A Trustworthy Biological Assets Governance System Using Decentralized Identity

Zhengkang Fang, Jing Yu, Shufen Fang, Shuo Wang, Weilin Chan, Zexin Gao, Keke Gai

***Abstract*—Along with the development of the biological industry, the system of biological assets governance has a higher-level requirement in asset identity verification due to the demands of digitization and *Financial Technology* (FinTech). In order to achieve broad scalability and adaptability in the governance of biological assets' identities, this work proposes a *Biological Assets Decentralized Identity Management* (BA-DID) system that develops a *Decentralized Identity* (DID)-based solution to addressing the verification issues in biological assets while considering multi-dimensional requirements. Blockchain technology is the fundamental infrastructure of the system. Authenticated certificates of the asset owners verify identity attributes. *Financial Service Institutions* (FSI) and governance agencies act as issuers, providing *Verifiable Credentials* (VC) for biological assets by using a group of attributes over a consensus. The asset owners hold VCs and can be available to other organizations as proof of the corresponding asset attributes. Our evaluations have demonstrated that the proposed approach has superior performance in biological assets verification, security, and maintenance.**

***Index Terms*—Decentralized identity, blockchain, biological identity governance, FinTech**

I. Introduction

The biological assets industry has been experiencing a vigorous advance in recent years, and an even big market growth is predicated by multiple countries. One of features of the biological assets industry is that a long business cycle and a heavy asset investment are needed, due to the characteristics of modern agriculture. For the purpose of the industry upgrade, financial services are considered one of significant impact factors for unlocking potential financial values as well as new products or services, such as offering biological asset mortgage loans [1]. *Financial Technology* (FinTech) is broadly believed to be a fundamental for facilitating embedding financial elements into the upgrade solutions, such as using blockchain-based trustworthy tracing systems for biological assets' quality governance. Biological assets' identity is its' foundation.

However, verifying biological assets' identities is a challenge in many cases of FinTech adoptions, when a product-level capability is required in agriculture. Massive quantity and distribution further enhance the complexity of the adoption. Lack of identity verification technology results in challenges of biological asset governance, such as loan frauds and disease controls [2], [3]. Even though prior studies have explored numerous work on smart agriculture, governing biological assets has been rarely addressed. Finding out adaptive and effective solutions to establishing a trustworthy biological governance system has an urgent demand.

In this work, we focus on the identity governance issues in biological assets. As a traditional method, centralized identity solution highly relies on the trustworthiness of the third party. The nature of this type of method needs guarantee absolute security of the trustful party, which unfortunately often encounters challenges in practice. Sensitive data, e.g., personal identity information or objects' attributes, are collected by the third party for the purpose of verification or authentications [4], such that this setting makes the service providers a visible target for both inside and outside attackers. Moreover, single point of failure issues also restrict the implementation of the centralized system [5], which weakens the robustness of the entire system. As a typical agriculture service requires continuous responses, these limitations are obstacles to achieving effective biological asset governance.

Our prior research has demonstrated that implementing *Decentralized Identity* (DID) technology is an effective alternative for addressing trustworthiness issues caused by the centralized identity setting [6], [7]. Blockchain technology is a technical option for constructing a DID system as fundamental infrastructure due to its characteristics of being tamper-resistant, transparent, etc. The setting of DID enables multiple participants to be involved in the verification of identities over a consensus, such that privacy can be protected by avoiding a single party collecting sensitive data from data owners. As only necessary data are collected by an individual participant, the risk of privacy leakage is reduced when considering the threat from single node attacks. The distributed ledger setting also solves the single-point of failure issues. Therefore, a blockchain-based DID is a technical option for constructing a trustworthy biological asset governance system.

In this paper, we have proposed a consortium blockchain-

Z. Fang, S. Fang, S. Wang, Z. Gao, and K. Gai are with the School of Cyberspace Science and Technology, Beijing Institute of Technology, Beijing, China, 100081. (emails: {3220221425, 3220221424, 3220215214, 3220221488, gaikeke}@bit.edu.cn); Z. Fang is also with Beijing Muguo Technology Co., Ltd., Beijing, China. K. Gai and Z. Gao is also with Yangtze Delta Region Academy of Beijing Institute of Technology, Jiaxing, Zhejiang, China.

