ISSN 2409-2665
Journal of Logistics, Informatics and Service Science
Vol. 9 (2022) No.2, pp. 131-144
DOI:10.33168/LISS.2022.0208

IoT Edge Cloud Platform with Revocatable Blockchain Smart Contract

Jaekyung Nam and Min Choi

Department of Information and Communication Engineering, Chungbuk National University, Republic of Korea

miniromi@knou.ac.kr

Abstract. The Internet of Things (IoT) is gaining popularity as a solution to rural concerns confronting sustainability crises, such as reducing revenue, export, and growth rates due to the aging of various industries. In this research, we design and implement an IoT edge cloud platform structure with blockchaining functionality that can solve the problems. Since the connection bandwidth and code spaces of smart farms are restricted, we configure the edge node for site management. Sensor modules for a site are manufactured and the edge is configured to manage the data of each sensor module. Moreover, a distributed coordination system is also supported using Kafka for data interworking between each edge node.

Keywords: edge platform, internet of things, blockchain

1. Introduction

The agriculture industry will confront tremendous challenges in feeding the predicted up to 10 billion within a few tens of years. The food organization expects that food production must be increased (Global Forecast to 2026). To solve those challenges, the agricultural sector must adopt smart farms. Smart farms, according to AIOTI, are applications of data collection, processing, analysis, and automation technologies across the entire value chain. As a result, most smart farms use IoT technology to transmit and receive real-time data from sensors and to control them. In addition, we can measure such as temperature, humidity, vibrations, or shocks during product transport (Region and Segment Forecasts 2019-2025). Therefore, the application of IoT technology in the agricultural sector can control the efficiency of resource use and significantly improve production capacity. One of them is processing and analyzing vast amounts of data coming from heterogeneous devices (Anusha et al., 2019). Furthermore, due to restricted computing, networking, and storage resources, overall energy and cost, and unreliable latency, processing all of this acquired data straight to a central server is wasteful and often impractical. The MQTT broker and Kafka server are the two major components of the edge platform used in this study. The MQTT broker is responsible for exchanging messages between various sensors and actuators in the smart farm, whereas Kafka reliably sends large amounts of data generated in the smart farm to the consumers. Blockchain technology (Atijosan et al., 2021; El Azzaoui et al., 2022) can track the provenance of food, assisting in the creation of trustworthy food supply chains and the establishment of trust between producers and consumers (Vanipriya et al., 2021). Deploying on a blockchain means storing EVM bytecode in a block. Since all nodes participating in a blockchain network have the same block, all nodes can hold and execute EVM bytecode. Examples of smart contracts in real life are as follows. The most representative example is a vending machine, which is implemented in the form of a Yuhan Automata, accepts coins, and provides change and goods. At this time, the establishment of the contract is made by the agreement of the offer and acceptance (possible tacitly), and can be regarded as a paid contract / down payment contract under the Civil Code. The contract is programmed to be executed automatically when certain contract conditions are met has slightly different characteristics. In this paper, we review the methods to solve the problems caused by the difficulty of contract cancellation/cancellation caused by the difference between these smart contracts and contracts under the general civil law.

2. Background

Apache Kafka provides publish or subscribe messaging model for data production and consumption. It facilitates real-time data access for real-time stream processing by permitting long-term data storage (Verma et al., 2021). Apache Kafka was built from the ground up to enable long-term data storage and data replay. It takes a novel approach to data persistence, fault tolerance, and replay. Therefore, this can be seen in how it handles scalability by allowing data access using cross-partition data sharing, topics/partitions, data offsets, and consumer group names for data replication persistence in clusters, increased data volume, and load. Moreover, Apache Kafka is also well suited for real-time stream processing applications because it is designed to act as a communication layer for real-time log processing. This makes Apache Kafka suited for applications that handle massive volumes of data in real-time and run on communications infrastructure. Every partition has one server that acts as the leader for all read/write operations within the server, while the other server acts as a follower of this leader. If a leader goes down or fails, by default one of the followers on the other server is chosen as the new leader. Producers can generate specific messages coming to selected partitions within a topic. Consumers can consume published messages based on topics. Messages are delivered to consumer instances within the subscribing consumer group. The TOPIC-partitioned data is transmitted to the server (KAFKA Cluster) and replicated to each broker in the cluster (each partition of the server). Afterward, at the request of the consumer group, each broker in the KAFKA cluster designed a system capable of distributing data and transmitting large amounts of data.

