Design and implementation of national record management system with blockchain

Ragouguelaba agoda koussema*
Graduate School of Science and Engineering, Doshisha University, Kyotanabe, Japan;

Hirohide Haga
Graduate School of Science and Engineering, Doshisha university, Kyotanabe, Japan;

ABSTRACT

Apart from financial transactions, blockchains are suggested for numerous application fields. While it is possible to mount generic blockchains for specific use cases, the implementation is generally lean and easy to adjust. Here we present the fundamental ideas of blockchain technology and examine a real use case for national record management over a blockchain technology. A fully customized, private, and authorized blockchain is implemented from scratch. On the basis of this use, we analyze and justify the necessity for the blockchain technology and describe the desire features of our system. We discuss in more detail about the implementation of our system. This article describes the design and implementation of national record management system by using customized blockchain. Furthermore, to provide the best user experience, we also built the web application interface with Java web application framework named PrimeFace. The implementation of a prototype revealed that the built blockchain technology from scratch is more suitable.

CCS CONCEPTS

• **Security and privacy** → Systems security; Distributed systems security.

KEYWORDS

blockchain, fundamental, implementation, web application

ACM Reference Format:
Ragouguelaba agoda koussema and Hirohide Haga. 2021. Design and implementation of national record management system with blockchain. In *2021 4th International Conference on Electronics and Electrical Engineering Technology (EEET 2021), December 03–05, 2021, Nanjing, China.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3508297.3508333

1 INTRODUCTION

This article describes the design and implementation of highly reliable and secure national record management system under relatively poor ICT (Information and Communication Technology) environment. To achieve this final goal, we have decided to use the blockchain technology. Firstly, used as decentralized and trust-less ledger for digital currencies, blockchain is adopted in many fields [2-4]. Many sectors, for data management, have focusing on the using the conceivable outcomes given by blockchain technology to realize decentralized, trust-less applications that do not depend on a single trusted party. Some governments are exploring the properties and benefits, as well as the deficiencies of this innovation for their particular utilize cases.

Data management can be defined as the organizing and maintaining data to meet ongoing information lifecycle needs. The data need to be stored and be secured for usage. It is important to have a traceability of all updates that will be made on the data.

The dependability and security of relational database management system (RDMS) are two of its important features. Resident data typically contains highly sensitive personal information. Several pieces of equipment and software must be installed in order to implement highly secure RDMS. Data transmission lines must be safeguarded against unauthorized access.

High level data exchange systems are disrupted by lack of communication infrastructure. Data management systems must be extremely secure, dependable, and simple to use. There are already technologies in place to implement such a highly secure and dependable system.

For our purpose, we evaluate the need for blockchain and the applicability of different types of consensus algorithms and permissions. A certain number of existing implementations, such as Multichain [8], Ethereum [9], and Hyperledger Fabric [10], were initially taken into consideration. In any case, due to versatility issues on the specified equipment and for accomplishing a lightweight and basic solution, a blockchain has been implemented from scratch. Furthermore, this solution does not involve only blockchain technology, but also requires user interaction with the web application developed with Java programming language and the processing of privacy-sensitive data. People can access website by using wireless communication infrastructure such as a mobile phone network.

The rest of the paper is structures as follows: in section 2, we introduce the blockchain technology. In section 3, we describe the underlying use case and its legal requirements, including related work on the subject. We further motivate the use of blockchain technology for the proposed application, we discuss existing blockchain implementations and describe our custom implementation in detail in section 5. Finally, in section we discuss the future work before concluding the paper.

2 OVERVIEW OF BLOCKCHAIN TECHNOLOGY

2.1 Definition of blockchain

Blockchain is a decentralized blockchain ledger that keeps track of all network transactions. With simple secure private keys, each

*Place the footnote text for the author (if applicable) here.

