Video Streaming Based on Blockchain State Channel with IoT Camera

Min-Hyuk Jeong and Sang-Kyun Kim*

Myongji University, Republic of Korea
E-mail: goldmunt@gmail.com
*Corresponding Author

Received 10 June 2021; Accepted 11 November 2021;
Publication 12 February 2022

Abstract

This paper proposes a system that provides video streaming services from the Internet of Media Things using blockchain and cryptocurrency (token). The user pays the token by the contract terms of the smart contract written on the blockchain through the distributed application (DApp). The IP camera, which is paid the token, streams the taken video to the user in real-time. To investigate the possibility of a blockchain camera streaming service, we upload a smart contract for streaming service on Etherium-based blockchain, and ERC20 tokens necessary for the transaction are created and implemented. To overcome the slow trading speed and the disability of proper refunding, the off-chain transaction, one of the blockchain scaling techniques, was applied and implemented in the system.

Keywords: Video streaming, Internet of Things, blockchain, smart contract, distributed applications.

Journal of Web Engineering, Vol. 21_3, 661–676.
doi: 10.13052/jwe1540-9589.2134

1 Introduction

As Industry 4.0 entered, the Internet of Things (IoT) and the blockchain emerged as the core technology of this new age. The IoT market has grown by an average of 22.6% each year due to the introduction of various IoT services [1]. The blockchain is also increasing due to blockchain 2.0, enabling contracts to be established and implemented through the smart contract. The Internet of Things and the blockchain are growing fast, but each has a problem to be solved.

At present, the IoT system is mainly implemented in a centralised form, and the interconnection is so low. One of the reasons for such low interconnectivity is the lack of interoperability [2]. The lack of interoperability is because Internet service providers or device makers commonly use no standard APIs or data formats. Therefore, most of the Internet services currently being provided are centralised services in-home IoT. This lack of interoperability can be resolved by standardising APIs and data formats used for inter-things communication.

Another reason for interfering with interconnectivity is the lack of a service that can securely share things (devices). Even if the interoperability problem is solved by establishing a standard, there is no way to share things one user owns with others. To solve this problem, we need to provide proper means to share things or their functionalities with other people for some time during which they do not use their things.

Most blockchain DApps currently have little difference in user experience compared to existing Internet-based services. Therefore, users use the existing Internet-based services that have proven to be reliable, rather than using the services provided by a DApp with the effort of purchasing the crypto-currencies and installing a crypto-currency wallet supported by the DApp. Blockchain killer content and service development combining the strengths of the blockchain with the actual demands of the user will benefit both the service provider and the consumer.

Previous works combine technologies for the Internet of Media Things and blockchain [3–5]. They, however, lack counting on the error-resilient design against any unexpected service interruption.

This paper proposes a novel media service and system, which combines the features and advantages of the Internet of Things and blockchain technology. By combining the blockchain trading system with the Internet of Things, a camera (thing), the owner can share the resource of the camera with other people or things when not using their camera. The user pays tokens (cryptocurrency) to watch the scene being taken by the camera. This paper proposes a usage scenario in which the camera scene is streamed for a time corresponding to the token paid and introduces the result of implementing the scenario. The design considers the off-chain transaction method to overcome the slow transaction speed of blockchain and enable refunds for incomplete services.

The composition of this paper is as follows. Section 2 discusses existing off-chain transaction techniques, and Section 3 explains a blockchain-based IoT camera video streaming scenario based on the state channel transaction method. In Section 4, we describe the blockchain-based IoT camera streaming system. Section 5 concludes this paper.

2 Off-Chain Transactions

Off-chain transactions denote trading outside the main blockchain. There are two reasons for using off-chain transactions [6].

The first reason is scalability. The block size is fixed, and there is a limit to the number of transactions processed per second. Therefore, if small transactions occur frequently, the number of queued transactions will increase, causing bottlenecks. Therefore, small and frequent transactions must be processed off-chain, and only the final result of the transaction is registered on the main blockchain [7].

The second reason is speed. Unlike general banking systems, blockchain transactions require a consensus process and take a long time to be confirmed. Use cases that require real-time processing are difficult to implement through blockchain. Off-chain transactions, e.g., state channels, sidechains, and childchains, are techniques to overcome these limitations of blockchain. This section discusses the three existing off-chain technologies and discusses the best off-chain methods for IoT camera video streaming systems.

