Research Article

International Journal of Advanced Computer Research, Vol 9(40)
ISSN (Print): 2249-7277 ISSN (Online): 2277-7970
http://dx.doi.org/10.19101/IJACR.COM16003

Bspace: a group workspace over the Ethereum blockchain with off-blockchain storage

Jae-Hwan Jin[1]*, Hyun-Min Eom[2] and Myung-Joon Lee[3]
Research Scholar, School of Electrical Engineering, University of Ulsan, Ulsan, South Korea[1]
M.S. Degree Student, School of Electrical Engineering, University of Ulsan, Ulsan, South Korea[2]
Professor, School of Electrical Engineering, University of Ulsan, Ulsan, South Korea [3]

Received: 23-June-2018; Revised: 29-August-2018; Accepted: 9-November-2018

Abstract
Recently, to store a large amount of data for the blockchain decentralized applications, off-blockchain data storage techniques interworking with blockchain have emerged. However, these off-blockchain data storage techniques essentially cause the issue that it is difficult to control the user authentication and the access rights for each user outside the associated blockchain. In this paper, we introduce an implementation technique for supporting group workspaces based on the Ethereum blockchain, which provides secure file management among group users. To store large size files in group workspaces, a system structure is also being proposed where the Ethereum blockchain can stably interwork with the associated off–blockchain storage. Also, based on the file access rights stored in the Ethereum blockchain, we propose a systematic access control process for files shared among group members in the off-blockchain storage. In addition, to show the validity of the proposed technique, we present an Ethereum decentralized application called Bspace for using group workspaces interworking with off-blockchain storage.

Keywords
Ethereum, Blockchain, Group workspace, Cassandra, Off-blockchain storage.

1.Introduction
Ethereum [1], a second-generation blockchain [2] technology, supports the development of decentralized applications [3] based on blockchain beyond processing financial transactions. Ethereum uses smart contracts deployed in the blockchain network to perform functions requested by decentralized applications [4]. So, the Ethereum applications can store various information in the blockchain network or reliably use the information, while transactions for each request are recorded in the blockchain network. However, Ethereum cannot contain large amounts of data in a single transaction, thus causing problems in storing large files. Generally, Ethereum has a size limit of about 780kb for one transaction, so it is not suitable for storing high-resolution images, music, video files, etc. in the Ethereum blockchain network. Recently, to solve this problem, various developers and applications introduced methods of storing files using off-blockchain storage [5].

*Author for correspondence

Usually, in these off-blockchain storage methods, a file is stored in the associated external storage and the hash key of the file, which is the key to access the file, is stored in the blockchain. But these off-blockchain storage methods also cause an essential problem that if a hash key of a file was lost, anyone can access the file stored in the external storage using the hash key without a separate authentication procedure. To solve this problem, some developers store encrypted data in external storage. But the encryption of larger size files requires much more computation and storage, and there is still a risk of file loss. In particular, group workspaces, where frequent file sharing among multiple users occurs, are typical examples of large file sharing problems. The blockchain-based group workspace [6] has the advantage of being able to reliably perform group operations among users based on transaction history. But the off-block chain storage does not support access control techniques in association with the access rights of the members of the group workspaces. So, to utilize the off-blockchain storage more securely, it is desirable to prevent the external storage from the direct access of unauthorized users in association with access control rights of group members. In this paper, as the extension of our previous work [7], we introduce an implementation technique for supporting group workspaces based on the Ethereum blockchain, which provides secure file management among group users. To this end, we propose the system architecture for interworking off-block storage in the Ethereum blockchain, and describe the user authentication process required in the system. Based on this, we design a smart contract to support a group workspace where multiple users share files. In addition, we propose an access control method based on an Ethereum blockchain for more stable and systematic file management among the users. In addition, to show the validity of the proposed technique, we present an Ethereum decentralized application called Bspace for using the functionality of group workspaces interworking with off-blockchain storage.