J. Yu is with the Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China. (email: yujing02@iie.ac.cn).

W. Chan is with the School of Computer Science and Technology, Beijing Institute of Technology, Beijing, China, 100081. (email: {chanweilin}@bit.edu.cn);

This work is supported by the National Key Research and Development Program of China (Grant No. 2021YFB2701300), the Open Topics of Key Laboratory of Blockchain Technology and Data Security, The Ministry of Industry and Information Technology of the People's Republic of China (Grant No. 20242217), and the project offered by Beijing Muguo Technology Co., Ltd.

Correspondence to Dr. J. Yu (yujing02@iie.ac.cn),

Co-correspondence to Dr. K. Gai (gaikeke@bit.edu.cn).

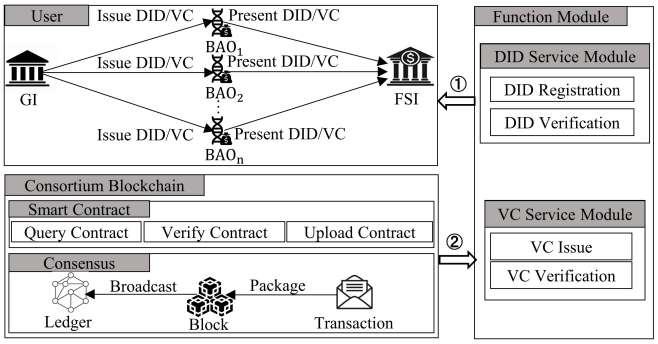

Fig. 1. High-level architecture of BA-DID, where 1 in fig denotes DID/VC Services and 2 denotes Blockchain Services.

based system that uses DID technology for governing biological assets, named *Biological Assets Decentralized Identity Management* (BA-DID). Our system focuses on governing the identities of biological assets, allowing a higher-level trustworthiness in protecting assets' identities and attributes during both authorization and verification processes.

The main contributions of this work are twofold.

- This work proposes a DID-based governance scheme for biological assets' identities for the first time. The proposed scheme uses blockchain technology as a fundamental infrastructure, and it solves trustworthiness issues caused by the single-node setting. Stakeholders take part in the process of asset identity verification by applying our proposed *Verifiable Credentials* (VCs), which facilitates technical supports for *Financial Service Institutes* (FSIs) to verify the underlying assets and form the critical fundamental of financial services.
- We have developed a business-level system that enables FSI to acquire verification of DID and VC. Our system has four key function modules: DID registration, DID verification, VC issue, and VC verification. Specifically, we address the characteristics of smart agriculture and design an applicable system for governing biological assets while considering various attributes of the asset. An empirical study has been given by presenting several crucial smart contracts in our work. ChainMaker is implemented in our system to build consortium blockchain

II. System Design

A. Design Objective

Reliability: The system should be highly reliable and provide continuing service during a single node failure. *Performance:* The system design should provide good response speed with limited latency and computational resource consumption. *Scalability:* The system should have scalability to support growing users and data volume.

B. System Overview

We propose a biological assets identity management model, BA-DID, which utilizes distributed digital identities and consortium blockchains. This model enables the registration and verification of DIDs under the supervision of governance institutions. All entities participating in the system can possess DIDs and present them to other participants as proof of their identity. The model also allows data owners to issue VCs, which can be presented to other organizations as evidence of relevant attributes. First, we define the participants of this model. Then, we outline the architecture of the BA-DID model. Finally, we describe the four main stages of achieving the desired functionality. The BA-DID system involves three entities. Each entity is briefly defined as follows.

BAO pertains to individuals or companies that possess diverse biological assets. To validate the identity and attribute information of biological assets, owners have the ability to provide the FSI with the DID and VC issued by the *Governance Institution* (GI). **FSI** is a professional organization that assumes the responsibility of offering a diverse range of financial services and products. FSI has the capability to request verification of the DID and VC from BA-DID, and upon successful verification, the identity and attribute information of the biological assets can be duly confirmed. **GI** assumes the crucial role of system regulation and is embodied by departments entrusted with regulatory responsibilities. In the context of the consortium blockchain, it possesses the authority to authorize the participation of other entities. GI serves as the issuer of DID and VC derived from the attribute data of the GI.