MQTT is servers publish/subscribe messaging transport protocol designed to be open, simple, and easy for clients to implement. Because of these characteristics, they are used in many contexts, including limited environments (Donzia et al., 2019). MQTT is important for many IoT use cases. Hence, MQTT has 3 defined quality of service (QoS) levels. QoS refers to a level that guarantees the quality of service. In this study, the QoS level was set to 0 because speed is prioritized over the reliability of data generated by sensors. After collecting the sensor data generated from the sensor module located in the smart farm through the edge platform, it is processed to be classified as an MQTT component by TOPIC (classified by temperature, sunlight, rainfall, etc.).

Edge computing represents a paradigm shift from centralized to decentralize. Improve QoE by using compute resources closer to the client (Guillermo et al., 2019). The environment built for this is called the edge platform which eliminates bottlenecks and potential points of failure and enables rapid recovery from failures. For this reason, edge platforms aim to reduce response time or latency by caching content (Swain et al., 2021). The edge platform can be used wherever edge computing is used (Keswani et al., 2018; Park 2021). Edge computing refers to the technologies that enable computation to be conducted at the network's edge (Gokhale et al., 2018; Hammad et al., 2022). As the number of cloud computing users grows, so does the throughput of the associated server and data center, causing a situation in which a delay develops in the process of analyzing and transmitting the acquired data. For the purpose of solving this problem, edge computing refers to processing data at the terminal level rather than a specific server or data center. In other words, edge computing represents a paradigm shift from centralized to decentralize. Since QoE is improved and processed using computing resources closer to the client, the load concentrated on the cloud system can be reduced and it has strong characteristics against fatal blows caused by server failure (Naresh et al., 2019; LG Tuiun). Edge computing eliminates bottlenecks and potential points of failure and enables rapid recovery from failures (Alandjani, G. O. 2021).. For this reason, edge computing aims to reduce response time or latency by caching content (Goyal et al., 2019; Shin et al., 2022). Edge computing can be used in places. Based on these characteristics, it is spotlighted as the core IoT technology of the 4th industrial revolution such as autonomous driving, aviation engine, AR/VR, biometrics and healthcare (Ahmed et al., 2018; El Azzaoui et al., 2022; Vanipriya et al., 2021; Gilmore 2014). Smart contracts for a Dapp developer, a blockchain application, writes a smart contract in a high-level language, compiles it with an EVM compiler, creates EVM bytecode and distributes the blockchain, and the EVM bytecode is executed in each EVM. This is similar to the relationship between a Java program, Java bytecode, and the Java Virtual Machine.

3. Design and Implementation

3.1. System architecture

This study does not constitute all environments used. It is designed for direct application to real agriculture using the Raspberry Pi Zero w with minimal resources. 1 GHz, single-core CPU, 512 MB RAM, very cheap. Therefore, it is very easy to use in agriculture. Each hardware operates on an independent IoT device and communicates with edge nodes using the MQTT protocol, which is relatively lightweight, compared to HTTP. Each hardware interacts with the edge platform and exchanges large amounts of data. Each sensor is not interconnected and operates through an edge platform. This means that data can be processed efficiently without unnecessary communication.

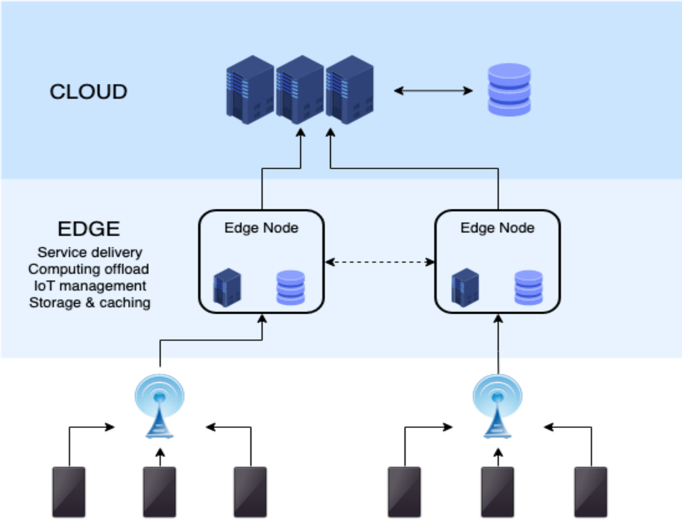

Fig. 1: Edge cloud computing architecture

The following addresses in Table. 1 is used to access RAM and read-only information. T_a is the ambient temperature of the object and T_{OBJ1} and T_{OBJ2} are the temperature of the object. The result has a resolution of 0.02C and is available in RAM. To is derived from RAM as:

$$\mathrm{To}[\mathrm{K}] = \mathrm{Toreg} * 0.02, \mathrm{or}\ 0.02\ \mathrm{K}/LSB \quad (1)$$

Table. 1: MLX90614 temperature sensor RAM addresses

RAM(32x16)		
Name	Address	Read access
Melexis reserved	0x00	Yes
…	…	…
Melexis reserved	0x03	Yes
Raw data IR channel 1	0x04	
Raw data IR channel 2	0x05	
T_A	0x06	Yes
T_{OBJ1}	0x07	Yes
T_{OBJ2}	0x08	Yes
Melexis reserved	0x09	Yes
…	…	…
Melexis reserved	0x1f	Yes

Fig. 2 depicts the interaction sequence among the components that make up this system. We achieved reliable and high performance machine independent interaction using pub/sub technology and REST API, providing services to multiple users at the same time by managing sessions for each user.

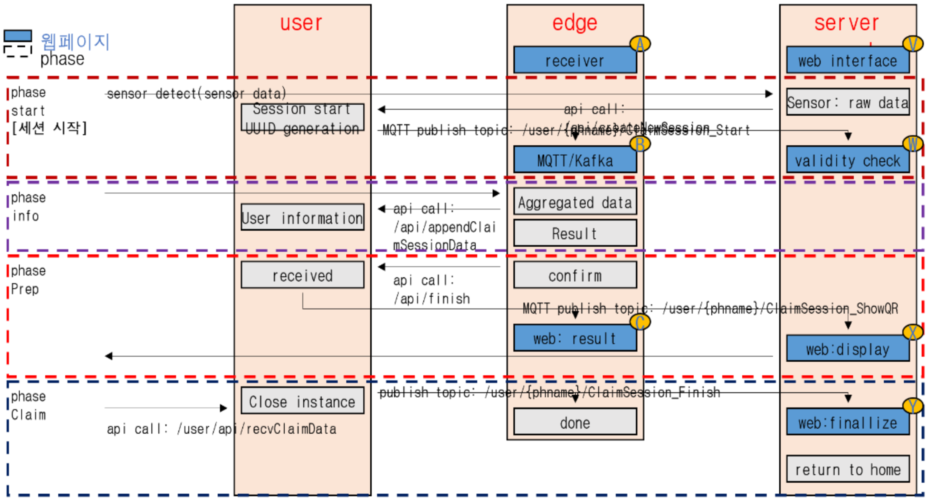

Fig. 2: operation sequence between edge and cloud server

In this research, a blockchain platform is applied to record facts about major events affecting crop harvest. We can provide alarms to managers from insights derived through time series analysis of data acquired from sensors. In addition, by recording major messages and events on the blockchain platform, it can be provided when a request for a proof-of-existence service occurs in the future. For this purpose, we designed and con-figured some conventional computing server and some blockchain server together.

