2017 23rd Annual Conference on Advanced Computing and Communications

A Limited-use Asset Management System on the Blockchain Platform with an Extended Open Assets Protocol

Takuma TAKEUCHI, Toshiya SHIMIZU, Ken KAMAKURA,
Takeshi SHIMOYAMA and Hiroshi TSUDA

FUJITSU LABORATORIES LTD.
4-1-1, Kamikodanaka, Nakahara-ku, Kawasaki, Kanagawa, 211-8588, Japan
Email: {takeuchi.takuma, shimizu.toshiya, kamakura.ken, shimo-shimo, htsuda}@jp.fujitsu.com

***Abstract*—Currency payment platforms on blockchains have widely spread through the e-commerce market all over the world. Colored Coins are ones of the platforms that can add information which expresses issuers and purposes of the coins. The purposes written in the information can express the usage such as coupons, but cannot prevent the coins from being paid for other purposes.**

First, in this paper, we propose an extended open assets protocol for limited-use assets. This protocol prevents the assets from being used for other purposes than the issuer's intended purposes. Second, we construct a limited-use asset management system on the blockchain platform with our proposed open assets protocol. Finally, we indicate the effects and use cases of our proposed system.

***Index Terms*—blockchain technology, digital currency, payment system, limited-use asset, open assets protocol.**

I. Introduction

Blockchains, as digital currency payment platforms on peer-to-peer network architectures, have widely spread through the market in the world. Bitcoin platform [12] is the most popular digital currency payment platform now. The total value of all Bitcoin as a digital currency reached more than 70 billion dollars in August, 2017. Digital currency payment platforms on blockchains feature the following: the platforms are impossible to be tampered; they are zero downtime platforms; moreover, they are decentralized.

Digital currency payment platforms on blockchains are used not only as platforms for currencies which can be used anywhere, but also those for limited purposes such as book coupons, regional promotion tickets, and so forth. An example of transaction platforms considering limited purposes is Colored Coin platform [14], which is regarded as one of Blockchain 2.0 platforms [15].

Colored Coins (Open Assets Protocol), for example, are extension of the Bitcoin protocol that can overlay some information on small amounts of Bitcoins. Then, Colored Coins can hold informations which express their issuers and purposes. The purposes described in the informations can express the usage of the coins, such as book coupons, regional promotion tickets, and so on.

An existing problem of Colored Coins is that the purpose described in the coins cannot prevent the coins to pay for other purposes. Certainly, when someone pays Colored Coins which have some limited purposes to purchase for other purposes, the transaction logs are visible to everyone. So, the facts that Colored Coins are used for other purposes can be found by everyone. However, the ordinary Colored Coins platforms cannot reject the settlement of the payment, which violates the intended limited purposes described on the coins.

In this paper, for solving the above issue, we first propose an extended open assets protocol for limited-use assets. This protocol verifies limited purposes described in limited-use assets transactions. If the assets are used for other purposes than the intended purposes, this protocol prevents the settlements of the assets. Second, we construct a limited-use asset management system on the blockchain platform with this protocol. Finally, we indicate the effects and use cases of our proposed system.

Section 2 contains some preliminaries of blockchain platforms and related works. In Section 3 we present an extended open assets protocol for preventing the utilization of limited-use assets for other purposes than intended ones. Section 4 describes a prototype limited-use asset management system on blockchain platforms with the above protocols. In Section 5 we explain about the benefits of our proposed system. Section 6 contains use cases of our proposed system. Section 7 is devoted to conclusions.

II. Preliminaries

A. Blockchain platforms

Blockchain platforms are composed of networks of nodes, which share a common data of transactions with consensus about the state of structure. These systems have many properties [1], [11], [12]. In this paper, we focus on the following two of them.

1) **Additional field on transactions**:
   When we issue a transaction on blockchains, we can fill in the following information on the transaction: the input address, the output address, the amount, and the

978-1-5386-6911-2/18/$31.00 ©2018 IEEE
DOI 10.1109/ADCOM.2017.00008

timestamp. Some kinds of blockchain platforms prepare additional field on transactions. The field is used for recording additional information other than the input addresses, output addresses, amounts, and timestamps. For example, Bitcoin transaction data have OP_RETURN as the additional field which can hold 80 bytes data.
Colored Coins [14] are ones of cryptocurrency technologies which are modified from Bitcoin protocol with the additional field. A "colored" coin is an amount of bitcoin repurposed to express another asset. There are several implementations of Colored Coins, and open assets protocol [6] is a standard implementation.

2) **Verification of transactions**:
When we issue transactions on blockchains, third parties verify whether the transactions are valid or not. The verification of transactions is called "mining" and such people are called "miners." If transactions are judged to be invalid by the mining process, their settlements are not executed on the blockchains. Examples of invalid transactions are the following:
   - The input account has no more than 2 BTC and the amount is 3 BTC,
   - Transactions which are issued without permission of their input addresses.

In Section 3, the additional field on transactions is used and the definition of "invalid" transactions is extended in our proposed protocol.

B. Related works

There have been various attempts to construct digital currencies payment system for some purposes. Micropayment Channels [8] and Lightning Networks [13] are off-chain payment systems between two or several people, which do not need the verification of transactions. These payment systems are used for the purpose of microcommerce because off-chain transactions do not require the transaction fee.

In recent years, private blockchain platforms and permissioned blockchain platforms [16] have been developed rapidly, such as Ethereum [5], [9], [17] and Hyperledger [10]. On permissioned blockchains, limited-use asset management systems can be configured for particular parties such as one or several corporations.

III. Our proposed open assets protocol

Using the two properties on Section 2.1, we propose an extended open assets protocol for limited-use asset management systems. The protocol consists of the following three features.

1) **the feature to publish usage information (PUI)**:
The feature publishes usage information. Specifically, we first assume that an account on blockchains is prepared as the account to publish the information of usage. Let us call it "usage publisher account" and denote it by P. Moreover, let us denote the output addresses by $R_1, R_2, \dots$, and denote the usage information by $u_1, u_2, \dots$. Then, the feature means that P broadcasts a transaction whose output address is R_i and that the additional field of the transaction holds the information u_i $(i = 1, 2, \dots)$.
In this paper, we call this feature **"Publish Usage Information"** (**PUI** as an abbreviation).
2) **the feature to publish limited-use assets (PLUA)**:
The feature publishes some limited-use assets. In particular, first let us denote the account which publishes a limited-use asset by A, the account receives them by B, and the intended usage of the limited-use asset by v. Then, the feature indicates that A broadcasts a transaction whose output address is B and that the additional field of the transaction holds the information v and the usage publisher account P's address.
In this paper, we call this feature **"Publish Limited Use Assets"** (**PLUA**).
3) **the feature to verify transactions with limited-use assets (VTLUA)**:
The feature verifies transactions on which accounts pay limited-use assets to other accounts. To be more specific, first let us assume that the account B pays a limited-use asset to the account R_i $(i \in \{1, 2, \dots\})$ and that the usage information of the asset is v. Then, the feature represents that the transaction verification by miners includes the following process in addtion to the normal transaction verification: the transaction is invalid if $u_i \neq v$. After the verification of VTLUA, the limited-use assets become ordinary assets, that is, the account R_i can use the assets for any purposes.
In this paper, we call this feature **"Verify Transactions with Limited Use Assets"** (**VTLUA**).

The following Fig. 1 describes the above three features.

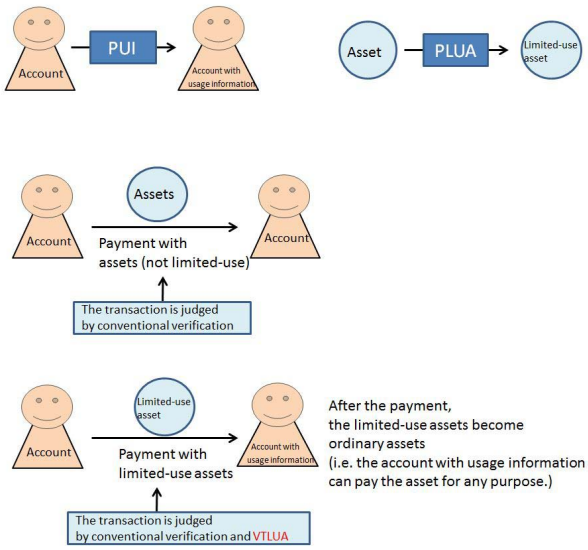


Fig. 1. an image of the three features

IV. Our proposed limited-use asset management system

In the previous section, we discuss the extended open assets protocol for limited-use assets. Then, in this section we indicate a prototype limited-use asset management system, which is modified from the Bitcoin platform.

Let us use OP_RETURN of Bitcoin transactions as the additional field of transactions. Moreover, let us use the feature of Bitcoin transactions: transactions are chained together [2]. Like the following Fig. 2, each Bitcoin transaction spends the money previously received in one or more earlier transactions, so the input of one transaction is the output of previous transactions. The unspent transaction output is called UTXO.

The three features in Section 3 are implemented in the following way.

1) **PUI**:
PUI is constructed by the following Transaction 1 ($i = 1, 2, \dots$) as Fig. 3:

Transaction 1: ($i = 1, 2, \dots$)

input address	the usage publisher account P (an address)
output address	account R_i (an address)
amount	As small as the transaction fee
data on OP_RETURN	usage information u_i (a string data)

2) **PLUA**:
PLUA is built by the following Transaction 2, which includes the following information as Fig. 4:

Transaction 2:

input address	A (an address)
output address	B (an address)
amount	The same as the asset's value
data on OP_RETURN	v (a string data)

3) **Two types of transactions**:
After PUI and PLUA, let us assume that account B broadcasts transactions which use UTXO of the above Transaction 2. The transactions can be classified into the following two:

(3a) The input address is B, the output address is another account.

(3b) The input address is B, the output address is also B.

For example, let us assume that account B which has 100 BTC pays 20 BTC to account R_1. When the transaction fee is 0.00001 BTC, B broadcasts a transaction whose amount is 20 BTC from B to R_1 and a transaction whose amount is 79.99999 BTC (equivalent to the charge of the payment) from B to itself. The former transaction is classified to (3a) and the latter is to (3b).

4) **VTLUA**:
Now by using VTLUA, let us verify the transaction that is broadcasted by B as Fig. 5.

- If the input address is B and the output address is another account, which is denoted by R (that is, the above case of (3a)), VTLUA is implemented as the following process:

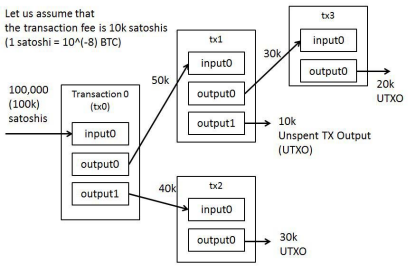


Fig. 2. an image of Bitcoin's Transactions (modified from [2])

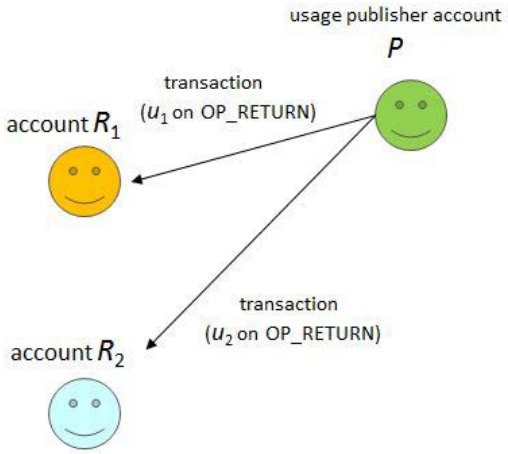


Fig. 3. an image of PUI in the limited-use asset management system

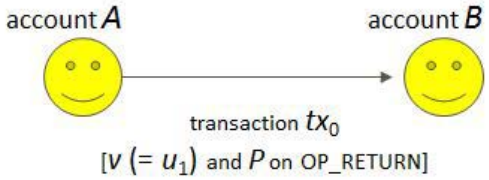


Fig. 4. an image of PLUA in the limited-use asset management system

(1) Get the string information on OP_RETURN of the UTXO of the transaction. This string information includes v and P.
(2) Get the string information on OP_RETURN of the transaction which is from P to R. This string information is denoted by u.
(3) Execute the normal verification of Bitcoin transactions. If the normal verification judges the transaction to be valid, go to the next step. Otherwise break out the transaction judgement.
(4) Verify whether the two string informations v and u_1 are equal or not. If they are not equal, make the transaction invalid. If they are equal, make the transaction valid.

- If the input address is B and the output address is also B (that is, the above case of (3b)), VTLUA is implemented as the following process:
  (1) Get the string information on OP_RETURN of the UTXO of the transaction. This string information is v.
  (2) Get the string information on OP_RETURN of the transaction which is from B to B. This string information is denoted by v'.
  (3) Execute the normal verification of Bitcoin transactions. If the normal verification judges the transaction to be valid, go to the next step. Otherwise break out the transaction judgement.
  (4) Verify whether the two string informations v and v' are equal or not. If they are not equal, make the transaction invalid. If they are equal, make the transaction valid.

The transaction verication of our proposal system is described in the flowchart Fig. 7. For comparison, the transaction verification of conventional system is described in the flowchart Fig. 6.

For example, let us construct a limited-use asset management system for book coupons, substitute the string "book shop" for u_1, substitute the string "game shop" for u_2, and substitute the string "book shop" for v. If the account B use a book coupon asset whose usage information is v (= "book shop") for buying a game from the account R_2, VTLUA judges the transaction to be invalid and the payment is not executed since $u_2 \neq v$.