The BA-DID architecture is illustrated in Fig. 1, comprising three distinct layers: the user layer, the functional module layer, and the blockchain layer. Acting as the foundational infrastructure for BA-DID, the consortium blockchain serves as a repository for storing the identity information of biological assets. It facilitates identity management through the deployment of smart contracts. The upper-level functional modules encompass the DID service and VC service. Specifically, the DID service module primarily handles the registration and verification of DID, whereas the VC service module assumes responsibility for the issuance and verification of VC. Within the application layer, the registration and verification of DIDs are accomplished by invoking the DID service interfaces provided by the functional modules. Furthermore, the attributes of biological assets are validated by utilizing the VC service.

C. Main Phases

1) DID Registration: As shown in the Fig. 2, the BAO first generates its key pair, (PUB_{BAO}, PRI_{BAO}), where PRI_{BAO} is securely stored by the BAO locally, and PUB_{BAO} is published to the blockchain network through on-chain transactions for subsequent encryption operations. Next, the BAO sends a registration request with DID to the trustworthy GI. Upon receiving the request, GI queries the PUB_{BAO} published on the blockchain and encrypts a challenge message for identity verification using PUB_{BAO}. Upon receiving the challenge message, the BAO decrypts it using its secret PRI_{BAO}. When the decrypted information matches the message sent by GI, GI verifies the BAO's identity. Subsequently, the BAO provides detailed identity proof,

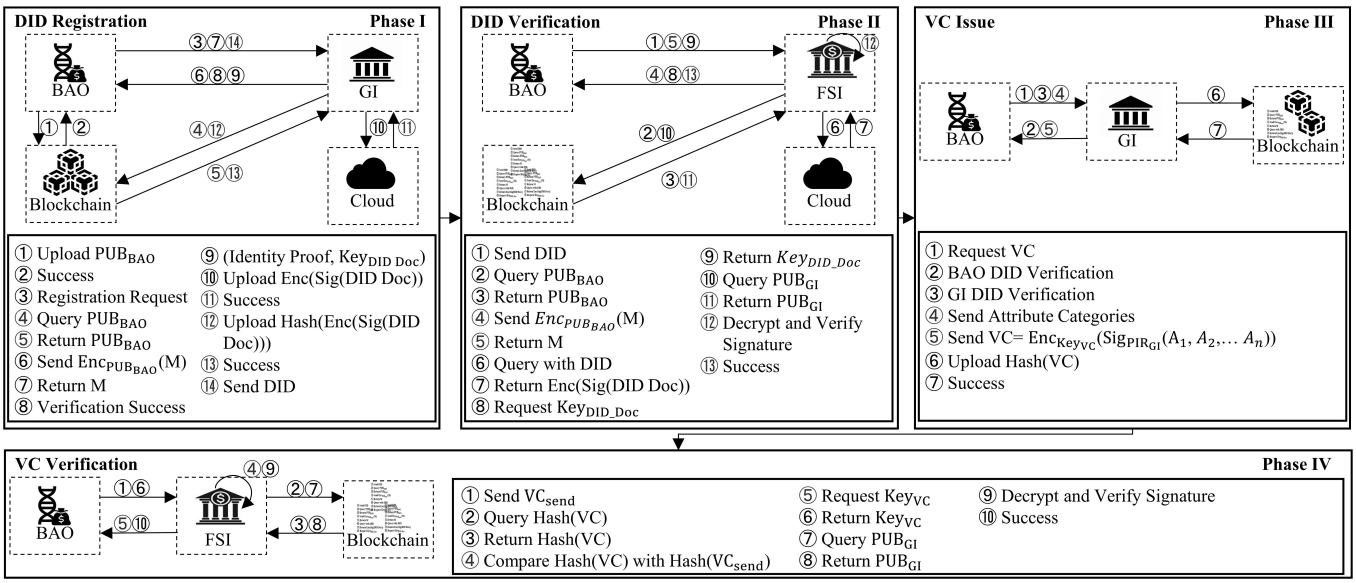

Fig. 2. Main phases of the Proposed BA-DID system.