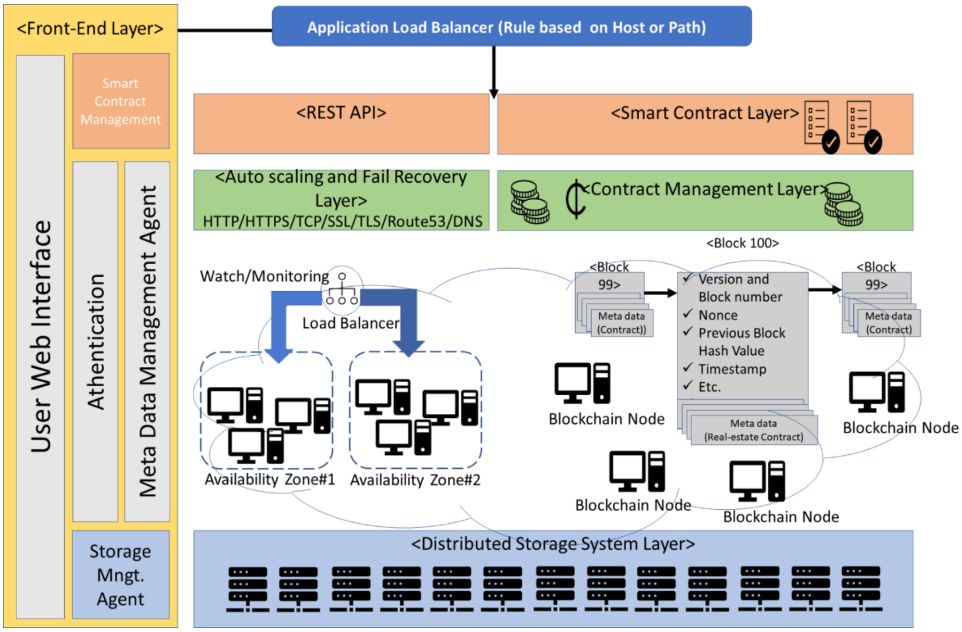

Fig. 2: operation sequence between edge and cloud server

Fig. 3 shows a structural diagram connecting these technologies such as server, storage, load balancing, autoscaling, and network configuration. The blockchain server implements W3C's DID and verifiable claim using distributed ledger technology and applies block-chain-based public key and pairwise DID concept so that identity and usage information cannot be found at the source. In addition, we also make use of off-chaining technique to improve blockchain scalability. The off-chaining is a technique to reduce the burden on the main chain(blockchain), if data supported by the back is stored in the main chain nodes, each node becomes difficult to withstand the load, so the TPS is lowered, the cost is inevitably higher, and the cause of performance degradation such as response time being. If a server operates in a separate layer, it moves transactions that were being processed on the main chain to off-chain. Thus, it reduces the load by summarizing only the most essential information and putting it in the main chain.

From the perspective of blockchain and cryptocurrency, the characteristics of smart contracts are as follows. A smart contract is a computer program executed by pre-written logic, stored and replicated on a distributed storage platform, executed on a computer network (on the same computer running the blockchain), and noting changes to the ledger. There are characteristics that can be applied (such as payment in cryptocurrency).

blockchains = distributed trustworthy storage
smart contracts = distributed trustworthy calculations.

In this way, even if the smart contract is automatically completed by the logic written in advance, it is necessary to cancel or cancel the contract if a dispute arises. In such cases, it is difficult for smart contracts and blockchain architectures that are automatically executed to effectively handle disputes. For example, despite signing/purchasing a contract with an arbitrary third party and paying the entire price, when actually receiving the service and service, it is different from the original contract or the essential function of the service and service during the defect repair or warranty period even smart contracts will be disputed if they are damaged beyond fulfillment. Nevertheless, due to the nature of smart contracts, once certain conditions are fulfilled, they are irrevocable and cannot be deleted from the blockchain. If one party cancels the contract and wants to restore it to its original state, the blockchain transaction will no longer be able to fulfill its role. In case of defects in expression of intention, mistake, fraud, or coercion, the parties may cancel the contract according to the law, or if the ground for breach of the contract is recognized, a problem may arise in which one party exercises the right to rescind.