2.1 State Channel

The state channel method is a method of creating a separate channel between traders. Some money is deposited in the (bank) smart contract of the main blockchain. There is no fee for any transaction occurring on an off-chain state channel, and there is no limit to the number of transactions. Figure 1 shows a sequence diagram of the basic process of trading tokens using the Raiden Network, one of the Ethereum-based state channel solutions [8].

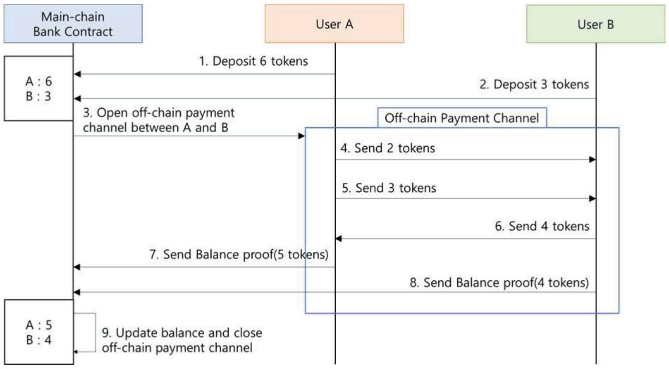

Main-chain
Bank Contract
User A
User B
1. Deposit 6 tokens
2. Deposit 3 tokens
A : 6
B : 3
3. Open off-chain payment
channel between A and B
Off-chain Payment Channel
4. Send 2 tokens
5. Send 3 tokens
6. Send 4 tokens
7. Send Balance proof(5 tokens)
8. Send Balance proof(4 tokens)
A : 5
B : 4
9. Update balance and close
off-chain payment channel

Figure 1 Token trading sequence diagram using Raiden Network.

User A and user B deposit some tokens (e.g., six tokens from user A and three tokens from user B) to the bank contract in the main blockchain (Figures 1.1 and 1.2). The bank contract verifies the deposit and opens an off-chain trading channel for users A and B (Figure 1.3). On the off-chain trading channel, user A sends two tokens to user B and then three tokens to send a total of five tokens (Figures 1.4 and 1.5). As a result, user A has one token left, and user B has eight tokens. Then, user B sends four tokens back to user A so that user A has five tokens and user B has four tokens (Figure 1.6). After the transaction, users A and B send their balances to the bank contract on the main blockchain, respectively (Figures 1.7 and 1.8). The bank contract verifies that the sum of user A and user B shares is equal to the sum of the shares before opening the chain (Figure 1.9).

Currently, representative solutions for state channels include Bitcoin's Lightning Network and Ethereum's Raiden Network. Transactions on the off-chain have the advantage of being made almost instantaneously with RESTful APIs. In addition, all off-chain transactions are secured by using hash values created by users' private keys. After the off-chain transaction is completed, the main blockchain goes through checking the result of the transaction once again, enabling secure transactions.

2.2 Sidechain

A sidechain is a way to create a separate blockchain that is connected to the main blockchain. There is a point of contact between the main blockchain and the sidechain, a bridge, and periodically updates transactions in the sidechain to the main blockchain. The sidechain can set independent properties without

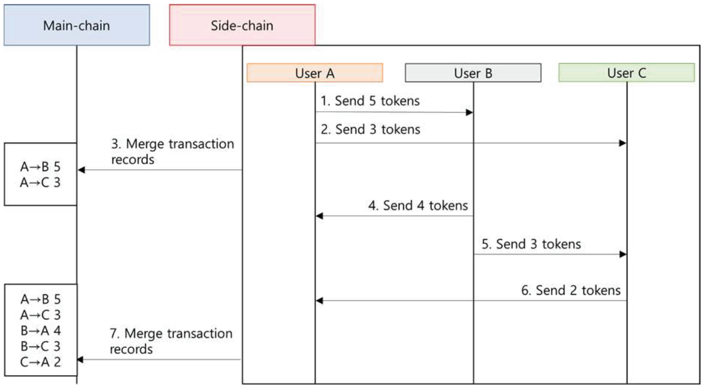


Figure 2 Token trading sequence diagram using sidechain.

following the main blockchain properties such as block size or consensus method. To use the sidechain, the user must deposit some money in the bank contract for the sidechain of the main blockchain. The deposit cannot be used for other services on the main blockchain but only on that sidechain. The sidechain is mainly applied to corporate, private chains or games on the blockchain [9, 10].