such as gender and address, to GI to verify its real identity. GI uploads the BAOs' PUB_{BAO} to an internal cloud server for subsequent encrypted communication and verification processes. Based on the identity proof provided by the BAO, GI generates a DID Doc and signs the document using its private key PRI_{GI} to ensure the integrity and trustworthiness of the DID Doc. Soon, GI used the symmetric encryption key Key_{DID_Doc} to negotiate with the BAO to encrypt the signed DID Doc to ensure its confidentiality. Finally, GI uploads the encrypted DID Doc to the cloud server and hashes it on the blockchain to ensure the integrity and immutability of the DID Doc, thus completing the entire Register DID phase.

2) DID Verification: When the BAO presents its DID, the FSI needs to verify the BAO's DID. The FSI queries the PUB_{BAO} from the blockchain and encrypts a challenge message using this key. When the BAO successfully decrypts and returns the message, it is considered to be the BAO themselves. Then, the FSI requests the encrypted and signed DID Doc from the cloud storage using the DID. It submits a verification transaction on the blockchain, inputting the signed and encrypted DID Doc to verify when it has been tampered with. The FSI requests the symmetric encryption key Key_{DID_Doc} used to encrypt the DID Doc from the BAO and queries the public key PUB_{GI} of the issuer GI on the blockchain. Finally, the FSI decrypts and verifies the GIs' signature to complete the DID verification process and obtain the attribute data within the DID Doc.

3) VC Issue: The BAO sends a request to the issuer GI to issue a VC. Upon receiving the request, the issuer GI and the BAO mutually verify their identities using their respective DIDs to ensure the authenticity and trustworthiness of both parties. This verification phase is consistent with the described DID Verification phase. Once the verification is successful, the BAO provides the issuer GI with the desired attribute categories for the VC, including identity information, attributes, or other claims. GI, based on the provided attribute category data from the BAO, signs the data using its private key PRI_{GI}^{DID} and encrypts it using the pre-negotiated symmetric encryption key Key_{VC} to construct a complete VC. Finally, GI sends the created VC to the BAO, and the hash value of the VC is uploaded to the blockchain, thereby completing the VC issuance process. The BAO can store the VC in a digital wallet or other digital identity management tools and provide it to verifiers for verification and use when needed.

4) VC Verification: The BAO presents a VC to the verifier FSI. The FSI calculates the current hash value of the VC and compares it with the corresponding hash value stored on the blockchain. When the comparison results match, the FSI requests the symmetric encryption key Key_{VC} negotiated with the VC issuer GI from the BAO to decrypt the VC. The FSI also requests the public key PUB_{GI} of GI from the blockchain to verify the signature, ensuring that the VC has not been tampered with and has been issued by a legitimate issuer. Based on the verified VC information, the verifier extracts the BAO's identity information, attributes, or other claims and can make corresponding decisions, such as granting access permissions or providing services.

D. Smart Contracts

The BA-DID primarily focuses on three contracts: Verify, Query and Upload.

Verify Contract: The *Verify* algorithm in the smart contract first retrieves the user-uploaded structured data, `data`, and the unique identifiers of it, `code`. The `data` can be VC or DID Doc. It performs separate checks for the validity of `data` and `code` and verifies when the `data` hash corresponding to `code` is already stored on the blockchain. When the checks

Algorithm 1 Verify

Require: `code`, `data`
Ensure: `status`
  if length of code = 0 or length of data = 0 **then**
    return `status` ← $fail$
  end if
  result ← ctx.get_state(verify, `code`)
  if result is not error **then**
    result_str ← convert result to UTF-8 string
    Initialize a new SHA256 hasher
    hasher.update(`data`)
    hashed_data ← hasher.finalize()
    hashed_data_hex ← format *hashed_data* as hex
    if *result_str* = *hashed_data_hex* **then**
      `status` ← $success$
    end if
  end if
  return `status`