2.Background

2.1Group workspace

A group workspace provides a service for a user group composed of a number of users to effectively perform resource sharing among group members. Group members can upload and download files or documents from a group workspace to share resources among themselves. Resources that belong to a group workspace can be stored in a variety of repositories, such as separate databases or cloud storage [8]. The group workspace manages the access rights information of the users in order to manage the shared resources effectively. Permissions on shared resources are generally divided into read/write privileges, and individual privileges, such as individual resources or full privileges, can be divided in detail. The owner of the group can specify this authority and the user can perform the operation based on the specified authority. Recently, a group management method using blockchain network technology has been studied to enhance the stability of the rights management and authentication of user groups.

2.2Cassandra distributed database

Cassandra [9] is an open source not only structured query language (NoSQL) distributed database designed to manage high volumes of data between many servers with high performance without a single point of failure. Cassandra can extend capacity horizontally without compromising performance by adding configuration nodes, and supports several programming languages such as Java, Python, PHP, and Ruby for use in various environments. Cassandra is often used to handle high-capacity, high-performance transactions that do not require the definition of complex relationships between data, and is now being used as a database in social network services such as Twitter and Facebook. The data of Cassandra are distributed and stored based on the hash value of the data called the partitioning key in the ring type structure composed of several nodes. In addition, Cassandra supports a variety of replication strategies to increase the reliability of data management by setting the number of nodes that act as replicas and the strategy in which replicas are placed in the ring. Clients using Cassandra can also specify a level of consistency when performing data queries to ensure that the values in the replicas are consistent.

3.Off-blockchain storage system interworking with Ethereum

In this section, the configuration of based off-blockchain storage system interworking the Ethereum blockchain is proposed. The proposed system uses the Cassandra distributed database as off-blockchain storage to support file storage service requested in Ethereum decentralized application.

3.1Off-blockchain storage solution for Ethereum

Off-blockchain storage solution is a framework that enables blockchain users to store their data outside the blockchain. In the framework, off-blockchain storage stores the actual file data, whereas hashed keys that can access the file data are kept on the Ethereum blockchain. To store files into Cassandra at the request of decentralized applications, we use the event-watch mechanism of the Ethereum smart contract. The Ethereum events help the application developers to access the Ethereum virtual machine logging facilities, which are used for calling JavaScript callbacks inside the user interface of the applications, allowing the developers to listen for the events with parameters specified in the smart contracts. The explained techniques can be depicted the system structure shown in *Figure 1*.

The proposed system consists of the gateway server [10], the representational state transfer (REST) application programming interface (API) server, and the Cassandra distributed database. The REST API server is directly connected to Cassandra, providing the RESTful web services server to store or retrieve files in Cassandra. The REST API server receives the hash key and the file requested by the user through the provided web service. After that, the REST API server accesses Cassandra after authenticating the user who invoked the Web service. The gateway server monitors events occurring in the smart contract and transmits them to the REST API server. To do this, the gateway server calls the Web service on the REST API server along with the parameters received by the event. The parameters of the event include the type of action, the information of the file to be accessed, and the requested user information. Cassandra is only managed by the REST API server to prevent external access. Cassandra stores files in key-value format, and any stored file can be accessed only through the hashed key of the file.

