BookChain: Library-free Book Sharing Based on Blockchain Technology

Jiajie Zeng, Xiaohai Dai, Jiang Xiao, Wenhui Yang, Weifeng Hao, and Hai Jin
National Engineering Research Center for Big Data Technology and System
Services Computing Technology and System Lab, Cluster and Grid Computing Lab
School of Computer Science and Technology, Huazhong University of Science and Technology, Wuhan 430074, China
Email: jiangxiao@hust.edu.cn

***Abstract*—Modern bookcrossing leverages the mobile networks to help readers share books via convenient connection, and thus expedites the dissemination of information. However, the lack of traceability has significantly hindered the wide adoption of mobile bookcrossing, leading to loss of books. Meanwhile, the managerial inefficiency of current mobile bookcrossing systems becomes another obstacle for advancing the book circulation. To overcome these problems, we propose to leverage the promising blockchain technology with the merits of its immutability and cryptographic smart contract. In this paper, we present *BookChain*, a traceable and efficient blockchain-based inner-campus book sharing system. *BookChain* stores the complete sharing data of an interested book permanently on blockchain, such that every reader can trace the borrowing history, which reduces the potential for loss of book. *BookChain* also introduces the use of smart contract to automate the circulation of books with minimal human intervention, resulting in the improvement of efficiency. The experimental results show the effectiveness and low-cost of the proposed system under high concurrent users.**

***Index Terms*—Distributed computing, Blockchain, bookcrossing, book sharing, smart contract**

I. Introduction

With the popularization of smart phones, the increasing scale of mobile network enhances the communication and interaction between people [1]. By means of mobile network, many traditional social activities can get transformed. One of the most interesting and meaningful transformation is bookcrossing [2] (a.k.a., book sharing). Bookcrossing originated from Europe in the 1960s [2], in a way that readers shared the books they have finished reading in public places, such as benches in the parks. People who have common interests in the shared book can take it away for further reading. After finishing reading, the readers can become the next sharers. This kind of book sharing enables free dissemination of knowledge. However, this offline sharing mode requires people to grasp the target book at the fixed locations, leading to constrained circulation range. Alternatively, the advent of network provides new opportunities to solve the problem. By recording the related information of shared books on the mobile network, it is convenient for people to find the status and location.

However, there remains many challenging issues that impede the process of bookcrossing [3], which extends beyond the capacity of mobile network. The most serious problem is the confidence crisis. According to the statistic [4], 30% amount of readers may take possession of a shared book, and will not comply the rules of sharing it with the next readers. Another severe obstacle is the inefficient management of books. In current mobile book crossing systems, the process of circulation is not fully automated, because it still requires an administrator to audit most requests of users, leading to low book circulation rate. Furthermore, the stranded reading materials in the public places may suffer from the sabotage or weather condition, or even permanently damaged for further circulation. Hence, there is rising need of enriching the traditional book sharing systems with traceability and efficiency.

To face the urgent demand, we explore the potential of aggregating the newly emerging technology blockchain [5] into conventional book crossing systems. The intuition motivated us is that, by fusing distributed consensus [6], P2P networking, cryptography-based ledger, and smart contract, blockchain is envisioned to provide the promising characteristics including immutability, transparency, and decentralization [7]. In particular, the history of a shared book will be recorded in the blockchain through consensus protocol, which can be accessed and traceable by every reader. Additionally, the efficiency of book sharing can be increased by the introduction of smart contract, which encoded the pre-defined sharing rules.

In this paper, we present *BookChain*, a blockchain-based system that enables safe, efficient, and autonomous books exchanging among students on campus. *BookChain* consists of three key components: mobile application for book exchange, distributed servers and blockchain nodes for processing requests, and the underlying storage facilities to record data. Core information of books (i.e., status and location) is recorded on the blockchain. In this way, the trajectory of books can be traced reliably, thus tackling the problem of confidence crisis. All the actions related to shared books are conducted by means of smart contract, which can facilitate the circulation of books as well solve the problem of inefficient management. We implement the *BookChain* based on the FISCO BCOS platform [8]. BCOS is a consortium blockchain that incorporates the advantages of Ethereum [9] and Hyperledger Fabric [10], whose effectiveness has been well validated in multiple enterprise application scenarios [11].

