ISSN:1738-6764, eISSN:2093-7504

Automated Tool for Building Hyperledger Fabric Blockchain Networks Using Ansible

Vungsovanreach Kong [1], Saravit Soeng [2], Kyung-Hee Lee [3] and Wan-Sup Cho [4,*]

[1] Department of Big Data, Chungbuk National University, Korea, Ph.D. Student; kv.sovanreach@gmail.com
[2] Department of Big Data, Chungbuk National University, Korea, Ph.D. Student; soengsaravit@gmail.com
[3] Department of Management Information Systems, Chungbuk National University, Korea, Ph.D; lee.kyunghee@gmail.com
[4] Department of Management Information Systems, Chungbuk National University, Korea, Professor; wscho@cbnu.ac.kr
* Correspondence

https://doi.org/10.5392/IJoC.2023.19.2.015

Manuscript Received 27 October 2022; Received 04 April 2023; Accepted 26 April 2023

Abstract: *Building a blockchain network is time-consuming and challenging. It requires extensive information technology (IT) knowledge and relevant skills. To simplify this task, global IT companies provide cloud-based blockchain services. In this paper, we proposed a cloud-based construction and management tool called smart blockchain network constructor (SBC) that could allow blockchain developers, operators and enterprises to easily deploy blockchain networks within their infrastructure and save time and cost. SBC employs Hyperledger Fabric, a well-known private blockchain platform. Ansible is an open-source IT automation engine that could support network-wide deployment. Instead of complex and repetitive text commands, SBC provides a user-friendly web interface that allows users to setup, deploy, and interact seamlessly with blockchain networks. To verify its usefulness and convenience, a blockchain network that could conduct food safety checks and an electronic voting decentralized app were built and tested with SBC. The process of blockchain network construction, which involves writing more than ten settings files and executing numerous command lines, can be replaced with simple input and click operations from a graphical user interface. A Hyperledger Fabric channel was used to enable independent and reliable tasks in both food safety and smart election tasks. As a result, an automated system was created to simulate the necessary manual work in constructing blockchain networks. We observed that the proposed automated system dramatically reduced the time consumption of blockchain network construction compared to manual construction. We strongly believe that this system will be highly beneficial to both blockchain operators and developers for their applications built on blockchain technology.*

Keywords: Blockchain; Ansible; Blockchain Network Construction Tool; E-voting App; Food Safety-Checking System

1. Introduction

Although initially this technology was focused on digital currencies, it has now expanded to various other industries owing to its shared, immutable, and decentralized method of operation [1]. Trust Internet powered by blockchain has changed the methods of conducting business and is transforming businesses across industries, including supply chains, internet of things, financial services, and healthcare.

From the software architecture viewpoint, blockchain is a new type of distributed system comprising distributed nodes that synchronize and share a distributed ledger and operate autonomously based on a predetermined consensus algorithm [2]. Blockchain-based applications have several advantageous characteristics, such as data immutability, transparency, integrity, fair access, and non-repudiation of transactions. The Ethereum and Hyperledger Fabric blockchain platforms support the creation, management, and operation of blockchain networks; they have been popularized as open-source products. This study is limited to research employing the Hyperledger Fabric framework [3]-[5].

However, developing a blockchain-based application using Hyperledger Fabric entails the complex process of designing and building a blockchain network to implement and deploy smart contracts [3-5]. It is a time-consuming and expensive process because it requires various components, such as operating systems, shell scripts, network infrastructure, and program distribution. Moreover, developers and administrators must manually execute script files repeatedly at all nodes participating in the blockchain network, and errors may occur during this process.

This study aims to develop a tool that could reduce the manual process of building a blockchain network and allow organizations to easily configure and manage blockchain networks using their own infrastructure through a user-friendly web interface. To achieve this, we developed an automation tool called the smart blockchain network constructor (SBC) to simplify the task of creating and deploying a blockchain. It has the following features:

- Network management: Users can set up computers as target nodes to deploy and run blockchain components.
- Software installation: Setting up a blockchain requires the installation of several software packages at each node, and manually performing this task on each computer is time-consuming. With SBC, users can easily install the required software with the requisite version on any computer in the network.
- Channel and chaincode management: Users can create channels and install or upgrade the chaincode through a convenient user interface (UI). Subsequently, they can verify whether the chaincode is working properly through the UI.
- Blockchain monitoring tool: Users can verify the status of the blockchain network, channels, blocks, and transaction history.
- Web UI: Users can easily manage and track the setup of the blockchain network.

To verify the usefulness of SBC, we used it to develop two blockchain applications: a food safety reinforcement system and a smartphone-based electronic voting (e-voting) decentralized application (DApp). In the food safety reinforcement system, a blockchain network was established to enhance the information regarding agricultural products, certify eco-friendly agricultural products, and maintain information transparency in the distribution process. It was observed that food safety data recorded in the blockchain ledger increased reliability, enabled quick tracking and response when a problem occurred with a specific agricultural product, and clearly indicated the responsible party. In the e-voting app, the voting results were recorded in a distributed ledger that was retained by each stakeholder, thereby increasing the credibility of e-voting. The reliability of the e-voting app was confirmed by using it for four university student representative elections. Additionally, by creating and operating a separate channel for each election, the safe storage of voting information and privacy were improved.

The remainder of this paper is organized as follows. Section 2 presents the related work. Section 3 presents the user requirements, architecture of the proposed system, and its system implementation. Section 4 describes two real-life applications of SBC. Finally, Section 5 concludes the paper and presents directions for future research.

2. Related Work

Similar to our research, there are a few popular automated tools for enhancing the deployment of blockchain networks. First and foremost, there is Amazon Managed Blockchain, a fully managed service that allows users to create and manage scalable blockchain networks using the popular open-source frameworks Hyperledger Fabric and Ethereum [6]. This cloud-based blockchain service streamlines the time-consuming and complex process of establishing and managing blockchain networks. Users can quickly deploy blockchain applications with Amazon Managed Blockchain, which includes built-in features such as node deployment, network management, and automatic scaling [7-8]. FabDep is another automated tool that automates the deployment and management of Hyperledger Fabric blockchain networks [9]. It employs Kubernetes and Docker to build a blockchain network that is easily scalable and manageable. FabDep provides a straightforward interface for deploying Hyperledger Fabric networks in the cloud or on-premises, allowing users to quickly spin up a new network with a few mouse clicks [10]. FabDep also includes monitoring and alerting capabilities to ensure that the network is running smoothly. Finally, there is Hyperledger Bevel, a blockchain platform designed for enterprise-level applications [11]. It gives you a modular framework for creating and deploying distributed ledger applications. It offers a wide range of features such as smart contract execution, consensus mechanism, privacy and confidentiality, identity management, interoperability, and scalability [12].

