the Perspectives of Applications, Challenges, and Opportunities," *IEEE Access*, vol. 7, pp. 117134–117151, 2019, doi: 10.1109/ACCESS.2019.2936094.

[12] D. di Francesco Maesa, P. Mori, and L. Ricci, "A blockchain based approach for the definition of auditable Access Control systems," *Computers & Security*, vol. 84, pp. 93–119, Jul. 2019, doi: 10.1016/J.COSE.2019.03.016.

[13] A. Badr, L. Rafferty, Q. H. Mahmoud, K. Elgazzar, and P. C. K. Hung, "A Permissioned Blockchain-Based System for Verification of Academic Records," in *2019 10th IFIP International Conference on New Technologies, Mobility and Security (NTMS)*, Jun. 2019, pp. 1–5. doi: 10.1109/NTMS.2019.8763831.

[14] A. Alammary, S. Alhazmi, M. Almasri, and S. Gillani, "Blockchain-Based Applications in Education: A Systematic Review," *Applied Sciences*, vol. 9, no. 12, p. 2400, Jun. 2019, doi: 10.3390/app9122400.

[15] R. Raimundo and A. Rosário, "Blockchain System in the Higher Education," *European Journal of Investigation in Health, Psychology and Education*, vol. 11, no. 1, pp. 276–293, Mar. 2021, doi:

10.3390/ejihpe11010021.

[16] S. Wang, Y. Yuan, X. Wang, J. Li, R. Qin, and F.-Y. Wang, “An Overview of Smart Contract: Architecture, Applications, and Future Trends,” in *2018 IEEE Intelligent Vehicles Symposium (IV)*, Jun. 2018, pp. 108–113. doi: 10.1109/IVS.2018.8500488.

[17] W. Zou *et al.*, “Smart Contract Development: Challenges and Opportunities,” *IEEE Transactions on Software Engineering*, vol. 47, no. 10, pp. 2084–2106, Oct. 2021, doi: 10.1109/TSE.2019.2942301.