J Inf Process Syst, Vol.17, No.5, pp.989~1003, October 2021
https://doi.org/10.3745/JIPS.03.0167

ISSN 1976-913X (Print)
ISSN 2092-805X (Electronic)

JOURNAL OF INFORMATION PROCESSING SYSTEMS JIPS

Implementation of Search Engine to Minimize Traffic Using Blockchain-Based Web Usage History Management System

Sunghyun Yu[*], Cheolmin Yeom[*], and Yoojae Won[*]

Abstract

With the recent increase in the types of services provided by Internet companies, collection of various types of data has become a necessity. Data collectors corresponding to web services profit by collecting users' data indiscriminately and providing it to the associated services. However, the data provider remains unaware of the manner in which the data are collected and used. Furthermore, the data collector of a web service consumes web resources by generating a large amount of web traffic. This traffic can damage servers by causing service outages. In this study, we propose a website search engine that employs a system that controls user information using blockchains and builds its database based on the recorded information. The system is divided into three parts: a collection section that uses proxy, a management section that uses blockchains, and a search engine that uses a built-in database. This structure allows data sovereigns to manage their data more transparently. Search engines that use blockchains do not use internet bots, and instead use the data generated by user behavior. This avoids generation of traffic from internet bots and can, thereby, contribute to creating a better web ecosystem.

Keywords

Blockchain, Data Sovereignty, EOS, My Data, Proxy Server, Search Engine, Self-Sovereign Model, Smart Contract

1. Introduction

With the current popularity of big data, AI, and cloud computing, interest in data sovereignty is also on the rise. Data sovereignty is the practice of granting rights to an individual's data, akin to the rights of an individual over his body or property, to determine when, where, how, and for what purpose his data will be used, and to know why it is used when necessary [1]. Many companies provide services using the aforementioned technologies. Among them, representative web search engine service providers maximize their profits by providing advertising services and customized services [2]. These customized services are provided by tracking browsing information as well as keywords retrieved from the search engine service, storing them in their database, and analyzing them. This information is then used to improve the quality of the customized services. However, the measure of quality is often determined by

※ This is an Open Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License (http://creativecommons.org/licenses/by-nc/3.0/) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.

Manuscript received February 4, 2020; first revison April 8, 2020; accepted April 14, 2020.

Corresponding Author: Yoojae Won (yjwon@cnu.ac.kr)
[*] Dept. of Computer Science Engineering, Chungnam National University, Daejeon, Korea (yoursaint@cnu.ac.kr, cjfals18@cnu.ac.kr, yjwon@cnu.ac.kr)

Copyright© 2021 KIPS

the profitability of the service instead of the extent of data sovereignty it affords. The same principle applies to methods of data storage as well. Several internet companies store and use data collected from users in their databases. As the storage database is centralized, the data owner is not at a position of control with regard to his data. This means that decision-making regarding manipulation and use of data is possible irrespective of the opinion of the sovereign and that the sovereign may not even be aware of the decision [3].

It is, thus, necessary to address these issues to ensure data sovereignty, especially to keep up with the increased interest in this topic. Additionally, existing web search engine services suffer from traffic problems. In such services, information is generally collected using internet bots, which are automated information gathering software. Internet bots are used to collect information on the web as they simulate the web surfing of a real user. Web search engine service providers use internet bots to gather information to build a database. This enables them to perform search engine functions like transmitting queries to the database and associating them to the most appropriate links. In order to provide such a service, the database must remain updated, requiring internet bots to visit various servers frequently. This generates a significant amount of traffic [4]. Consequently, a server in a poor environment may be paralyzed by the traffic generated by several internet bots, making it difficult for it to provide services to real users [5]. In fact, according to a 2018 report, the proportion of bot traffic is as high as 42.2% of the total web traffic [6]. With the rapid increase in the number of devices connected to the Internet [7], efforts to reduce the traffic generated by Internet bots are becoming indispensable for the efficient use of web resources, making solving the aforementioned problems imperative. This study proposes a method that bypasses the use of internet bots and makes web resources more efficient via a website usage record management system that guarantees data sovereignty and a website search engine built using the website. The central idea is to use blockchain technology, Smart Contracts, and proxies to build a website usage record management system, to build a website search engine using the data collected by this system and to provide it as a web service. Using this method, the user can first trace and control his/her web usage records to ensure data sovereignty. Second, the use of Smart Contracts allows users to control and collect website usage records and thereby increases efficiency by automating the processing of payments between users [8]. Third, if a web search engine is constructed using the collected data, a database can be built without the use of internet bots. This can contribute to the reduction of the overall web traffic. In the following sections, we introduce related studies and describe the detailed method of construction of the proposed website usage record management system and the method of implementation of the proposed website search engine using the database.

2. Related Work

2.1 Self-Sovereign Model

The self-sovereign model is presented in detail following an overview of the centralized model. Existing centralized models suffer from the problem that the control over user’s data resides between the issuer and the verifier, thus implying that decisions about the data can be made without the involvement or the cognizance of the user (Fig. 1).

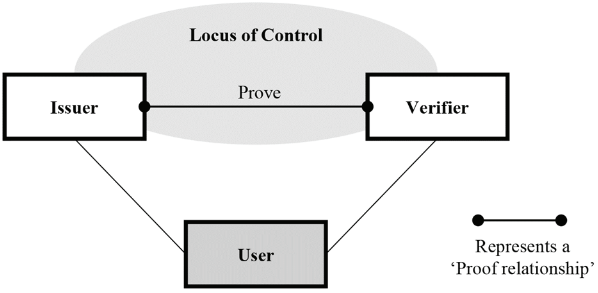


Fig. 1. Centralized model in the field of identity verification.