Our research is distinguished by its ability to be deployed on user-owned servers rather than cloud infrastructure. This means that users can configure and run the blockchain network on their own server by configuring server connection profiles like IP address and port. This feature provides users with a level of control and autonomy not found in existing blockchain tools, which typically require users to deploy and operate on an enterprise-cloud infrastructure. The tool provides a more flexible and convenient solution for users who prefer not to rely on third-party cloud infrastructure providers such as Google cloud, Amazon cloud, and Azure cloud by allowing them to deploy the blockchain network on their own local server. Because of this flexibility, users have more control over their own data and infrastructure and can tailor their deployment to meet their specific needs and requirements. Because users can configure the blockchain network on a local server, they can avoid potential latency and bandwidth issues that may arise when operating on cloud infrastructure. This is especially important for users who need quick and dependable access to their data and applications.

Overall, the tool's deployment on the user's local server provides a significant advantage over existing blockchain tools, making it a compelling choice for users who value autonomy, control, and customization in their blockchain deployment.

This section briefly describes the process of constructing a Hyperledger Fabric-based blockchain network, which comprises complex construction steps, as shown in Figure 1 [3].

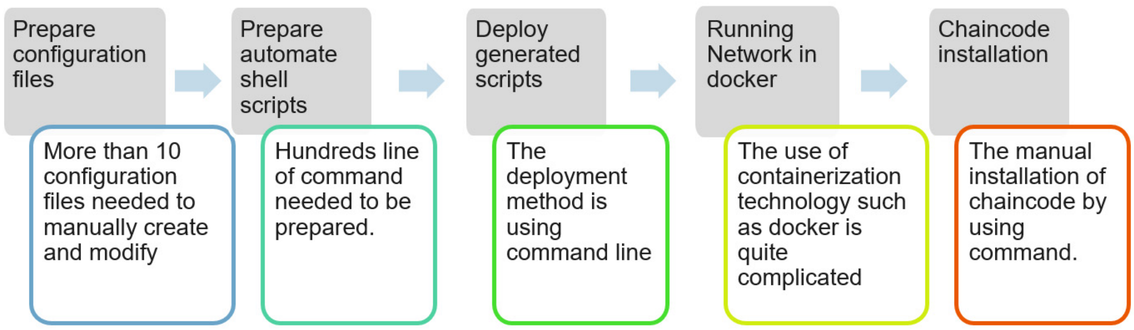

Figure 1. Detail process for constructing and deploying a Hyperledger Fabric-based blockchain network.

To start a Hyperledger Fabric blockchain network, trusted nodes must be established using the membership service provider component [13]. A cryptogen binary file is used to generate the required certificates for the peers and orderers. The cryptogen tool uses a YAML configuration file to obtain the required certificates. The cryptoconfig.yaml is the default configuration file. After execution, the cryptogen file generates certificates and keys, which can be used by the nodes to join the blockchain network.

After the certificates and keys are generated, the configtxgen tool is used to generate some important blockchain artifacts, such as channel.tx, genesis block, and anchor peer transactions. The channel contains the blockchain ledger and it is the mechanism that allows related peers to join a business network. The genesis block is the first block in a blockchain. It is used to bootstrap the ordering service, and it contains the channel configuration. The anchor peer transactions specify the anchor peer of each organization in the channel.

To start the blockchain network, the docker compose command is used to run the Docker containers. Each organization comprises a Fabric certificate authority (CA) container, a Fabric peer container, and a CouchDB container. The ordering service can be configured using SOLO or Kafka systems.

Once all nodes are running, the Docker command line interface (CLI) container is used to create and join the channel and update the anchor peers in each organization [14]. Next, smart contracts are installed at the peers of each organization and instantiated using the Docker CLI container. An endorsement policy is applied when smart contracts are instantiated. The endorsement policy defines the node that is permitted to endorse transactions when smart contracts are invoked. Finally, the Hyperledger Fabric blockchain network starts running and can communicate with the application programming interface (API) layer, where users can use the client software development kit (SDK) to communicate with the blockchain network.

Configuration management is the process of systematically managing, organizing, and confirming changes. The configuration is designed to eliminate errors and confusion caused by version updates, mismanagement, deployment, or maintenance. Configuration management provides tools for handling and managing packages, configuration files, and other settings to prepare the servers for their allocated tasks. Without configuration management, this process will be time-consuming and prone to human error. The automation of configuration management eliminates a significant amount of manual work and improves reliability and consistency.

Most system administrators tend to overlook the preparation for system deployment and configuration. Instead, they prepare shell scripts or employ pre-configured system images. Manually writing custom scripts may result in human errors, and a custom script that is written by an individual might not be easy to understand for others. Furthermore, the custom script might be ambiguous, and its nature might be unpredictable.

Automation of configuration management aims to eliminate all manual work. By using an automated configuration, operations such as upgrading, monitoring, and status checking can be performed rapidly, and the system administrator can track all changes and errors. Thus, an automated configuration preemptively eliminates many hassles and makes the process highly efficient and productive.

3. System Design and Implementation

This section describes the design and implementation of the SBC tool proposed to build and manage blockchain networks. First, user requirements are introduced, and then the use cases of SBC are presented. Thereafter, the system architecture and implementation are described, and finally, the construction and execution of the e-voting blockchain using SBC are explained.

3.1 User Requirements

In this study, we categorized users into two types: blockchain operators and developers. To elucidate the functional requirements expected by these users, some statements by them are listed in Table 1. Thereafter, the different user types are defined in more detail using use case and sequence diagrams.

Table 1. User statements describing the expected functional requirements from a blockchain construction and management tool.

User type	Task
Blockchain operator	Assign hosts and setup an organization in my own infrastructure. Pre-install the required software and verify software installation. Transfer all blockchain related configuration files to all hosts. Deploy blockchain related components to the target hosts. Start blockchain network at all the target hosts. See network health status at all running peers.
Blockchain developer	Install, upgrade, and test chaincode via the UI. Download the connection profile for dApp development. Explore the number of block and transaction history by organizations.