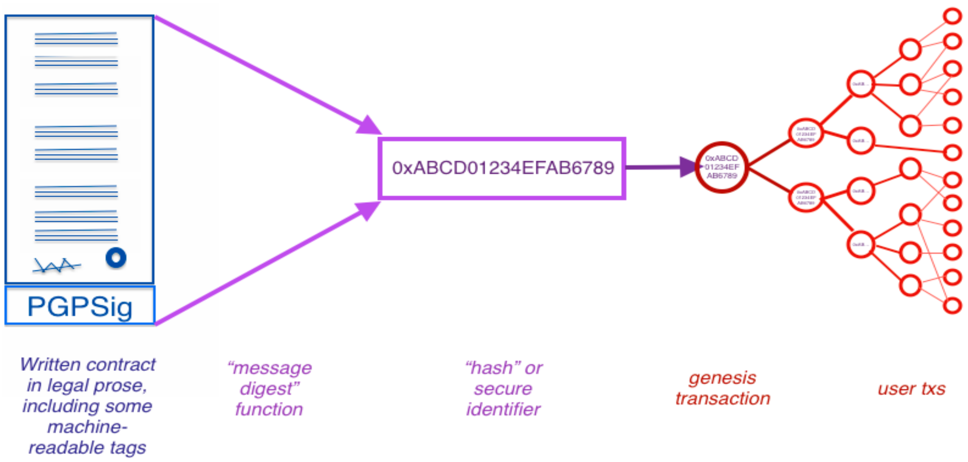

Fig. 3: Ricardian contract

To solve this problem, the concept of a Ricardian contract (Grigg 2004) has been proposed as shown in Figure 3. It is a digital document in which the parties define the terms and content of the contract, and is written as a human-readable document, which is encoded and cryptographically signed. And approve since the Ricardian contract is in a form that can be easily read by both humans and programs, there is a mutual inconsistency between what an individual understands in concluding a smart contract and what a contract is actually concluded between blockchain nodes through a smart contract. a way to avoid it.

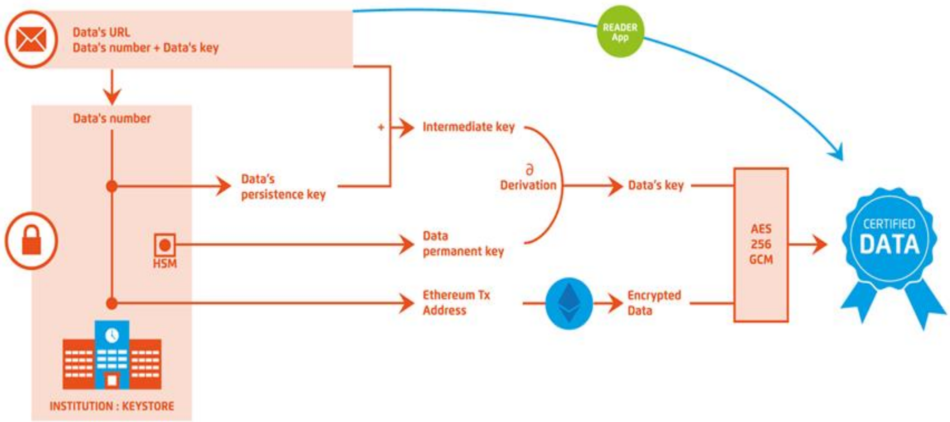

Fig. 4: BC diploma's smart contract termination

In the case of BC diploma of Fig. 4, an architectural example for realizing the termination of a contract, each contracting party creates a transaction using three sets of keys: a key, a persistence key, and a permanent key. Because access control (key) and "right to delete" (persistence key) must be guaranteed in accordance with the GDPR regulations that define, or a way to cancel it should be provided. Therefore, in this respect, BCdiploma's case can be seen as a case of thinking about the cancellation/cancellation of blockchain smart contracts, etc., ahead of other countries.

3.2. Software architecture and experiments

Our server's runtime is built of Node.js, which is based on the JavaScript language, and a package suited for Node.js is set so that the server may function properly. The whole system consists of an edge platform with multiple mqtt clients and multiple edge nodes. Each client connects to the edge platform to send and receive data. Table 2 shows the modules used in this research. The edge node is mainly composed of two components, the MQTT component and the Kafka server. The MQTT component is a broker and subscriber to the MQTT protocol. This allows for immediate data transfer as well as other operations after receiving the data. In this study, specific data is sent to Kafka after receiving data from a component for reliable data storage and transmission.

Table 2: Libraries and modules used

	modules	version
Run time	Node.js	v14.17.3
Web server	Express.js	v4.17.1
Mqtt client	mqtt.js	v0.46.1
Mqtt broker	aedes.js	v4.2.8
Database	MongoDB	v4.4.6
Database client	mongodb.js	v4.1.1
Kafka server	Apache kafka	v2.8.0
Kafka client	kafka.js	v1.15.0
Visualization tool	chart.js	v2.4.0