The self-sovereign model addresses the aforementioned problem by placing the user at the center of the control locus, such that both the "issuer" and "verifier" agencies can send and receive data via the user, and the data are verified using a distributed ledger (Fig. 2). This model enables individuals to actively manage and control their information and use it in various fields as per their discretion [3,9]. In this study, the self-sovereign model adopts an approach that is different from the previously proposed centralized collection systems. The collected data lie at the heart of the locus of control so that a user can see when, how and who uses their data and where it is stored. The record is then managed via a register using a distributed ledger. In this study, the register's role is fulfilled by blockchain.

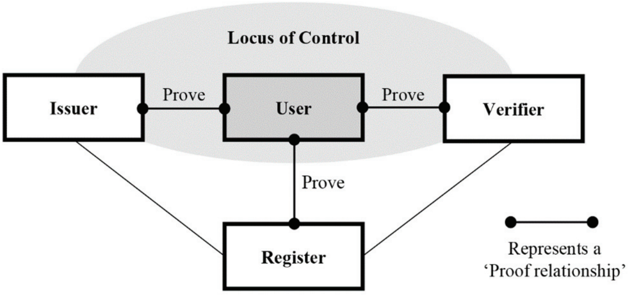


Fig. 2. Self-Sovereign model in the field of identity verification.

2.2 Blockchain

A blockchain is a type of distributed database that uses distributed ledger technology to prevent the falsification of data records via arbitrary manipulation [10] and stores data, such as the transaction history of all users, in the network using data distribution processing technology. As multiple nodes store data in a blockchain, instead of a central server, attacks on it are futile unless the majority of the participants' data are tampered with.

Blockchain is composed of interconnected blocks. A block comprises two parts: a block header and a transaction. The block header comprises six pieces of information: the hash value of the previous header, the version, and the time. If a block contains basic information, then there is key information present in the transaction [11] (Fig. 3).

Blockchain has been used to construct singular global financial units based on decentralization, beginning with the first-generation blockchain, Bitcoin, however, it has not gained wide acceptance in the financial sector owing to its slow transaction speed, low scalability, and lack of positive consensus.

However, the second-generation blockchain, Ethereum, has become a platform, not just for financial transactions, but also for contract automation using Smart Contracts. Recently, third generation blockchains have been popularized and applied throughout the society based on positive changes in consensus algorithms, transaction processing speeds, and availability in various programming languages.

This study uses blockchain to construct a decentralized system. Proof-of-work between nodes is used to establish data integrity, and Smart Contracts are used to store or manage data on the blockchain. All histories in the system, such as blocks created or transactions for processing data using Smart Contracts, are stored in blocks and linked together in the order in which they are produced, thus enabling traceability of the data. In addition, tokens generated during the proof-of-work of the blockchain can be used to reward system contributors. These tokens are also the terms of the Smart Contracts for collection and are used by sovereign owners of the collected data to receive just compensation.

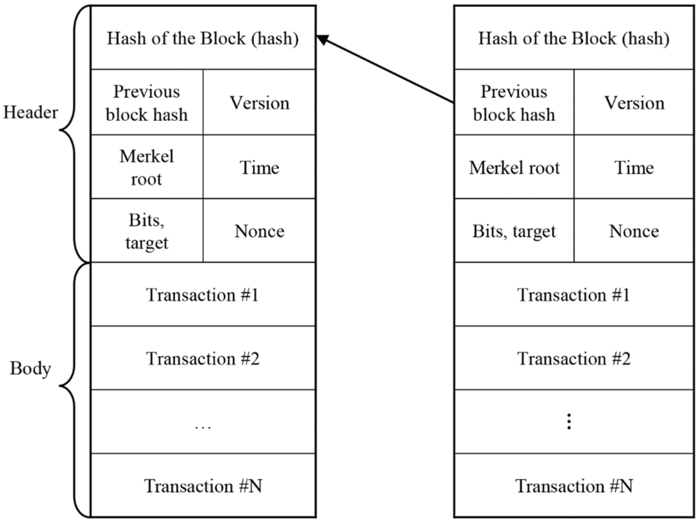


Fig. 3. Block structure of blockchain.

2.3 EOSIO

EOS is a third-generation cryptocurrency that uses delegated proof-of-stake (DPoS), which is a type of proof-of-stake (PoS). In the pure PoS format, the number of shares owned by a miner is directly proportional to its likelihood of mining new blocks [12]. The difference between PoS and DPoS lies in the fact that DPoS chooses block producers (BPs) to achieve consensus by delegating their stakes, which is environmentally friendly and guarantees high performance, because it is more energy efficient than consensus through proof-of-work, as is common in other consensus algorithms. Based on blockchain technology, EOS is a decentralized platform designed to support decentralized applications that can develop Smart Contracts about 200 times faster than second generation cryptocurrencies without any fees. It transforms complex addresses into accounts that are easy to understand. This makes EOS faster than other blockchains for decentralized applications (DApps) [13]. The reasons for choosing EOS over other blockchain protocols in this study are as follows. The difference between EOS and other technologies is the use of DPoS as previously explained. As described previously, it is environmentally friendly and exhibits high performance in transaction processing. Consensus algorithms, such as proof-of-work used in traditional blockchain technology, require a lot of time and resources. People using web search services are not generous enough to wait for the required amount of time and are unwilling to allow their computer resources to be unnecessarily consumed for the process. This is a major reason to choose EOS over other blockchain technologies to collect, manage, and link web usage records in less time. Further, if DPoS is used via EOS, it enables anyone to provide services using the data on the blockchain while simultaneously allowing users to vote for service providers who provide higher quality or desired services. Service providers that are aware of this will, in turn, work to improve the quality of their services in order to receive greater number of votes. This will naturally help service providers to enhance the overall quality of the system.