EEET 2021, December 03–05, 2021, Nanjing, China

ACM ISBN 978-1-4503-8516-9/21/12…$15.00
https://doi.org/10.1145/3508297.3508333

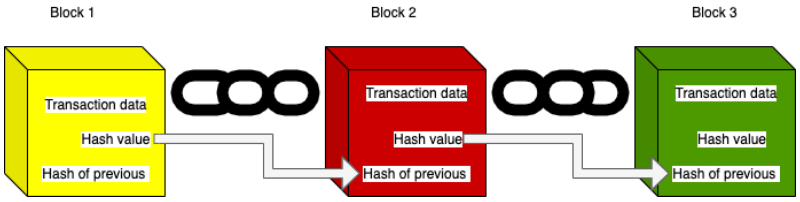

Figure 1: Conceptual illustration of blockchain database

node stores all information on the network. Each block includes information such as the previous block's hash value, data to be stored in the current block, and a nonce value. To create each block, the user must choose an acceptable nonce value for each block and changing the nonce requires a significant amount of computing power. This is the primary explanation for blockchain's ability to provide a highly stable and dependable data management system. Therefore, virtually falsification is impossible. This is the main reason why blockchain provides highly secure and reliable data management system. And blockchain has an attractive function which are very different from the traditional databases. Every node in the network store all blockchain. Therefore, even some accidents such as power failure or cracker's attack destroy the database of one node in the network, all data will be recovered by using the data stored in the other nodes. This is why blockchain has high robust nature. Figure 1 is a conceptual illustration of blockchain database.

2.2 Types of blockchains

Public blockchains, private blockchains, and consortium blockchains are the three major types of blockchains.

- Public Blockchain: This form of blockchain is open to the general public, and anyone can join the network as a node. Users may or may not be compensated for taking part. They are not held by anyone and are available to the public for participation. Bitcoin is the best example of a distributed blockchain. Any time a user makes a transaction, it is mirrored in the block's copy for all. A decentralized blockchain is used by Bitcoin.
- Private Blockchain: As the name suggests, a private blockchain is one that is only accessible by a consortium or group of individuals or organizations that have agreed to share the ledger. Only the owner has the authority to make improvements to it. Federated blockchains are more scalable (high scalability) and provide greater transaction privacy.
- Consortium Blockchain: A consortium blockchain is a cross between public and private blockchains. There are federated blockchains, which are managed by a community of people. There are two or more administrative nodes in a group. In contrast to public blockchains, no one can participate in the transaction verification process without the permission of administrative nodes. This type of blockchain is commonly used in the banking industry, for example.

The existence of our target system (resident record management system) precludes the use of public blockchain. We would use a private blockchain or a consortium blockchain.

- Explanation of our system design
- Implementation of blockchain technology with Java programming from scratch

3 RELATED WORK

At its most basic, blockchain technology is a data store distributed across a network among participants who can reach consensus on the validity of transactions without the need of a central authority to mediate or authenticate. The technology's initial applications were focused on cryptocurrencies in public permissionless networks. It was primarily used in financial applications, the technology can be used to track changes to any physical or digital asset, including data that is valuable to individuals or organizations.

As we look at blockchain technology more closely, we will see four main features: immutability, cryptographic digests, cryptographic signatures, and distributed networks. Each component protects against a specific aspect of unauthorized data changes made with valid user credentials or by hackers. Integrating these blockchain innovations into database allows mainstream systems to profit from the essential security advantages of blockchain with minimal or no adjustments.

Digital ledger has been used to protect trusted records by financial institutions. It has been used too in the healthcare system to share and manage data. Healthcare data are highly sensitive, there is a need to protect it from an illegal access.

Estonia, for example, I implementing a technology called Keyless Signature Infrastructure (KSI) to protect all government data. KSI generates hash values, which are numeric values that uniquely reflect large quantities of data. The hash values are spread through a private network of government computers and stored in a blockchain [1].

In the work [7] the authors implement a decentralized record management system to handle electronic health records using blockchain technology.

Some authors [6] recommended the use of smart contracts to handle clinical trial authorization information on a permissioned Ethereum blockchain and a private IPFS network to store the data structure.