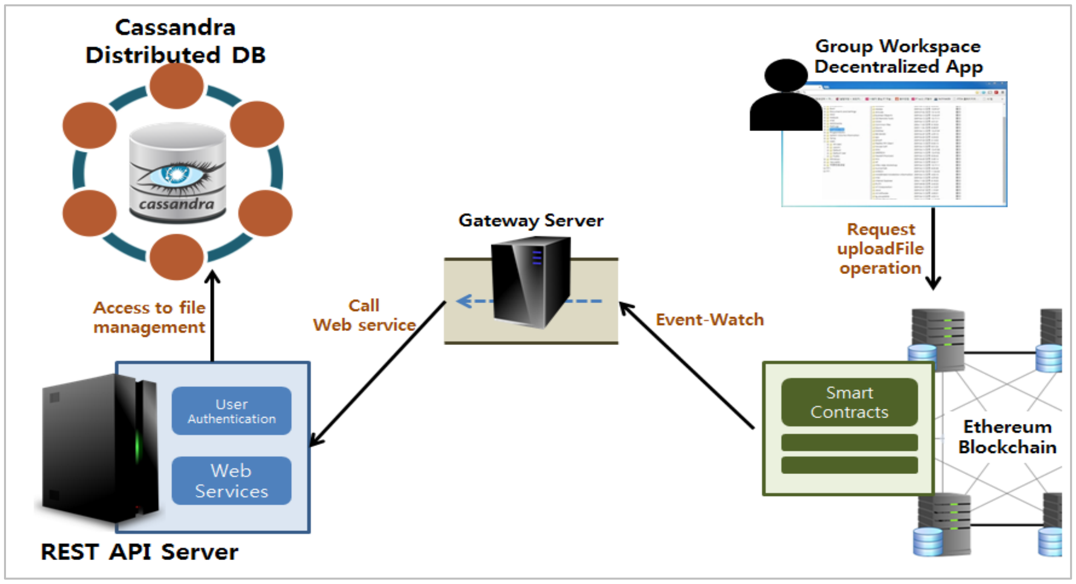


Figure 1 System structure for interworking with off-blockchain storage

3.2Process for interworking off-blockchain storage with Ethereum

Depending on the size of the file to be uploaded, the decentralized application can request that the file be stored in the Ethereum blockchain or the Cassandra database. In general, the Ethereum blockchain can store several documents or low-resolution images, whose size are from tens to hundreds of kilobytes. So, large size files should be stored in Cassandra using the proposed off-blockchain storage solution. *Figure 2* shows the activity diagram for the process of storing large files using off-blockchain storage. When a user requests a file upload, the decentralized application creates a signature using the user's account and executes the file upload function of the smart contract along with the information of the file. The file upload function of smart contract generates the file access key by hashing the user's account information, upload time, and information of the file. Then, the smart contract generates a file upload event using the signature of the user and the access key of the file. The generated event is monitored at the gateway server, and the signature and the access key transmit through the REST API server's access token generation service. The REST API server generates the access token based on the parameters and the request time when the access token generation service is invoked. The access token is used by the user when requesting a file storage service. The token has the expiration time so that it can be valid only for a certain time with the REST API server. The generated access token is then returned to the gateway server as a response to the web service, and passed back to the smart contract. The smart contract that received the token passes the token to the decentralized application by generating the corresponding event. The decentralized application invokes the Web service for uploading the large size file directly to the REST API server using the received access token. Through this process, the large-sized file is passed to the REST API server, which is then stored with the file access key into the Cassandra distributed database.