When a sensor publishes data on a specific topic, the MQTT component receives it and classifies it into direct processing data and data processing through Kafka. Many sensor data take constant values except under special circumstances. However, under special circumstances, it exhibits unusual values. In this case, it is important to reliably transfer the desired data between successive sets of data. This is done through Kafka within the Edge platform, where the received data is shared on the edge platform and data is shared with many consumers that consume the data. Server processing data and send to different services. When data is sent through Kafka, consumers of such as databases and web servers consume the data immediately and proceed as follows:

```
> db.temperature.find({})
{ "_id" : ObjectId("61397193 6f5b64ad6b9ee464"), "1631154579442" : 25.69 }
{ "_id" : ObjectId("613971936f5b64ad6b9ee465"), "1631154579556" : 25.67 }
{ "_id" : ObjectId("613971946f5b64ad6b9ee466"), "1631154580529" : 25.75 }
{ "_id" : ObjectId("613971956f5b64ad6b9ee467"), "1631154581550" : 25.73 }
{ "_id" : ObjectId("613971966f5b64ad6b9ee468"), "1631154582529" : 25.73 }
{ "_id" : ObjectId("613971976f5b64ad6b9ee469"), "1631154583584" : 25.73 }
{ "_id" : ObjectId("613971986f5b64ad6b9ee46a"), "1631154584566" : 25.67 }
{ "_id" : ObjectId("613971996f5b64ad6b9ee46b"), "1631154585527" : 25.79 }
{ "_id" : ObjectId("6139719a6f5b64ad6b9ee46c"), "1631154586560" : 25.79 }
{ "_id" : ObjectId("6139719b6f5b64ad6b9ee46d"), "1631154587521" : 25.75 }
{ "_id" : ObjectId("6139719c6f5b64ad6b9ee46e"), "1631154588528" : 25.81 }
{ "_id" : ObjectId("613971d36f5b64ad6b9ee46f"), "1631154643989" : 25.69 }
{ "_id" : ObjectId("613971d46f5b64ad6b9ee470"), "1631154644121" : 25.79 }
{ "_id" : ObjectId("613971d56f5b64ad6b9ee471"), "1631154645058" : 25.75 }
{ "_id" : ObjectId("613971d66f5b64ad6b9ee472"), "1631154646065" : 25.73 }
{ "_id" : ObjectId("613971d76f5b64ad6b9ee473"), "1631154647088" : 25.75 }
{ "_id" : ObjectId("613971d86f5b64ad6b9ee474"), "1631154648072" : 25.81 }
{ "_id" : ObjectId("613971d96f5b64ad6b9ee475"), "1631154649077" : 25.73 }
{ "_id" : ObjectId("613971da6f5b64ad6b9ee476"), "1631154650072" : 25.69 }
{ "_id" : ObjectId("613971db6f5b64ad6b9ee477"), "1631154651095" : 25.79 }
```

Fig. 5: Temperature sensor data logged in edge cloud systems

Figure 5 shows the logging of the traffic that each IoT node sends to the server. Each IoT node transmits data measured by the node along with its ID. Using pub/sub technology and REST API, we established stable and high-performance machine-independent interaction while giving services to several users at the same time by maintaining sessions for each user.