2.4 Smart Contract

The Smart Contract is a concept originally proposed by Nick Szabo in 1994 to create a contract using digital instructions and automatically execute the contract according to the terms [14]. Digital contracts are clear in terms of contractual results and can be implemented at right angles. The Bitcoin script embodies this on the basis of blockchain and automatically performs transactions based on pre-set conditions. However, allowing loops within a contract makes it vulnerable to DOS attacks. Smart Contracts can now be used to solve the aforementioned problem in Ethereum, and to write contracts in a high-level language. It can further be used as a means of creating various applications beyond simple financial contracts, on the blockchain platform [15]. In this study, Smart Contracts are used to record the user's web usage records collected by proxy in EOS's multi-index database, leaving the history on the blockchain and investigating the preferences of the websites accessed. We also include tokens in the terms of our Smart Contract, as described above, to ensure that users receive fair compensation from the service providers who wish to access their data. Similar to the system proposed in this study, the ones constructed using Smart Contracts can be easily expanded to deliver various services.

2.5 Proxy

A proxy is a computer system or application that allows a client to indirectly access other network services through it. It is called a proxy because it participates in communication on behalf of both the server and client as a relay. Therefore, this technology can keep the proxy user anonymous and prevent data leakage and possible threats. Additionally, by inserting relevant scripts, associated services can be provided via software installed on the proxy server, without installing any software on the user's computer [16]. In this study, proxy is used to provide a web proxy environment. The web proxy is implemented so that the users can log in to their blockchain wallet using a session. The proxy stores a log of the web surfing activity performed using its service in a transaction and sends it to the blockchain network. This generates a Smart Contract and if the user is logged into his blockchain wallet, then he himself becomes the actor of that record will be able to check his history or manage his data. Users may also obtain appropriate compensation based on the terms of Smart Contracts.

2.6 MongoDB

MongoDB is a new type of database designed to overcome the limitations of relational databases. It is a NoSQL database that has been designed to store, process, and manage big data, typically characterized by 3V (volume, variety, velocity) [17]. Unlike existing databases, the schema is free and stores the data in a document in JSON format. When the data are stored, the user employs collections to group similar documents together. This grants the system greater query efficiency than relational databases.

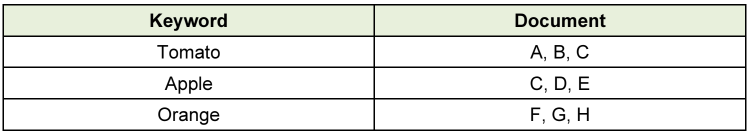
Any type of data can be stored on the MongoDB database, and it has the advantage of being intuitive and exhibits excellent reading and writing performances. However, its performance suffers when a transaction is required. In this study, MongoDB acts as an intermediary to increase the convenience of providing and using data stored in the blockchain, such as data stored in multi-index databases or transaction histories recorded in blocks. It is used to process various types of data and render them accessible in a convenient form.

2.7 Elastic Search

Elasticsearch is an open source real-time distributed search engine developed based on Apache Lucene to support JSON-based unstructured data distributed search and analysis. It concurrently employs real-time search service and distributed and parallel processing, thereby enabling the implementation of various functions via the application of a plug-in type [18].

Elasticsearch allows the user to store, search, and analyze vast amounts of data within a short period of time. Querying using a relational database involves scanning and mapping all available tables. Therefore, its performance becomes worse with an increasing number of documents. In fact, its performance as a search engine is not satisfactory owing to its slow speed even if the number of documents exceeds merely 10,000.

As is apparent from the table, if the keyword "tomato" is used in a search, the documents that are known to contain the word "tomato" in them (namely, A, B, and C) are returned (Fig. 4).



Keyword	Document
Tomato	A, B, C
Apple	C, D, E
Orange	F, G, H

Fig. 4. Table of relations between keywords and documents.

If a user inquires using the keyword "Incheon Terminal," the relational database will only search for the phrase "Incheon Terminal" and not for information about "Incheon Terminal." However, in the case of Elasticsearch, the term "Incheon Terminal" is divided into two keywords, "Incheon" and "Terminal," and all sentences including any of the two keywords, not necessarily together, are returned.

Elasticsearch supports a full query and relevance rating system. Therefore, the searched results are returned in descending order of score. When a user searches for "Incheon Terminal," the document containing "Incheon Terminal" appears ahead of documents containing "Incheon" or "Terminal."

Because of this feature, Elasticsearch is useful in providing services that need to search the contents of a database in real time or to with the use of keywords [19]. In this study, Elasticsearch is used as a search engine in a website search engine service by linking with blockchain and MongoDB, which processes and owns the data. MongoDB processes the data according to the data format of Elasticsearch, and Elasticsearch receives and processes the data to build the data structure for the search engine service. Once built, the search engine is updated periodically with data written to the blockchain. It also processes the query from the implemented search engine site and returns the results to the user.

3. Website Usage History Management System

In this paper, we propose a website record management system using blockchains and Smart Contracts and a collection system that operates with the use of proxy servers. The proposed system also uses the data stored in the system's database to provide search engine services. This solves the problem of excessive traffic from existing custom services and website search engines

3.1 EOS Private Multi-Host Multi-Node

In this system, each proxy server and search engine service, which is used for collection, possesses a single blockchain node. Each node is connected to a P2P to configure the EOS private multi-host multi-node.

In EOS, nodes are divided into BP nodes and non-BP node. BPs act as subjects to create blocks that are joined to the blockchain, and non-BPs do not create blocks. In this study, three nodes—one BP and two non-BPs—are employed. The two BPs provide proxy service and web search engine service, and their services are evaluated. The non-BP node is a history node in the management system that keeps and reads records such as transactions. Both the BPs receive and record transactions sent from users. The records are used to create blocks in a prescribed order. The BPs use the records to create relevant blocks and mutually check each other. When finished, each block is connected to the preceding block. This interaction is executed using P2P communication. The non-BP node is not capable of participating in the block check and receives the generated block data. When a user connected to the non-BP seeks transaction transfer, it acts as a relay between the user and the BP nodes (Fig. 5).