3.2 Use Cases

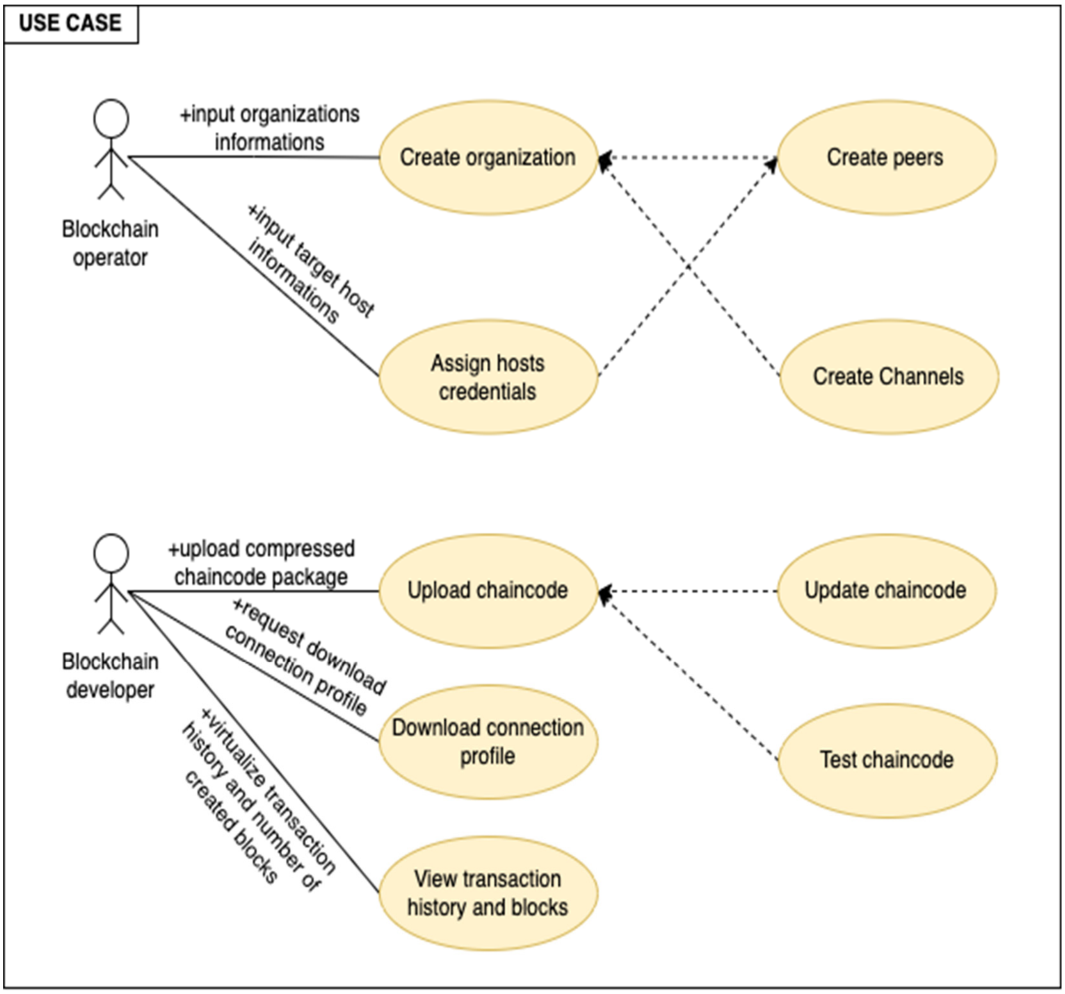

Figure 2. Detail use cases and functionalities of SBC for both blockchain operators and developers.

Our system comprised two types of users: blockchain operators and developers. Figure 2 shows the various use cases of the system. Blockchain operators can create organizations, peers, and channels to run the blockchain network. Additionally, they can assign a specific host machine to run the blockchain network. Blockchain developers can upload their chaincode packages to the network and download the connection profile of the network for developing DApps. Additionally, they can use the blockchain explorer tool to check the transaction history and the number of blocks that have been created.

3.3 System Architecture

Figure 3 presents an overview of the SBC architecture, which comprises the following four main layers:

- Blockchain construction web application
- Ansible automation engine
- Hyperledger Fabric network
- E-voting web application

The blockchain construction web application layer is the top layer, which is used for user interactions. This layer provides all functionalities required to construct the blockchain network through a web interface. Communication between web applications and the Ansible automation engine [15] is performed via the REST API.

The Ansible automation engine layer is between the web application and blockchain network. It comprises the Ansible inventory files used to store host information, Ansible playbooks used to store the tasks, and various configuration templates related to Hyperledger artifacts, chaincode, and docker compose templates.

The Hyperledger Fabric network layer is the main layer, which is constructed using the Ansible automation engine [14]. It comprises the Hyperledger Fabric components such as an ordering service, organization peers, CouchDB for all organization peers, CA for each organization, and installed smart contracts.

After building a blockchain network for a specific purpose using the SBC, DApps can be developed using the Fabric client SDK-based REST API. The development of the food safety and e-voting DApps using the SBC is described in Section 4. Because the food supply chain comprises multiple channels wherein various companies participate, the creation and management of channels is important. Additionally, the election blockchain can be reused for each election; however, a separate blockchain network must be built for each election. When a new election is launched, a separate channel to support it can be easily created in Hyperledger Fabric using the SBC tool.