In summary, the contributions of our work are as follows:

- To the best of our knowledge, this is the first work to address unreliability and circulation inefficiency of mobile

978-1-7281-5212-7/19/$31.00 ©2019 IEEE
DOI 10.1109/MSN48538.2019.00051

booksharing systems by the introduction of blockchain technology.
- We design and implement *BookChain*, a traceable and efficient bookcrossing system with the power of blockchain to achieve inner-campus book sharing by using a dapp.
- Experimental results demonstrate that *BookChain* can effectively process requests of bookcrossing under large-scale concurrent users with low-cost.

The rest of the paper is organized as follows. Section II introduces the problems in bookcrossing. The design of *BookChain* is presented in Section III. Section IV shows the implement details of the system. Section V gives evaluations of the system. Section VI concludes the paper.

II. Problems In Bookcrossing

The emergence of Internet has accelerated the popularity of mobile bookcrossing, providing a more convenient way of sharing books among readers. After readers register online and encode the books, the subsequent circulation of the books can be tracked. Nevertheless, some problems still exist in bookcrossing.

The primary problem in bookcrossing is that users do not trust other participants in the activity (i.e., confidence crisis) [12], which results in low participation rate. According to the statistics on the *tspl* bookcrossing website [1] : there are 6268 registered users and 4205 books on the website, only 10% of users have online records of participating in bookcrossing. Not everyone is willing to contribute their precious books to bookcrossing because they think other readers cannot properly preserve the books in the circulation.

Another problem of bookcrossing is lacking effective supervision [13] and protection measures (i.e., inefficient management). Whether the readers report the book' status to the owner depends entirely on the their integrity, which is very easy to cause the loss of books. Readers often ignore the protection of shared books when reading them, thus some returned books contain damage: pages-missing and scribble notes.

III. Overall Design

Through the above analysis of critical issues that impedes the development of conventional bookcrossing systems, we envision that blockchain offers the promises to tackle the problems in bookcrossing. In this section, we describe *BookChain*, a new inner-campus bookcrossing system conjoined with blockchain.

A. Overview

Based on the reading requirements of college students, we design the bookcrossing system with four fundamental objectives: (1) students can easily share their books and fully utilize the idle books, (2) students can borrow books directly without any intermediaries in order to achieve high efficiency circulation, (3) enable the information of books circulation transparent and build the dependable bookcrossing, and (4) the storage of circulation information needs to be highly traceable to minimize the probability of book loss.

[1]http://www.tspl.cn

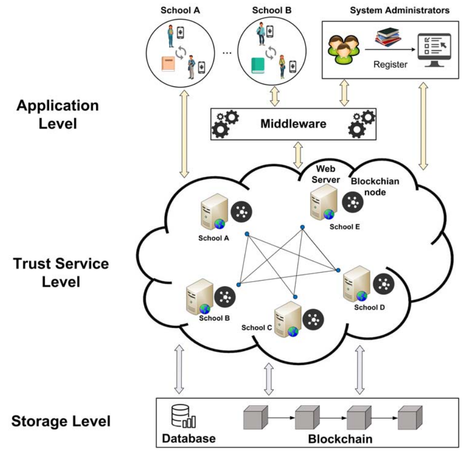


Fig. 1: The architecture of *BookChain*

The system architecture of *BookChain* is illustrated in Fig. 1, which consists of three levels: Application, Trust Service, and Storage. The Application Level comprises a set of applications designed for end-users to manipulate the system conveniently. The Trust Service Level provides the core functions of bookcrossing to users (i.e., borrowing books). The Storage Level stores the data securely and permanently.