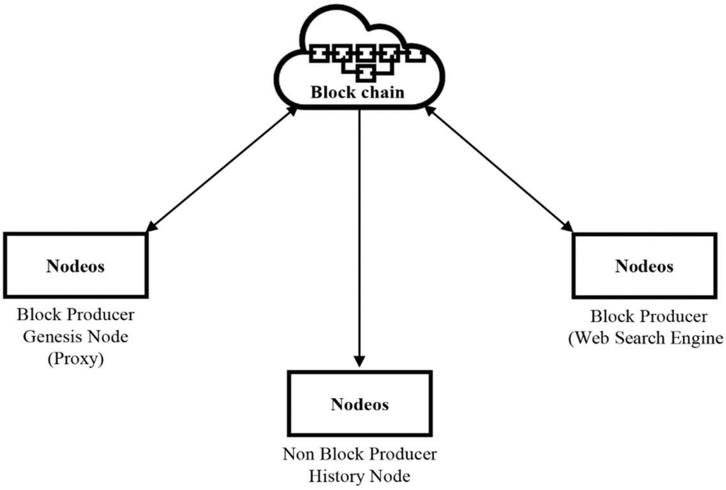


Fig. 5. Node composition in this study.

3.2 Collection System

The collection system consists of the user, a proxy server destination server, a blockchain, and a database. In this study, the proxy server is implemented at Node.js, and the destination server refers to the server of the website that the user wishes to access. When the website is accessed via the proxy server, the proxy server sends the transaction to the blockchain using a Smart Contract and records the contents. The contents are, in turn, stored for a certain period of time in the multi-index database inside the blockchain, called EOS::TABLE in EOS. Stored contents can be checked in the corresponding multi-index database within the period of time that it is stored for and can be used as a confirmation of the transaction transfer record even in the block created by the agreement (Fig. 6).

3.3 Management System

The management system is a system that operates between the blockchain and the user and manages collected information by creating a profile that determines the disposal of user information and collected information in advance for general users using Smart Contracts. It also enables historical tracking of recorded data using transaction records stored in blocks. For service provider-type users, they are responsible for providing data for third party application services (Fig. 7).

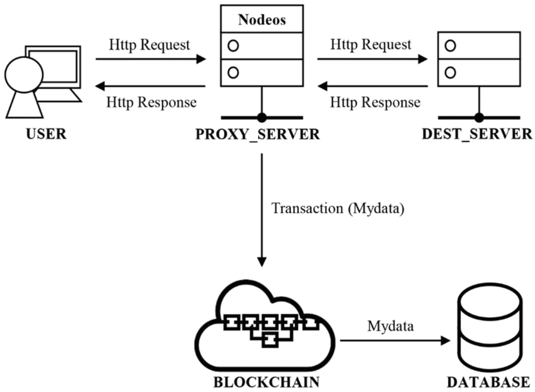


Fig. 6. Operating procedure of the collection system.

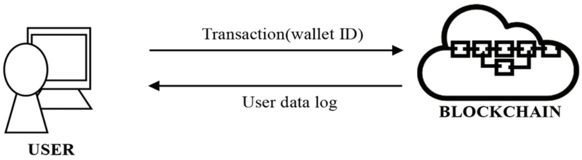


Fig. 7. Operating procedure of management system.

3.4 Search Engine using This System

We propose a search engine using this system to utilize the collected data based on user behavior stored in the system database to build a web search engine service. This type of search engine does not require the services of Internet bots. The collected data are stored in the blockchain following the aforementioned process and is not immediately suitable for use in the application. This is because data compiled using the transaction logs stored in a block has a more complicated structure than data stored in multi-index databases. To address this issue, the data are primarily stored in a DB with a flexible schema for synchronization, and the data stored in the database are synchronized with the search engine to optimize search efficiency. In this study, we use MongoDB as the DB and Elasticsearch as the search engine (Fig. 8).

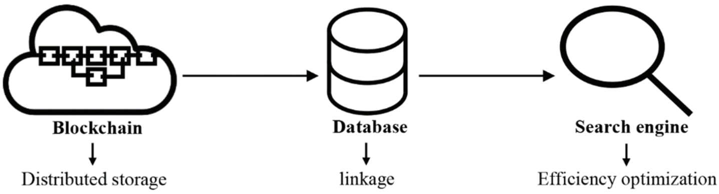


Fig. 8. Operating procedure of the search engine using this system.

4. Experiments

Users are capable of creating their own wallet account on the proposed system based on their name, coin information, and user profile. When a user creates a wallet and uses the proxy server and the search engine, the search engine collects his information, reflects it in the search results, and rewards users for this contribution. In this case, if a third party service provider provides customized service using the collected information, a Smart Contract is formed between the information provider and the service provider to reward the user. Search engines provide search capabilities and collect information about searches, wallet names, search keywords, and site addresses, and store these items in blocks. Users can also mark pages by "liking" them, which indicates the pages that they find useful. The multiple metrics inherent in this collected data can be used to calculate usability scores according to which the search results may be ranked. This helps the search engine service provide the user with the top ranked results.