The sidechain operation is shown in Figure 2. The process of each user depositing some money in the bank contract of the main blockchain is omitted for brevity.

User A sends five tokens to user B (Figure 2.1), and user A sends three tokens to user C in the sidechain (Figure 2.2). When the sidechain meets the transaction history update cycle, the sidechain updates the transaction history on the main blockchain (Figure 2.3). User B then sends four tokens to user A (Figure 2.4) and three tokens to user C (Figure 2.5), then user C sends two tokens to user A (Figure 2.6). Since the transaction history update cycle is back, the sidechain will update the transaction history on the main blockchain (Figure 2.7). Updating transaction details in the sidechain to the main blockchain can be set differently for each sidechain, such as uploading periodically in one transaction or updating each critical event.

The most actively used sidechain solution is the Loom Network, which is widely used as a blockchain for games. The Rootstock project is also a sidechain that enables the use of smart contract features through Bitcoin. Since a sidechain is also a blockchain, a consensus process and a block generation process are required. All transactions are done through the blockchain, ensuring safety.

2.3 Child Chain

The child chain refers to a child blockchain connected to a parent chain (main blockchain). A child chain can have other descendant chains with a tree structure. The parent chain only has a block header hash of the child chain. The child chain updates the block header hash in the parent chain with a specific block generation cycle [11]. Therefore, to search for transactions occurring in the child chain, it is possible to search down the tree structure through the hash of block headers recorded in the chains.

Figure 3 shows how the child chain records transactions. There are blockchains A and B, the child chains of the main blockchain, and blockchain A has blockchain C, the child chain. Figure 3 has a tree structure consisting of three layers, including the main blockchain. The block's hash containing the transactions in the blockchain C is recorded in transaction 4 of the blockchain A block. Blockchain A and Blockchain B also include the transactions that occurred in their blockchains and write the hash of the block's header to one of the transactions in the main blockchain.

A child chain can also have its child chain, taking the form of a tree structure. As the child chain tree becomes deeper and continues to be created, transactions per second (TPS) can be infinitely extended. A typical solution is Plasma running on Ethereum.

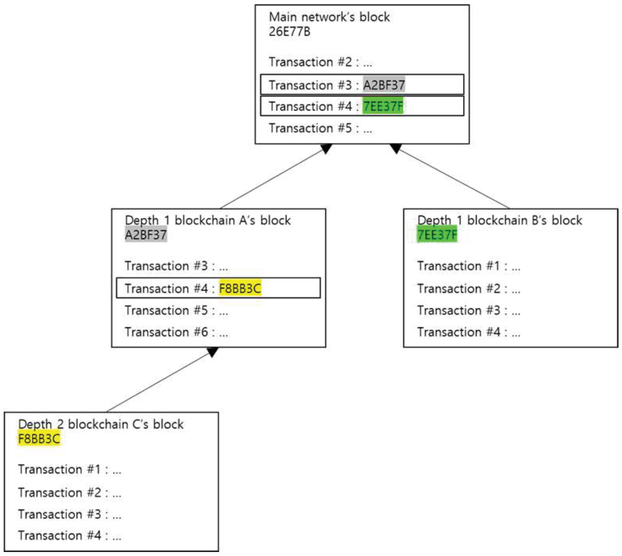


Figure 3 Child chain transaction record method.

2.4 Comparisons of Off-Chain Transaction Methods

We choose the most appropriate technology for the proposed IoT camera video streaming system among the abovementioned three off-chain methods. The items to compare are the transaction speed and the possibility of refund in case of error. When streaming video, it should be possible to divide the video packet into smaller units (e.g. one second) to make transactions in real-time. In the event of a network error or failure of video streaming, only the last transaction up to that moment should be recorded.