However, there exist several challenges when we reconcile blockchain technology into traditional bookcrossing:

- *How to make user communicate with blockchain network easily?* There are no handy tools to operate the blockchain, due to its inherent design defects. User needs to repeatedly type the commands to obtain the information on blockchain, including the status of blocks and transactions. Although we can use advanced programming languages (e.g., Solidity) to write smart contracts, the dapp still cannot directly invoke the functions in smart contracts, which is unfriendly to dapp developers.
- *How to appropriately deploy blockchain nodes in the system?* The empirical solution is that every user run a blockchain node on his smartphone, but it has two limitations: 1) Not every smartphone has enough computing power to run a blockchain node, since it will consume large storage space and occupy CPU time frequently; 2) The time of reaching consensus among users will increase significantly with the growth of users, because they need to spend more time to communicate.
- *How to quickly and accurately obtain the data in the storage level?* According to the design goals of the system, we need to store books' information on the blockchain to enhance data integrity. Nevertheless, the retrospective search is inefficient because of the nature of blockchain (i.e., chain-based structure [14]). This will be the bottleneck of the system when the user frequently queries books information. Moreover, we need to provide

sufficient search results to users so that they can choose books they are interested in.

Hereby, we present the design details in accordance to the three levels at a top-down approach, providing the solutions to the above challenges:

- We design a user-friendly middleware platform for developers to manipulate the blockchain easily. The platform not only provides web pages for users to view the status of blockchain, but also encapsulates numerous API interfaces in a *Restful* style to make the development of dapp more conveniently (III-B).
- We deploy a full node of *BookChain* in each school server. The school server will aggregate the requests from its students and faculty, and send transactions on blockchain in a delegated way. It is worthy to note that this approach is distinct from SPV nodes in Bitcoin [15], where each user needs to maintain a light ledger (III-C).
- *BookChain* employs a database as the secondary storage of the system. Book status is stored on blockchain, while the inherent information of books is recorded on the database to accelerate the search speed. Furthermore, we use the database as a cache to store the data on smart contracts (III-D). We also introduce a tokenizer that can extract the stems from the words to broaden the search results, the concrete description will be given in Section IV.

B. Application Level

The application level at the top of BookChain is designed for users to participate in bookcrossing with the friendly interface. There are two types of users in the system, so we design two applications for them respectively. The Android dapp is designed for users who want to participate in bookcrossing, and the website is only for school administrators to maintain the books in the system. We also design a middleware platform that provides flexible tools for users or developers to monitor the state of the blockchain.

Android dapp: Through the *Graphical User Interface* (GUI), user can view the book's information in detail, including the image and brief introduction. In addition, users can borrow any available book by scanning the QR code on it, and return it through the personal books management function. The dapp also provides the collection function for users to keep their favorite books in a list which can constantly trace the book's status.

Website: The website is designed for administrators of each school to manage books. Once a user donates a book to the school, the administrator will log in the website to enter the information of the book and generate a QR code which contains a unique ID for it. The administrator maintains books on the web page, including updating books' information when it changes.

Middleware platform: In the traditional development progress of blockchain applications, developers need to callback the function repeatedly through the command line to observe the running status of the smart contracts, which significantly reduces the development efficiency. To mitigate the situation, we design a middleware platform for developers to operate the blockchain friendly. The platform encapsulates *Restful-style* API interface, which is more convenient while developing dapps. Transaction data can be coded and decoded through the interface, and data details on blockchain can be displayed on multiple devices including web pages, mobile terminals and so on.

C. Trust Service Level

The trust service level is the core module of our system, which includes blockchain network and servers. This level connects the other two levels to provide the complete service for users. In this subsection, we describe three main components in this level: Web server, Smart contract, and Blockchain network.

Web server: The web server is capable of parsing the HTTP protocol. When receiving HTTP requests sent by dapp or website, the web server will process them and return HTTP responses, such as sending back HTML pages or *JSON* (JavaScript Object Notation) data. The server will call the smart contract remotely to provide bookcrossing services for users. Besides, the web server also provides the basic functionality for the system, including user registration and login, checking if the borrowed books expire. In order to achieve load balancing for large users, we assign the web server to each school, and requests from users are sent to the corresponding school's server.