Our proposed system can be constructed on platforms modified from other blockchain platforms, for example, Ethereum [5], [9], [17], Hyperledger [10], and so on.

V. Benefits of our proposed system

As described in the previous sections, our proposed system prevents the utilization of limited-use assets for other purposes

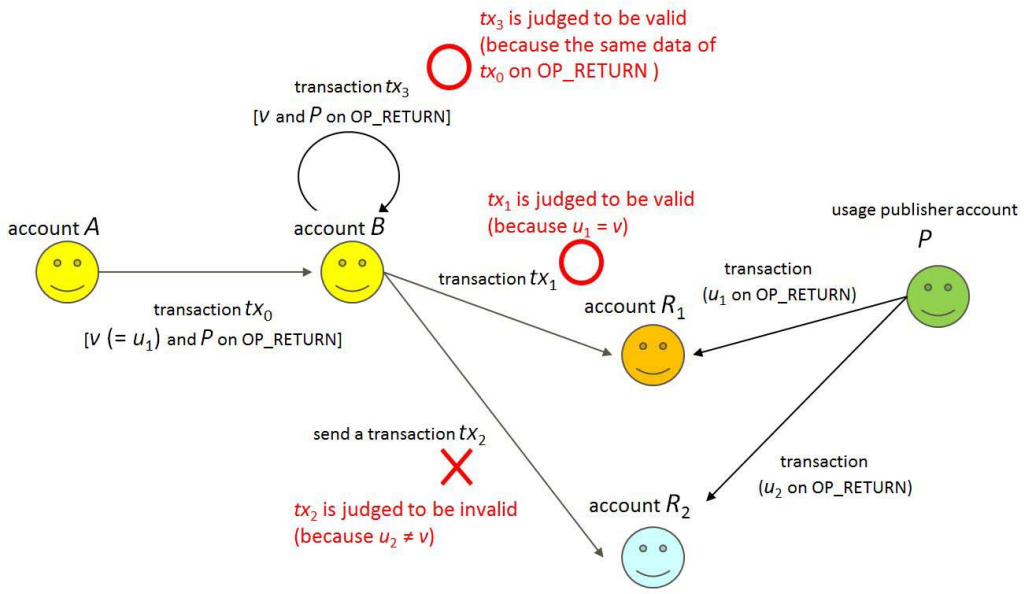


Fig. 5. an image of VTLUA in the limited-use asset management system

than the intended ones. If someone broadcasts transactions which means that limited-use assets for specific purposes are used for different purposes from the specific ones, the miners' verification determines that the transactions are invalid and they are not settled on blockchains.

A benefit of our proposed system is feasible due to only the mechanism of our proposed blockchain systems and it doesn't require external computing resources and databases. Moreover, this effect doesn't need the additional work of anyone who is concerned in the payment of limited-use assets because miners, who are third-party people, produce the benefit.

Another benefit is that our proposed system is an enhanced version of public digital currencies payment system, such as Bitcoin, because limited-use assets in our system can be used for arbitrary purpose after the assets are used for the intended purpose.

VI. Use cases

In the previous sections, we explained about the detail of our proposed system and the effects of it. Then, in this section we indicate some use cases of our system as the following:

- Digital Rights Management. Our system can be used as access control technology to restrict usage of various coupons for shops, such as book coupons, game coupons, music coupons, and so on. When our system are used for managing music coupons, the system can be applied into the coupons of blockchain music services such as Ujomusic [7].
- Regional promotion tickets. Our system can be a regional currency system to promote regional development. For example, if the limited-use assets which is used only for shops in a particular region, the assets are promotion coupons of the region.
- Voting rights system. Our system is also applicable to limited-use assets which are not money, such as voting rights.

VII. Conclusion

In this paper, we have proposed an extended open assets protocol for limited-use assets. We have also indicated an example of this protocol on the modified Bitcoin platform.

As future work, we intend to construct a limited-use asset system on off-chain transactions such as Micropayment Channels [8] and Lightning Networks [13]. Off-chain transactions are settled without blockchain miners' verification. So our proposed system cannot be applied in a simple way because our system needs miners' verification as VTLUA.

References

[1] Antonopoulos, A. M. Mastering Bitcoin: unlocking digital cryptocurrencies. O'Reilly Media, Inc., 2014.
[2] Bitcoin Developer Guide [Online; accessed May 30, 2017]: https://bitcoin.org/en/developer-guide#block-chain-overview
[3] Bitcoin.org - Unspent Transaction Output, UTXO [Online; accessed May 30, 2017]: https://bitcoin.org/en/glossary/unspent-transaction-output
[4] Bitcoin Wiki - Proof of Work [Online; accessed May 30, 2017]: https://en.bitcoin.it/wiki/Proof_of_work
[5] Buterin, V. A next-generation smart contract and decentralized application platform. white paper, 2014.
[6] Charlon, F. Open Assets Protocol [Online; accessed May 30, 2017]: https://github.com/OpenAssets/open-assets-protocol
[7] Consensys - Ujomusic [Online; accessed May 30, 2017]: http://ujomusic.com/

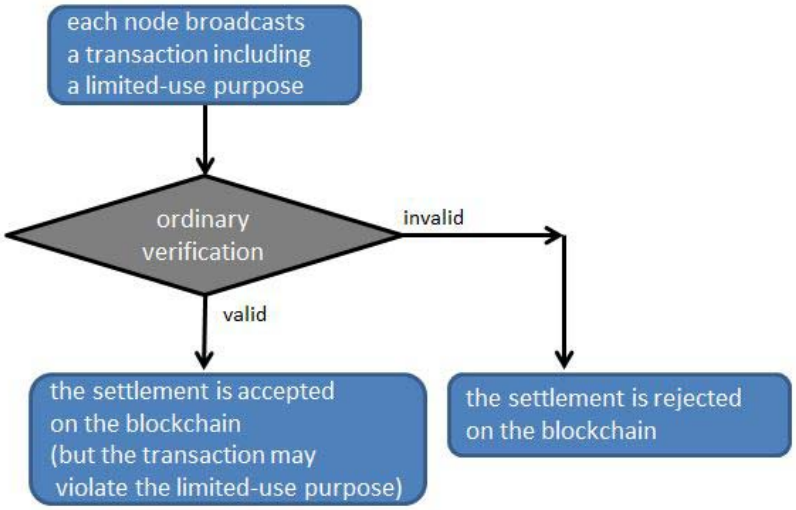


Fig. 6. conventional verification flowchart of transactions by blockchain miners

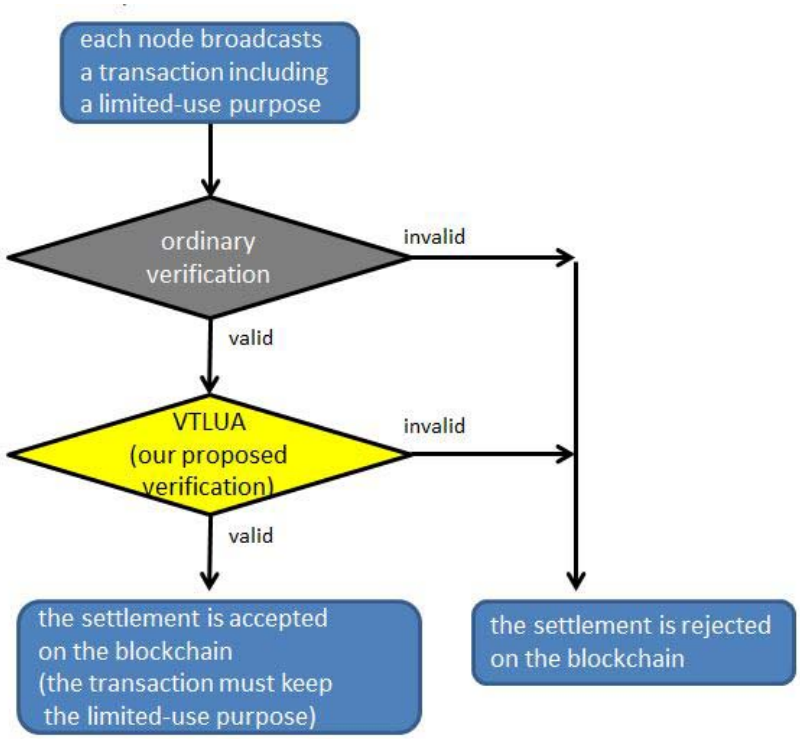


Fig. 7. our proposed verification flowchart of transactions by blockchain miners

[8] Decker, C. & Wattenhofer, R. A fast and scalable payment network with bitcoin duplex micropayment channels. In Symposium on Self-Stabilizing Systems (pp. 3-18). Springer International Publishing, 2015.

[9] Ethereum Project [Online; accessed May 30, 2017] : https://www.ethereum.org/

[10] Hyperledger Project [Online; accessed May 30, 2017]: https://www.hyperledger.org/

[11] Jain, R. Blockchains: The Revolutionary Trust Protocol, BEL Keynote at 22nd Annual International Conference on Advance Computing and Communications (ADCOM 2016), 2016.

[12] Nakamoto, S. Bitcoin: A Peer-to-Peer Electronic Cash System [Online; accessed May 30, 2017], https://bitcoin.org/bitcoin.pdf, 2008.
[13] Poon, J., & Dryja, T. The bitcoin lightning network, 2015.
[14] Rosenfeld, M. Overview of colored coins. White paper, bitcoil.co.il, 2012.
[15] Swan, M. Blockchain: Blueprint for a new economy. O'Reilly Media, Inc., 2015.
[16] Swanson, T. Consensus-as-a-service: a brief report on the emergence of permissioned, distributed ledger systems, 2015.
[17] Wood, G. Ethereum: A secure decentralised generalised transaction ledger. Ethereum Project Yellow Paper, 2014.

2019 IEEE International Conference on Blockchain (Blockchain)

Ownership preserving AI Market Places using Blockchain

Nishant Baranwal Somy
IIT Kharagpur, India
somy1997@gmail.com

Kalapriya Kannan
IBM Research, India
kalapriya.kannan@in.ibm.com

Vijay Arya
IBM Research, India
vijay.arya@in.ibm.com

Sandeep Hans
IBM Research, India
shans001@in.ibm.com

Abhishek Singh
IBM Research, India
abhishek.s@in.ibm.com

Pranay Lohia
IBM Research, India
plohia07@in.ibm.com

Sameep Mehta
IBM Research, India
sameepmehta@in.ibm.com

***Abstract*—We present a blockchain based system that allows data owners, cloud vendors, and AI developers to collaboratively train machine learning models in a trustless AI marketplace. Data is a highly valued digital asset and central to deriving business insights. Our system enables data owners to retain ownership and privacy of their data, while still allowing AI developers to leverage the data for training. Similarly, AI developers can utilize compute resources from cloud vendors without loosing ownership or privacy of their trained models. Our system protocols are set up to incentivize all three entities - data owners, cloud vendors, and AI developers to truthfully record their actions on the distributed ledger, so that the blockchain system provides verifiable evidence of wrongdoing and dispute resolution. Our system is implemented on the Hyperledger Fabric and can provide a viable alternative to centralized AI systems that do not guarantee data or model privacy. We present experimental performance results that demonstrate the latency and throughput of its transactions under different network configurations where peers on the blockchain may be spread across different datacenters and geographies. Our results indicate that the proposed solution scales well to large number of data and model owners and can train up to 70 models per second on a 12-peer non optimized blockchain network and roughly 30 models per second in a 24 peer network.**

I. Introduction

A number of AI Marketplaces [1]–[6] are being set up as collaboration hubs to enable different stakeholders in the AI value chain to connect, develop, and monetize AI assets i.e. data and models, in a secure manner. Additionally, these marketplaces aim to accelerate innovation, promote responsible use of AI, and fair distribution of value generated by the development and use of AI assets. For example, consider a group of k hospitals each of whom have a certain amount of patient healthcare data, but lack expertise to jointly build AI models. At the same time, AI developers in academia and industry generally lack access to patient healthcare data. In this setting, an AI marketplace can enable collaboration between the hospitals (i.e. data owners) and AI developers to securely build models to assess patient health. Moreover, cloud vendors can contribute GPU compute resources to train these models and any model building blocks can be reused by other developers. Finally, these models can be discovered and consumed by businesses including clinics, insurance, and pharmacy companies.

A critical factor impeding the success of both centralized and decentralized AI marketplaces is that they do not guarantee data and model privacy [7]–[9]. As a consequence, both data and model owners can easily lose ownership of their assets and are unable to derive value from them in a sustainable manner. Moreover, large scale sharing of data or models may not be feasible due to regulatory constraints [10] and also because owners might lose competitive intellectual property and economic advantage. Additionally, centralized AI marketplaces are dependent on a trusted central entity to maintain a verifiable audit trail of data sharing and training, which has the potential to create digital monopolies, increase costs, and is open to malpractice.