4.1 Scenario

Fig. 9 depicts a diagram for explaining the operating procedure of the aforementioned search engine based on the collection, storage, and utilization methods described previously. The whole diagram depicts a scenario of the website usage record system—the upper row exhibits the operation of the blockchain-based search engine, and the lower row depicts the procedure of the storage method and the utilization method. First, the user searches for the desired information by sending a query to the blockchain-based search engine via the web-based UI. Subsequently, the user's web surfing history is collected and stored via the collection system. The collected information is recorded on the blockchain using a Smart Contract. Therefore, if a user accesses the proxy server via this process, his information can be stored in the

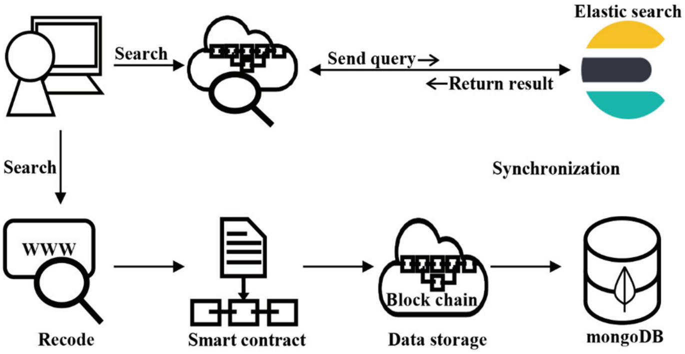
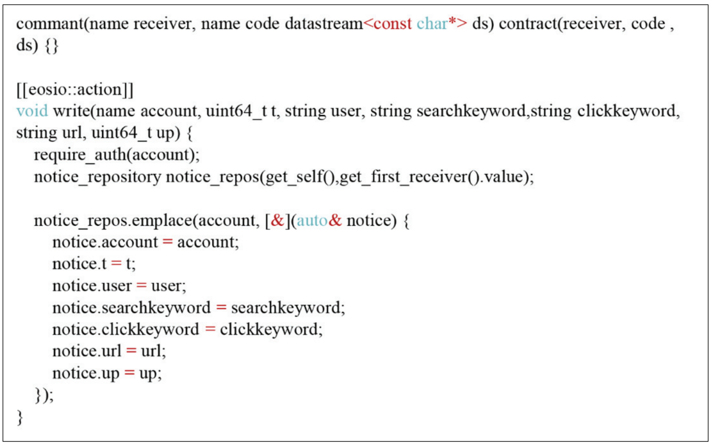


Fig. 9. Operating procedure.

blockchain, preventing its leakage into the destination server through the embedded JavaScript and thus avoiding serious compromise of user experience. At this point, the Smart Contract is executed via transaction transmission using eos.js of the proxy server and is saved in the blockchain. This information is also synchronized with the database to enable easy perusal of transaction records stored in the blockchain. Via this process, we construct a search engine (server) based on the database.

4.2 Collected Data Recording using Smart Contract

To record website usage records using Smart Contract, contract creation is required. EOS uses C++ to compose contracts, and the user can access the contract by compiling it, and subsequently converting it to web assembly, and uploading it to the blockchain network (Fig. 10). This system is based on the 1.6.x version of EOSIO.CDT (Contract Development Toolkit).



```
commant(name receiver, name code datastream<const char*> ds) contract(receiver, code ,
ds) {}

[[eosio::action]]
void write(name account, uint64_t t, string user, string searchkeyword,string clickkeyword,
string url, uint64_t up) {
   require_auth(account);
   notice_repository notice_repos(get_self(),get_first_receiver().value);

   notice_repos.emplace(account, [&](auto& notice) {
      notice.account = account;
      notice.t = t;
      notice.user = user;
      notice.searchkeyword = searchkeyword;
      notice.clickkeyword = clickkeyword;
      notice.url = url;
      notice.up = up;
   });
}
```

Fig. 10. Part of the contract created for the operation of the smart contract.

An EOS Contract is composed of eosio::contract, eosio::action, and eosio::table. In this system, eosio::table records the name of the wallet that transmitted the transaction, the time of transmission, the title of the site, and the web address of the site. In eosio::action, an action of transmitting information is encoded. In this action, it is verified that the transaction sender and the owner of the record are the same when the wallet information is used. Subsequently, the repository of the table that was created first is searched, and the data are recorded using the emplace function if the corresponding index is empty.

The implemented contract is compiled using eosio-cpp and registered using cleos. This makes it possible to use contracts registered in blockchains based on transactions.

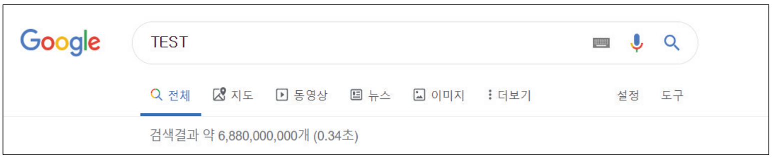
In the code that implements the proxy server, another code is included to collect the URLs visited by the user and transmit this information to the server. Collected URLs are processed in the table form of the created contract and transmitted using Eos.js. When Eos.js transmits a transaction to the blockchain network, the contract is activated and written to eosio::table, which leaves a transaction record in the block.

4.3 Collection of Website Usage using Collection System

The proxy server is used to collect website usage records. First, the proxy server transmits transactions to the blockchain network, and a Smart Contract that stores the collected information is created in advance. Via this process, the collected website usage records can be recorded in the blockchain. The unblocker module is used to build a proxy server using Node.js and insert the scripts.

First, the user connects to the destination server via the proxy server. No separate program installation is required for this step. The proxy_server sends a request to the destination server and receives a response corresponding to the request of the user. At this point, website usage record collection and JavaScript for proxy service are performed and the response is sent to the user.