Smart contract: Since smart contract does not require human involvement, it can achieve a fully automated process as long as the requirements are met. The smart contract can also save time and reduce costs. So we use smart contract to replace the role of a centralized institution like the library to realize decentralized book sharing. The proposed smart contract provides many functions, such as membership registration, borrowing, and returning books. It also records the information of users and books, we will discuss it in the Storage Level (III-D). These functions are publicly available on the blockchain, so all web servers can call the corresponding functions to process requests.

Blockchain network: The university is a relatively enclosed environment, so we tend to employ a consortium blockchain and use *PBFT* (Practical Byzantine Fault Tolerance) [16] as the consensus algorithm. However, it is impractical for each student to run a blockchain full node on his/her smartphone. On the one hand, the computing power of the smartphone is far less than the specialized server, running a full node on the smartphone may consume majority resource, which could disable other normal function. On the other hand, the time complexity of *PBFT* is $O(n^2)$, which will grow significantly as participants increased. Therefore, it becomes a major challenge for a university with thousands of students. To overcome this, we leverage each school to maintain a full node which as a delegate of its students and faculties to send transactions on blockchain networks. Nodes between schools can reach a consensus of these transactions to achieve bookcrossing.

D. Storage Level

The storage level consists of two modules: database and blockchain. With the interface provided by the storage level, the service level can easily retrieve and update the data in it.

Blockchain: We use blockchain to store the important information in the system, such as books status and users' contact info. This data indicates the book location and how we can communicate with the owner. The loss of this data will cause the book missing because we do not know how to get the book. With blockchain technology, this data are maintained by all schools, each school has a copy which can avoid a single point of failure. The data is also consistent among schools through the consensus protocol. Once the data recorded on the blockchain, it is immutable. This allows users to trust each other in the system without a third party like the library.

Database: The nature of blockchain is a distributed ledger technology, we can utilize it to store books data in the system. However, the poor performance of searching data on blockchain becomes the bottleneck of the system, since book information is retrieved frequently (e.g., title, picture). In order to accelerate the performance, we apply a relational database to store basic information of books and partial users' information. Another reason why we use databases is the significant overhead of the storage on blockchain, the more information we store, the more *gas* will be consumed by the smart contracts. From the view of security, the basic information of books is the inherent attributes, its meaningless to tamper it, so we do not need to securely store this data.

IV. Implementation

In this section, we present the running process and implementation details of *BookChain*. We have implemented *BookChain* based on FISCO BCOS which is a permissioned blockchain.

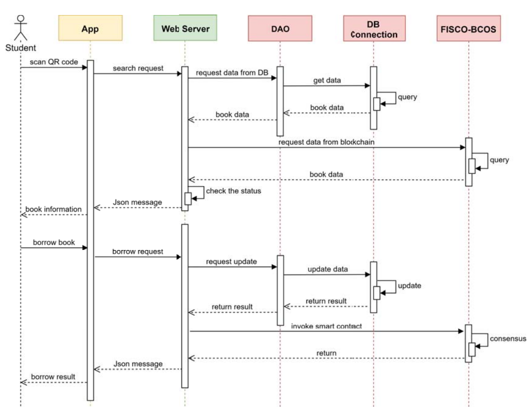


Fig. 2: The flow chart of borrowing books

A. Web Server

We use Apache Tomcat as the web server to receive the *HTTP* requests from the components in Application Level (i.e., dapp and website). Fig. 2 illustrates how the web server interacts with other components within the system when a user borrows the book. After a user scans the QR code on the book by using dapp, the dapp will send a query request to the corresponding web server. The web server will get the basic data of books from database through DAO which is an interface to access the database, then it will further get the location information from FISCO BCOS blockchain via invoking smart contract. If the book is available, the user can borrow it after the web server updates the correlative information in the database and reaches consensus on blockchain network. Any results from the web server to the dapp are in *JSON* type which can be easily decoded and transformed to user-friendly page data.