The off-chain technology that can perform transactions at the fastest speed is the state channel method. The state channel can complete transactions in real-time using a RESTful API on the channel between traders. Therefore, once a channel is established, transaction processing is possible in real-time regardless of the TPS of the main block. The sidechain can speed up the transaction by setting the consensus process and block creation rate. Still, it is difficult to process as fast as the state channel supporting real-time transactions because all transactions must be written to the block. Child chains, like sidechains, process transactions at a similar speed to sidechains because all transactions must be recorded in blocks. However, the deeper the blockchain tree, the more transactions are registered on the main blockchain, so it takes longer to finalise than the sidechain. In addition, the state channel method is more robust to service errors than the sidechain or child chain, where all transactions are recorded in the block.

3 State Channel Scenario for IoT Camera Video Streaming System

We used the Raiden network based on the Ethereum network to create a state channel for IoT camera video streaming.

Figure 4 shows the scenario of applying Raiden Network to the IoT camera video streaming system proposed in this paper. We skipped the process of asking the user for the cost per hour of streaming service through DApp and getting the streaming address from the camera contract. We assume that the cost of receiving streams from Camera_1 is three tokens per second. Currently, the user has decided to receive streaming for 30 seconds from Camera_1.

The smart contracts of the user and the camera each send a deposit to the bank contract on the main blockchain to open the state channel. To receive video streaming for 30 seconds (3 tokens/s), the user deposits

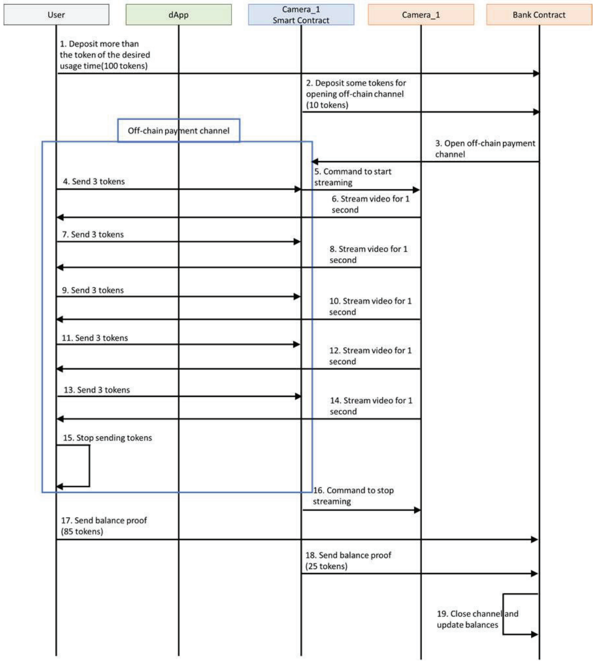

User
dApp
Camera_1
Smart Contract
Camera_1
Bank Contract
1. Deposit more than the token of the desired usage time(100 tokens)
2. Deposit some tokens for opening off-chain channel (10 tokens)
Off-chain payment channel
3. Open off-chain payment channel
5. Command to start streaming
4. Send 3 tokens
6. Stream video for 1 second
7. Send 3 tokens
8. Stream video for 1 second
9. Send 3 tokens
10. Stream video for 1 second
11. Send 3 tokens
12. Stream video for 1 second
13. Send 3 tokens
14. Stream video for 1 second
15. Stop sending tokens
16. Command to stop streaming
17. Send balance proof (85 tokens)
18. Send balance proof (25 tokens)
19. Close channel and update balances

Figure 4 Video streaming sequence diagram with IoT camera using Raiden Network.

100 tokens, more than 90 tokens (Figure 4.1). The camera's smart contract deposits ten tokens to establish a channel (Figure 4.2). After confirming the deposit, the bank contract opens the state channel (Figure 4.3). The user starts transmitting three tokens per second over the state channel (Figure 4.4). After confirming the token deposit, the camera's smart contract instructs the camera to stream (Figure 4.5), and the camera starts streaming one second of video (Figure 4.6). The user continues to transmit three tokens per second (Figures 4.7, 4.9, 4.11, 4.13), and the camera continues to stream one second of video (Figures 4.8, 4.10, 4.12, 4.14). If the token transfer is interrupted for some unexpected reason (Figure 4.15), the smart contract will immediately stop streaming (Figure 4.16). The user and the camera's smart contract send their proof of equity to the bank contract on the main blockchain (Figures 4.17, 4.18). Bank contracts verify their shares and close the state channel (Figure 4.19).