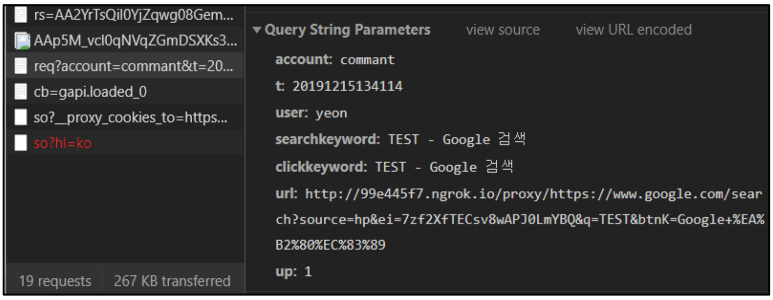
The user continues web surfing based on the response received. At this time, JavaScript collects the URLs accessed by the user via the proxy site (Fig. 11).



검색결과 약 6,880,000,000개 (0.34초)

Fig. 11. Access via proxy site of the user client.

The URLs collected by JavaScript are transmitted back to the proxy server via the GET method (Fig. 12). The proxy server receives this and transmits the transaction to the blockchain utilizing Node.js and Eos.js.



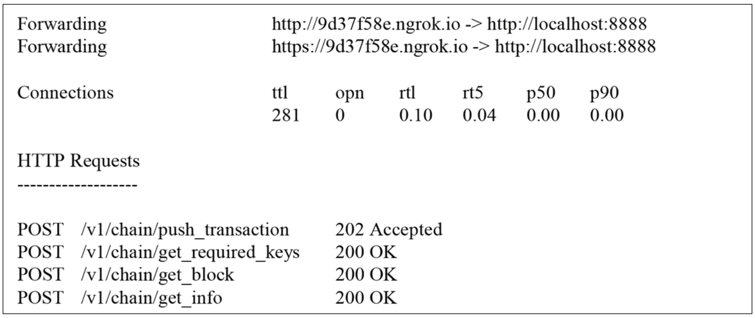
```
rs=AA2YrTsQil0YjZqwg08Gem...
AAp5M_vcl0qNVqZGmDSXKs3...
req?account=commant&t=20...
cb=gapi.loaded_0
so?__proxy_cookies_to=https...
so?hl=ko

19 requests | 267 KB transferred
```

```
▼ Query String Parameters     view source     view URL encoded
   account: commant
   t: 20191215134114
   user: yeon
   searchkeyword: TEST - Google 검색
   clickkeyword: TEST - Google 검색
   url: http://99e445f7.ngrok.io/proxy/https://www.google.com/sear
   ch?source=hp&ei=7zf2XfTECsv8wAPJ0LmYBQ&q=TEST&btnK=Google+%EA%
   B2%80%EC%83%89
   up: 1
```

Fig. 12. Client sends data to proxy server using GET method.

Transactions sent to the blockchain network execute the Smart Contracts to record website usage, and the Smart Contracts record this information in the eosio::table, a chain and multi-index database (Fig. 13).



```
Forwarding                    http://9d37f58e.ngrok.io -> http://localhost:8888
Forwarding                    https://9d37f58e.ngrok.io -> http://localhost:8888

Connections                   ttl     opn     rtl     rt5     p50     p90
                              281     0       0.10    0.04    0.00    0.00

HTTP Requests
-------------------

POST   /v1/chain/push_transaction      202 Accepted
POST   /v1/chain/get_required_keys     200 OK
POST   /v1/chain/get_block             200 OK
POST   /v1/chain/get_info              200 OK
```

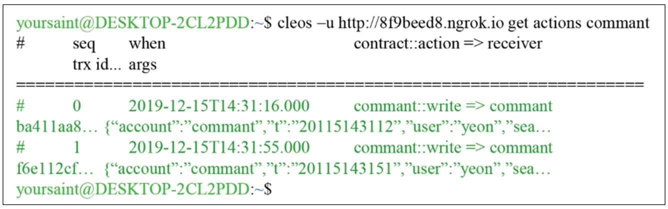
Fig. 13. Screen displaying a blockchain node web server accepting the transaction sent by a proxy server.

4.4 Results

If the blockchain node web server approves the transaction sent by the proxy server using Eos.js, the transaction is transmitted to the blockchain network (Fig. 14).

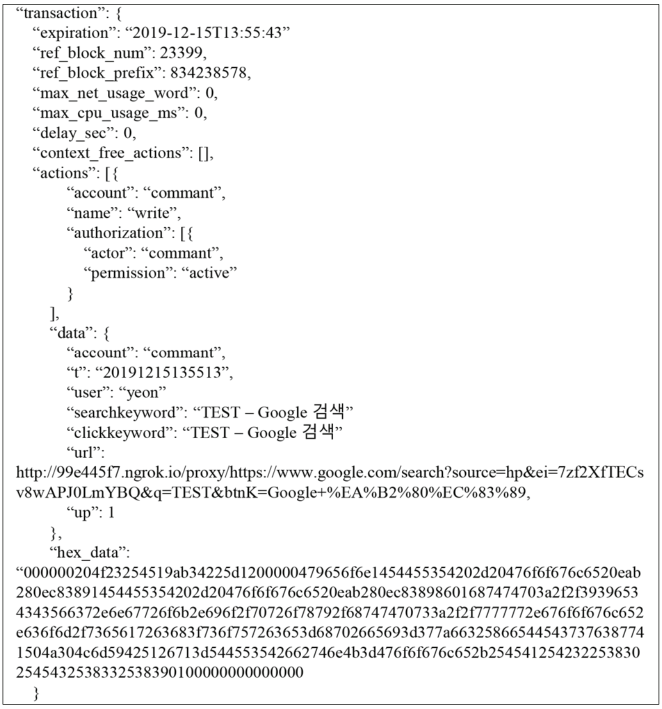
This transaction triggers the Smart Contract, and the data are recorded in the eosio::table; furthermore, the history is recorded as a transaction record in the generated block (Fig. 15).

If the data are recorded in a block in this fashion, the data can be checked via the search engine using this system and a search function may be performed (Fig. 16).