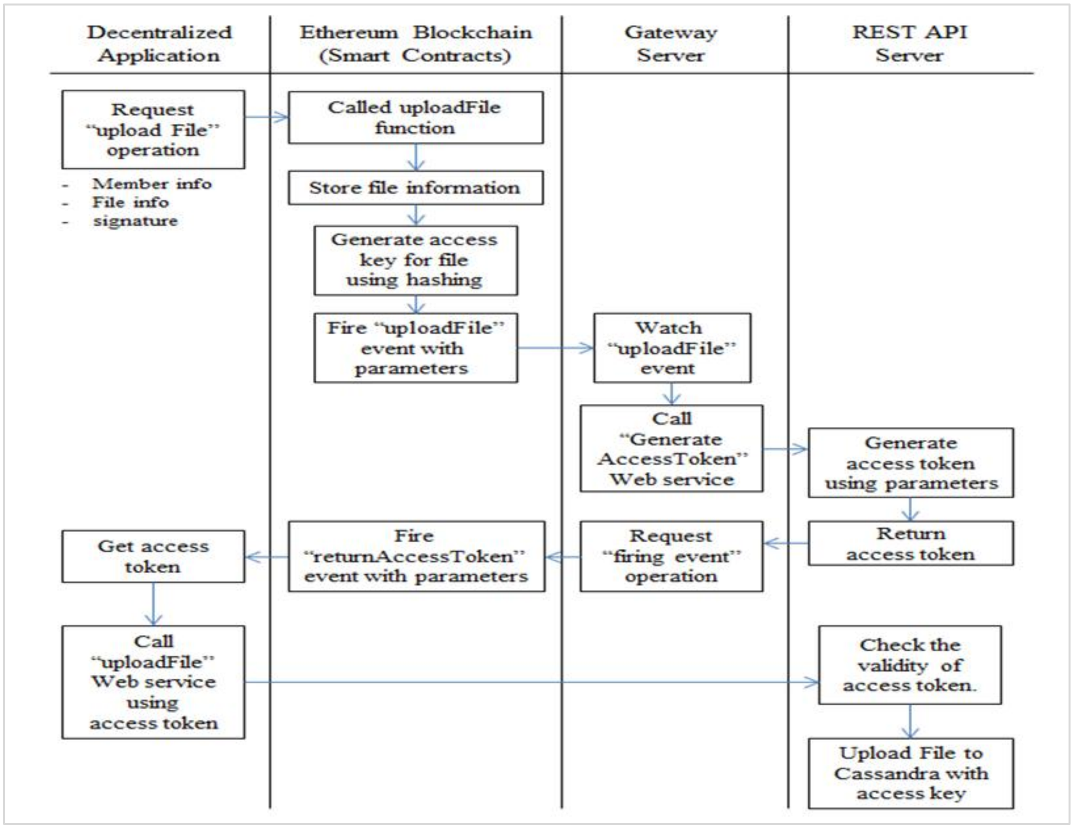

Decentralized Application
Ethereum Blockchain (Smart Contracts)
Gateway Server
REST API Server
Request "upload File" operation
- Member info
- File info
- signature
Called uploadFile function
Store file information
Generate access key for file using hashing
Fire "uploadFile" event with parameters
Watch "uploadFile" event
Call "Generate AccessToken" Web service
Generate access token using parameters
Return access token
Request "firing event" operation
Fire "returnAccessToken" event with parameters
Get access token
Call "uploadFile" Web service using access token
Check the validity of access token.
Upload File to Cassandra with access key

Figure 2 Activity diagram for uploading large file using the proposed system

3.3User authentication in off-blockchain storage

The proposed system uses two authentication processes as shown the *Figure 3* to prevent off-blockchain storage access by unregistered users.

Note that a smart contract is able to control the execution of functions based on the user account of the decentralized application. This works through the execution guard as the modifier of the solidity, so that user accounts not registered in the smart contract cannot execute functions related to off-blockchain storage. In addition, the REST API server uses a separate user authentication process to prevent the REST API server from requesting web services directly outside the Ethereum blockchain. The Web services provided by the REST API server are invoked with the signature sent as a parameter when the user requests a function in the smart contract. Based on this, the REST API server verifies the user's signature in association with the user's account. The signature that has been verified is regarded as created by the user, and the requested function is performed. To this end, the REST API server manages the account information of the user for each decentralized application. In this way, the proposed system can prohibit the access of unregistered users because only the verified users can access Cassandra.