Existing blockchains take a considerable time for transactions to be registered and agreed upon in a block. However, video streaming services using Raiden Network can execute the transaction in real-time, so it doesn't pay for the service at once. As a result, errors can be detected and reacted in real-time. If the token is continuously transmitted through the state channel and the token transmission continues, the service provider continues the service. If an unexpected situation occurs and the token transmission is stopped, the service provider may stop streaming immediately. Because the video keeps being transmitted as long as the token is received, no refund process is needed for the unexpected service interruption. Also, if the service provider does not provide the video service properly, the user can stop the token transfer immediately.

4 Implementation of IoT Camera Video Streaming System

The following items must be configuring the IoT camera video streaming system, IP cameras, camera's smart contracts, a bank contract, a DApp, and tokens that comply with the ERC20 Standard. The structure of the IoT video streaming system is shown in Figure 5.

Two IP cameras are shooting in different locations, and each camera is connected to a smart contract on their Etherium blockchain. Each smart contract and the DApp are bound to a bank contract to make deposits, open/close an off-chain channel, and confirm the final balances. The DApp, which controls two cameras, allows users to communicate with the camera of their choice. The IoT camera video streaming system was implemented on the Etherium Robsten test net.

The cryptocurrency required for the service transaction was newly created as Media Thing Token (MTT), conforming to the ERC20 Standard. Figure 6

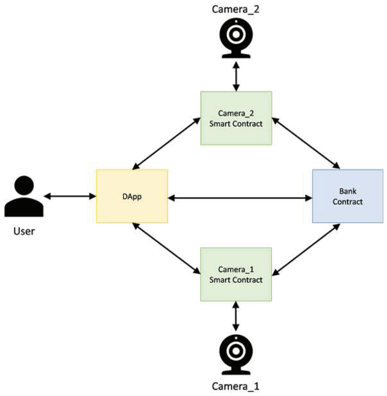
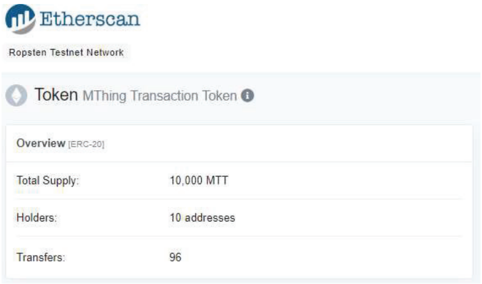


Figure 5 Structure of IoT camera video streaming system.



Etherscan

Ropsten Testnet Network

Token MThing Transaction Token

Overview [ERC-20]

Total Supply:	10,000 MTT
Holders:	10 addresses
Transfers:	96

Figure 6 MTT information.

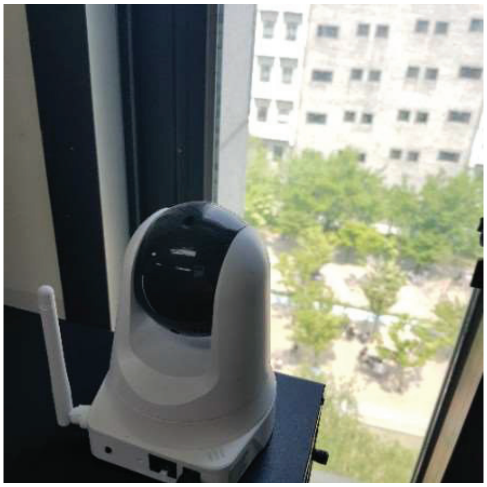
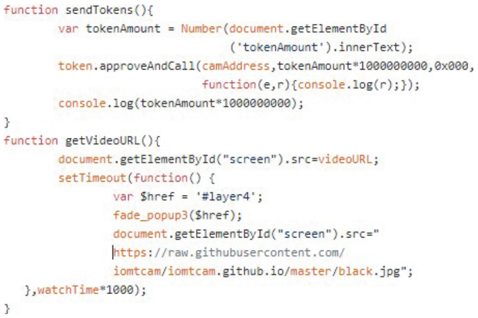


Figure 7 IP camera setup for campus view.

shows the MTT information uploaded to the Etherium Robsten test net. The information indicates that 10,000 MTTs have been created, ten owners are currently issued, and 96 transactions have occurred.