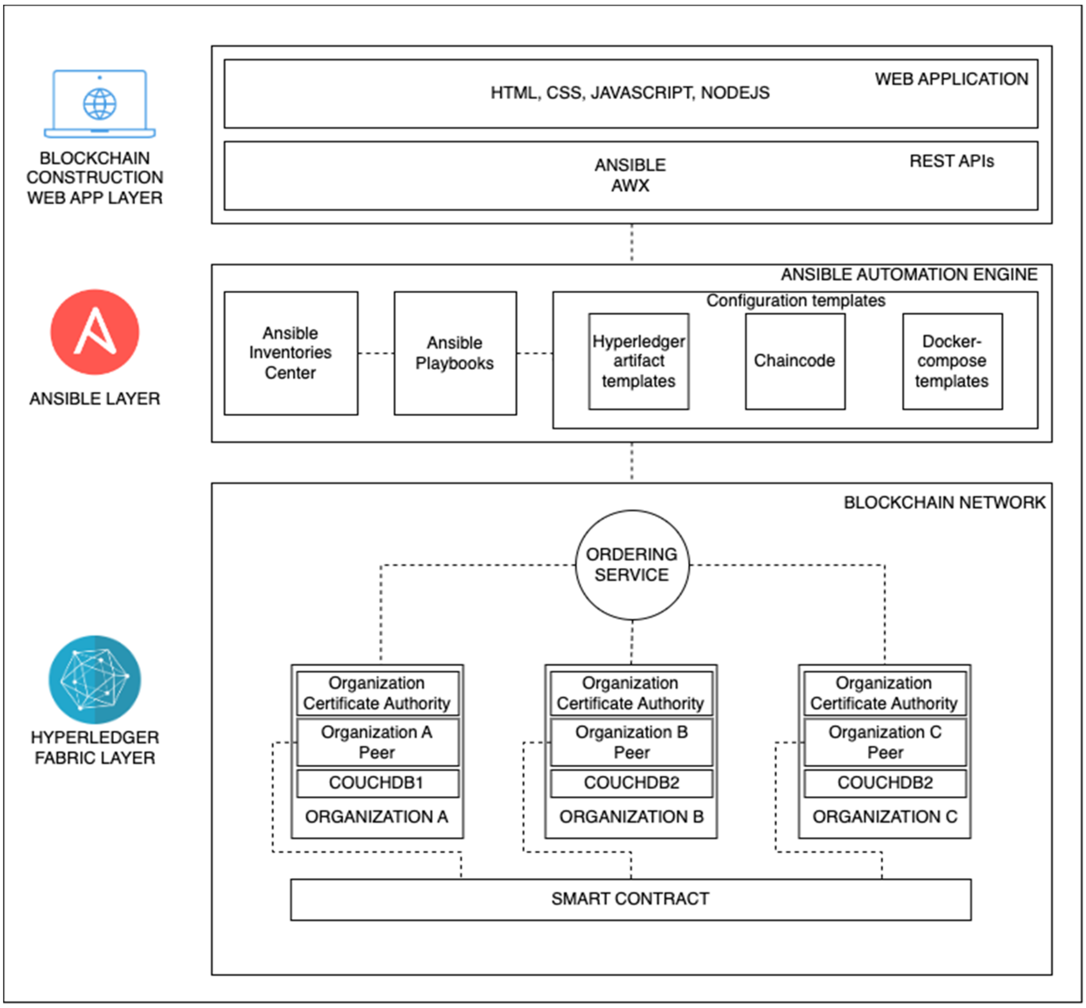

Figure 3. An end-to-end diagram describes the detail system architecture of SBC from user interaction GUI to blockchain network.

3.4 System Implementation

This section describes the components required to implement the proposed SBC system including heardware and software components.

3.4.1 Hardware Components

A prototype of the proposed system was developed using a desktop computer. The system was tested using a sample use case of three computers: one control and two managed nodes. The control and managed nodes comprised Intel CoreTM i7-6700 CPUs @ 3.40 GHz and Ubuntu version 16.04 LTS operating system. The control and managed nodes comprised 32 and 16 GB of RAM, respectively. The control node requires a higher RAM capacity as it runs the Ansible engine.

3.4.2 Software Components

The software comprises three parts. The first part includes components related to the Ansible automation engine [15]. We used three sample inventory files: hosts_solo.yml, hosts_dedicated_orderer.yml, and hosts_kafka.yml. The role of each host was specified in these inventory files. These roles could be root_orderer, orderer, peer, root_peer, explorer, zookeeper, or kafka_broker. The second part includes components related to Hyperledger Fabric, such as Hyperledger Fabric CA, Hyperledger Fabric peer, and Hyperledger explorer. The third part includes components related to the web application implementation.

To interact with the Ansible automation engine, we used Ansible AWX [15], which is an open-source version of the Ansible tower [15]. The AWX is built to run on top of Ansible and provides the REST endpoints for client-side interactions. We made HTTP requests to the Ansible AWX endpoints to execute the Ansible playbooks. For the front-end, we employed client-side technologies such as HTML, CSS, and JavaScript. Figure 4 shows the architecture of the web application specific to food distribution.

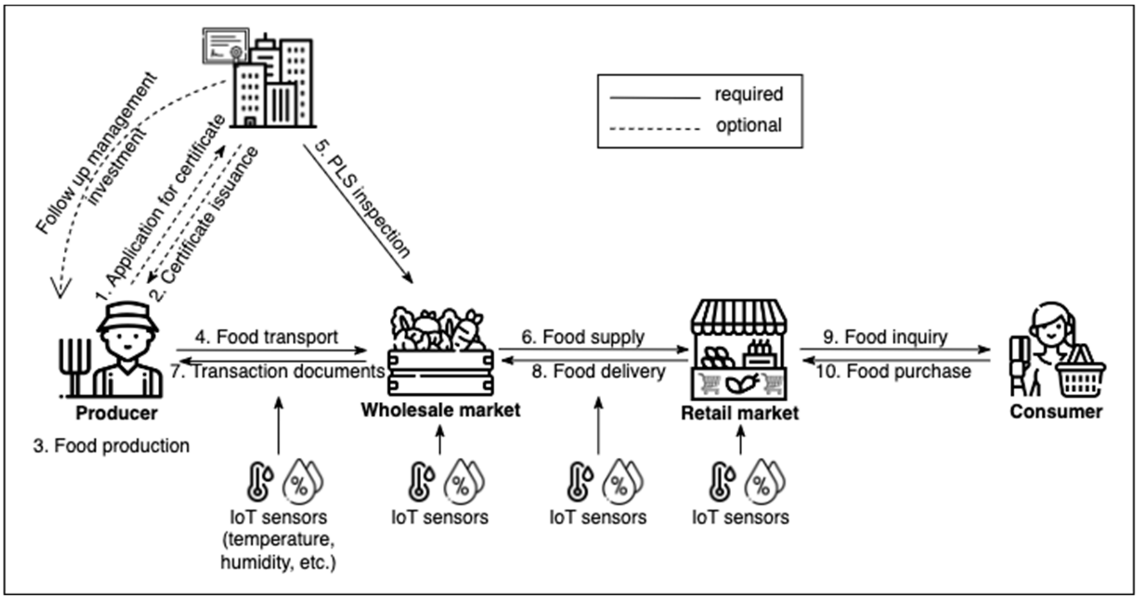

Figure 4. A descriptive scenario of the food safety-checking system built on top of blockchain network that was constructed using SBC.

4. Running Examples

This section described two real-world implementations, Food Safety and E-Voting systems, where blockchain networks were constructed using SBC.

4.1 Food Safety Blockchain

Food products are distributed through several channels. They may be sold without an intermediate channel in a direct transaction between the producer and consumer or sold through intermediate channels such as consumer groups, producer organizations, specialized stores, or ecommerce platforms. Figure 4 shows a complex case of food distribution that involves various trade channels. It shows the channels between producers and consumers, which include food certification and inspection agencies, wholesalers, and retailers. In each channel, data related to food safety, such as temperature, humidity, certificate of origin, quality inspection and certificates, and contract documents, are generated and delivered. These data must be shared with numerous parties in real-time and managed in a trustworthy manner.

Additionally, each company involved in the supply chain may want to share the ledger within its own channel in a distinct manner. In Hyperledger Fabric, different types of food supply chains can be configured using channels, and it is crucial to create and operate them rapidly. In a supply chain, certain companies may not want to share information with competitors; therefore, the flexibility of creating and managing channels is essential. Such flexible management requirements have increased the demand for tools that support quick and convenient creation and change of channels. This perspective emphasizes the requirement of a tool that can conveniently create and manage blockchain networks and channels.

In Figure 4, there is a channel wherein five nodes (institutions and enterprises) are connected, and multiple such channels can be created. Each node shares a distributed ledger and the data can be recorded or viewed on the distributed ledger through smart contracts. The data recorded in the distributed ledger cannot be changed and are synchronized with the ledgers of all nodes. The proposed tool makes it convenient to create and manage these blockchain networks and channels.