This work presents the design and architecture of a blockchain based solution that preserves the privacy and ownership of AI assets in a decentralized AI marketplace that has no trusted central entity. Our system considers three classes of market participants: data owners (DO), model developers or owners (MO) and cloud owners (CO) and allows them to collaboratively train AI models on available datasets using federated learning [11]. In our system, data privacy is ensured by splitting each dataset across multiple COs so that no single entity on the blockchain has access to the entire dataset. Each CO that holds a data subset then participates in multiple rounds of training using federated learning in order to build the model. Model privacy is guaranteed by training models that are encrypted using fully homomorphic encryption, so that model predictions are unusable without the decryption key [12]. Our system has been implemented using the open source Hyperledger Fabric [13] wherein all stakeholders interact with the system through chaincode functions (equivalently smart contracts on Ethereum network). We present the design of these functions, which incentivize truthful recording of all transactions on the blockchain including splitting and distribution of datasets and the scheduling and execution of multiple rounds of training across COs. This ensures that the system provides verifiable evidence of expected behavior or wrongdoing and dispute resolution, thus building trust with all stakeholders. For instance, the system allows an MO to easily verify that the data as proposed by the DO is indeed the data used to train her model. Similarly, an MO can verify that each CO participating in federated learning has indeed submitted a unique intermediate model based on a round of training on its data, as opposed to submitting a copy of a trained model from

978-1-7281-4693-5/19/$31.00 ©2019 IEEE
DOI 10.1109/Blockchain.2019.00029

another CO.

We have deployed our solution across a blockchain network composed of multiple organizations spread across three different locations with each organization contributing up to 24 peers. We execute transactions related to collaborative training of AI models and present experimental performance results that demonstrate the latency and throughput of these transactions under different network configurations. Our experiments show that our system scales well and can support training of up to 70 models/second with sub second latencies observed for recording information in the blockchain.

The rest of the paper is organized as follows. Section II presents the design and architecture of our system including details of its chaincode functions. Section III presents experimental evaluation results, followed by sections IV and V, which present related work and conclusions respectively.

II. System Overview

Privacy preserving environments for data sharing and model training when data, models, and compute resources are offered in a trustless setting, require both efficient protocols and platforms capable of truthfully recording and verifying the sequence of actions by different stakeholders.

Stakeholders. Our system models three marketplace stakeholders: *data owners* (DO), *cloud owners* (CO), and *model owners* (MO). The DOs own large proprietary datasets (e.g. healthcare, self-driving cars, compliance data, etc.), which are especially valuable to train accurate AI models. They wish to monetize this data and sell it for AI training in a safe, secure, and transparent manner multiple times without loosing its ownership. By being part of a marketplace, DOs can increase the outreach and monetary gains that they can derive from their data, which may otherwise be utilized minimally or lie unused in their datalakes. The COs are cloud service providers, who wish to sell storage and GPU compute resources needed to train AI models. By being part of a marketplace, the COs can increase their customer base and offer subscription-based storage and compute services at competitive prices. The MOs are enterprise or freelance AI developers who have the skills and experience to develop sophisticated AI models, but lack the data needed to train the models. The MOs eventually wish to monetize their trained models and therefore do not want to loose ownership of their models in the process of training them. By being part of a marketplace, MOs can obtain access to diverse datasets that meet their training requirements, GPU compute resources at competitive prices from COs, and ultimately lower the overall cost of training AI models. Figure 1 shows the different marketplace stakeholders, each of whom is motivated to participate in the marketplace based on their personal economic gains.

System Operation. There are two distinct phases of system operation for training AI models so that both the DOs and MOs retain privacy of their assets, i.e, Data distribution phase and Collaborative model training phase.

Data distribution phase. The protocol begins with DOs expressing their intent to share their datasets in the marketplace

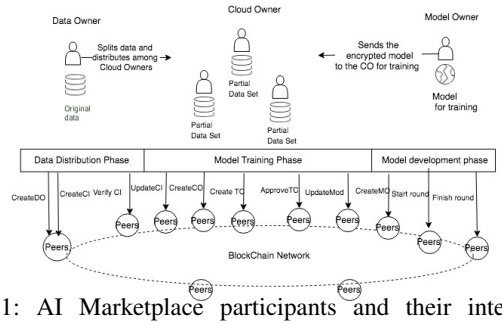


Fig. 1: AI Marketplace participants and their interaction with Blockchain via different chaincode functions (i.e. smart contracts). We have implemented 15 chaincode functions to interact and store information in the blockchain.

for AI training. However, any exposure of the whole dataset can lead to data leaks to other parties in the system. In order to preserve the privacy of their datasets, the DOs can potentially share encrypted datasets, in which case AI training would need to be performed on encrypted data. However when MOs wish to monetize trained models, model inferencing would again require data to be fed in encrypted form. This implies that MOs may need to depend on DOs for encryption of new input data for inference or possibly decrypting the prediction response. Our system uses an alternative novel approach wherein the dataset is split into multiple subsets and stored across COs so that no single CO has full access to the dataset and each CO holds a small fraction of the overall data. The DO enters into an offline contract with a set of COs and securely transfers the data subsets off-chain to the individual CO's. The COs record the receipt of data subsets on blockchain, which is verified and acknowledged by the DO. While recording information on blockchain, each entity uses pseudo identifiers for anonymization and the actual identities of entities is not revealed. Thus only the DO knows the identities of all COs that store its data and none of the COs know each other.

Collaborative training phase. Once a dataset is distributed to the COs, it is available for purposes of AI training in the marketplace. We refer to these data subsets as privacy preserving data subsets (PPDS). An MO develops a model with the help of data samples that DOs expose and obtains permission to train the model on the entire dataset. Since the dataset is distributed across multiple COs, our system employs federated learning to train the AI models, wherein each CO contributes compute resources towards training. However, in federated learning, all COs obtain access to the final trained model, which implies that MOs would loose ownership of their models during the training phase. In order to avoid this, the MOs encrypt their models using homomorphic encryption and the training proceeds in rounds. During each round, the MO shares the running version of the encrypted model with all COs. Each CO trains the encrypted model using its own data subset and returns back the trained model to the MO. The MO aggregates the individually trained models are shares the updated version with the COs for a next round of training, eventually obtaining a final trained encrypted model. Training

of homomorphically encrypted models is a well studied area [14] [15] [16]. We do not delve into the details of the model encryption but choose an approach that seamlessly blends homomorphic encryption with federated learning.

Additionally, since MOs have access to partially trained models at the end of each round, they can potentially learn characteristics of data stored by each CO. In order to avoid this, during each each round, a random set of COs holding a dataset are utilized for training. In this manner, while all data subsets are eventually utilized for training over multiple rounds, it becomes difficult for MOs to decipher the data characteristics of individual COs, thereby guaranteeing complete integrity of data ownership.

There are a number of challenges in realizing the above steps in a trusteless setting using blockchain. Specifically, *how does one ensure that the market operation is transparent to all parties? How can one build a trusted platform for data sharing and collaborative training such that participants can record actions without exposing data and models? How can the protocol ensure that parties cannot collude with each other? How can all parties record their actions such that the system automatically provides evidence of expected behavior or wrongdoing and dispute resolution?*

Our system leverages Blockchain to enable all stakeholders to truthfully record the sequence of events during data sharing and training. Blockchain has the advantage of enabling trust between different non-trusted entities in a marketplace. For details of how blockchain works, one can refer to [13] that provides a comprehensive overview of an open source blockchain implementation, the Hyperledger Fabric. To provide complete transparency and preserve the ownership in a trusted manner our system uses blockchain for recording and validating all operations. We assume that all stakeholders subscribing to the protocol will either contribute nodes to the blockchain network to facilitate their requests and integrate the APIs to interact with the blockchain for all required events as specified in the protocol. In this paper, we turn our attention to designing the system and protocol that would enable all the stakeholders to participate in AI training without the fear of losing ownership. With our system, existing techniques for model encryptions and data splitting can be easily plugged in. Therefore, we delve less on individual mechanisms of optimized data splitting or model encryptions and focus completely on designing a system that would enable ownership preservation.

Existing works like [7] [17] uses the blockchain to both store data and train models. However, our system decouples the storage of data and models from the blockchain (ie., peers in the underlying blockchain network do not store data or train models). This design brings in twin benefits. Firstly, unlike [7] the network nodes need not perform redundant model training operations to provide proof of work or achieve consensus. This eliminates the computation and storage overhead from network nodes. Secondly, the communication with blockchain can be designed as shorter messages using blockchain for recording purposes only. This naturally enables existing data and model platforms to plug-in and integrate into any blockchain network

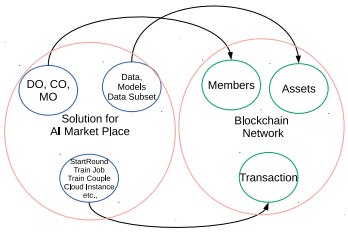


Fig. 2: Mapping between solution and blockchain based entities

seamlessly. Figure 1 shows the interactions of the participating parties with the blockchain network. The interactions are defined in the subsequent sections.

A. The protocol and Interaction with Blockchain

In this section we walkthrough the protocol and interaction with the blockchain. Figure 1 illustrates the interaction of different stakeholders with the blockchain. As mentioned earlier, we assume that the different stakeholders will use our API's to truthfully record the events in the blockchain. Our design ensures that blockchain performs a few critical validations and verification and the stakeholders will not be able to proceed without these validations. This makes blockchain relevant and bypassing the blockchain interactions would not be feasible.

1) Blockchain members and assets: As illustrated in figure 2, our solution for AI marketplace leverages capabilities of the underlying blockchain network to define members, assets, and transactions. Members are essentially participants or stakeholders in the AI marketplace which includes COs, MOs, and DOs. Real world resources are modeled as assets in the blockchain, which in our case includes the Data, Data Subsets, and Models. Transactions (or chaincode functions) enable members to perform a set of predefined operations on the assets related to collaborative training. Member access to perform operations on assets is controlled by the access control list (ACL).

New members are created through the Create calls. A sample representation of DO in the blockchain is illustrated in the listing 1 (members are treated as assets in the blockchain). For example, *createDO* call returns a unique identifier and allows a DO to register and join the marketplace. Similarly other create calls are used to create MO and CO entities and their identifiers are recorded in blockchain.

Listing 1: DO definition

```
asset DO identified by id {
  o String id
  o String name optional
  o String organization optional
  o String howMany optional
}
```

Our solution defines 10 different types of assets to store different entities and their states. We enumerate a few critical assets and describe in detail their purpose throughout the protocol. Assets are essentially represented as data structures within blockchain and used by COs, MOs and DOs to record information during different steps of collaborative training.

For instance, corresponding to each CO, there exists a an asset called *Cloud Instance* (CI), which links the CO to the Data chunk it holds (including data subsets and their replicated counterparts). Three main components in the CI are the subset ID, the CO identifier, and a field that tracks the status of the Data Subset. It should be noted that the actual identifiers of COs, MOs, and DOs (i.e. an ip address or a URL), is never actually stored on the blockchain. Therefore, no parties can query the blockchain and obtain the identifier to establish a direct contact with CO. The blockchain provides a pseudo-random identifier for every member, which is used to store and refer to the members for purposes of querying.

Another asset called *Train Couple* (TC) is used to represent a model training instance, which associates a model with a data subset. A TC has a DSS and a model object as its member variables. The status field indicates whether the model is ready to be trained or has completed the training phase. The listings 2 and 3 show the CI and TC asset definitions respectively as represented in the blockchain.

Listing 2: CI metadata

```
asset CloudInstance
identified by id {
  o String id
  --> CO co
  --> DSS dss optional
  o CIStatus status optional
  o String nonce optional
  o String hash optional
  o Integer rounds optional
}
```

Listing 3: TC metadata

```
asset TC identified by id {
  o String id
  --> DS ds
  --> Mod mod
  o TCStatus status
  o Integer rem
  o Boolean paid
  o Integer round
}
```

Similar to assets, our solution includes about 37 different transactions for enabling collaborative training between different stakeholders in a trustless setting. About 15 of these are called directly by different members while the remaining ones are called from within other transactions. We enumerate a few critical transactions below.

The *StartRound* transaction (listing 4) is used by the MOs to begin the training process wherein blockchain chooses a random set of data subsets which are trained by each cloud owner independently for federated learning.

Listing 4: Chaincode for StartRound Transaction

```
func (t *DataMarketChaincode) StartRound(stub shim.ChaincodeStubInterface, args
      []string) peer.Response {
 //Initiate a TC
 var tc TC
 tcid := args[0]
 tcBytes, err := stub.GetState(tcid)
 if tcBytes == nil {
   fmt.Printf("tc with id %s do not exists\n", tcid)
   return shim.Error("tc do not exists")
 }

 //Check if the TC is approved by the DO so that model training can start.
 if tc.Status != "APPROVED" {
   fmt.Printf("TC with id %s not yet approved\n", tcid)
   return shim.Error("tc not yet approved")
 }
 //Track the number of rounds in the Blockchain
 tc.Round += 1
//Get random list of DSS to train in each round and initialize the 'rem' to the
     number. Each CO updates rem by subtraction when its training is complete
 tc.Curdssidlist = nil
 var r int = (tc.Round+1) % 2
 var ds DS
 dsid := tc.Dsid
 dsBytes, err := stub.GetState(dsid)
 tc.Rem = 0
 for i := 0; i < len(ds.Dssidlist); i++ {
   var dss DSS
   dssBytes, err := stub.GetState(ds.Dssidlist[i])
   if i%2 == r {
     tc.Curdssidlist = append(tc.Curdssidlist, dss.Id)
     tc.Rem += len(dss.Ciidlist)
   }
 }
 fmt.Printf("Going to store tc\n")
 //Store the states in blockchain
 err = stub.PutState(tcid, newTcBytes)
 return shim.Success(newTcBytes)
}
```

For sake of brevity we skip other transactions but describe them in the context of protocol in the subsequent sessions.

B. Data Distribution and Acknowledgement in Blockchain

One of the first steps involved collaborative training is the splitting and replication of a dataset by the DO among different COs. In our solution, both these steps are performed intelligently to ensure that the ownership of the dataset is preserved and that malicious COs do not falsely claim to have trained their models using their data subsets, without actually training them.