Algorithm 2 Query

Require: `code`, `type`
Ensure: `value`
  if length of `code` = 0 **then**
    return `value` ← null
  end if
  query_type ← ctx.arg_as_utf8_str(`type`)
  if length of *query_type* = 0 **then**
    return `value` ← null
  end if
  pk_vec ← ctx.get_state(*query_type*, `code`)
  if *pk_vec* is valid **then**
    `value` ← convert *pk_vec* to UTF-8 string
  end if
  return `value`

Algorithm 3 Upload

Require: `code`, `data`
Ensure: `status`
  `code` ← ctx.arg_as_utf8_str(`code`)
  if length of `code` = 0 **then**
    return `status` ← fail
  end if
  info_data ← ctx.arg_as_utf8_str(`data`)
  if length of *info_data* = 0 **then**
    return `status` ← fail
  end if
  Initialize a new SHA256 hasher
  hasher.update(*info_data*)
  hashed_info ← hasher.finalize()
  ctx.put_state(*upload_hash*, `code`, *hashed_info*)
  return `status` ← success

as mentioned earlier fail, an error is returned. It then queries the stored `data` hash from the blockchain and unpacks the result into *result_str*. The `data` is hashed using the SHA-256 algorithm. The hash value of `data`, *hashed_data*, is then compared with the hash value obtained from the blockchain query, *result_str*. A verification `status` failure is returned when they are not equal; otherwise, a verification `status` success is returned.

Query Contract: The *Query* algorithm in the smart contract can query PUB or DID Doc after signature encrypted and hashed on blockchain. It first retrieves the user-uploaded `code`. For PUB query, DID, as the `code`, is generated locally by the user before registration, and the user also generates a public-private key pair for signing the DID Doc and VC. When the user uploads the PUB, the DID is a unique identifier. Therefore, it is possible to query the user's public key on the blockchain using the DID. For DID Doc query, the `code` used for querying is also DID. During execution, the validity of the *code* needs to be checked. When a valid `code` is obtained, the function retrieves the stored public key from the blockchain using it as a field and returns the deserialized result *data_str*.

Upload Contract: The *Upload* algorithm in the smart contract is used to upload the PUB or the hash values of the DID Doc and VC to the blockchain for storage. This is done for subsequent integrity verification of the PUB, DID Doc and VC. First, the unique identifier of the user-uploaded PUB, DID Doc or VC code is retrieved. The user-uploaded PUB, DID Doc or VC, *info_data* is obtained. The validity of code and *info_data* is verified. When successful, *info_data* is hashed using the SHA-256 algorithm to obtain *hashed_info*. It is then serialized and recorded in the ledger.

III. Experiment Evaluations

A. Experiment Configuration

We leveraged the ChainMaker framework and deployed the consortium blockchain on three *virtual machines* (VMs) running on the same physical machine. The physical machine was equipped with an i7-11800H CPU, 32GB of RAM, a 1TB hard disk, and a Windows 11 operating system. The three VMs were run using VMware. Each VM had the same configuration with 8GB of RAM and 50GB of disk space, running the Ubuntu operating system. We utilized the ChainMaker BaaS open-source platform to create the blockchain and deploy smart contracts. The consensus algorithm used on the consortium blockchain is RAFT. The RAFT algorithm is currently the most widely used non-Byzantine fault-tolerant consensus algorithm. It relies on a voting mechanism and log replication to achieve node consensus. Nodes vote to participate in the leader selection, who handles all requests and then replicates them to other nodes in the form of logs.

During the evaluation process, we set up control groups with task contract invocation counts ranging from 100 to 700, with intervals of 100. We analyze the performance of the BA-DID system using two parameters: gas consumption and contract execution time. The attribute data within the VCs to be uploaded to the chain can be customized based on user requirements. Moreover, the size of data uploaded, queried, and verified by the contract may also impact the performance

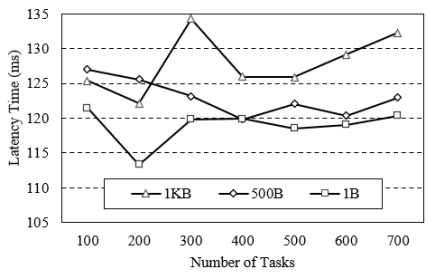

Fig. 3. Latency Time under Setting 1.