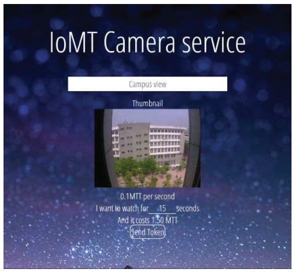
As shown in Figure 7, two IP cameras were set to take the campus and the laboratory scenes. The IP cameras start streaming when they receive commands from the DApp.

We implemented the DApp that connects two IP cameras to smart contracts. The DApp supports transactions between users and smart contracts. The Web3 library was used to call the smart contract function according to the user's behaviour. It receives the payment through the *getVideoURL_CostPerMinute()* function defined in the MPEG-IoMT International Standard [12] and receives the address of the Etherium wallet through



```
function sendTokens(){
        var tokenAmount = Number(document.getElementById
                                ('tokenAmount').innerText);
        token.approveAndCall(camAddress,tokenAmount*1000000000,0x000,
                            function(e,r){console.log(r);});
        console.log(tokenAmount*1000000000);
}
function getVideoURL(){
        document.getElementById("screen").src=videoURL;
        setTimeout(function() {
                var $href = '#layer4';
                fade_popup3($href);
                document.getElementById("screen").src="
                https://raw.githubusercontent.com/
                iomtcam/iomtcam.github.io/master/black.jpg";
   },watchTime*1000);
}
```

Figure 8 Functions implemented in DApp.



IoMT Camera service

Campus view

Thumbnail

0.1MTT per second

I want to watch for 15 seconds

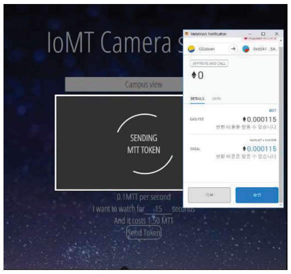
And it costs 1.50 MTT

Figure 9 The landing page of DApp.

the *getWalletAddress()* function [13]. Finally, the token is sent through the *sendTokens()* function [13]. The DApp also controls IP cameras connected. At this time, the *getVideoURL()* function defined in the MPEG-IoMT is used for requesting streaming to the IP camera [12].

Figure 8 shows an example implementation of the *sendTokens()* and *getVideoURL()* functions among the functions used to configure the DApp. The *sendTokens()* function passes the number of tokens that the user has for the bank contract. The video streaming is requested to the camera via the *getVideoURL()* function.

Figures 9, 10, and 11 show examples of implemented video streaming DApp GUI. When a user selects a camera to receive a video streaming service on the landing page, the camera displays a thumbnail of the shot scene (Figure 9). When the user inputs the desired time in the blank space, the cost is calculated and displayed. Press the ’Send Token’ button to send the calculated token (Figure 10). Users watch the video being captured in real-time from the corresponding camera once the transaction is completed (Figure 11).



♦0.000115
변환 비용을 찾을 수 없습니다

♦0.000115
변환 비용은 찾을 수 없습니다

0.1MTT per second
I want to watch for 15 seconds
And it costs 1.50 MTT

Figure 10 Screenshot of the token transaction.

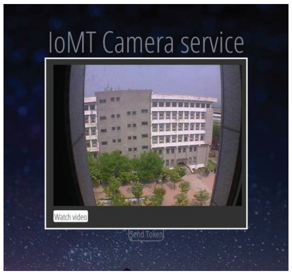


Figure 11 Video scene streaming from IP camera.

Using this system took an average of 20 s∼30 s to deposit the tokens and start receiving the video streaming service. The average processing time is similar to when the deposit transaction is written to the Ethereum block, and the block is created through the consensus process. In the case of service interruption caused by an error either on the service provider's side or user side, the smart contract of the camera can react on it instantly to stop the off-chain transaction and finalise the service.