4 AVAILABLE IMPLEMENTATIONS

A variety of available implementations were considered for implementing private and permissioned blockchain. The sections that follow describe the relevant implementations and our reason for not selecting them.

Bitcoin [2] and its client Bitcoin Core5 are intended for financial transactions. Because of its limited space, low throughput and high

delay, the Bitcoin network is unsuitable for storing any significant amount of meta data [5].

The consensus algorithm of Bitcoin is only concerned with double-spending of bitcoins, not with data. Although it is technically possible to modify this cryptographic algorithm, the design of bitcoin does not lend itself well to our use case.

Since Bitcoin and its most popular implementation are infeasible, we ruled out any attempts at modifying the protocol. In most cases, modifying the existing source code would take more time and resources than building a blockchain from scratch.

MultiChain [8] is a blockchain implementation that is both private and permissioned. As with Bitcoin, it is primarily used for financial transactions, which means that the consensus algorithm would need to be significantly modified. While this is likely easier than it is with Bitcoin, the platforms currently supported by MultiChain are a significant limitation for our use case. MultiChain is currently only supported on 64-bit systems and requires several additional software dependencies. As a result, we cannot use MultiChain as the foundation for our implementation.

Ethereum [9] enables the creation of smart contracts in a variety of programming languages, including Solidity6. These smart contracts are capable of representing transactions that are practically arbitrarily complex. These are not limited to financial transactions, as is the case with Bitcoin, but also allow for the representation of states and state changes, as our use case requires. Despite Ethereum's flexibility, previous research has discovered that even simple smart contracts are prohibitively expensive [11]. Additionally, costs are unpredictable due to frequent changes in the cost structure of executed operations and fluctuations in the exchange rate to Euros7 [14, 15]. While the cost issue could be resolved by utilizing a private blockchain, Ethereum's complexity far exceeds the requirements for our use case. When a value needs to be stored in a smart contract, it is updated using a modified Merkle Patricia Trie, which is relatively time consuming, as demonstrated in [16]. Additionally, it has been demonstrated that this results in a slow execution speed as the volume of data increases, which is detrimental to our use case, particularly from the perspective of the users.

OpenChain15 is a private blockchain that is optimized for energy efficiency, network communication, and block rate. As a result, it is based on a client-server model rather than a peer-to-peer network, and its consensus algorithm is proof of authority rather than proof of work. Because the goal of this work is to create a decentralized and trustless model, a centralized approach such as the one used by OpenChain's proof of authority consensus algorithm is not applicable. Nominating a designated authority responsible for transaction validation would violate the desired security and trust properties. The same is true for all blockchains that achieve consensus via Proof of Authority (PoA), such as Corda [17].

HAWK [18] is a blockchain that prioritizes privacy. Additionally, several blockchain implementations, such as Tendermint19, Stellar [19], EOS20, and OmniLedger [20], use BFT as a consensus algorithm. In comparison, the NEO blockchain21, which is based on BFT, can be considered stable, but its complexity, which enables the execution of smart contracts similar to those found on Ethereum, exhibits the same issues as the latter.

Due to the unique requirements of our use case and to gain insight into the entire blockchain development and implementation process, we chose to build our own custom blockchain tailored to the specific legal requirements and requirements.

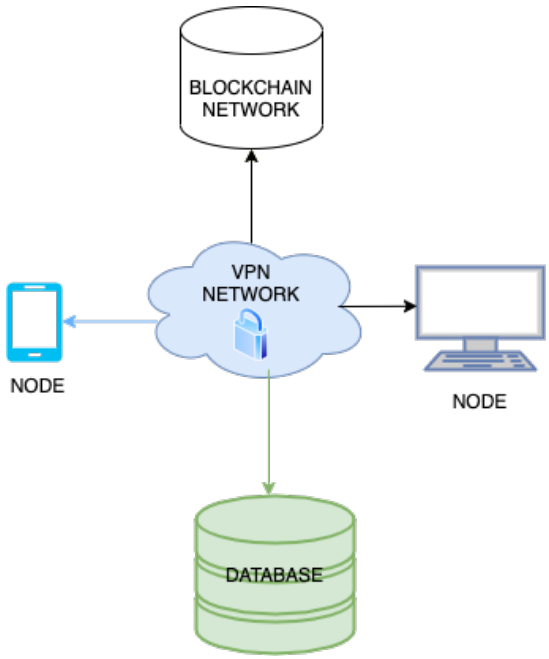

Figure 2: Components

5 DESCRIPTION OF IMPLEMENTATION OF BLOCKCHAIN WITH JAVA PROGRAMMING

Our work was to implement a resident records management system by encrypting with SH-256 cryptography and implementation of blockchain from scratch.