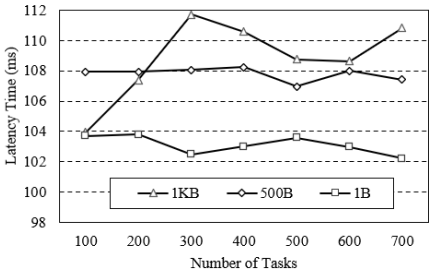

Fig. 4. Latency Time under Setting 2.

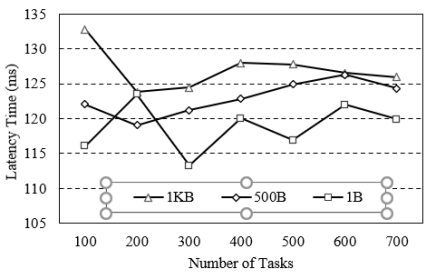

Fig. 5. Latency Time under Setting 3.

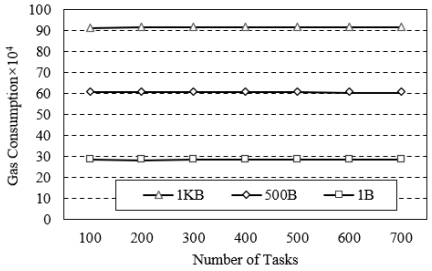

Fig. 6. Gas Consumption under Setting 1.

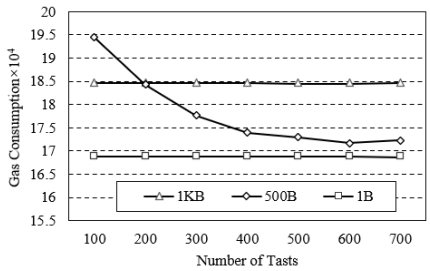

Fig. 7. Gas Consumption under Setting 2.

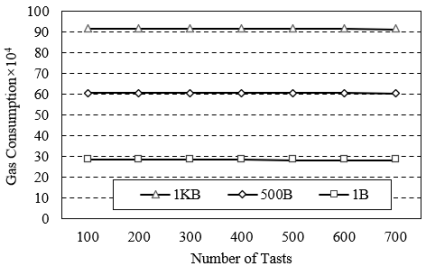

Fig. 8. Gas Consumption under Setting 3.

TABLE I
EXPERIMENT SETTINGS

Settings	# of Invoke	Data Size	Smart Contract
1	100-700	1KB, 500B,1B	Upload
2	100-700	1KB, 500B,1B	Query
3	100-700	1KB, 500B,1B	Verify

of the blockchain. Therefore, we set up three control groups with data sizes of 1 byte, 500 bytes, and 1 kilobyte to observe the performance of the blockchain. Experimental settings are shown in the table below.

B. Experiment Results

Fig.s 3 and 6 illustrate the latency and gas consumption for Setting 1, while Fig.s 4 and 7 show for Setting 2, and Fig.s 5 and 8 are for Setting 3. We shall explain each experimental result graph in sequence in the following.

In Fig. 3, the *Upload* contract execution latency was approximately 130ms for 1KB of data, 125ms for 500B of data, and 120ms for 1B of data, demonstrating stability across different task volumes. In Fig. 4, the *Query* contract execution latency was approximately 105ms for 1KB of data, 107ms for 500B of data, and 103ms for 1B of data, showing stability across different task volumes. In Fig. 5, the *Verify* contract was executed with a data size of 1KB, 500B, and 1B, and the latency remained stable around 130ms, 126ms, and 120ms for different task volumes, respectively.

In Fig. 6, the *Upload* contract gas consumption remained stable at approximately 915,000 for 1KB of data, 608,000 for 500B of data, and 286,000 for 1B of data across different task volumes. In Fig. 7, the *Query* contract gas consumption remained stable at approximately 185,000 for 1KB of data, 172,000 for 500B of data, and 169,000 for 1B of data across different task volumes. In Fig. 8, the *Verify* contract gas consumption remained stable at approximately 915,000 for 1KB of data, 606,000 for 500B of data, and 285,000 for 1B of data across different task volumes.