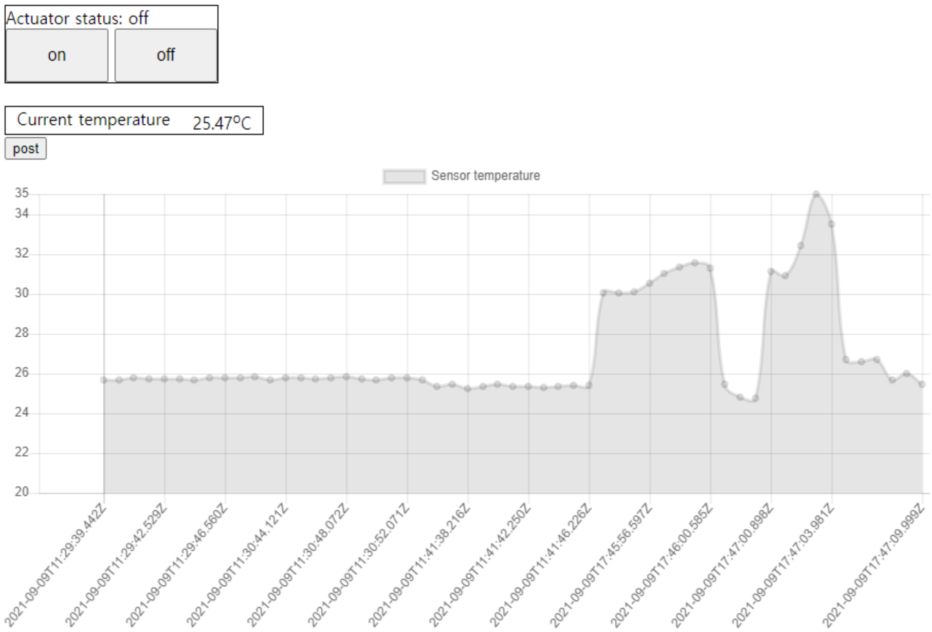

Fig. 6: Web dashboard and acquired data

The web dashboard in Fig. 6 utilizes the sensor data delivered to the DB through the edge platform to visualize it through a graph tool on the web, communicates with the edge platform through a web socket, and checks the status of the smart farm in real-time. The x-axis draws a graph with the date value scraped from the DB by the server, and the Y-axis draws a graph with the temperature value. In the upper left corner are values indicating the state of the actuator and buttons that can be adjusted manually. At the bottom, the current farm temperature value is displayed.

Fig. 7 shows an experiment to evaluate the transmission rate according to the data size. In this experiment, performance evaluation was performed with a total of 10 million records. We can see that the maximum value is when the record size is 500 Bytes. Therefore, when transmitting data, it seems desirable to divide data into 500 Byte data size with the maximum data rate and transmit it when transmitting large data on the MQTT Kafka platform.

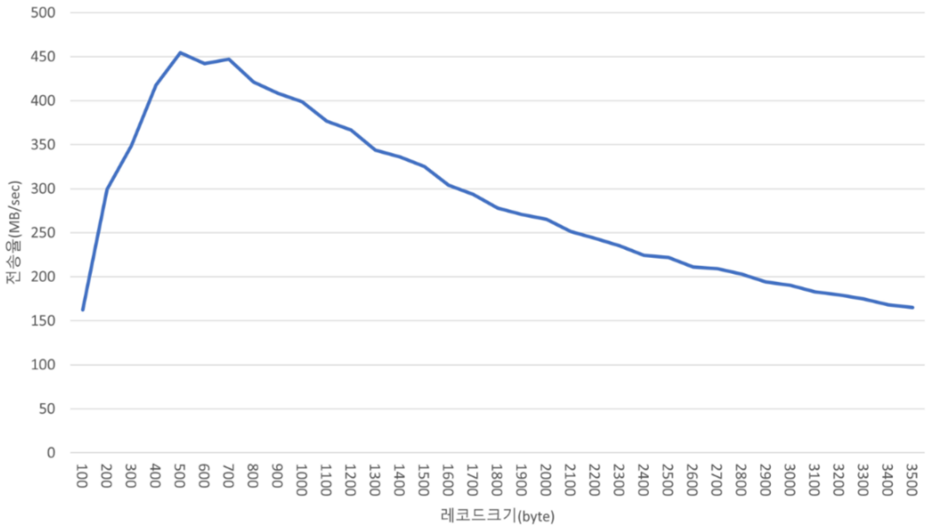

Fig. 7: Performance depending on record size

4. Conclusions

Internet of Things can be built easily and quickly through the edge platform using MQTT and KAFKA implemented in this study. In smart farms, efficiency can be increased based on massive sensor data, and sales can be expected to increase as production increases. In addition, as the time and technology required building a smart farm decreases the burden of continuous change and the introduction of technology is expected to decrease. Through the smart farm edge platform technology using MQTT and KAFKA developed in this study, we can expect to increase the sales of the farm and contribute to the continuous development of big data-based services in the future. In addition, the edge platform is expected to reduce the cost of sensor replacement by contributing to compatibility in real farms where sensors need to be constantly replaced.

Acknowledgement

This research was supported by Chungbuk National University Korea National University Development Project (2022)

References

Ahmed, N., De, D., & Hussain, I. (2018). Internet of things (IoT) for smart precision agriculture and farming in rural areas. *IEEE Internet of Things Journal*, 5(6), 4890-4899.

Alandjani, G. O. (2021). Leveraging Vulnerabilities in Sensor Based IoT Edge

Computing Networks. *International Journal of Future Generation Communication and Networking*, NADIA, 14(1), (pp. 11-20), DOI: 10.33832/ijfgcn.2021.14.1.02.