Data Splitting. As explained previously, the DO makes an off-chain agreement with a certain number of COs and sends the data subsets to them securely. In our solution, no information about the actual dataset itself is recorded on the blockchain to ensure its privacy. The DO splits the dataset such that each CO has a small subset of the entire dataset. Moreover, the splitting is done in a skewed fashion i.e. no subset has all the classes of the data or range of values. Also, no subset has majority of the data from a single class. The first condition ensures that no single CO can derive meaningful information about the dataset, while the second condition ensures that using multiple queries or using multiple model iterations, an MO cannot interpret the data characteristics.

In terms of steps recorded on blockchain, the DO creates a CI instance by using the transaction 'CreateCI' for each chunk of data that gets distributed. The initial status is set to 'Free' and all the COs are notified. The COs actively wait for events on blockchain and upon receiving a notification, each CO prepares itself to receive the data offchain from the DO. Thus each CO knows the identity of the DO and vice-versa, however none of the COs know each other.

A fraudulent CO can claim to have received a different data subset than the one provided by the DO. In order to avoid this scenario, our protocol requires that the CO declare the hash of the data subset it receives on blockchain. The CO records the hash values of its data subset in the asset CI. It uses the transaction $JoinCI$ to update the hash value on blockchain. The DO consults the blockchain and verifies the hash declared by the CO against its own computation of the hash on the data subset that it distributed to the CO. If there is a mismatch, the DO can tag the CO as fraudulent. The DO uses the transaction $VerifyCI$ to update the status in the CI to 'verified' if the DO sees a matching hash, otherwise it remains set as non-verified. This step ensures that CO receives no faulty data subset. Once a DO verifies the hash of a CO's data subset and marks the CI status as verified, the data chunks become ready for training.

Data replication. In order to save GPU compute resources, a fraudulent CO can also claim to have trained the model on its subset without actually performing the required training

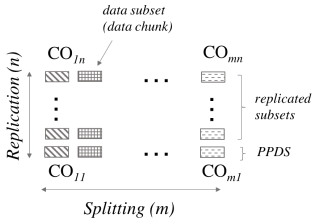


Fig. 3: Illustration of how a dataset is distributed over mn cloud owners. The dataset is split into m PPDS (horizontal), with each subset replicated n times (vertical). In practice, the replication can be a function of space, time, required level of consensus, etc and may vary for each data subset.

(by supplying a dummy model). To prevent this scenario, our solution also requires that each data subset be replicated (i.e. copied) across multiple COs as illustrated in figure 3. We use the term replicated subsets to refer to data subsets that are replicated. The replicated subsets serve a different function compared to PPDS. The former ensures that model training was indeed performed correctly on each data subset, while the later is used for the purpose of preserving ownership without exposing the full dataset. During federated learning, model training rounds are performed by all the mn COs that hold a subset of a dataset. Thus, if a fraudulent CO reports an inaccurately trained model, this can be compared against other COs that hold the same replicated subset. As long as the number of fraudulent COs is $< n/2$, any party can verify whether the CO actually trained the model using its data subset or provided a fake model.

In terms of steps recorded on blockchain, for each dataset, 'CreateDSS' is called to have an entry for each of the data chunks in blockchain, i.e. both replicated and PPDS. After each round of training, each CO records a hash of its partially trained model onto the blockchain. However, as the hash values of models trained by COs that hold replicated data subsets would be the same, a fraudulent CO can still read these hash values from the blockchain and report these back again as proof of a trained model. To overcome this problem, each CO appends a random nonce to its partially trained model and then computes the hash. The CO then reports both the hash and the nonce onto the blockchain. Since no two COs can produce the same nonce, the hash reported by each CO is guaranteed to be different. Thus if a fraudulent CO copies any other CO's hash and/or nonce and reports it as its own, then the CO reporting the same hash at a later time on blockchain can be concluded as fraudulent. As long as the number of fraudulent COs is $< n/2$, the correct model corresponding to each data subset will be available via consensus. Additionally, any party that has access to the partially trained model (e.g. MO) can append the nonce reported by the CO to the model and compute its hash to verify the hash reported by the CO.

C. Training via Federated learning steps on blockchain

Once a DO shares and verifies the data chunks held by different COs, the corresponding COs are ready to participate in the training process via federated learning. The DO publishes details related to its dataset and a contract binding which an MO can lookup these details and use the dataset for training. When an MO is convinced about the dataset and the associated contract, it expresses its intent to train an AI model on the dataset. The training of models via federated learning proceeds in rounds. Since the dataset is distributed across multiple COs, during each round of training the MO supplies the current running model to all COs. At the end of the training round, the MO receives updated models from all the COs and aggregates these into a single federated model (generally by averaging), which it then supplies to all COs for a new round of training. The MO also decides the termination criteria for training the model, i.e. it evaluates metrics on the federated model to determine if the model requires further training.

In terms of the sequence of operations recorded on blockchain, models are essentially represented as assets and an MO uses 'CreateMod' to create an instance of the Model object on blockchain. The model object holds the following information: Model Owner ID, Model ID, Model Type, Model URL, Training Method and Model Hash. The model type specifies the type of AI model (e.g. DT, Neural Network, etc.,). Model URL points to the current version of the model that is obtained after certain number of rounds of training. The training method specifies parameters related to how the training has to be performed by each CO The hash value field holds the hash declared by a CO after a round of training. This is used for verification by the MO when it downloads the trained models for aggregation.

An MO expresses intent to develop and train a model by creating a TC object in the blockchain using the smart contract 'Request TC'. It mentions the dataset and the model object while creating the TC. The 'status' field in the TC object tracks the different phases of training. A notification is raised by the blockchain when a TC is created, which notifies the DO that a model owner wishes to train on the dataset. The DO then uses the 'ApproveTC' transaction to approve the model training by setting the status field to "APPROVED". This allows the DO to verify if the same MO has submitted multiple prior requests and approve model training. An asset called Train Job (TJ) is created by the same transaction used by the DO to approve the TC for tracking the progress of individual jobs (training of each of the datasets). Therefore, for each CI there exists a corresponding TJ.

Randomization of PPDS. The training begins with an MO invoking the 'StartRound (SR)' smart contract transaction. The SR transaction takes as input the TC and identifies the dataset (and eventually PPDS) that can be used for training the model in that round. While federated learning in general allows training on all the PPDS held by different COs, our solution uses a random set of PPDS during each round of training. For the chosen PPDS all the replicated units are considered. A random selection of PPDS for each round of training ensures that MOs cannot deduce any meaningful information about the data held by COs using the partially trained models obtained after each round. However, training

over multiple rounds ensures that all the PPDS are eventually utilized for training. At the end of SR, blockchain raises notification for each CO whose PPDS and replicated units has been selected for that round.

The COs listens for events published on the blockchain. If a notification corresponding to the start round transaction arrives with TC ID associated with a CO, the corresponding CO participates in a round of training. The CO examines the model object provided in the TC to obtain information about the model and the associated training method. The URL for the training program is encrypted with the keys provided in the training file. The CO downloads the training program and starts training. The trained models are stored in specific object stores. The location of the trained model is signed using the public key provided by MO in training file. The TJ is updated with the hash value of the model and encrypted model location using the smart transaction 'UpdateTJ'. The rem field in the TC is used to track whether all CO's have updated their respective TJs. When the rem matches the total number of PPDS selected, the MO is notified that the training is completed.

Training consensus. As explained in the previous section, a malicious CO may provide a model that is not accurately trained in order to save GPU resources. The consensus across models trained on replicated data subsets helps avoid this problem. For each PPDS, there exists a set of replication subsets held by other COs. Therefore the MO can check the contents stored in blockchain and verify whether the models trained on replicated subsets yield the same hash values. Incase the hash values are different, then the trained model corresponding to a data subset is obtained via consensus across the respective replicated subsets. This cross validation of models across COs ensures that training on PPDS is valid and that COs do not maliciously declare that they have trained a model without actually training it accurately.

The MO listens for notifications about round completion (i.e. when rem matches the number of PPDS that has updated the TJ) and upon receiving downloads the individually trained models uploaded by the COs on cloud object storage. The MO uses its key to decrypt the URL (encrypted by the CO using the key available in the training file that belongs to the MO). After downloading the model file, it computes the hash value and compares it with the hash value reported by the CO to ensure that the version the CO claims to have uploaded is the same as the one downloaded by MO.

After successfully downloading all the models trained by respective COs, the MO aggregates these into a single federated model and evaluates metrics on the model. Based on the metrics, the MO can invoke the SR again and continue to perform the training process or terminate the training.

D. Trusted Verifiability of Participant actions

This section shows how the actions of any participant can be verified using transactions stored on the blockchain.

How does MO verify that CO has the dataset that it is claiming to have? In our system, a dataset is split across multiple COs and each data subset is also replicated across multiple COs. Thus multiple CIs exist for the same data subset. When a data subset is used for training, all copies of the same data subset is used for training (albeit in different CO). Each CI will have the hash value of its data subset. A consensus is used to check if the result reported by the CO owning the same subset are same. For example, consider a data subset replicated across 5 COs. Thus all COs will train on the same data subset. A consensus on correct training can be set to verify the output from at least two COs. The MO can wait for the response from 2 COs and verify the model. If the models match, then the data subset that CO claims to have is correct.

How can we ensure that 2 COs having same data subset do not copy each others hash? In the setting where a data subset is replicated among multiple COs, one CO can copy the hash of a data subset to its CI and falsely claim to have a copy of the data subset. To avoid this, each CO computes the hash on the data subset appended by a nonce, before publishing its hash. It also publishes the nonce used in the CI. Therefore a DO can verify if any two COs reported the same hash or not.

How does CO ensure that DO has given it the correct dataset? In order to establish that it has not received a bad data subset or an in-correct one (in case a DO colludes with another CO), a CO can verify the hashes uploaded by other COs that have the same subset. To identify same subsets, the CO computes the hash of the data it has and compare it with others. It adds the nonce declared by another CO to its data subset and recalculates the hash on its subset. If the hash matches with the one uploaded by the CO whose nonce was used, then it is likely that both COs were given the same subset of the dataset. Thus, a CO can cross verify whether it has received the correct data subset from a DO.

How does DO verify that CO has got the correct dataset? The CO upon receiving a data subset updates the CI with the hash of its subset along with the nonce information. DO recomputes the hash along with the nonce of the CO and verifies if both match and sets the status to 'VERIFY'. Only the DO is provided permission to update the CI.

How does an MO ensure that the partially trained model uploaded by the CO is the same that it has downloaded? A CO trains the model received from a MO on its data subset. Once the training is over, the CO updates the TJ with the hash of the model. Since several COs posses the same data subset, a malicious CO can copy the hash value of the model provided by a CO that actually spends resources to train a model. In order to avoid this, a nonce is added by a CO to the model before computing its hash. When the model url becomes available, the MO downloads the model and computes the hash by appending the nonce provided in the TJ. It the hashes match, then the model uploaded by the CO is the same as the one it has downloaded.

How does MO ensure that CO simply doesn't copy the hash from another CO having same data subset? A CO adds a nonce to model to compute the hash and no two COs

can declare the same nonce. Therefore if a CO copies the hash computed by another CO, then MO can recompute the hash by using the model uploaded by CO and the nonce declared by it. If the hashes match then the model uploaded by the CO is same as the one it has declared.

How does DO ensure that the MO doesn't control which data subsets will be used in a training round? The data distributed to the COs is recorded in the blockchain through the CI. In this case, an MO can easily query the blockchain and get the list of CIs and attempt to identify the distribution of the data subsets. In our system, the selection of COs to execute model training in each round is performed by the blockchain through a chaincode function 'start Round'. Although invoked by the MO, this chaincode function which holds the logic of picking up a random subset of COs to conduct a round of training, is carried out by the underlying blockchain which is completely agnostic of the MO. This is also one of the advantages of using a blockchain.

How does DO ensure that MO cannot access the data shared with CO? The COs hold the data subsets and can therefore collude with a MO to share the data. If a subset of COs collude with the MO, the MO can gain access to a part of the dataset, in which case the data may be compromised and the ownership is lost. However this is not possible since the MO only has access to the blockchain and does not know the identity of any COs. Each CO is effectively represented as a random hash value in the blockchain. Therefore, an MO does not obtain access to the actual identify or location of the CO.

III. Experiments, Observation and Results

We have implemented our system - 15 transactions in all, using chain code functions. Using these transactions one can perform all operations of the protocol and complete model training activities. We have evaluated the protocol for two important metrics: latency and throughput, typical of distributed systems that require scalability. Our experiments were performed with permissioned blockchain network (Hypledger fabric version 1.2.0-rc1) components deployed as Docker containers running atop Soft-Layer [18] servers. Each component was provisioned a separate server with 32 cores, 64GB RAM and ran Ubuntu16.04. We use Hyperledger Caliper (or Caliper) [19] as the benchmarking tool. Caliper allows users to measure the performance of a blockchain implementation with a set of predefined use cases and produces reports containing a number of performance indicators, such as tps (Transactions Per Second), transaction latency, etc.

All our experiments use standard default settings of configuration parameters that comes with the Fabric. This is to emulate the general behavior of the blockchain and one can expect better responses in optimized settings. The block size for all our experiments was 500 and the block timeout was 1s. The default block formation policy was considered as 2:3:1. The transaction submission rate were 500tps,1000tps and 1500tps. We have tried with other higher transaction rates, but the metrics felt sharply down. We ran each experiment 30 times and in each run a total of 100,000 transactions were submitted. We report the average across all the runs.