Based on the experimental results, it is evident that the data size has a clear impact on latency and gas consumption. As the data size increases, both latency and gas consumption increase. This is due to the increased consumption of computational and network resources caused by the contracts' larger amount of data being processed. In addition, different contracts also exhibit variations in latency and gas consumption. The *Upload* and *Verify* contracts show significantly higher latency and gas consumption compared to the *Query* contract. It is known that latency and gas consumption during the execution of smart contracts are related to contract complexity, data size, operation types, storage, and retrieval costs. Data querying is considered a more straightforward contract, resulting in lower latency and gas consumption than the other two contracts. Little difference is observed when comparing the latency and gas consumption of the same contract under different task volumes. This indicates that our system is scalable. The abnormally high gas consumption for tasks 100 and 200 in Fig. 7 may be attributed to frequent modifications of the on-chain state during the execution process.

IV. RELATED WORK

Biological Assets Governance (BAG) has attracted considerable attention from scholars in recent years. In the management of plant genetic resources, The work [8] established an intelligent and comprehensive plant genetic resources database management system according to international standards. Some scholars have also focused on supervising and predicting the production process of biological assets using machine learning, especially image recognition technology,

to improve production efficiency. For example, The work [9] introduced the applications of image recognition, *Internet of Things* (IoT), and database in biological assets management and production. These enhanced agricultural product output and quality. Li *et al.* [10] proposed a framework for automatic localization, analysis, and visualization of poultry farms using remote sensing imagery.

These works had demonstrated the importance and significance of exploring BAG. Our work was different to the above work focusing on improving efficiency in production and management. We focused on addressing identity management issues, thereby achieving efficient and trustworthy authentication technology. It provided trusted identity and attribute services completely controlled by user to other applications within the BAG system.

Identity Management (IdM) technology had developed from isolated identity, centralized identity, federated identity [11] to user-centric identity. Recently, self-sovereign identity had emerged as the possible direction for IdM [12]. Self-sovereign identity was a truly decentralized, entirely owned and controlled identity by individuals. Blockchain, with its decentralized and tamper-resistant nature, is considered a promising infrastructure to ensure secure storage and access control [13], [14], [15]. With the emergence of blockchain, the realization of self-sovereign identity had found a breakthrough [16].

DID was a form of self-sovereign identity characterized by guaranteed data authenticity, protection of user privacy and security and strong portability. There are currently many DID solutions implemented using blockchain technology. Cui *et al.* [17] proposed a blockchain-based multi-WSN authentication scheme for IoT. They established a blockchain network among different types of nodes, creating a hybrid blockchain model comprising both local and public chains. In contrast, our approach did not employ a hybrid blockchain model. We achieved efficient identity management through the design of identity verification architecture and smart contracts, and we not only focused on identity verification. The study [6] proposed a blockchain-based DID model to address the identity authentication problem in financial loans involving collateral. Other studies [18], [19] utilized smart contracts to transform information that needs to be stored on the blockchain into verified transactions. These works also used the DID model to build IdM solutions and had achieved excellent performance in different application fields. Our work applied the DID model to a new application field of biological assets IdM. We found that DID model could also perform well in it.

V. Conclusion

In recent years, the management of biological assets identities has garnered significant attention. We proposed a universal solution for biological asset identity management named BA-DID. BA-DID was able to issue and verify DIDs and VCs through smart contracts executed on a consortium blockchain. The system did not rely on centralized identity service providers, allowing users to have full control over their own data and reducing the risk of large-scale data breaches. The decentralized architecture also helped mitigate the risk of system collapse due to centralized node failures. The immutability of the blockchain ensured the credibility and traceability of DIDs and VCs. We implemented the system on the ChainMaker consortium blockchain framework and evaluated its performance through latency and gas consumption under various loads and data sizes. The system demonstrated good performance and scalability.

References

[1] E. Kadanal and K. Emine. Agricultural loan and agricultural production value in turkey. *ALINTERI J. AGRIC. SCI.*, 35(1):93–98, 2020.