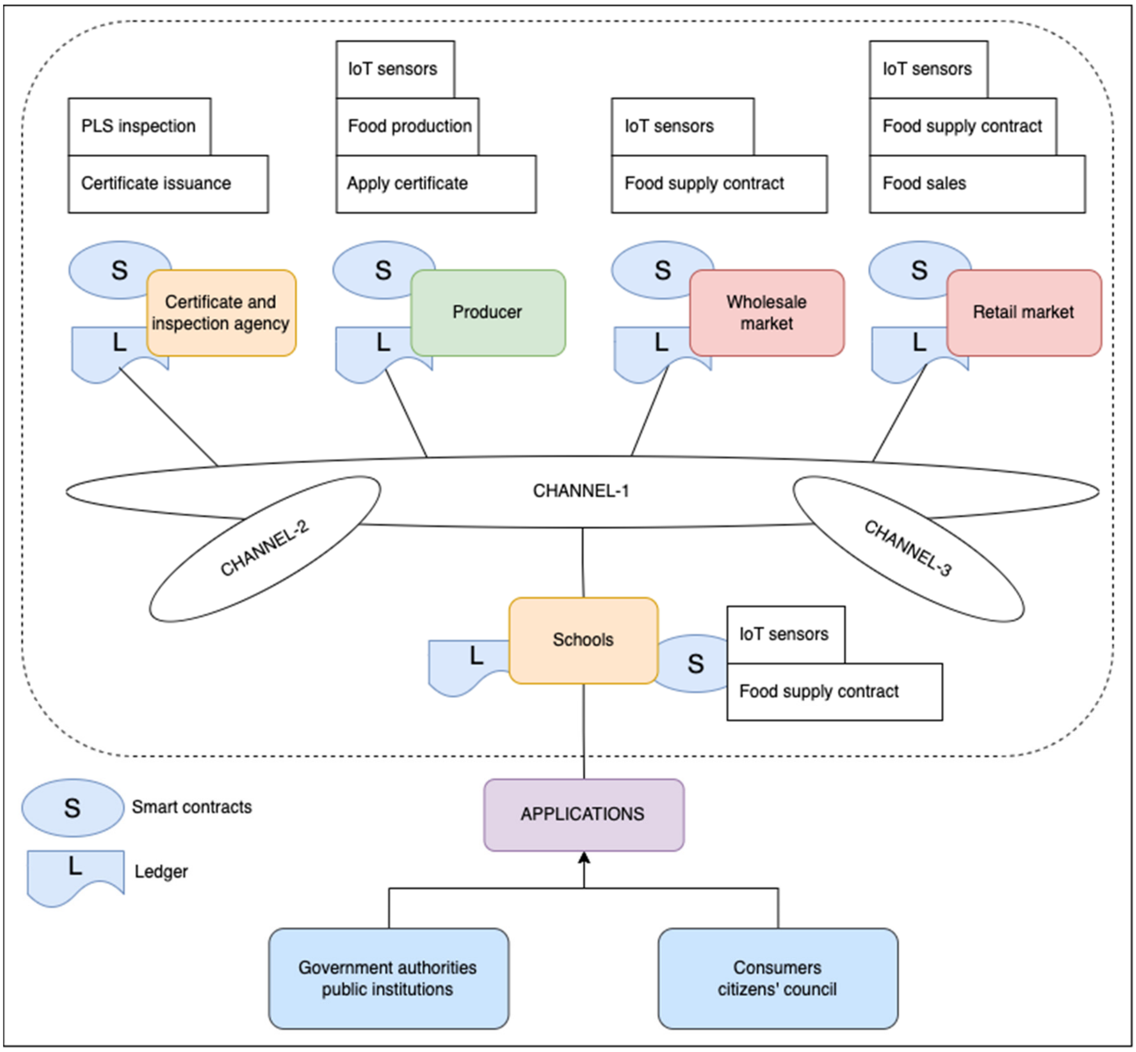

Figure 5. Detail visualization of the blockchain network built using SBC for a food safety-checking system.

4.2 E-Voting Application

We developed a blockchain-based e-voting DApp [16]. E-voting is a representative blockchain application because voting results must be highly reliable. The e-voting performed during this study involved electing a representative of the university student council, wherein thousands of students voted using smartphones, and the voting results were shared among several candidates and managers to ensure reliability. In a real-world case, we built a blockchain network using three organizations and six peers on three nodes. Moreover, the process could be repeated for each election with a different subject; however, it required configuring a separate channel in Hyperledger Fabric to minimize overhead owing to reconstruction, while also maintaining reliability.

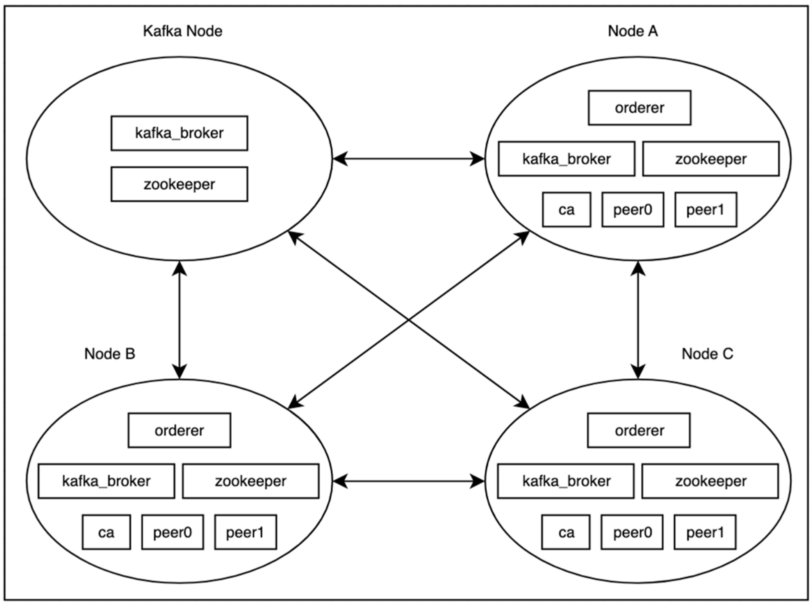

Figure 6. Hyperledger Fabric network built using the proposed blockchain construction and management tool.