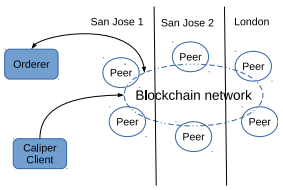


Fig. 4: Ilustration of a blockchain network used to study performance of our protocol. The number of peers shown in the picture is only for illustration and in reality is scaled upto 24 in each location.

Peers in the blockchain were setup in different locations to faithfully reproduce experiments closer to real world scenarios. We had considered upto 24 peers in each location and 2 data center (DC) locations viz., San Jose and London. Experiments were conducted with nodes located within a Data center (Single DC) or across data center locations (2 DC setup). In the 2 DC setup we have two configurations. One where the geo locations were San Jose 1 and San Jose 2 and another where peers are distributed between between San Jose 1 and London geographies. The number of peers were equally distributed between two locations. For example, in a 4 peer network 2 DC setup- 2 peers are located in San Jose location and 2 peers are located in London. Figure 4 illustrates the network setup used in our experiments in different locations. We have one orderer and one client both in the San Jose location.

A. Effect of Increasing number of Peers

We study the effect of increasing peers in a single DC and 2 DC setup. Figure 5 and Figure 6 shows the different transactions per second and latencies observed with increasing number of peers in a 2 DC setup.

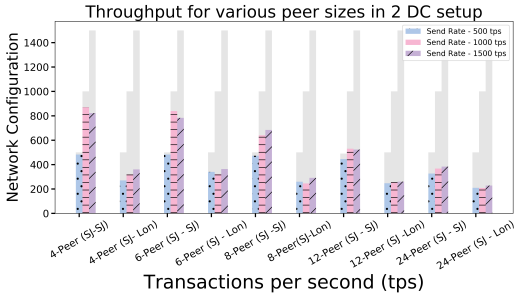


Fig. 5: Effect on Throughput with increasing peer sizes distributed across 2 geographically distributed data centers.

Our experimental results indicates that the throughput falls by about 1.5 times when the number of peers are doubled. For instance, increasing the peers from 4 to 6 reduces the average throughput by about 30% compared to a 4-peer configuration. While this is expected due to the endorsement times and varying queue sizes, the result is also due to the large

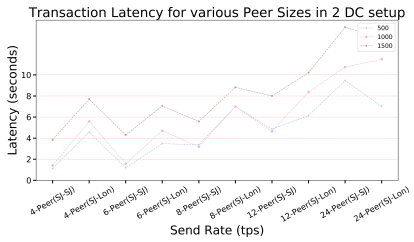


Fig. 6: Effect on Latency with increasing number of peers distributed across 2 geographically distributed data centers

ping latencies observed acorss the machines in these different locations.

Latency is measured end-to-end, from the time of submitting a call to the blockchain to the time it was committed to the ledger by the peers. Figure 6 shows the average latencies computed for different peer configurations. With increase in peer size spread across even 2 different locations the latency increases by about 4-6 times. Our measurements of the ping times between the client and different peers in different locations show that the ping latencies vary in the range of 300ms to 3s. As blockchain peers have to communicate with the client during the endorsement and commit phases, communication costs play a significant role in latencies on blockchain. With high network speeds one can expect the latencies to reduce, however latency has a significant impact on the throughput.

B. Effect of Send Rates

We measure the effects of varying Send Rates. Send Rate emulates different loads that each peer gets from their clients. We experiment with three different send rate 500tps, 1000tps, 1500tps and observe the throughput and latencies of the blockchain system. We have tried experiments with higher send rates but the throughput falls sharply and therefore pivot our observations around these numbers.

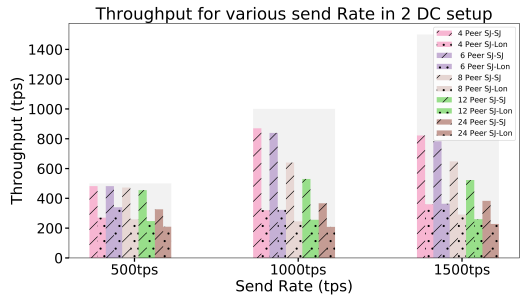


Fig. 7: Effect on Throughput with increasing send rates of transactions

Figure 7 shows the observed throughput averaged over all transactions for varying peer sizes. As our objective was to measure the 'limit' of the blockchain to handle incoming rates (i.e. incoming rate handled by blockchain without drop in throughput), we average the rate over 30 different rounds for all transactions. In the Figure 'SR-TP' shows the gap between the throughput and the 'send rate'. It can be seen that the

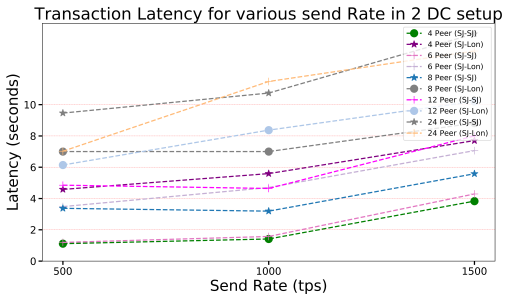


Fig. 8: Effect on Latency with increasing send rates of transactions

blockchain can scale effortlessly for send rate of 1000tps and falls sharply for 1500tps. The sharp fall is due to the fact that as peers increase, time to endorse and commit also increases significantly reducing the throughput rates.

Given these throughput rates we try to answer the question about the number of model trainings that our system can support. Considering a single DO, CO and MO training a data set it takes about 15 transactions to complete the entire model training. It does not include the model training time which is done by the cloud owners and not by the blockchain. On average about 950 transactions are supported (4 DC being the best) for 'send rate' of 1000. This implies that about 65 model trainings per second could be supported easily without additional optimization. Compared to both existing networks like Bitcoin [20] that supports roughly 7tps and number of models reported by current online systems [9] (about 200 models a month) our system is well equipped to scale for large scale model training.

Figure 8 shows the average latency (for all transactions) for different send rates. Latency increases sharply when the transaction 'send rate' is more than 1000 tps. This is due to increase in number of transactions within the blockchain. The blockchain components (such as orderer and endorsers) slow down, spending their time in book keeping of records. Further, the queue sizes for the orderer increase as the transaction input rate increases. Transactions spend time in the kafka queue (queue within the orderer of blockchain that stores the transactions for making blocks) before getting committed. This blockchain setup can support approximately 1000 tps after which the latency increases sharply.

C. Transactionwise comparisons

We study transaction wise impact due to increasing peers and different send rates. Figure 9 (a) shows the different throughput for each smart contract in the solution. Similarly, latency for all transactions across two different network settings (1-DC and 2-DC) is presented in Figure 9 (b).

All smart contracts follow the similar trend in the number of transactions observed for a particular peer distribution. The best observed case is one of 4 peers in a single DC which can support up to 1000(tps). The throughput reduces significantly for the 24 peer network spread across 2 DCs. As mentioned

earlier this reduction is due to the ping times observed between the peers and the client. In short, individual transactions have similar throughput or latencies. This is key to the design as no single API would block or create undue delays in the transaction execution leading to a bottleneck.

The average time taken to finish all the 15 transactions is about 15s in a 4 peer 1-DC setup. Thus, for a model trained the total time spent on the blockchain is roughly 15 seconds. This time, compared to the actual model training time which can span hundreds of minutes is insignificant compared to the benefits obtained.

D. Comparison to centralized transaction system

We have also performed experiments considering an alternative centralized system that can provide similar functionality using a traditional database. Operations in the blockchain are mapped to either insert operations (where information is recorded) and procedural sql (where logic is available) in database. For the experiments, we use single node open source database Postgres [21] setup and use Pgbench [22] to perform the experiments. Pgbench reports about 2000-3600 tps for a send rate of 100K transactions using 8 clients and each client generating workload using 8 threads. The latency as observed using Pgbench ranges from 700-3000 ms (including the connection time). These observations are on a highly optimized single node transaction systems and accommodates roughly twice the number of transactions than the proposed blockchain system. However, trust is left to the external intermediaries. Our system at a reasonable drop in transactions compared to well optimized Postgres provides inherent trust and outweighs the costs involved.

In summary, we see that the proposed protocol can scale efficiently for training large number of AI applications (supporting 1000tps in a 4-DC setup, equivalent to 65-70 models trainings per second). A model training using our system introduces approximately 15s of delay which remains insignificant compared to the large time scales involved in training AI applications. Even when compared to traditional database systems that provide similar functionality, the performance levels are comparable, thus enabling an affordable system that ensures privacy and ownership of AI assets in an otherwise trustless environment.

IV. Related work

Our system can be contrasted against prior frameworks for AI marketplaces along the dimensions of design, privacy and data leaks.

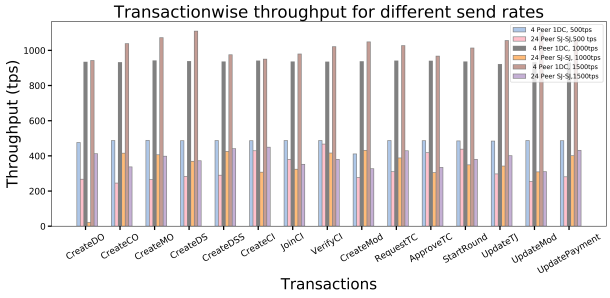
Kaggle [9] is one of the earliest centralized ventures, which provides a marketplace platform for data owners and model developers to collaborate. However both data and models are publicly available to participants. In our distributed design, data and models are maintained by individual stakeholders while blockchain ensures transparent execution of training of models on data. Moreover the design relieves blockchain peers from performing AI training related operations, which enables plug-and-play of our system over any blockchain network.

SkyChain [17] solution is specific to medical AI services. Both model and datasets are uploaded to the SkyChain database and SkyChain provides the infrastructure to train the model. Data and models being core AI assets in the marketplace, one would expect their value and monetary benefits to grow with every usage. [23] is a blockchain based decentralized database for storing personal data in encrypted fashion. Buyers can pay DAT tokens to buy the ”Keys” to the datasets. However, these marketplaces expose either the data or model or both to the participants, resulting in loss of value over time. Ocean protocol [8] is another blockchain based AI marketplace and uses a reputation based system to remove fake Data and dishonest participants for the systems. Our system can operate across multiple administrative domains providing complete privacy and ownership of AI assets. With every application using the data or model, our platform has the potential to provide monetary benefits to all stakeholders. Droplet [24] operates using token transfer. Data is encrypted using symmetric encryption. However the data is exposed to services that perform training and is therefore prone to leakages.

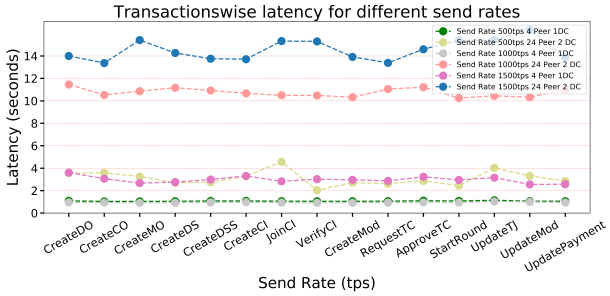
Closest to our work is the Danku protocol [7] proposed by Algorithmia, which enables the operation of an AI marketplace wherein blockchain peers are utilized for both training and data storage. However, multiple peers execute the training of the same model by downloading data, leading to redundant computation and storage utilization. OpenMined [25] focusses on exposing data from end users who own very small amount of data. Our solution allows large data sets to be trained without relying on the data owners to provide computational resources. Computable labs [26] is yet another AI marketplace that trains models, however the data is completely exposed to the buyers who can potentially leak the data to other buyers. Our approach employs lightweight solution where the assets are stored off-chain. The blockchain peers are not overloaded with the training or storage tasks. In [7], with the increase in the number of training jobs, peers would spend significant amount of time in training, which renders it inefficient for large data sets or training that requires longer durations. Our unique proposition is to guarantee complete ownership of data and models without exposing them to any participants in the system.

V. Conclusions

Lack of data and model privacy leading to the subsequent loss of value and ownership have impeded the growth of both centralized and decentralized AI marketplaces. We present a novel mechanism for protecting the privacy and ownership of assets in a decentralized and trustless AI marketplace using blockchain. Our system chaincode functions are set up to incentivize all participants to truthfully record their actions on the distributed ledger, so that the underlying blockchain system holds verifiable evidence of expected behavior, wrongdoing and dispute resolution. Our implementation using the Hyperledger Fabric shows that our system can support large scale



(a) Transactionwise Throughput scaling for select peer sizes and send rates



(b) Transactionwise Latency for select peer sizes and send rates

Fig. 9: Throughput and Latency impact for different transactions

model training and provides a viable alternative to centralized AI systems that do not guarantee data or model privacy.

In future, we intend to include more comprehensive algorithms for homomorphic encryptions of models and utilizing encrypted models in federated learning. We are also working towards an evaluation mechanism where the stakeholders can transparently evaluate the models and data and get paid for their contributions.