Algorithm 1 getStem

Input: $String\ originwords$
Output: $String\ transformedwords$
1: $String[]$ *wordList* $\leftarrow$ strip *originwords* by spaces
2: $String[]$ *tempList* $\leftarrow$ remove the stop words in *wordList*
3: int *l* $\leftarrow$ length of *tempList*
4: **for** $i = 0;\ i < l;\ i++$ **do**
5:     // check if the word is in the dictionary dic
6:     **if** $dic.containsKey(tempList[i])$ **then**
7:         $tempList[i] \leftarrow dic.get(tempList[i])$;
8:         continue;
9:     **else**
10:         // use porter algorithm to get the stem
11:         $tempList[i] \leftarrow porter(tempList[i])$;
12:     **end if**
13: **end for**
14: **return** $String$ *transformedwords* $\leftarrow$ splice *tempList* with spaces

---

B. Tokenizer

The tokenizer is widely used in search engines and information retrieval systems to extract the stem from a word, which can expand the scope of search terms in order to gain more search results. The traditional tokenizer based on rules (e.g., Porter Stemming [17]) can gracefully handle common words with negligible errors. However, the spared books of college students are mostly professional, and the rules-based tokenizer is not applicable for dealing with specialized vocabulary and special symbols (e.g., C++, and C#). To overcome this drawback, we implement a novel tokenizer for books searching based on Porter Stemming algorithm.

Algorithm 1 describes how to get the stems from a series of words. The algorithm first preprocesses the words by stripping the spaces between any two of them and removing the stop words which appear frequently (e.g., the), then calculates the length of the words array (Line 1-3). The main loop processes the words array. In each step, if the word exists in the dictionary that contains the pre-defined words, the algorithm will replace the word according to the dictionary (Line 7). When comes to the missing word, the algorithm will call the *poter* function to get the stem of it (Line 11). Finally, the stems of search field will be spliced and returned, and the database will query the books' information based on the stems.

C. Smart Contract

The smart contract is responsible for recording and updating the information of students and books, as well as providing the borrow and return function of books to users. When the web server receives the coming requests of bookcrossing, it will invoke the contract to add or update the information of students and books, for example, user registration or borrowing books. The bookcrossing contract consists of *StudentInfo* used for recording students' information, *BookInfo* used for recording the donor of books, and *BookStatus* used for recording the status (e.g., location) of books. This information is shared on the blockchain.

D. Authority Management

In this subsection, we demonstrate the authority management of the system. There are three roles in our system, namely super administrators, schools, and students, which are described as follows:

- **Super Administrators:** The super administrator is elected by the majority of nodes or the organizations which might have a long-established reputation, for example, departments of the school. The super administrator has all permissions such as verify the identity of organizations and grant permissions to different organizations. They also need to maintain and upgrade the system to perform normally.

- **Schools:** The schools are responsible for invoking and executing the smart contracts to store the books' or users' information on the blockchain. When a book is transferred from one student to another, school needs to invoke the contracts to update the book information in the blockchain.

- **Students:** Students are the main participants in bookcrossing activity. However, they can not access and update data in the blockchain directly. Students need to send requests to the homologous web server using dapp, and the web server will get the information on blockchain in a delegated way.

E. Consensus

The consensus process is carried by all authorized schools which are responsible for submitting the request of the transaction such as borrowing a book. Fig. 3 shows the process of whole consensus. When a student sends a request of borrowing books to schools, schools commit the transaction of updating book's information and this transaction is packaged into blocks by a leader who is elected by all nodes. Then the leader broadcasts the block to other school's blockchain nodes called follower. The followers verify all transactions in the block and

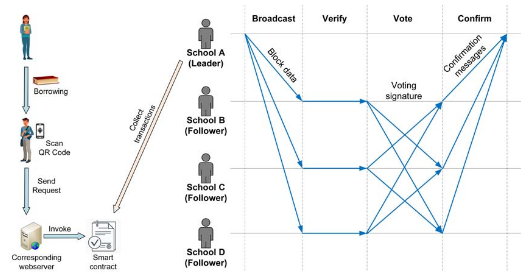


Fig. 3: The consensus process for bookcrossing

broadcast voting signature of the block to other nodes if the transaction are valid. After receiving more than 2/3 voting signatures, then each node broadcasts a confirmation message to other nodes. The block is recorded into the blockchain after all nodes receive more than 2/3 confirmation messages. After that, the book status is changed and stored on the blockchain.