4.3 The user interface of Smart Blockchain Network Constructor

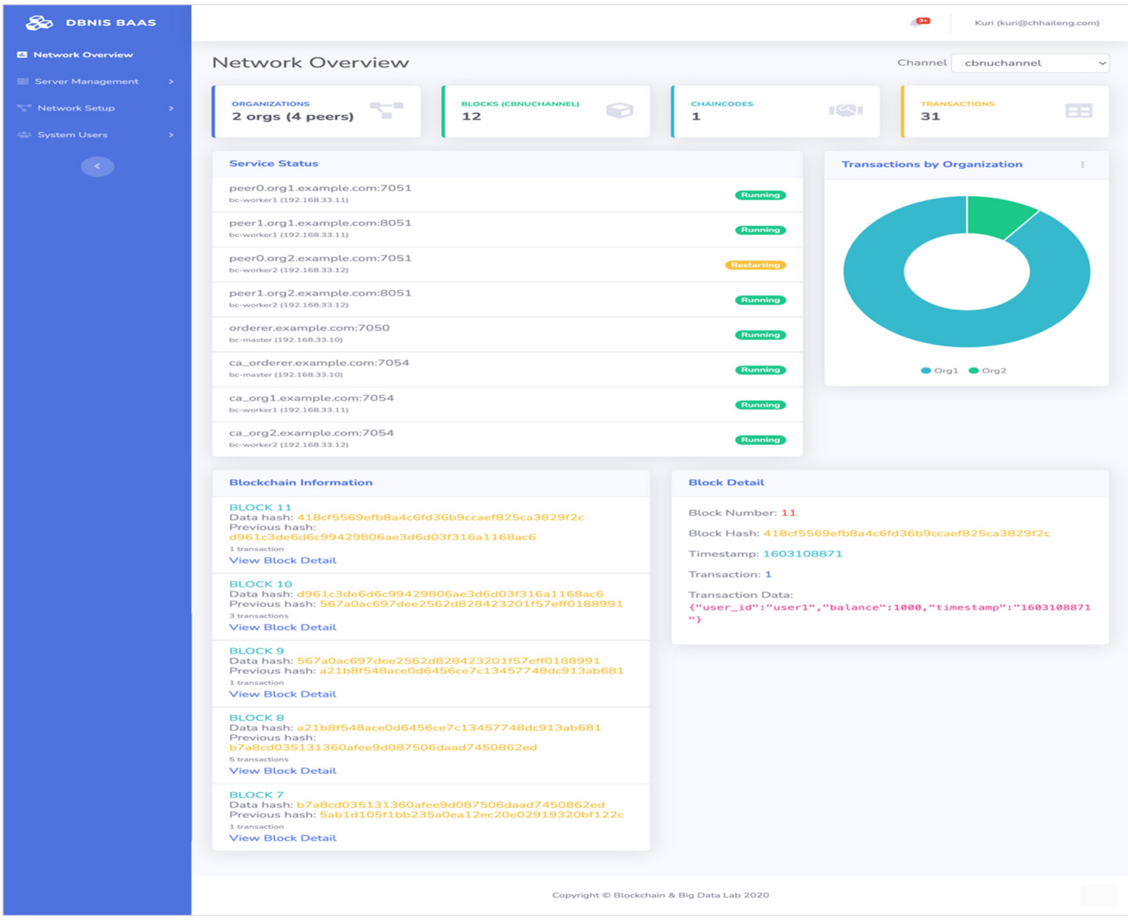

Figure 7. The main dashboard of SBC includes brief information on organizations, blocks, chaincode, transactions, and service status.

Figure 7 shows the dashboard of the SBC UI. The top of the dashboard displays the numbers of organizations constituting the blockchain network, nodes, current blocks, and the installed chaincodes. The service status and blocks information are shown at the bottom.

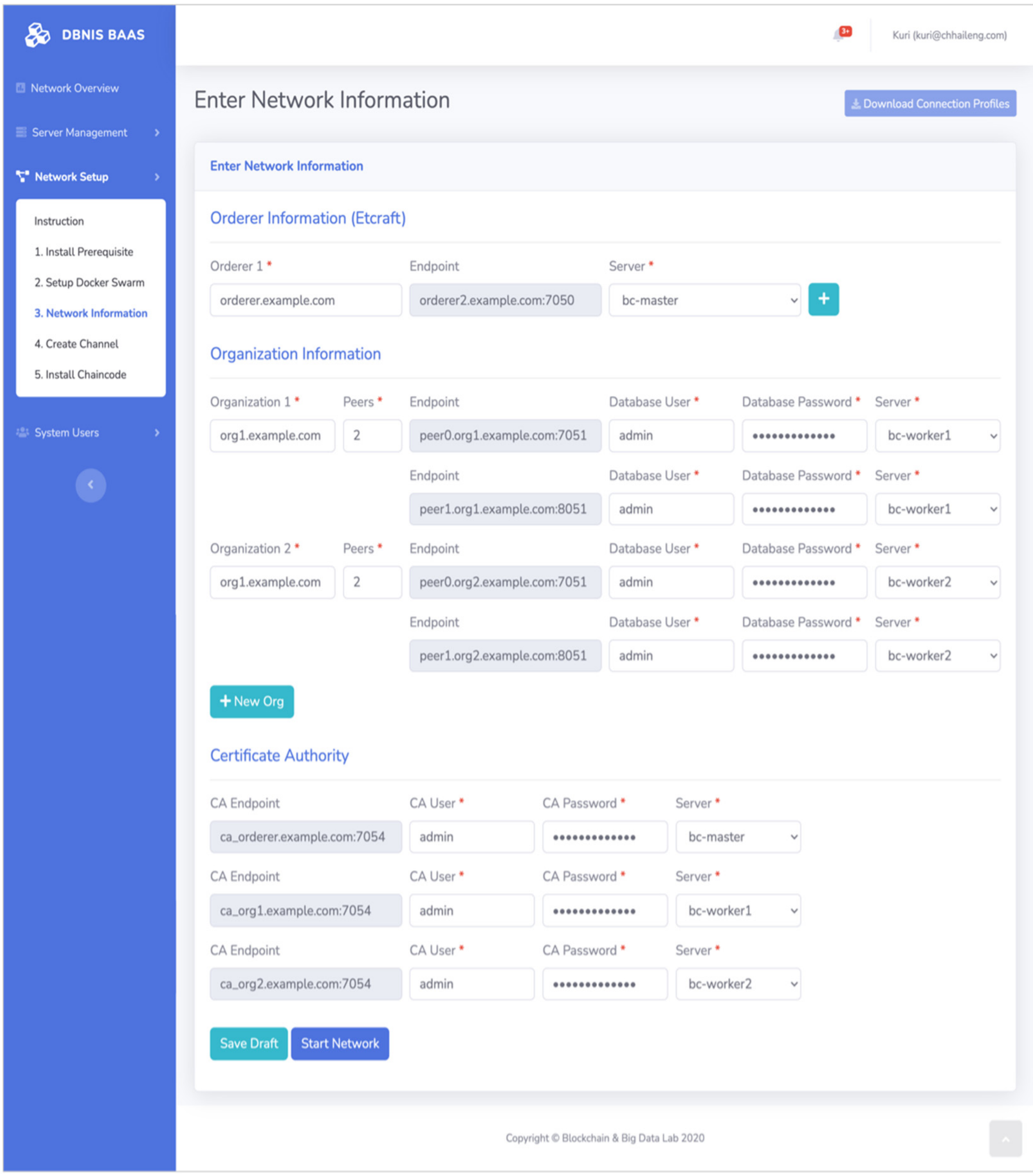

Figure 8. Blockchain network information list of each unit such as orders, organizations, and certificate authority.

Figure 8 shows the network configuration, wherein the creation process of two peers in each of the two organizations can be observed.