5 Conclusion

The IoT camera video streaming scenario was proposed and was implemented the IoT camera video streaming system that operates. With standardised APIs and smart contracts in blockchain 2.0, the DApp can access IoT cameras on the Internet and receive video streaming services regardless of platform.

We examined off-chain transactions to overcome a low TPS and support a proper refund mechanism when service errors occur. By comparing the three off-chain transaction methods, we selected the state channel method as the most suitable off-chain transaction for the IoT camera video streaming system. A new video streaming system scenario was proposed by applying the state channel method to the IoT camera video streaming system.

In the future, it is necessary to develop a more complex media service scenario that supports a plurality of media things such as a microphone, a display, and a speaker, rather than a simple form of a DApp-smart contracts-camera.

Acknowledgement

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korean government (MSIT) (1711107567, Study on human factors of 360°VR sensorial effects in consideration of user learning styles).

References

[1] Korea Internet and Security Agency. *Internet Business Survey of 2018*, 2018 (accessed April 3, 2020), https://www.kisa.or.kr/eng/usefulreport/surveyReport_List.jsp.

[2] M.H. Jeong and S.-K. Kim., Discovery, connection and transaction of Internet of media things, *Proceedings of the Korean Institute of Broadcast and Media Engineers*, pages 197–200, 2018, (In Korean).

[3] K. Lee, M.H. Jeong, and S.-K. Kim, Transaction process on Internet of media things. *6th EEECS 2018*, July 2018.

[4] M.H. Jeong and S.-K. Kim, Internet of media things camera streaming system based on blockchain, *The Korean Institute of Broadcast and Media Engineers Summer Conference*, 2019.

[5] S.-K. Kim, Video streaming system based on Internet of media things and blockchain, *International Symposium on Multimedia and Communication Technology*, August 2019.

[6] S. Faust, S. Dziembowski, and K. Hostáková, General state channel networks, *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, October 2018, DOI:10.1145/3243734.3243856.

[7] D. A. Nagy, I. A. Seres, L. Gulyás and P. Burcsi, Topological analysis of bitcoin's lightning network, *arXiv preprint*, 2019, arXiv:1901.04972.

[8] Raiden network, *Raiden network*, 2019 (accessed April 3, 2020), http://raiden.network/.

[9] J. de Kruijff and H. Weigand. Understanding the blockchain using enterprise ontology. *International Conference on Advanced Information Systems Engineering*, 10253:29–43, May 2017, DOI:https://doi.org/10.1007/978-3-319-59536-8_3.

[10] P. Robinson, S. Johnson, and J. Brainard, Sidechains and interoperability, *arXiv preprint*, 2019, arXiv:1903.04077.

[11] Y. Kwon, S. Kim and S. Cho, A survey of scalability solutions on blockchain, *2018 International Conference on Information and Communication Technology Convergence (ICTC)*, pages 1204–1207, October 2018, DOI:10.1109/ICTC.2018.8539529.

[12] S.-K. Kim and S.M. Chun. Text of ISO/IEC FDIS 23093-3 Media data format and API. Standard, ISO/IEC JTC1/SC29/WG11, Geneva, Switzerland, March 2019.

[13] S.-K. Kim and M. Choi. Text of ISO/IEC FDIS 23093-2 Discovery and communication API. Standard, ISO/IEC JTC1/SC29/WG11, Geneva, Switzerland, March 2019.

Biographies

Min-Hyuk Jeong received the B.S. and M.S. degrees in Computer Engineering from the Myongji University in 2016 and 2018, respectively. He is currently pursuing a PhD degree in Computer Engineering at Myongji University, under the supervision of Prof. Sang-Kyun Kim. His research interest lies in image processing, 4D media, VR, the Internet of Things, and blockchain.

Sang-Kyun Kim. received BS, MS, and PhD degrees in Computer Science from the University of Iowa in 1991, 1995, and 1997. During 1997–2007, he stayed at Samsung Advanced Institute of Technology to work on driving simulators and digital content management. Currently, he has been a professor of the Undergraduate School of Convergence Software at Myongji University (Republic of Korea) since 2007. His research interests include digital content (image, video, and music) analysis and management, fast image search and indexing, colour adaptation, 4D media, sensors and actuators, VR, Internet of Things, blockchain, and multimedia standardisation.