V. Evaluation

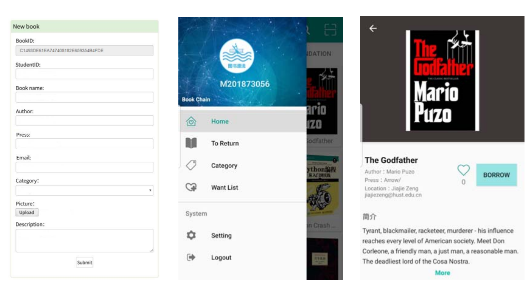
We conduct several experiments to evaluate the performance of *BookChain*. From the analyze of bookcrossing activity, we can conclude that the querying books is the most frequent action in bookcrossing, followed by borrowing and returning books. Therefore, the response times of these actions are the key indices to evaluate the performance of *BookChain*.

A. System Configuration

We test the proposed system with four machines, each of them operates one BCOS node. The machine has two 24-core Intel Xeon 8260 2.4GHz CPUs, 128GB DRAM, and 7.2TB HDD, with CentOS 7.6 operating system. We use Tomcat 7.0.94 as the web server and MySQL 5.7.27 as the database.

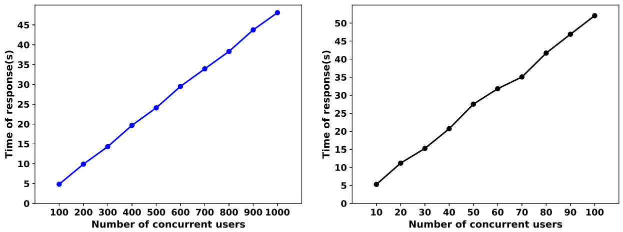
B. System Functions

Fig. 4 shows the interfaces of the system. Administrators can add a donated book to the system through the web pages as shown in Fig. 4(a). Fig. 4(b)-(c) depict the screenshots of bookcrossing dapp. Users can view the personal information and book details through the GUI, and borrow the book by clicking the 'BORROW' button.



(a) Add a new book (b) Overview of dapp (c) Book details

Fig. 4: The interfaces of *BookChain*



(a) Query performance (b) Borrowing performance

Fig. 5: The performance of *BookChain*

C. System Performance

We first test the query performance of the system, the result is shown in Fig. 5(a). As the concurrent users who send the query request of intended books vary from 100 to 1,000, the total response time after sending query requests range from 4.84s to 44.09s, which grows linearly with the request amount. In other words, the system can process approximately 22 requests per second on average, basically satisfying the user requirement in bookcrossing compared to 20 concurrent users at bookcrossing.com [2].

Fig. 5(b) shows the response time of borrowing books requests with different scale of concurrent users. However, *BookChain* can only handle two borrowing requests per second, which is substantially imperceptible to the user. The reason of such mediocre performance may be the fussy bookcrossing's business logic in web server and the network delay between server and database, since borrowing activity involves data changes.

D. Transactions Costs

We measure the *gas* cost of different transactions in bookcrossing smart contract, and Table I illustrates the results. Adding new books consumes maximum *gas* in bookcrossing, because the contract needs to create new storage space to store the book data. Compared with returning books, borrowing books expends twice as much *gas* due to the modification of more data, since book location and contact details must be changed when the user borrows it. The *gas limit* in each block of BCOS blockchain is set as three hundred million, so each block can roughly pack 3,566 adding book transactions or 7,071 return book transactions. The results show that BCOS blockchain can handle thousands of transactions per second, since it generates a new block every 500ms.

VI. Conclusion

In this paper we first discuss the status and challenges of existing mobile bookcrossing systems. To surmount the difficulties, we propose *BookChain*, a campus-oriented decentralized book sharing system based on BCOS blockchain. *BookChain* enables students efficiently share their spare books with others on campus by using the android dapp, without any intermediaries (e.g., libraries). The data and status information of books are stored on the smart contract, as is the user information. The bookcrossing (i.e., borrow and return books) makes up the transactions on the blockchain, and we make each school to run a full node of blockchain which acts as the delegate of its students to reach consensus. The information of bookcrossing is open and transparent, which enhances the trust among all borrowers. Besides, the system can track the circulation history of books and minimize the probability of book missing.