Blockchain is meant to secure storage of all data included each block. Hashes, which function as a data security mechanism, are difficult to manipulate and thus save all important information. It is decentralized since each node has full access to and modifications to the data. This includes any modifications that are made to every hash, so that the information cannot be leaked or corrupted. Every classified data in the network may thus be securely kept in blocks, accessible and recorded by each node.

5.1 Overview

Its major components are shown in Figure 2. To send and receive data, participants have a node installed in their premise that runs the blockchain node software and web application. Virtual private networks (VPNs) connect all nodes to the internet so that they can communicate with one another without using public IP addresses. To facilitate the clearing process, servers are installed on the premises of each utility.

5.2 Nodes

The nodes are implemented in java 11. Figure 3 illustrates the fundamental architecture of our node implementation. The participants

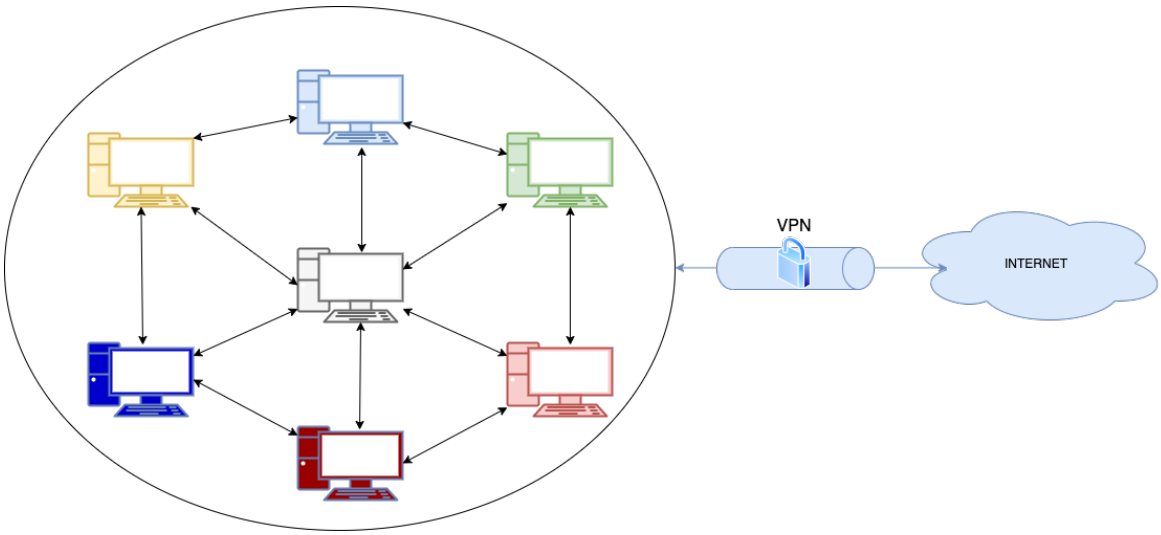

Figure 3: Fundamental architecture of nodes

will need to provide private key and public key, as well as user name to communicate with each other. These are required. The nodes must communicate with each other so that everyone has the same state of the blockchain. The pee-to-peer approach has established itself to ensure that this works with a number of participants. With this approach, all nodes in the network have the same status and communicate with each other without a central authority. As soon as a node receives new information, such a new transaction or a new block, it then sends the information to all other nodes. Each node is both a server and a client, used to synchronized the data of each node.