REFERENCES

[1] SingularityNET. (2017). [Online]. Available: https://singularitynet.io/

[2] ATMatrix. (2018). [Online]. Available: https://www.atmatrix.org/

[3] BurstIQ. (2014). [Online]. Available: https://www.burstiq.com/

[4] Neureal. (2017) Open-source, peer-to-peer, ai supercomputing, live data stream prediction powered by blockchain, infinitely scalable. [Online]. Available: https://docs.google.com/document/d/1kOJx7clG2V4Tevhg wndRDievXpVaAciPzjmqGxI0CtA/view

[5] Deepsee.io. (2017). [Online]. Available: http://deepsee.io

[6] Fysical. (2018). [Online]. Available: https://fysical.org/

[7] A. B. Kurtulmus and K. Daniel, "Trustless machine learning contracts; evaluating and exchanging machine learning models on the ethereum blockchain," *CoRR*, vol. abs/1802.10185, 2018. [Online]. Available: http://arxiv.org/abs/1802.10185

[8] Ocean Protocol Foundation, "Ocean protocol: A decentralized substrate for ai data and services," BigchainDB GmbH and DEX Pte. Ltd, Tech. Rep., March 2018. [Online]. Available: https://oceanprotocol.com/

[9] KAGGLE. (2010). [Online]. Available: https://www.kaggle.com

[10] GDPR. (2016) EU general data protection regulation. [Online]. Available: https://www.eugdpr.org

[11] J. Konen, H. B. McMahan, F. X. Yu, P. Richtarik, A. T. Suresh, and D. Bacon, "Federated learning: Strategies for improving communication efficiency," in *NIPS Workshop on Private Multi-Party Machine Learning*, 2016. [Online]. Available: https://arxiv.org/abs/1610.05492

[12] X. Sun, P. Zhang, J. K. Liu, J. Yu, and W. Xie, "Private machine learning classification based on fully homomorphic encryption," *IEEE Transactions on Emerging Topics in Computing*, pp. 1–1, 2018.

[13] Hyperledger. (2016). [Online]. Available: https://www.hyperledger.org/

[14] S. Hardy, W. Henecka, H. Ivey-Law, R. Nock, G. Patrini, G. Smith, and B. Thorne, "Private federated learning on vertically partitioned data via entity resolution and additively homomorphic encryption," *CoRR*, vol. abs/1711.10677, 2017. [Online]. Available: http://arxiv.org/abs/1711.10677

[15] C. Gentry, "Fully homomorphic encryption using ideal lattices," in *Proceedings of the Forty-first Annual ACM Symposium on Theory of Computing*, ser. STOC '09. New York, NY, USA: ACM, 2009, pp. 169–178. [Online]. Available: http://doi.acm.org/10.1145/1536414.1536440

[16] N. Dowlin, R. Gilad-Bachrach, K. Laine, K. Lauter, M. Naehrig, and J. Wernsing, "Cryptonets: Applying neural networks to encrypted data with high throughput and accuracy," 03 2016.

[17] SkyChain. (2017) SkyChain: The future of artificial intelligence in healthcare. [Online]. Available: https://cryptoslate.com/coins/skychain/

[18] SoftLayer Technologies, Inc. (2011). [Online]. Available: http://www.softlayer.com/

[19] Hyperledger Caliper. (2017). [Online]. Available: https://www.hyperledger.org/projects/caliper

[20] Bitcoin. (2016). [Online]. Available: https://en.wikipedia.org/wiki/Bitcoin_scalability_problem

[21] PostgreSQL. (1996) Postgresql: The world's most advanced open source relational database. [Online]. Available: https://www.postgresql.org/

[22] Pgbench. (1996). [Online]. Available: https://wiki.postgresql.org/wiki/Pgbench

[23] Datum. (2017). [Online]. Available: https://datum.org/

[24] Droplet. (2017). [Online]. Available: https://dropletchain.github.io/

[25] OpenMined. (2017). [Online]. Available: https://www.openmined.org/

[26] Computable Labs. (2017). [Online]. Available: https://www.computable.io/

2019 IEEE International Conference on Blockchain (Blockchain)

ArtChain: Blockchain-enabled Platform for Art Marketplace

Ziyuan Wang, Lin Yang, Qin Wang, Donghai Liu, Zhiyu Xu, Shigang Liu
Blockchain Innovation Centre
Swinburne University of Technology

***Abstract*—Blockchain is an emerging technology that has the potential to revolutionize the global industry and create a trusted relationship in a multi-party business network. There are a number of practical use cases where blockchain has been applied. One specific area is the Art industry, where it is a natural fit in the way that art forensics and transactions are conducted, tracked and recorded. This motivates us to develop the ArtChain platform to assist the Art Industry. In this paper, we present ArtChain, which is an integrated trading system based on blockchain. It includes the front end, the back end, the services, the smart contract, the chain connection and the deployment scripts from the bottom to the top. To the best of our knowledge, this is the first deployed blockchain-enabled art trading platform in Australia. It provides a transparent yet privacy-preserving, and tamper-proof transaction history for registration, provenance, and traceability of art assets. Our objective analysis and evaluation show that the ArtChain platform is applicable and practical. For the interest of other researchers, our system implementation related resources are open-sourced on Github[1].**

I. Introduction

Blockchain, also known as distributed ledger technology (DLT) [1], is designed to support verification-driven transaction services within a generally un-trusted ecosystem. The design of blockchain technology ensures that no one business entity can modify, delete, or even append any record to the ledger without consensus from other network participants, ensuring the immutability of data stored on the ledger. Blockchain is now being used in several industry applications such as blockchain-enabled traceability and provenance for food safety [2] documentation and cross-organization workflow management in trading and logistics [3].

With $200 billion of annual trading, the art market is one of the largest unregulated markets in the world, accounting for one-third of the amount of crime just behind drugs and guns [4]. Tens of millions of dollars are transferred with little or no documentation and transparency. Current challenges and issues in the art market are: (1) lack of transparency on prices and ownership history (provenance) and inadequate control of transaction data due to the information asymmetry; (2) the authenticity and appraisal of high-value works of art is difficult; (3) lack of the value of artworks at the primary art market and transparency trading at the secondary auction market (both online and offline); (4) lack of recognition, public attention and care for a large number of artists; (5) it is difficult for the artists to get royalty payment from the secondary market.

Blockchain technology possesses a natural fit to improve the transparency, keep records and reduce illicit activities in the art market, due to its inherent properties [5] [6]. In this paper, we present our project work, called *ArtChain*, a blockchain-based art trading system, which has been piloted and operated as a working product in practice. It is expected to provide a complete solution towards these challenges by creating a new ecosystem for the art keeping, trading and transferring. *ArtChain* fundamentally builds up the underlying architecture of blockchain to support a commercial-level trading platform centered around art assets. The core value proposition of the platform lies in:

- *Privacy Protection* Shared ledger along with permissioned control ensures the transparency of each transaction which guarantees the privacy protection in art trading and provenance.
- *Traceability* Real-time tracking of individual artworks combined with the blockchain ledger assists in the fight against counterfeit artworks.
- *Irreversibility* The on-chain registration of collectors offline assets provides an immutable digital record of the artwork, which guarantees the true ownership, the provenance and the value of the artwork.
- *Transparency* Publicly displaying artworks to a wider range of professional investors, leveraging the openness of art ecosystem.

II. Blockchain Solution towards Art Trading

In this section, we start with the rationale behind the use of blockchain for an art marketplace, and then discuss the benefits of this blockchain-enabled platform.

A. Rationale behind Using Blockchain

The major entities or participants in our solution are described in the following.

- *Artist* Established artists along with new generation artists all have equal opportunities for professional evaluation and to publish their artworks. Published items will be available for trade.
- *Art gallery* After artwork is registered on the blockchain system, it can be tracked and located in real-time giving an additional level of security to galleries.

[1] https://github.com/ArtChainGlobal

978-1-7281-4693-5/19/$31.00 ©2019 IEEE
DOI 10.1109/Blockchain.2019.00068

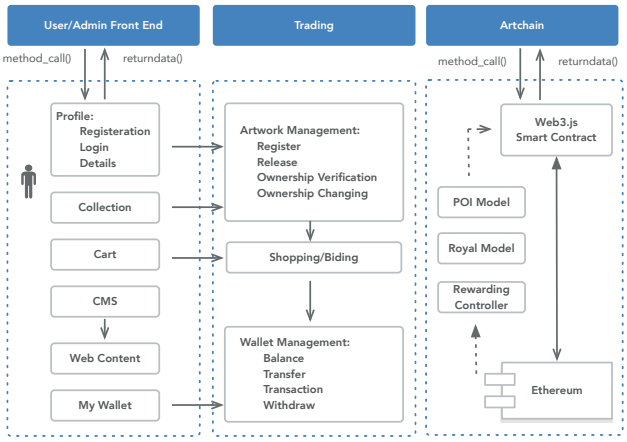


Figure 1. System architecture

- *Auction house* Data stored on blockchain can be synchronized with auction houses, opening up new channels for a greater global audience participation. More traffic equals more opportunities for all involved.
- *Collector* Online synchronization of collectors offline art assets. This proves ownership and sets the provenance of the pieces for future generations ensuring the value is preserved. Access to a global database with extensive filtering capabilities.

B. Benefits of a Blockchain-enabled Art Marketplace

Firstly, the artwork authenticity and traceable data can be simply achieved. Provenance is crucial when it comes to collecting art. Not having a record of the ownership history for a masterpiece often raises suspicion that it could be stolen or fake, hence a distributed ledger can be used to trace the transfer of ownership over a period of time, and serve as a decentralized database securing provenance data and other important information related to artworks. This allows for quick and indisputable ownership transfer in trading.

Secondly, royalty payment from the secondary market for artists can be achieved. A 5% royalty will be payable to visual artists on certain commercial sales of their work. This entitlement is created by the Resale Royalty Right for Visual Artists Act 2009.However, due to the difficulty in tracking the resales in the current art market, often artists do not necessarily get the royalty [7].

Thirdly, Blockchain audit trail helps in detecting tax evasion and money laundering. Add-on analytics or AI services can predict the current value of an artwork based on shared transparent data. This helps primary market valuation, which is more difficult and more speculative than secondary market due to a lack of market history.

Furthermore, our solution is designed to become an open, expandable infrastructure orientated towards the art industry. This means that participants will have the opportunity to develop an extensive range of art-related applications for specific scenarios based on the foundation of ArtChain.

III. System Overview

In this section, we first present the foundational principles the architecture is based on, the high-level architecture and its main components. Then, we present basic data model design and the trading process of the platform. In addition, we discuss the trust and security issues.

A. High-level Architecture Design

We first evaluate several blockchain platforms to inform our decision on which platform to apply. Based on the business requirements and technical assessment we decide to use the Ethereum private blockchain and Proof of Authority (PoA) [8] as the consensus algorithm. Initially, we considered to use Hyperledger Fabric to implement our system due to its capability, popularity and maturity. However, it lacks support in native token, which is a key business requirement in our design as the art trading platform hopes to integrate the payment process and the ownership transfer process. We design and implement a utility token called ACGT to achieve the high performance requirement. Refer to Section IV-A Tokenization for more details.

Here, we adopt microservices architecture for the following benefits: (1) Allows quick parallel development of various components in the application landscape; (2) Reduces discussion time between various groups developing various components; (3) When done properly, provides clean reusable interfaces; (4) When done properly, reduces handshaking in interfaces; (5) Reduces the risk and time of integration/chain testing. The architecture design is shown in Figure 1.

There are three layers in the system: the user front end, the trading back end, and the ArtChain blockchain layer.

- *User Front End:* includes the following functions: managing Profile for user registration, login and user details; displaying art Collection; shopping Cart; user Wallet; and CMS (Content Management System) to create and manage web content.
- *Trading Back End:* consists of Artwork Management, Shopping/biding, and Wallet Management. Artwork Management includes artwork registration, ownership verification and ownership transfer. Artists or collectors conduct the registration of their artworks through the assessment system of professional institutions within ArtChain. Their works of art will then be eligible for trading and participating in the ecosystem.
- *ArtChain:* including the following components: (1) *Royalty model:* responsible for artists royalty payment in the resale of their artworks. (2) *POI model:* manage *Proof of Interaction* (POI) agreements, which are used as incentives to grow the ecosystem of applications.

Figure 2. Data model: *User* Class

More details are described in Section IV-A. (3)*Rewarding Controller:* based on POI model to manage the rewarding to participants. The details are business confidential information, which is out of the scope of this paper.

B. Design of Data Model

There are three major groups of data objects stored in the distributed ledger as illustrated as follow:

- *User:* contains all information related to a user's profile, login, wallet, and auction events attended. An artist is also a user, with additional information and verification. The detail is shown in Figure 2
- *Artwork:* consists of details, tag, history of ownership transfer, and order details. These class represents the workflow related to the masterpiece. The detail is shown in Figure 3.

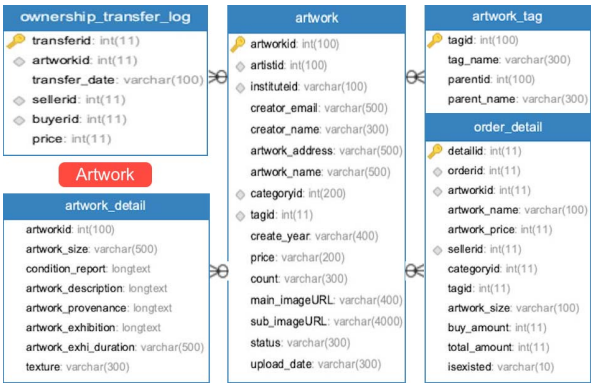


Figure 3. Data model: *Artwork* Class

Figure 4. Data model: *Trading* Class

- *Trading:* combines an order with the artwork and the buyer's basic information and the shipping address. The detail is shown in Figure 4.

The trading process is shown in Figure 5. When the user's trading request is received, Profile Services and Trading Services are triggered to retrieve the customer info and trade info. After checking the trading conditions, Payment Services are responsible for handling payment. Then Reward Services are called to request and receive the reward information. In the end, Shipping Services handle the shipping information.