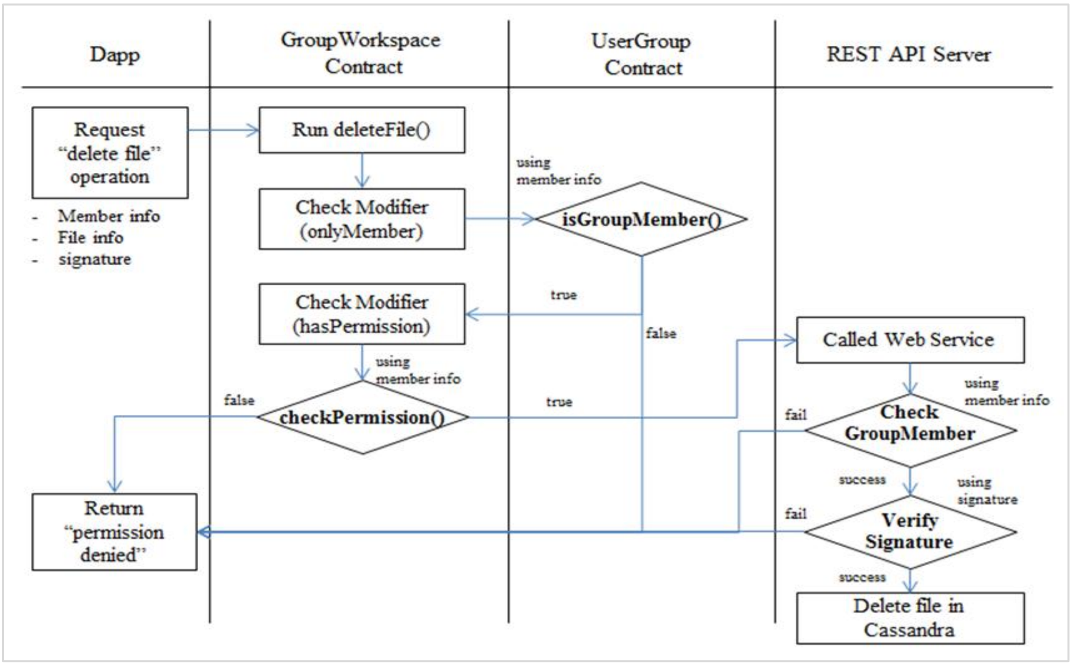


Figure 3 Activity diagram for authentication process

4.Group workspace based on an Ethereum blockchain

The group workspace provides file-sharing services among the members of the user group. In this section, we describe the developed Ethereum-based group workspace called Bspace, which uses the proposed off-blockchain storage solution.

4.1Ethereum based group workspace

Group workspaces provide group management and shared resource management functionalities as group services. The services provided by the group workspace are described in *Table 1*.

Table 1 Basic functions for group workspace

Service type	Service name	Service description
User group management service	CreateGroup	Create a user group, designate the requester as a group owner, and create a group workspace for the group
	DeleteGroup	Delete user group, group workspace and group member information
	GetGroupInfo	Returns user group information such as group owner, group name, and description
	AddGroupMember	Register the requested user as a group member.
	DeleteGroupMember	The specified group member is deleted from the user group
Shared resource management service	GetFileList	Returns file and folder list information for a specific path
	UploadFile	Upload the specified file to a specific path
	DownloadFile	Download the file in a specific path
	DeleteFile/Folder	Delete the specified file and delete the credential information
	CreateFolder	Creates a folder in a specific path
	setPermission	Register authority for specific file / folder of specific user

The owner of the group can create a user group and configure the members to use the group workspace of the group. Group members have rights to files based on the access control information approved by the group owner. When a group member requests a smart contract function that requires file access, it performs authentication in the smart contract execution guard based on the access control information of the member. We have extended the REST API server of the proposed off-blockchain storage solution to support group authentication. The REST API server receives information from the smart contract when there are changes to the group or the group members. Based on this, the REST API server synchronizes user group and group member information with the smart contract. Then, when the user requests a web service that requires access to off-blockchain storage, the REST API server can confirm that the requesting user is a member of the group based on the group information and signature included in the web service.

4.2Smart contracts for group workspace

The services provided by the proposed group workspace are performed through the smart contracts for group workspaces. *Figure 4* shows the types of the smart contracts and their main roles in Ethereum-based group workspace.