TABLE I: Gas cost of transactions

Action	**Deploy Contract**	**Add Book**	**Borrow Book**	**Return Book**
Gas Cost	3,324,074	5,255,982	84,110	42,425

Acknowledgment

This work is supported by National Science Foundation of China under Grant No. 61702203, Hubei Provincial Natural Science Foundation No. 2018CFB133. Jiang Xiao is the corresponding author.

References

[1] J. Shi, X. Wang, J. Liu, M. Zhang, and M. Huang, "Content-centric community-aware mobile social network routing scheme," in *Proceedings of the 14th International Conference on Mobile Ad-Hoc and Sensor Networks (MSN)*. IEEE, 2018, pp. 55–60.
[2] "Bookcrossing," https://en.wikipedia.org/wiki/BookCrossing.
[3] M. Corciolani and D. Dalli, "Gift-giving, sharing and commodity exchange at bookcrossing. com: new insights from a qualitative analysis," *Management Decision*, vol. 52, no. 4, pp. 755–776, 2014.
[4] "Free book-crossing movement gathers momentum in moscow," https://www.mos.ru/en/news/item/21332073/.
[5] M. Crosby, P. Pattanayak, S. Verma, and V. Kalyanaraman, "Blockchain technology: Beyond bitcoin," *Applied Innovation*, vol. 2, no. 6, pp. 71–87, 2016.
[6] Z. Zheng, S. Xie, H. Dai, X. Chen, and H. Wang, "An overview of blockchain technology: Architecture, consensus, and future trends," in *Proceedings of 2017 IEEE International Congress on Big Data (BigData Congress)*. IEEE, 2017, pp. 557–564.
[7] Z. Zheng, S. Xie, H.-N. Dai, X. Chen, and H. Wang, "Blockchain challenges and opportunities: A survey," *International Journal of Web and Grid Services*, vol. 14, no. 4, pp. 352–375, 2018.
[8] "BCOS whitepaper," https://github.com/bcosorg/whitepaper/.
[9] "Ethereum," https://www.ethereum.org/.
[10] E. Androulaki, A. Barger, V. Bortnikov, C. Cachin, K. Christidis, A. De Caro, D. Enyeart, C. Ferris, G. Laventman, and Y. Manevich, "Hyperledger fabric: a distributed operating system for permissioned blockchains," in *Proceedings of the 13th EuroSys Conference (EuroSys)*. ACM, 2018, pp. 30–45.
[11] J. Hu, D. He, Q. Zhao, and K.-K. R. Choo, "Parking management: A blockchain-based privacy-preserving system," *IEEE Consumer Electronics Magazine*, vol. 8, no. 4, pp. 45–49, 2019.
[12] R. Eidenbenz, Y. Li, and R. Wattenhofer, "Reading up on bookcrossing." *International Journal of the Book*, vol. 10, no. 2, pp. 11–26, 2013.
[13] D. Dalli and M. Corciolani, "Releasing books into the wild: Communal gift giving at bookcrossing. com," *NA—Advances in Consumer Research*, vol. 35, no. 346, pp. 1040–1059, 2008.
[14] M. Muzammal, Q. Qu, and B. Nasrulin, "Renovating blockchain with distributed databases: An open source system," *Future generation computer systems*, vol. 90, pp. 105–117, 2019.
[15] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," https://bitcoin.org/bitcoin.pdf/, 2008.
[16] M. Castro and B. Liskov, "Practical byzantine fault tolerance," in *Proceedings of the Third Symposium on Operating Systems Design and Implementation (OSDI)*. USENIX, 1999, pp. 173–186.
[17] P. Willett, "The porter stemming algorithm: then and now," *Program: electronic library and information systems*, vol. 40, no. 3, pp. 219–223, 2006.