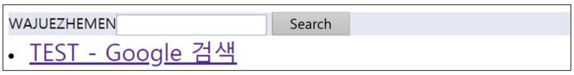
```
yoursaint@DESKTOP-2CL2PDD:~$ cleos –u http://8f9beed8.ngrok.io get actions commant
#         seq       when                                contract::action => receiver
          trx id... args
==================================================================
#         0         2019-12-15T14:31:16.000             commant::write => commant
ba411aa8… {“account”:”commant”,”t”:”20115143112”,”user”:”yeon”,”sea…
#         1         2019-12-15T14:31:55.000             commant::write => commant
f6e112cf… {“account”:”commant”,”t”:”20115143151”,”user”:”yeon”,”sea…
yoursaint@DESKTOP-2CL2PDD:~$
```

Fig. 14. Screen displaying transaction sent by the proxy server.



```
“transaction”: {
   “expiration”: “2019-12-15T13:55:43”
   “ref_block_num”: 23399,
   “ref_block_prefix”: 834238578,
   “max_net_usage_word”: 0,
   “max_cpu_usage_ms”: 0,
   “delay_sec”: 0,
   “context_free_actions”: [],
   “actions”: [{
         “account”: “commant”,
         “name”: “write”,
         “authorization”: [{
            “actor”: “commant”,
            “permission”: “active”
         }
      ],
      “data”: {
         “account”: “commant”,
         “t”: “20191215135513”,
         “user”: “yeon”
         “searchkeyword”: “TEST – Google 검색”
         “clickkeyword”: “TEST – Google 검색”
         “url”:
http://99e445f7.ngrok.io/proxy/https://www.google.com/search?source=hp&ei=7zf2XfTECs
v8wAPJ0LmYBQ&q=TEST&btnK=Google+%EA%B2%80%EC%83%89,
         “up”: 1
      },
      “hex_data”:
“000000204f23254519ab34225d1200000479656f6e1454455354202d20476f6f676c6520eab
280ec83891454455354202d20476f6f676c6520eab280ec838986016874747 03a2f2f3939653
4343566372e6e67726f6b2e696f2f70726f78792f68747470733a2f2f7777772e676f6f676c652
e636f6d2f7365617263683f736f757263653d68702665693d377a6632586654454373763877 4
1504a304c6d59425126713d544553542662746e4b3d476f6f676c652b254541254232253830
25454325383325383901000000000000000
   }
```

Fig. 15. Example of transaction content in a block.



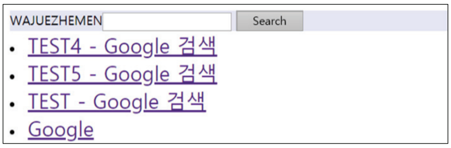
WAJUEZHEMEN [ ] Search

- TEST - Google 검색

Fig. 16. Screen displaying transmitted data being reflected in the search engine.

In a database, information increases with the increase in the amount of data written in a block. In a blockchain, the amount of information stored in a block is proportional to the quantity of transactions transmitted for the Smart Contract, which increases as more of the user's web behavior is recorded (Fig. 17).

Generally, search engines index the retrieved web documents based on the popularity of the pages on the web [20]. In the proposed system, we provide a more efficient search engine service by processing increased amounts of data and calculating the corresponding usability scores. The number of responses returned by a search engine increases with the amount of information stored in its database. Consequently, the usability score calculated based on a high amount of data can be used to recommend more useful information to the user (Figs. 18 and 19).

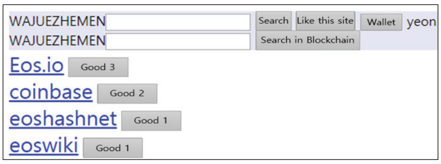


- TEST4 - Google 검색
- TEST5 - Google 검색
- TEST - Google 검색
- Google

Fig. 17. Increase in search engine data via service use.

Wallet id	keyword	Url (Example)	Like score
A,C,D	Eos	Eos.io	3
A,C	Eos	coinbase	2
A	Eos	eoshashnet	1
B	Eos	eoswiki	0

Fig. 18. Example of constructing a database using processed collected information.



Eos.io Good 3
coinbase Good 2
eoshashnet Good 1
eoswiki Good 1

Fig. 19. Website search engine built using a configured database.

5. Conclusion

In this study, we proposed a system that manages users' website records using blockchain and constructs a search engine based on this data. The management of information, reasonable payments for using the information, and minimizing unnecessary web traffic from automated programs, were also discussed. This study enables a more transparent management of website usage records, thus enabling users to check and intervene as per their requirement, and greater data sovereignty, by instituting reasonable compensation for data usage through Smart Contracts. Furthermore, the architecture maintains records of the owner of the information and its contents in the blockchain using Smart Contracts that can help to review the database in case of controversy and fraud.

A website search engine has also been proposed that operates based on users' web usage records, and thus, avoids the generation of indiscriminate traffic caused by automated programs that usual search engines rely on. This helps create a better web ecosystem. Additionally, it improves user experience by reducing the consumption of resources for service maintenance and providing a more user-friendly service.

An increase in the number of users causes an increase in the quantity of web records collected by the service, which, in turn, increases the number of generated transactions. This contributes to the further advancement of the service. Moreover, as data gathered from actual web surfing activities record more meaningful links than those gathered from bots and crawlers used by existing search engine sites, the proposed search engine is capable of provide more meaningful data.

As the proposed system depends on the behavior of its users to enrich its services, it can only provide meaningful service when the number of participants exceeds a certain threshold. However, its varied advantages offset this disadvantage. Therefore, if it is complemented meaningfully and developed using supplementary measures such as providing seed data, the proposed system can usher in an undisputed age of data sovereignty, which will, in turn, contribute to a healthier web environment.