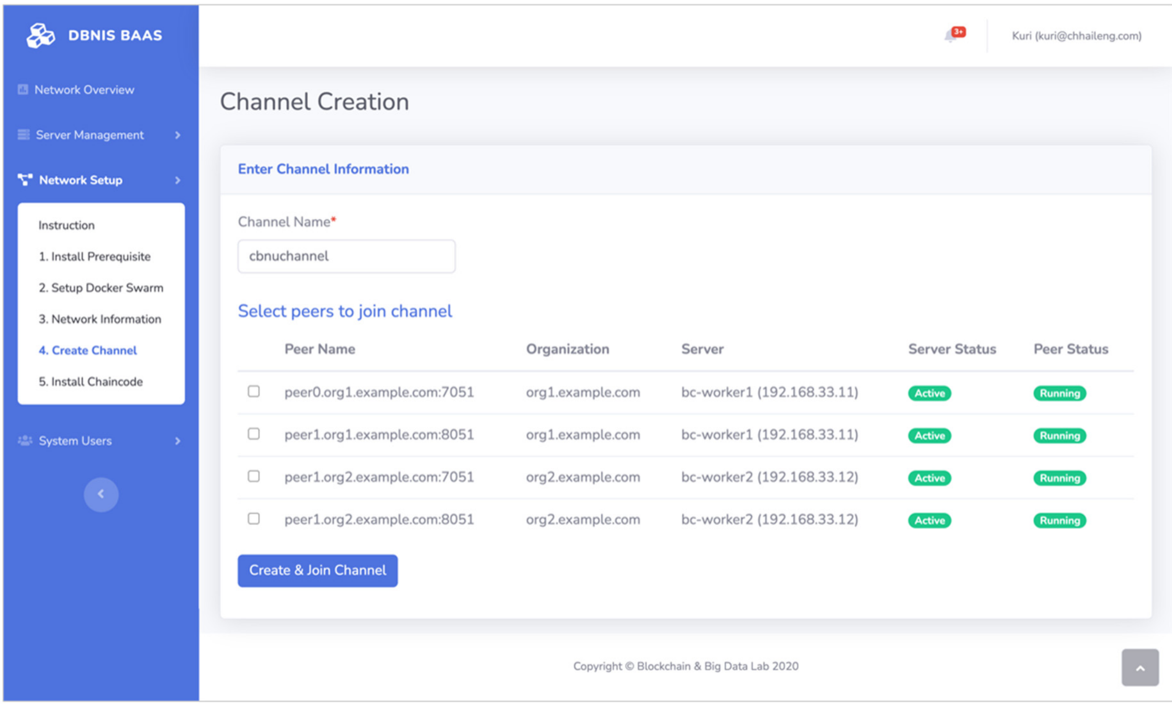

Figure 9. Network channel configuration for creating and joining peers to the organization channel.

Figure 9 shows the UI used for channel configuration such as creating and joining the channel.

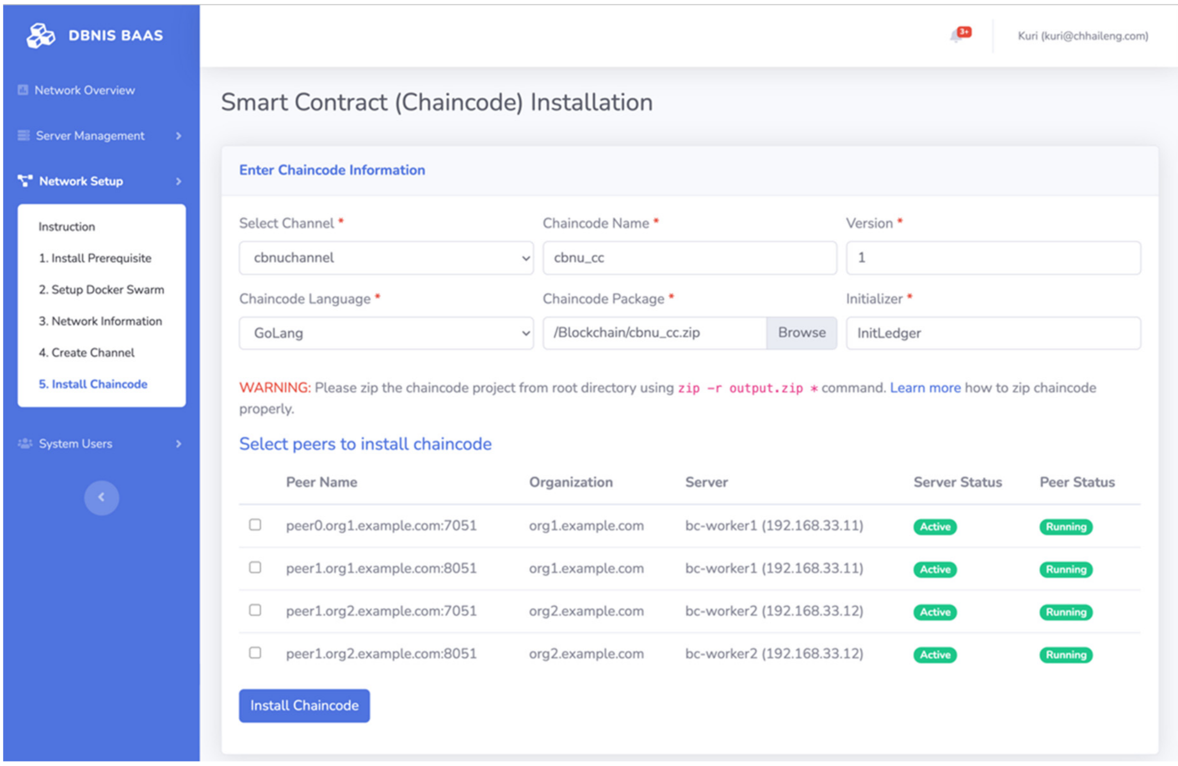

Figure 10. The installation of chaincode packages into peer machines of blockchain network.