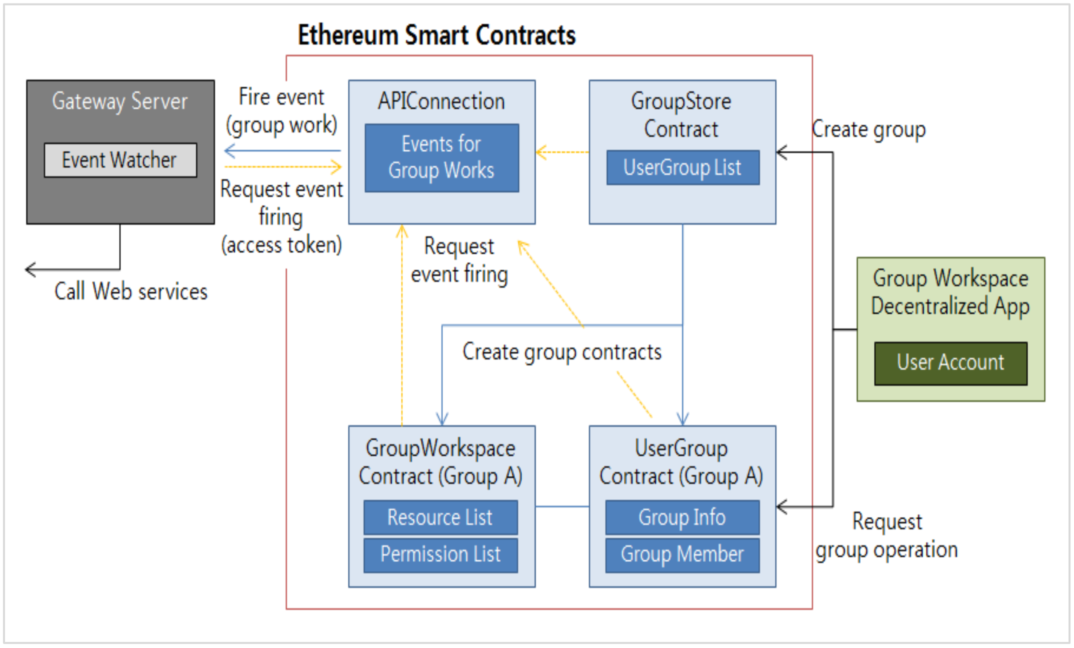


Figure 4 Smart contracts for group workspace service

The GroupStore contract creates a new user group and manages the contract address for all user groups. A user can check the contract address of the group that the user participates or all groups through the functions provided by the GroupStore contract. When a group owner requests to create a new group through the GroupStore contract, the UserGroup contract and the GroupWorkspace contract are created together. The UserGroup contract has basic information and the member information for the user group. When the group authentication is performed, the execution guards of the UserGroup contract can confirm that the user is a member of the group. The GroupWorkspace contract manages shared resources and access control information. In addition, group members can send requests for access to off-block chain storage using the file upload and download functions of the GroupWorkspace contract. The APIConnection contract has event firing functions for creating / deleting groups, adding / deleting group members, uploading / downloading files, and so on, which can be requested from smart contracts to REST API servers. Each smart contract function is associated with the REST API service through the appropriate event firing function of the APIConnection contract. *Table 2* shows the main functions provided by each contract.

Table 2 Main functions of proposed smart contracts

Smart contract	Function name	Parameters	Execution guard
GroupStore	createGroup	group name, group description, group type, group owner	-
	deleteGroup	group address	onlyOwner
	getJoinedGroupList	user address	-
	getAllGroupList	-	-
UserGroup	getGroupInfo	group address	-
	getGroupMemberInfo	member address	onlyMember
	addGroupMember	member address	onlyOwner
	deleteGroupMember	member address	onlyOwner
	getGroupMemberList	-	onlyMember
	isGroupMember	user address	-
GroupWorkspace	setPermission	member address, resource id, permission	onlyOwner
	hasPermission	member address, resource id, permission	onlyMember
	uploadFile	member address, file information, parent folder id	hasPermission
	downloadFile	file id	hasPermission
	getFileList	folder id	hasPermission
	createFolder	member address, folder name, parent folder id	hasPermission
	getFileInfo	file id	hasPermission
	deleteFile	file id	hasPermission

We performed a file sharing test using the proposed off-blockchain storage interworking system and smart contracts for group workspaces. In the test, we used 100 files whose sizes are up to 1GB from 1KB, and checked whether files were stored in the off-blockchain storage through file upload and download functions. As a result of the test, we confirmed that only users with valid access rights stored the files in the target place and all of the files were correctly stored in the off-blockchain storage.