5.3 Data management

Blockchain technology cannot store a large size of data. Therefore we divide the data in two: essential and non-essential data. An essential data can be “name”, “reference number”, “social security number”; and a non-essential data can be “date of birth”, “place of birth”. The administrator can select some fields as essential, these fields will be packaged into a block and stored in the blockchain, while unchecked non-essential perform the SHA256 procedure and store their hash in the block with the essential fields. All of this information is stored in RDMS. The data in the blockchain makes the essential data tamper-proof and traceable. The non-essential is stored the RDMS to reduce redundancy, and the hash result of the non-essential data is stored in the block. The SHA2546 procedure is done, and the result is compared to that previously saved in the blockchain. If they are same, then the data has not been altered [12].

It is possible to have data of any length hashed using the SHA256 algorithm [13]. This unique numerical representation of data is known as a “hash value”.

Therefore, if data has been updated, no matter how little of a change, the final hash value will be entirely different. The hash value of data may be used to check its integrity. Figure 4 is the SHA-256 encryption code with Java programming language.

5.4 Web application

A web application is available for user interaction. User-friendliness and abstraction between the blockchain’s underlying complexity and the high-level user interaction are the primary goals of this application. There are nodes and database connected to the application. It is used to send new transactions to the network and to receive information about incoming portions. A list of blocks is requested, and transactions related to a particular user are then filtered out. As a result, users can see how many portions the have at any given time.

5.5 Network setup

Nodes communicate with each other over, the web app over TCP/IP protocols. We use the WebSocket protocol which is essentially a TCP-based protocol. It first initiates a special request through the HTTP/HTTPS protocol for handshake and then creates a TCP connection for exchanging data, and then the server and the client communicate in real time through the TCP connection.

The node’s API defines three main types of messages:

(i)Block, (ii) Transaction and (iii) status. It is possible for a block message to propagate a newly mined block or to respond to a block request from another node. The web app sends transaction messages, which cause portions to be shifted. As well as being used for debugging, status messages are used to inform users of the node’s current status.

In our permissioned blockchain, the public keys of all nodes are known to one another and messages with invalid signatures are discarded.

In order to establish a secure connection between the server and the web app, HTTPS is used. Over private virtual network (VPN), all components of our system communicate with one another. The VPN simplifies communication on the network layer because the nodes are located on the different premises, where unchanging IP addresses are not guaranteed and network address translation (NAT) is common.

```
public String calculateBlockHash() throws UnsupportedEncodingException{
    String hashData = prevBlockHash+Long.toString(timeStamp)+Integer.toString(nonce)+transactions;

    MessageDigest digest = null;
    byte[] bytes = null;

    try {
        digest = MessageDigest.getInstance("SHA-256");
        bytes = digest.digest(hashData.getBytes("UTF_8"));
    }catch (NoSuchAlgorithmException | UnsupportedEncodingException ex){
        logger.log(Level.SEVERE, ex.getMessage());
    }

    StringBuffer buffer = new StringBuffer();
    for(byte b : bytes){
        buffer.append(String.format("%002x", b));
    }
    return buffer.toString();
}
```

Figure 4: : SHA-256 encryption Java code

6 CONCLUSION

Using blockchain to manage data for national records was the topic of our presentation. We described our system's architecture, as well as some the specifics of how it was implemented. As a result, we came up with the following conclusion: while Byzantine Fault Tolerance algorithms can improve scalability when the number of users is small, TLS handshakes for secure communication links introduce additional delays; one communication thread is not enough for asynchronous peer-to-peer communication; resynchronization of node states is difficult, especially in small networks with high entropy levels. A blockchain is difficult to implement from scratch, but it is doable for our use case.