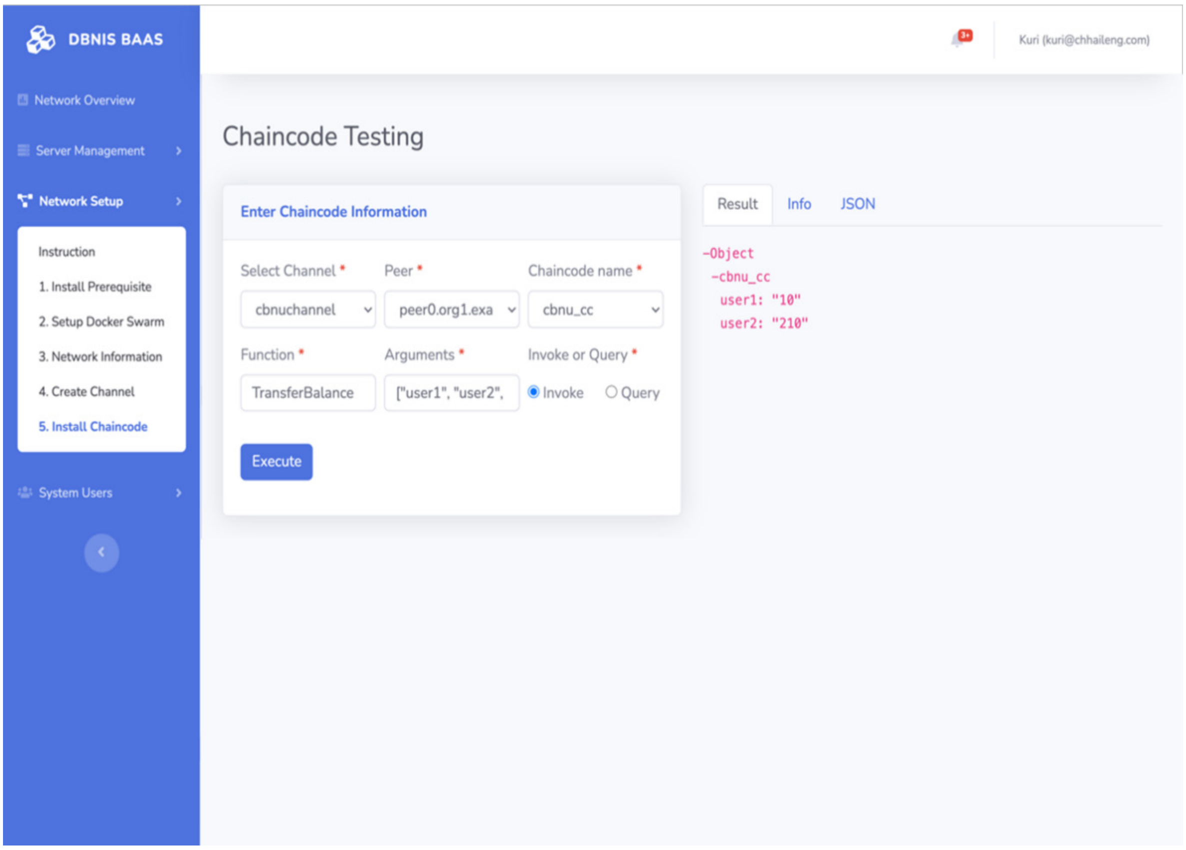

Figure 11. Chaincode testing by providing peer, function, and arguments information which indicates whether the installed chaincode works.

Figure 10 shows the UI used for installing a smart contract on the blockchain network and Figure 11 shows the UIs used for running and testing the installed chaincode.

5. Conclusions

This paper proposed a blockchain construction and management tool, called SBC, based on the Hyperledger Fabric and Ansible automation engine. It simplifies the setup and maintenance of the Hyperledger Fabric blockchain by reducing setup time, skill knowledge gaps, and human error. This study can significantly benefit blockchain operators, developers, and researchers aiming to implement blockchain-based applications rapidly. SBC provides a user-friendly web application that allows users to easily build and manage blockchain networks.

The usefulness of the proposed tool in real problems was verified by building two applications: a food safety-checking system and an e-voting app. The food safety blockchain system required building various types of food supply chains as blockchain networks (channels), and the proposed tool was effective in enabling the construction and management of such flexible blockchain networks. The blockchain-based e-voting app was used in four elections at the Chungbuk National University, and for each election, a new channel was created to store the voting results safely [16]. For conducting multiple elections, the administrator could use the tool to minimize the time and effort required to create and prepare separate blockchain networks and channels for each election. In future work, we plan to integrate the SBC with various products that support Hyperledger and provide continuous version upgrades.

Conflicts of Interest: The authors declare no conflict of interest.

References

[1] S. Underwood, "Blockchain beyond bitcoin," Communications of the ACM, vol. 59, no. 11, pp. 15-17, 2016, doi: https://doi.org/10.1145/2994581.

[2] X. Xu et al., "*A taxonomy of blockchain-based systems for architecture design*," in 2017 IEEE international conference on software architecture (ICSA), IEEE, pp. 243-252, 2017. doi: https://doi.org/10.1109/ICSA.2017.33.

[3] C. Cachin, "Architecture of the hyperledger blockchain fabric," in Workshop on distributed cryptocurrencies and consensus ledgers, vol. 310, no. 4: Chicago, IL, pp. 1-4, 2016.

[4] J. FRANKENFIELD. "Blockchain-as-a-Service (BaaS) Meaning and Major Players," 2022. Investopedia.com. https://www.investopedia.com/terms/b/blockchainasaservice-baas.asp

[5] Hyperledger Foundation. "Fabric CA Developer's Guide." Hyperledger Foundation. https://github.com/hyperledger/fabric-ca (accessed September 03, 2022).

[6] Amazon.com. "Amazon Managed Blockchain," 2023. https://aws.amazon.com/managed-blockchain/

[7] A. Long, D. Choi, and J. Coffman, "*Using Amazon Managed Blockchain for ePHI An Analysis of Hyperledger Fabric and Ethereum*," in 2022 IEEE World AI IoT Congress (AIIoT), IEEE, pp. 276-282, 2022. doi: https://doi.org/10.1109/AIIoT54504.2022.9817198.

[8] M. K. Elghoul, S. F. Bahgat, A. S. Hussein, and S. H. Hamad, "Secured Cloud-based Framework for Electronic Medical Records using Hyperledger Blockchain Network," Egyptian Computer Science Journal, vol. 46, no. 2, 2022.

[9] Fabricdeployer.com. "A Fabric Auto Deployment Tool," 2023. https://www.fabricdeployer.com/

[10] H. Singh. "DEPLOYING HYPERLEDGER FABRIC ON KUBERNETES," 2023. https://medium.com/debutinfotech/deploying-hyperledger-fabric-on-kubernetes-90dbd7ae10dd

[11] Hyperledger Foundation. "Hyperledger BEVEL," 2023. https://www.hyperledger.org/use/bevel

[12] Hyperledger Foundation. "An accelerator for delivering production-ready blockchain solutions," 2023. https://hyperledger-bevel.readthedocs.io/en/latest/index.html

[13] Hyperledger Foundation. "Hyperledger Fabric CLI," Hyperledger Foundation, 2022. https://github.com/hyperledger/fabric-cli

[14] D. Merkel, "Docker: lightweight linux containers for consistent development and deployment," Linux j, vol. 239, no. 2, p. 2, 2014.

[15] H. Sharma. "Ansible Tutorial for Beginners," 2022. Intellipaat.com https://intellipaat.com/blog/tutorial/devops-tutorial/ansible-tutorial/

[16] N. In-woo. "Chungbuk National University Conducts the Nation's First Blockchain Electronic Voting," Seoul Newspaper, 2022. https://www.seoul.co.kr/news/newsView.php?id=20191129500053