5.Conclusion

In this paper, we propose a large file management method in association with the Ethereum blockchain. The proposed method uses the Cassandra distributed database, which plays the role of off-blockchain storage to store a large number of files. To do this, we presented the system structure and user authentication process for storing large files using off-blockchain storage. In the system structure, the REST API server and the gateway server are used to transmit the user's file upload request to Cassandra. Because the user's request contains the signature of the user account for the off-blockchain authentication process, off-block storage can be used to prevent unauthorized user access. In addition, we presented Bspace, a group workspace based on Ethereum blockchain using the proposed method. The proposed group workspace provides file management service among group users using the proposed off-blockchain storage technology. With the proposed technology, developers of decentralized applications can solve security problems which arise in integrating the off-blockchain storage with the Ethereum decentralized applications, providing a large-capacity file storage service more easily and stably. In addition, developers are able to implement a variety of high-level services on the Ethereum blockchain such as file sharing service and access privilege control among multiple users as in the proposed group workspace without system structural limitations.

Acknowledgment

This research was partially supported by the Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (NRF-2016R1D1A3B03931976).

Conflicts of interest

The authors have no conflicts of interest to declare.

References

[1] Wood G. Ethereum: a secure decentralised generalised transaction ledger. Ethereum Project Yellow Paper. 2014; 151:1-32.

[2] Kosba A, Miller A, Shi E, Wen Z, Papamanthou C. Hawk: the blockchain model of cryptography and privacy-preserving smart contracts. In symposium on security and privacy 2016 (pp. 839-58). IEEE.

[3] https://www.stateofthedapps.com/. Accessed 29 May 2018.

[4] Bogner A, Chanson M, Meeuw A. A decentralised sharing app running a smart contract on the ethereum blockchain. In proceedings of the international conference on the internet of things 2016 (pp. 177-8). ACM.

[5] Linn LA, Koo MB. Blockchain for health data and its potential use in health it and health care related research. In ONC/NIST use of blockchain for healthcare and research workshop 2016. Gaithersburg, Maryland, United States.

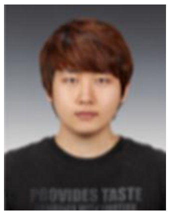
[6] Jin JH. Managing user groups and access rights on Ethereum blockchain. International Journal of Reliable Information and Assurance. 2017; 5(2):13-8.

[7] Jin JH. Providing group workspaces using Cassandra database over the Ethereum blockchain. International Journal of Wireless and Mobile Communication for Industrial Systems. 2018; 5(2):7-12.

[8] Lee HC, Park JE, Lee MJ. C3ware: a middleware supporting collaborative services over cloud storage. The Computer Journal. 2013; 57(2):217-24.

[9] Lakshman A, Malik P. Cassandra: a decentralized structured storage system. ACM SIGOPS Operating Systems Review. 2010; 44(2):35-40.

[10] Eom H. M., Jin J. H., and Lee M. J. A technique for sending email in association with the truffle Ethereum development framework. The International Journal of Private Cloud Computing Environment and Management.2018; 5:1-6.



Jae-Hwan Jin is a Ph.D. student in the School of Electrical Engineering at University of Ulsan in South Korea. He received his M.S. degree from University of Ulsan in South Korea. His research interests include Beacon-based Service, Cloud Storage Service, Collaborative System and Blockchain-based Service.
Email: jjhok2000@gmail.com

Hyun-Min Eom is an M.S. degree student in the School of Electrical Engineering at University of Ulsan in South Korea. He received his B.S. degree from University of Ulsan in South Korea. His research interest includes Blockchain-based service.

Email: lemony3383@gmail.com

Myung-Joon Lee is a professor in the School of Electrical Engineering at University of Ulsan in South Korea. He received his Ph.D. from Korea Advanced Institute of Science and Technology in South Korea. He has co-authored more than 200 research publications including numerous works on Collaborative System, Distributed System, Biological System and Blockchain-based Service.
Email: mjlee@ulsan.ac.kr