C. Trust Establishment

ArtChain co-operates with specialized or high-profile partners in the primary or secondary markets of the art industry, including museums, art galleries, and auction houses, to establish the **original** ledger nodes and provide core functions such as validating, ordering and generating blocks of transactions. These ledger nodes and other agent or routing nodes work together to protect the blockchain network.

We use Proof-of-Authority (PoA) as the trust model of ArtChain network. PoA is well-suited to regulated industries where entities are responsible for maintaining the network (known as authorities), rather than remain anonymous as in mining-based chains.

For our practical purpose, well-known museums and art galleries are acting as authorities in ArtChain network to conduct authenticity and price assessment for an artwork. They are called SuperNode. Supernodes perform validating, block generation and publishing.

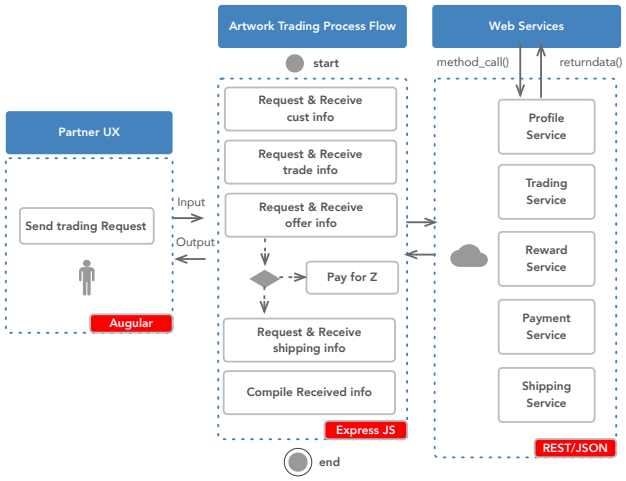


Figure 5. Trading Process

Any node attempting to engage in malicious conducts or falling under attack will be immediately detected by other nodes in the network once it shows unusual behaviour (e.g., sending illegal transactions, traffic attacks, and data tampering). The network will immediately isolate the particular node and send out warnings. ArtChain deploys ledger nodes throughout the primary and secondary art markets, including internet companies, cloud service providers and a large number of collectors of works of art and artists, which, from a probability point of view, can eliminate the possibilities where the majority of nodes fall under attack or collude to engage in malicious conducts.

Initially, we set up 100 nodes to provide sufficient redundancy and fight against 51% attack. Currently, they are deployed at AWS and Ali cloud and not activate all at the same time. We monitor the nodes behaviour and dynamically replace the crashed nodes. We plan to extend the deployment to be more decentralized on other clouds. In this regard, an important issue to consider is the trade-off between decentralization and performance.

IV. Scalable Blockchain Implementation

In this section, we present the implementation of ArtChain network. We first introduce tokenization and how it works in our system. Then we describe the design and implementation of the upgradable smart contract for the purpose of improving function and fixing bugs. Finally, we discuss how to preserve privacy and confidentiality as required by regulations and business needs.

A. Tokenizaiton

Tokenization refers to converting an asset into a digital token on the blockchain system, so that ownership of the asset can be transferred via smart contracts. Smart contracts have functions for automatic transactions, formulas for calculating asset prices and other specific features [9], [10]. Tokenization is not simply the creation of a token. Instead, it is about the design of the whole system, including understanding the various rights and issues.

There are two types of token in ArtChain: the security token ACG[2] and the utility token ACGT.

ACG token comes with the essential technical features of digital currencies, including a steady issue curve, free trading, immunity to double-spending attacks, and traceable transaction history. These features are secured through the ledger architecture and smart contracts. We develop relevant E-wallets for corporate or institutional users, which incorporate all essential functions for interactions with the applications on ArtChain.

ACG token provides incentives for maintaining the ArtChain network and the ecosystem of ArtChain applications.

- *Network Maintenance:* the consistency of ArtChain network is jointly assured by ledger nodes. Ledger nodes will have the opportunity to be awarded with ACG as block rewards and transaction fees, to encourage them to contribute to the security and stability of the ArtChain network.
- *Ecosystem of Applications:* ArtChain will award users with newly added ACG in positive correlation within a certain cycle based on a number of indicators such as their frequency of interaction with the ArtChain ecosystem, levels of contribution, influence and the number of ACG coins they hold. All indicators of ecosystem incentives are quantifiable and verifiable, which are collected and calculated by ledger nodes. Incentives will be allocated under *Proof of Interaction* (POI) agreements.

The ratio of the incentives for ArtChain network maintenance and the incentives for the ecosystem of ArtChain applications will be dynamically adjusted by using a negative feedback mechanism to maintain the balance and stability of the ArtChain network and the ecosystem of applications. The specific indicators and algorithms will be published before any main relevant applications go online, and will be implemented and operated through open rules of contracts. Relevant institutional users of the ecosystem (art galleries, museums, auction houses and artists) will be consulted.

The utility token ACGT is only used internally to facilitate payment in art trading. It is a kind of stable coins, which are designed to have a stable price or value over a period of time, therefore, less volatile. These coins aim to mimic the relative price stability of fiat currencies on one hand, but keep the core values of cryptocurrencies such as decentralization and security, on the other hand. Each ACGT token is collateralized by an equal amount of fiat currency (1 AUD) held by

[2]https://etherscan.io/token/0x984c134a8809571993fd1573fb99f06dc61e216f

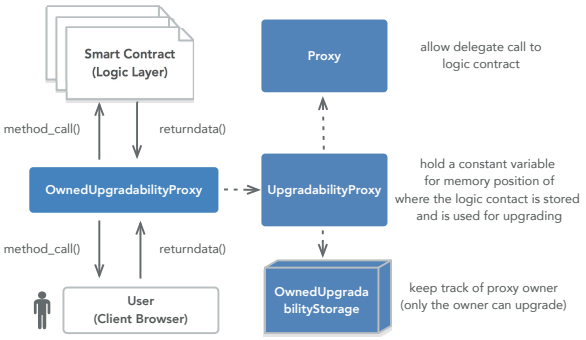


Figure 6. Zepplin proxy architecture pattern

a central custodian. Holders are guaranteed to redeem their token at any point for the stable value denominated in fiat.

B. Upgradable Smart Contracts

Smart contract, once deployed into the blockchain, is immutable literally. In consideration of bug fix and function improvement, lots of work has been done to propose an upgradeable design pattern of the smart contract [11].

The typical methods include:

- Separate logic and data
- Partially upgradable smart contracts system
- Separate logic and data in key-value pairs.
- Eternal storage with proxy contract

Among these methods, the proxy mechanism is most flexible and guarantees a 100% upgradable mechanism i.e., the logic could be completely modified while remaining the existing data state. In our system, we refer to Zeppelin's proxy patterns [12] and implement the so-called Unstructured Storage Pattern. The contract structure is shown in Fig 6.

By using this pattern, the system achieves following features:

- General user is unaware about upgrade of the contract.
- Implementation of logic contract is 100% upgradable.
- Data is stored in proxy contract. New data fields could be added by the upgraded logic contract, without touching existing data structure.

This design has also been chosen for the ZeppelinOS smart contract system, and gone through a full security audit.

C. Initializing Issue

In our implementation, we also address the initializing issue. This is a long-standing problem of upgradability solutions for Ethereum. Our aim is to create an upgradable logic contract, and we practically deploy a proxy contract which delegates to a pre-existing deployment of logic contract on the blockchain. Therefore the proxy contract has no chance to establish the initializing steps in the constructor of logic contract, and thus we need to do something special in order to correctly initialize the proxy contract.

Our workaround for it is to use an initializer function instead of the constructor, and make sure it is only executed once for a necessary initializing process. Other proposals could be found as Initializer Contracts [13] [14].

D. Safety Control

Transaction security: ArtChain network assures the security of users' accounts and funds by using blockchain consensus, digital signatures and end-users encrypted wallets. The artwork trading platform provides security services that are likened to those offered by financial institutions. It integrates data, applications and transactions in blockchain clouds through the efficient integration of data storage, network and other resources, so as to create a secure transaction environment.

Financial Management: ArtChain maintains high standards of integrity and ethical business conduct and is in compliance with relevant laws and regulations, as well as self-regulatory principles of the industry. We also implement a component to conduct the regulatory duty of Know-You-Customer (KYC).

E. Privacy and Confidentiality

ArtChain makes public all ledger nodes and their state in the network in real time. The transaction history (block content) and state information in ArtChain are publicly visible. However, in case of any privacy requirements for some transactions, such privacy information will be processed.

In the data model design, we carefully decide what data to be stored on-chain and what off-chain. The design has been evolved along business needs and regulatory needs. Currently, the on-chain data store contains information on artist, the hash of ownership, price, and history. The hash of ownership protects the privacy for owners who do not want to be known to the public, as well as in compliance with privacy regulations, such as the General Data Protection Regulation (GDPR)[3].

V. Performance Evaluation

We conduct an extensive performance testing of the system. We identify that the performance bottleneck of the system is the low-level I/O efficiency of the Ethereum client, i.e. Geth[4] in our system.

ArtChain private chain, based on POA consensus and 5-second block interval and deployed on 6 cloud nodes (8x2.5GHz CPUs, 32G memory), supports up to 1500 TPS, i.e., 1500 raw transactions on the chain, far more superior to Ethereum mainnet (about 15 TPS nowadays). Integrated user actions, like post new artwork or top up tokens, are usually comprised of a series of transactions/queries on the chain. ArtChain on average processes about 40-70 user actions per second.

[3]https://eugdpr.org/
[4]https://geth.ethereum.org/

A. Environment Setup

The private chain is composed of 3 Ali cloud servers (8x 2.5GHz CPUs, 32GB memory, 64GB hard disc), and 3 AWS cloud servers (8x 2.5GHz CPUs, 32GB memory, 8GB hard disc). Geth version 1.8.17, startup parameter is tuned as:

- --targetgaslimit 4294967295: increase the gas limit to 0xFFFFFFFF to seal as many transactions as in one block. Note this need to coordinate with the gasLimit in the genesis.json file when creating the chain.
- --txpool.lifetime 24h --txpool.accountslots 65536 --txpool.globalslots 65536 --txpool.accountqueue 64 --txpool.globalqueue 65536: increase $txpool$ so that it stores as many transactions both account specifically and globally as we submitted.

This paper employ web3.js[5] to communicate with the chain, and to monitor its performance, Wireshark[6] is applied to capture packet for analysis.

B. Throughput Analysis

Basically, blockchain throughput is limited by: a) How long it needs to generate a block, and b) How many transactions can be sealed in a block. And theoretical throughput = (number of transactions in a block)/(block interval). But in a large-scale network, the throughput is also restricted by the broadcast speed. An explicit example is Ethereum mainnet, with network congestion, its throughput dramatically degrades as nodes need more time to keep synchronized. This is why Ethereum is considered to have issues on network scalability.

As for our private chain, we tried following steps to tune up the system performance: (1) Speed up the block generation by changing the block interval when generating the $genesis.json$. To summary, the chain with 1-second interval shows the best performance, but 5-second is also acceptable. (2) Improve the gas limit of the chain. It does not shows significant improvement on the performance, because the gas limit is not the bottleneck of the system.

As our chain is only maintained by 6 cloud servers, we can ignore the effect of network scalability mentioned above. As long as the transactions are sealed, the nodes always have adequate time to keep sync. On the contrary, it is the node's hardware configuration that determines the system performance. We observe frequently crash of Geth client on the node with only 8GB memory originally. Using the node with 32GB memory, the performance is significantly improved, but the crash still occurs in certain scenarios.

Geth is thought as a memory monster whose design follows a "I use up what I have" idea, and will use up all available memory on the server. By default, our node servers disable the swap and will kill Geth process if it tries to use up the memory. Unfortunately, this always occurs. We observed it used up 8Gb memory when trying to create 70 new accounts. A suggestion is to enable the swap on the node, in terms of the sacrifice of the performance. Note the AWS cloud server has only 8GB disc space, and so the swap space is restricted on AWS servers.

[5]https://web3js.readthedocs.io/en/1.0/

[6]https://www.wireshark.org/

C. Test Results

1) Raw transaction test: The chain is configured with 5-second block interval and we get that (1) Transaction carries data of 50 bytes, it is a typical value for general transactions. (2) Establishes 2000 transactions in about 6-8 seconds per node. (3) Establishes 20000 general transaction queries in 2-5 seconds per node.

2) API based test: APIs such as $check\_user()$, $check\_transaction()$ and $check\_artwork()$ only query information from the chain and do not include any practical transactions. So they show as high throughput as general queries. It only depends on the processor and network performance. APIs such as $buy\_tokens()$, $post\_new\_artworks()$ and $freeze\_tokens()$ combine a series of queries and transactions, and these operations usually depend on the result of the precedent, so those APIs have bad parallel performance. For example, $post\_new\_artwork$ includes 16 low-level operations:

- 2x eth.sendTransaction
- 1x personal.unlockAccount
- 1x eth.estimatesGas
- 2x eth.gasPrice
- 6x eth.getTransactionReceipt
- 2x eth.subscribe
- 2x eth.unsubscribe

Some test results are listed below:

- Establish 116 times API $post\_new\_artworks()$ within about 18-20 seconds per node.
- Establish 58 times API $buy\_tokens()$ within about 5-6 seconds per node.

API $add\_new\_user()$ contains a low-level operation of $personal.newAccount$, which uses significant memory and CPU cycles. A typical result is listed below:

- Establish 64 times API $add\_new\_user()$ within about 20 seconds per node.