REFERENCES

[1] https://www.mckinsey.com/business-functions/mckinsey-digital/our-insights/using-blockchain-to-improve-data-management-in-the-public-sector
[2] S. Nakamoto, Bitcoin: A Peer-to-Peer Electronic Cash System. Technical report (2008). https://bitcoin.org/bitcoin.pdf
[3] F. Tschorsch, B. Scheuermann, Bitcoin and beyond: a technical survey on decentralized digital currencies. IEEE Comm. Surv. Tutor. 18(3), 2084-2123 (2016)
[4] K. Christidis, M. Devetsikiotis, Blockchains and smart contracts for the Internet of Things.
[5] IEEE Access. 4, 2292-2303 (2016)
[6] F. Knirsch, A. Unterweger, G. Eibl, D. Engel, in Sustainable Cloud and Energy Services: Principe and Practices. Chap. 4, ed. By W. Rivera. Privacy-Preserving Smart Grid Tariff Decisions with Blockchain-based Smart Contracts (Springer, Cham, 2017), pp. 85-116
[7] Mathis Steichen and al., Blockchain-based, Decentralized Access control for IPFS. 2018 IEEE International Conference on Blockchain.
[8] André Henrique Mayer and al., Electronic health records in a Blockchain: A systematic review, SAGE (September 2019).
[9] G. Greenspan, MultiChain Private Blockchain – White Paper. Technical report, Coin Sciences Ltd (2015). http://www.multichain.com/download/Multichain-White-Paper.pdf
[10] G. Wood, Ethereum: A Secure Decentralized Generalized Transaction Ledger. Technical report, Ethereum (2017). https://ethereum.github.io/yellowpaper/paper.pdf
[11] E. Androulaki, A. Barger, V. Bortnikov, C. Cachin, K. Christidis, A. De Caro, D. Enyeart, C. Ferris, G. Laventman, Y. Manevich, *et al.* Hyperledger Fabric: A Distributed Operating System for Permissioned Blockchains, in: Proceedings of the Thirteenth EuroSys Conference, ACM, 2018, p. 30.
[12] A. Unterweger, F. Knirsch, C. Leixnering, D. Engel, in 2018 9th IFIP International Conference on New Technologies, Mobility and Security (NTMS). Lessons learned from Implementing a Privacy-Preserving Smart Contract in Ethereum (IEEE, Paris, 2018)
[13] M. Di Pierro, What is the blockchain?, Comput. Sci. Eng. 19 (5) (2017) 92-95.
[14] D. Rachmawati, J. Tarigan, A. Ginting, A. comparative study of message digest 5 (md5) and sha256 algorithm, in: journal of Physics: Conference Series, Vol. 978, IOP Publishing, 2018, p. 012-116
[15] J. Bucko, D. Palova, M. Vejacka, in Central European Conference In Finance And Economics. Security and Trust in Cryptocurrencies (Technical University of Kosice, Herlany, 2015), pp. 14-24
[16] A. Unterweger, F. Knirsch, C. Leixnering, D. Engel, Update: Lessons Learned from Implementing a Privacy-Preserving Smart Contract in Ethereum. Technical report, Center for Secure Energy Informatics, Salzburg University of Applied Sciences, Austria (Nov 2017). http://www.en-trust.at/papers/Unterweger18a-t.pdf
[17] F. Knirsch, A. Unterweger, K. Karlsson, D. Engel, S. B. Wicker, Update: Evaluation of a Blockchain-based Proof-of-Possession Implementation. Technical report, Center for Secure Energy Informatics, Salzburg University of Applied Sciences, Austria (2018). http://www.en-trust.at/papers/Knirsch18a-t.pdf
[18] M. Hearn, Corda: A distributed ledger, Version 0.5. Technical report. Corda. Accessed September 2018 (2016). https://www.corda.net/content/corda-technical-whitepaper.pdf
[19] A. Kosba, A. Miller, E. Shi, Z. Wen, C. Papamanthou, in 2016 IEEE Symposium on Security and Privacy (SP). Hawk: The Blockchain Model of Cryptography and Privacy-Preserving Smart Contracts (IEEE, San Jose, 2016), pp. 839-858
[20] D. Mazières, The Stellar Consensus Protocol: A Federated Model for Internet-level Consensus. Technical report, Stellar Development Foundation (2016). https://www.stellar.org/papers/stellar-consensus-protocol.pdf figu