Acknowledgement

This work was supported by the Institute for Information and Communications Technology Planning and Evaluation (IITP) grant funded by the Korean Government (MSIT) (No. 2019-0-01343, Training Key Talents in Industrial Convergence Security).

References

[1] Z. N. Peterson, M. Gondree, and R. Beverly, "A position paper on data sovereignty: the importance of geolocating data in the cloud," in *Proceedings of the 3rd USENIX Workshop on Hot Topics in Cloud Computing (HotCloud)*, Portland, OR, 2011.

[2] A. Rangaswamy, C. L. Giles, and S. Seres, "A strategic perspective on search engines: thought candies for practitioners and researchers," *Journal of Interactive Marketing*, vol. 23, no. 1, pp. 49-60, 2009.

[3] T. Bouma, "Self-Sovereign Identity, Shifting the Locus of Control," 2019 [Online]. Available: https://trbouma.medium.com/self-sovereign-identity-shifting-the-locus-of-control-10da1c8757ad.

[4] M. Zineddine, "Search engines crawling process optimization: a webserver approach," *Internet Research*, vol. 26, no. 1, pp. 311-331, 2016.

[5] P. Owezarski, "On the impact of DoS attacks on Internet traffic characteristics and QoS," in *Proceedings of the 14th International Conference on Computer Communications and Networks*, San Diego, CA, 2005, pp. 269-274.

[6] Distil Networks, "The 2018 Bad Bot Report," https://resources.distilnetworks.com/white-paper-reports/2018-bad-bot-report.

[7] R. Khan, S. U. Khan, R. Zaheer, and S. Khan, "Future internet: The internet of things architecture, possible applications and key challenges," in *Proceedings of 2012 10th International Conference on Frontiers of Information Technology*, Islamabad, Pakistan, 2012, pp. 257-260.

[8] F. Idelberger, G. Governatori, R. Riveret, and G. Sartor, "Evaluation of logic-based smart contracts for blockchain systems," in *Rules and Rule Markup Languages for the Semantic Web*. Cham, Switzerland: Springer, 2016, pp. 167-183.

[9] A. Tobin and D. Reed, "The inevitable rise of self-sovereign identity," The Sovrin Foundation, Provo, UT, 2017.

[10] H. W. Kim and Y. S. Jeong, "Secure authentication-management human-centric scheme for trusting personal resource information on mobile cloud computing with blockchain," *Human-centric Computing and Information Sciences*, vol. 8, article no. 11, 2018. https://doi.org/10.1186/s13673-018-0136-7

[11] S. Nakamoto, “Bitcoin: a peer-to-peer electronic cash system,” 2008 [Online]. Available: https://bitcoin.org/en/bitcoin-paper.

[12] G. T. Nguyen and K. Kim, “A survey about consensus algorithms used in blockchain,” *Journal of Information Processing Systems*, vol. 14, no. 1, pp. 101-128, 2018.

[13] EOSIO Developer Portal [Online]. Available: https://developers.eos.io.

[14] N. Szabo, “The idea of smart contracts,” 1997 [Online]. Available: https://www.fon.hum.uva.nl/rob/Courses/InformationInSpeech/CDROM/Literature/LOTwinterschool2006/szabo.best.vwh.net/idea.html.

[15] K. Christidis and M. Devetsikiotis, “Blockchains and smart contracts for the Internet of Things,” *IEEE Access*, vol. 4, pp. 2292-2303, 2016.

[16] Y. Li, P. Han, C. Liu, and B. Fang, “Automatically crawling dynamic web applications via proxy-based JavaScript injection and runtime analysis,” in *Proceedings of 2018 IEEE 3rd International Conference on Data Science in Cyberspace (DSC)*, Guangzhou, China, 2018, pp. 242-249.

[17] M. T. Gonzalez-Aparicio, A. Ogunyadeka, M. Younas, J. Tuya, and R. Casado, “Transaction processing in consistency-aware user’s applications deployed on NoSQL databases,” *Human-centric Computing and Information Sciences*, vol. 7, article no. 7, 2017. https://doi.org/10.1186/s13673-017-0088-3

[18] Sematext Blog, Elastic Search: Distributed, Lucene-based Search Engine, May 2010. https://sematext.com/blog/solr-vs-elasticsearch-differences/

[19] M. S. Divya and S. K. Goyal, “ElasticSearch: an advanced and quick search technique to handle voluminous data,” *Compusoft*, vol. 2, no. 6, pp. 171-175, 2013.

[20] P. Gupta, S. K. Singh, D. Yadav, and A. K. Sharma, “An improved approach to ranking web documents,” *Journal of Information Processing Systems*, vol. 9, no. 2, pp. 217-236, 2013.

Sunghyun Yu **https://orcid.org/0000-0001-6616-2713**

He received his bachelor’s degree in computer engineering from Chungnam National University in 2020. Currently, he is pursuing his master’s degree in computer engineering from Chungnam National University.

Cheolmin Yeom **https://orcid.org/0000-0003-1704-9081**

He received his bachelor’s degree in computer engineering from Chungnam National University in 2019. Currently, he is pursuing his master’s degree in computer engineering from Chungnam National University.

Yoojae Won **https://orcid.org/0000-0002-7706-5983**

He received his B.S. and M.S. degrees from the Department of Computational Statistics at Chungnam National University, Korea, in 1985 and 1987, respectively. He received his Ph.D. from the Department of Computer Science Engineering at Chungnam National University, Korea, in 1998. He worked on wireless Internet information security at Electronics and Telecommunications Research Institute from February 1987 to February 2001. He worked on mobile security at AhnLab from March 2001 to August 2004. He worked on incident handling and was in charge of management planning at Korea Internet & Security Agency from September 2004 to February 2014. Currently, he is currently a professor in the Department of Computer Science Engineering at Chungnam National University.