[2] F. H. Bas et al. Insurance fraud: The case in turkey. In *Contemporary Issues in Audit Management and Forensic Accounting*, volume 102, pages 77–97. Emerald Publishing Limited, 2020.

[3] S. Zhu, Y. Chen, and W. Wang. Risk assessment of biological asset mortgage loans of china's new agricultural business entities. *Complexity*, 2020:1–12, 2020.

[4] D. Pramod. Privacy-preserving techniques in recommender systems: state-of-the-art review and future research agenda. *Data Technologies and Applications*, 57(1):32–55, 2023.

[5] S. Wang, H. Sheng, Y. Zhang, D. Yang, J. Shen, and R. Chen. Blockchain-empowered distributed multicamera multitarget tracking in edge computing. *IEEE Trans. Ind. Informatics*, 20(1):369–379, 2024.

[6] T. Xie, T. Zhang, K. Gai, and K. Xu. Cross-chain-based decentralized identity for mortgage loans. In *KSEM*, pages 619–633, 2021.

[7] T. Xie, K. Gai, L. Zhu, Y. Guo, and K.K.R. Choo. Cross-chain-based trustworthy node identity governance in Internet of Things. *IOTJ*, 10(24):21580 – 21594, 2023.

[8] E. Doyche, P. Malinov, N. Velcheva, and Z. Duchev . A genebank architecture : A distributed system for management of plant genetic resources. In *2020 IEEE 10th International Conference on Intelligent Systems*, pages 580–583, 2020.

[9] R. R. G. Rubia, J. Angel Ida Chellam, T. N. Prabhu, C. Kathirvel, M. Sivaramkrishnan, and et al. Machine learning approaches for smart agriculture. In *2022 6th International Conference on Computing Methodologies and Communication*, pages 1054–1058, 2022.

[10] Y. Li, R. Das, C. Duong, T. Lim, T. Haithcoat, and et al. Automated detection of poultry farms from aerial images for actionable ai system toward biosecurity applications. In *2023 IEEE Applied Imagery Pattern Recognition Workshop*, pages 1–4, 2023.

[11] Y. Cao and L. Yang. A survey of identity management technology. In *2010 IEEE International Conference on Information Theory and Information Security*, pages 287–293, 2010.

[12] A. AlBadi, F. Hajamohideen, and D. AlSaqri. A review on blockchain techniques used for identity management system: Privacy and access control. In *International Conference On Systems Engineering*, pages 361–375, 2023.

[13] K. Gai, Y. Wu, L. Zhu, Z. Zhang, and M. Qiu. Differential privacy-based blockchain for industrial Internet of Things. *TII*, 16(6):4156–4165, 2019.

[14] K. Gaiand J. Guo, L. Zhu, and S. Yu. Blockchain meets cloud computing: A survey. *IEEE Communications Surveys & Tutorials*, 22(3):2009–2030, 2020.

[15] Y. Miao, K. Gai, L. Zhu, K.K.R. Choo, and J. Vaidya. Blockchain-based shared data integrity auditing and deduplication. *IEEE Transactions on Dependable and Secure Computing*, PP(99):1, 2023.

[16] M. Shuaib, N. H. Hassan, S. Usman, S. Alam, S. Bhatia, A. Mashat, A. Kumar, and M. Kumar. Self-sovereign identity solution for blockchain-based land registry system: a comparison. *Mobile Information Systems*, 2022(99):1–17, 2022.

[17] Z. Cui, F. XUE, S. Zhang, X. Cai, Y. Cao, W. Zhang, and J. Chen. A hybrid blockchain-based identity authentication scheme for multi-wsn. *IEEE Transactions on Services Computing*, 13(2):241–251, 2020.

[18] K. Gai, Y. Zhang, M. Qiu, and B. Thuraisingham. Blockchain-enabled service optimizations in supply chain digital twin. *IEEE Trans. Serv. Comput.*, 16(3):1673–1685, 2022.

[19] K. Gai, Y. Wu, L. Zhu, K.K.R. Choo, and B. Xiao. Blockchain-enabled trustworthy group communications in uav networks. *IEEE Trans. Intell. Transport.*, 22(7):4118–4130, 2020.