Anusha, A. Guptha, G., Sivanageswar R. G., & Tenali, R. K. (2019). A model for smart agriculture using IoT. *International Journal of Innovative Technology and Exploring Engineering (IJITEE)*, 8(6).

Atijosan, A., Essien, E. Badru, R., &Alaga, T. (2021). Land suitability evaluation for cassava production using integral value ranked fuzzy AHP and GIS techniques. *International Journal of Hybrid Innovation Technologies*, 1(2), 43-62, DOI:10.21742/IJHIT.2021.1.2.03

BC diploma. The digital credentials on the blockchain, https://www.bcdiploma.com/en.

Crop Type (Vegetables Fruits Flowers) Equipment (HV AC LED Grow Lights Irrigation Systems Material Handling Equip-ment Control Systems)", Input Type and Region - Global Forecast to 2026, 2021.

Donzia, S. K. Y., Kim, H.-K., & Hwang, H. J. (2019). A software model for precision agriculture framework based on smart farming system and application of IoT gateway. *Springer International Publishing*, 49-58.

El Azzaoui, A., Choi, M. Y., Lee, C. H., & Park,J. H. (2022). Scalable lightweight blockchain-based authentication mechanism for secure VoIP communication. Article number: 12(8).

Gafni, A. (1988). Rollback mechanisms for optimistic distributed simulation systems. *Distributed simulation.*

Gilmore, B. (2014). The next step in internet evolution: The internet of things. *Internet of Things cmswire*.

Gokhale, P., Bhat, O., & Bhat, S. (2018). Introduction to IOT. *International Advanced Research Journal in Science Engineering and Technology (IARJ SET)*, 5(1).

Goyal, S., Mundra, U., & Shetty, S. (2019). Smart agriculture using IoT. *International Journal of Computer Science and Mobile Computing*, 8(5), 143-148.

Grigg, I. The Ricardian contract. *Proceedings of the First IEEE International Workshop on Electronic Contracting*.

Guillermo, J. C., García-Cedeño, A., Rivas-Lalaleo, D., Huerta, M., & Clotet, R. (2019). IoT architecture based on wireless sensor network applied to agricultural monitoring: A case of study of cacao crops in Ecuador. *Advances in Information and Communication Technologies for Adapting Agriculture to Climate Change II*, 42-57.

Hammad, M., Iliyasu, A. M., Elgendy, I. A., & Abd El-Latif, A. A. (2022). End-to-end data authentication deep learning model for securing IoT configurations. Article number, 12(4).

Keswani, B., Mohapatra, A. G., Mohanty, A., Khanna, A., Rodrigues, J. J. P. C. Gupta, D. (2018). Adapting weather conditions based IoT enabled smart irrigation technique in precision agriculture mechanisms. *Neural Computing and Applications*.

LG Tuiun, https://www.donga.com/news/It/article/all/20211223/110932008/1.

Naresh, M. & Munaswamy, P. (2019). Smart agriculture system using IoT technology. *International Journal of Recent Technology and Engineering (IJRTE)*, 7(5).

Park, H. D. (2021). A structure of intelligent smart farm control with correlated interface using edge computing. *International Journal of Hybrid Information Technologies*, 1(1), 51-60. doi:10.21742/IJHIT.2021.1.1.04.

Share & Trends Analysis Report By Type (Aggregate Systems Liquid Systems) By Crops (Tomatoes Lettuce Peppers Cu-cumbers Herbs) By Region And Segment Forecasts 2019 – 2025, Jan 2020.

Shin, S., Eom, S., & Choi, M. (2022). Soft core firmware-based board management module for high performance blockchain/fintech servers. Article number: 12(3).

Swain, A., Chatterjee, S., & Vishwanath, M. (2021). Hydroponics in vegetable crops: A review. *The Pharma Innovation Journal*, 10(6), 629-634.

Vanipriya, C. H., Maruyi, S. M., & Gupta, G. (2021). Artificial intelligence enabled plant emotion xpresser in the development hydroponics system. *Second International Conference on Aspects of Materials Science and Engineering (ICAMSE*).

Verma, S. & Gawade, S. D. (2021). A machine learning approach for prediction system and analysis of nutrients uptake for better crop growth in the hydroponics system. *2021 International Conference on Artificial Intelligence and Smart Systems (ICAIS)*.