3) Test on different block interval: We tested on different block intervals of 1 second, 2 seconds, 5 seconds and 15 seconds. The comparison of API based throughput with different block intervals is summarized in Figure 7.

As all the APIs are called at the same time during the test, we observe actually all transactions are sealed in one block. Our chain is fully capable of guarantee that. So besides the block interval difference, those calling procedures need almost the same processing time on the network and processes. That is why the different block interval practically results in different throughput.

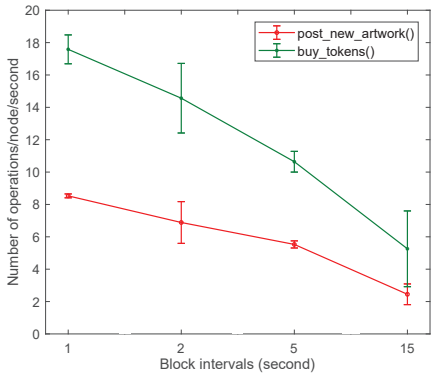


Figure 7. Test under different block interval

4) Test on node crash: Geth client crashes under certain scenarios. What we notice is that transactions like $personal.newAccount$(included in API $add\_new\_user()$) make Geth consume lots of memory. Got in tests:

- On the node configured with 32GB memory, Geth could support up to 70 concurrent $add\_new\_user()$ calls.
- Geth crash when submitting more than 70 $add\_new\_user()$ calls. It is killed by OS after using up all 8GB memory.

We then try to enable 32GB swap on the server, and find that Geth succeeds processing 96 concurrent $add\_new\_user()$ calls. During the process, it used up 32GB physical memory, and then 9.1GB swap memory. As a result, it uses as long as 914 seconds to establish all the calls. As a comparison, it needs only about 20 seconds for 64 calls without using swap memory. Performance degradation is obvious.

D. Bottleneck of the System

Based on our performance test, we find out that: (1) The performance bottleneck of our system is at the Geth IO execution. (2) The way to improve the system performance is to improve node hardware configuration.

According to [15], Ethereum uses LevelDB as the database to store key/value. The key to accessing database is irregular on account of the discreteness of hash. The LevelDB has an excellent performance in reading/writing continuously, while bad for the random key. Therefore, the time t for accessing LevelDB would be longer as the amount of data storage increases. In fact, the test results show that if n is large enough, the value of t will increase and the eciency will degrade largely for some data which not hit LevelDB cache at times.

Geth consumes huge memory on certain transactions, e.g., $personal.newAccount$, and will crash when receiving multiple concurrent memory-consuming transactions. A suggestion is to enable swap memory on the node to improve system stability, at the sacrifice of the performance (See performance degradation when memory is swapped). We suggest 4GB of swap space on the nodes, based on the performance test result. This improves the system stability and does not degrade the performance significantly.

VI. Related Work

In this section, we mainly review the work that closely related to this work, for more work about blockchain and the related applications, please refer to [1], [16].

Art as Digital Assets: Arts can be regarded as the digital asset to be stored on the blockchain platform. The blockchain inherently holds the property of authenticity, traceability, and irreversibility which can perfectly protect the digital assets for each masterpiece. Usually, the blockchain-based solution marks each masterpiece with an ID, may denote as token in smart contract. Similarly, many protocols are designed to trade the nonfinancial assets in form of tokens on the blockchain platform

Blockchain Solution: Since digital assets need properties both on authenticity and security, blockchain becomes the primary selection for the requirements. There are three options, including private blockchain, consortium blockchain and public blockchain. Due to the high security of the assets, the most suitable solution is the consortium methods, which relatively has a better trade-off between performance and security. The asset-based property is deployed on the application-layer of the blockchain, regulated by the rules defined in smart contract. There are plenty of applications successfully executed on top of blockchain [17] and subsequently the infrastructure [18] becomes more complete along with the development. Our solution provides a trading infrastructure for art, and it provides an paradigm for other high value commodities.

Privacy Protection: For the precious digital art assets, it is fundamental to protect the privacy of assets. There are two kinds of privacy in research, including identity privacy and transaction privacy. Identity privacy publicly links the real identity and transaction scripts, and there are several behavioral analysis strategies, including anti-money laundering and know your customer (KYC) to present the usage graph. Transaction privacy means the plain contents on the ledger, including the plain transferring value, account direction, and so on. Some adversaries may draw attentions to watch even monitor some accounts with huge amounts of property. Furthermore, there are several methods to achieve the high level protected blockchain. [19] employed the mixer to obfuscate the relationships among people. Maxwell proposed the Confidential Transaction and firstly achieved the implemented with the range proof scheme. [20] [21] finished the privacy preservation protocols based on ring signature. [22] sealed the plain amounts by using Paillier cryptosystem. [23] provides a complete solution to make the sensitive information unreadable for the public. Our

system relies on the original chain security and provides protection on the layer of web servers and back-end. This design decision comprehensively considers the performance and security for the whole integrated system as a trade-off.

VII. Conclusions

In this paper, we presented the ArtChain, which is an platform designed with registration, tracking, protection, and provenance for artworks enabled by blockchain technology. We also discussed how to design, implement and deploy the blockchain platform in operation as a working product in practice. The proposed blockchain implementation and Experimental results showed that our system towards artworks can provide a complete blockchain-based solution with the property of irreversibility, authentication, traceability and transparency.

In future, we plan to work on anti-counterfeiting for the original works of art by integrating with the smart modules of IoT, and activating relevant smart hardware and other functionality (e.g. positioning/location tracking) as required by artists or collectors.

References

[1] Z. Zheng, S. Xie, H.-N. Dai, X. Chen, and H. Wang, "Blockchain challenges and opportunities: a survey," *International Journal of Web and Grid Services*, vol. 14, no. 4, pp. 352–375, 2018.

[2] F. Yiannas, "A new era of food transparency powered by blockchain," *Innovations: Technology, Governance, Globalization*, vol. 12, no. 1-2, pp. 46–56, 2018.

[3] Z. Wang, D. Y. Liffman, D. Karunamoorthy, and E. Abebe, "Distributed ledger technology for document and workflow management in trade and logistics," in *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*. ACM, 2018, pp. 1895–1898.

[4] M. Zeilinger, "Digital art as monetised graphics: Enforcing intellectual property on the blockchain," *Philosophy & Technology*, vol. 31, no. 1, pp. 15–41, 2018.

[5] M. McConaghy, G. McMullen, G. Parry, T. McConaghy, and D. Holtzman, "Visibility and digital art: blockchain as an ownership layer on the internet," *Strategic Change*, vol. 26, no. 5, pp. 461–470, 2017.

[6] L. Lotti, "Contemporary art, capitalization and the blockchain: On the autonomy and automation of arts value," *Finance and Society*, vol. 2, no. 2, pp. 96–110, 2016.

[7] A. Whitaker, "Artist as owner not guarantor: The art market from the artists point of view," *Visual Resources*, vol. 34, no. 1-2, pp. 48–64, 2018.

[8] S. De Angelis, L. Aniello, R. Baldoni, F. Lombardi, A. Margheri, and V. Sassone, "Pbft vs proof-of-authority: applying the cap theorem to permissioned blockchain," 2018.

[9] M. Utz, S. Albrecht, T. Zoerner, and J. Strüker, "Blockchain-based management of shared energy assets using a smart contract ecosystem," in *International Conference on Business Information Systems*. Springer, 2018, pp. 217–222.

[10] G. Blossey, J. Eisenhardt, and G. Hahn, "Blockchain technology in supply chain management: An application perspective," in *Proceedings of the 52nd Hawaii International Conference on System Sciences*, 2019.

[11] J. Tanner, "Summary of ethereum upgradeable smart contract rd," https://blog.indorse.io/ethereum-upgradeable-smart-contract-strategies-456350d0557c, accessed: 2018-11-30.

[12] S. Palladino, "The parity wallet hack explained," *July-2017.[Online]. Available: https://blog. zeppelin. solutions/on-the-parity-wallet-multisig-hack-405a8c12e8f7*, 2017.

[13] T. Wang, "A unified analytical framework for trustable machine learning and automation running with blockchain," in *2018 IEEE International Conference on Big Data (Big Data)*. IEEE, 2018, pp. 4974–4983.

[14] K. L. Brousmiche, A. Durand, T. Heno, C. Poulain, A. Dalmieres, and E. B. Hamida, "Hybrid cryptographic protocol for secure vehicle data sharing over a consortium blockchain," in *To appear in Proceedings of IEEE Blockchain 2018*, 2018.

[15] H. Zhang, C. Jin, and H. Cui, "A method to predict the performance and storage of executing contract for ethereum consortium-blockchain," in *International Conference on Blockchain*. Springer, 2018, pp. 63–74.

[16] I.-C. Lin and T.-C. Liao, "A survey of blockchain security issues and challenges." *IJ Network Security*, vol. 19, no. 5, pp. 653–659, 2017.

[17] B. Qin, J. Huang, Q. Wang, X. Luo, B. Liang, and W. Shi, "Cecoin: A decentralized pki mitigating mitm attacks," *Future Generation Computer Systems*, 2017.

[18] M. Wang, Q. Wu, B. Qin, Q. Wang, J. Liu, and Z. Guan, "Lightweight and manageable digital evidence preservation system on bitcoin," *Journal of Computer Science and Technology*, vol. 33, no. 3, pp. 568–586, 2018.

[19] D. L. Chaum, "Untraceable electronic mail, return addresses, and digital pseudonyms," *Communications of the ACM*, vol. 24, no. 2, pp. 84–90, 1981.

[20] S. Noether, A. Mackenzie *et al.*, "Ring confidential transactions," *Ledger*, vol. 1, pp. 1–18, 2016.

[21] S.-F. Sun, M. H. Au, J. K. Liu, and T. H. Yuen, "Ringct 2.0: A compact accumulator-based (linkable ring signature) protocol for blockchain cryptocurrency monero," in *European Symposium on Research in Computer Security*. Springer, 2017, pp. 456–474.

[22] Q. Wang, B. Qin, J. Hu, and F. Xiao, "Preserving transaction privacy in bitcoin," *Future Generation Computer Systems*, 2017.

[23] E. B. Sasson, A. Chiesa, C. Garman, M. Green, I. Miers, E. Tromer, and M. Virza, "Zerocash: Decentralized anonymous payments from bitcoin," in *2014 IEEE Symposium on Security and Privacy*. IEEE, 2014, pp. 459–474.

Blockchain Enabled Distributed Storage and Sharing of Personal Data Assets

Jānis Grabis
Management Information Technology
Riga Technical University
Riga, Latvia
grabis@rtu.lv

Vlado Stankovski
Department of Computer Science
University of Ljubljana
Ljubljana, Slovenia
Vlado.Stankovski@fri.uni-lj.si

Roberts Zariņš
Management Information Technology
Riga Technical University
Riga, Latvia
roberts.zarins@gmail.com

***Abstract*—Personal data are important information assets. Data sharing has potential of creating value to data owners as well as causing security and privacy concerns. Distributed storage solutions have emerged as an approach adhering to the Privacy-by-Design principles and in combination with blockchain technologies enable data and value exchange within communities of Internet users. The paper elaborates an approach for efficient distributed data storage and sharing, where access control is provided using the blockchain technologies and data searching and retrieval are facilitated using a knowledge base. A conceptual model and data management processes are elaborated and a prototype is developed. The prototype is used in experimental studies to compare data storage usage and data retrieval speed for the proposed approach and on-chain storage.**

Keywords—personal data, privacy-by-design, distributed storage, blockchain, smart contracts

I. INTRODUCTION

Internet and World Wide Web have enabled massive distribution and sharing of data including personal data. Data sharing has potential of creating value to their consumers as well as their owners. However, Internet users have realized that unlimited sharing of the personal data could lead to security breaches. The personal information assets can be managed in a centralized or decentralized manner. In the former case access to information is often controlled by service providers and the individuals are not completely confident about the destiny of their data. Additionally, the individuals might have limited opportunities for benefiting from sharing their personal data. Decentralized and distributed storage solutions are designed without a single point of control [1]. Similarly, blockchains and smart contracts have emerged as technologies enabling information and value exchange in distributed environments [2]. Moreover, distributed cloud storage solutions should allow the individuals to deal with large volumes, velocity and veracity of the data [3]. In this respect, it is paramount to achieve high Quality of Service for distributed cloud storage operation. Unfortunately, blockchains have limitations concerning both storage volume and information processing speed.

The objective of this paper is to elaborate a method for efficient distributed storage and sharing of personal data assets within a community of users. The individuals should have a complete control over the way personal data are used and opportunity to benefit from sharing these data. The personal data are stored in a distributed storage and the individuals may choose which storage to use. The access to these data is controlled using a blockchain and smart contracts define access conditions. To improve data retrieval efficiency a concept of knowledge base is used. The knowledge base defines communities of the users and facilities information search operations. The privacy by design principles [4] are followed to ensure that the distributed personal data storage and sharing solution referred as to MyDataExperience adheres to users' privacy and information sharing needs.

The conceptual model of MyDataExperience as well as algorithms for data storage, sharing and searching are elaborated. A prototype of the solutions is implemented and used to evaluate data management efficiency. The Ethereum blockchain and InterPlanetary File System are used in the implementation. The evaluation objective is to compare the proposed solution with the on-chain storage of personal data according to the data storage usage and retrieval speed.

The main research contributions are:

- Blockchain controlled access to distributed personal data according to conditions specified in smart contracts;
- Development of the knowledge base as a mediator for efficient querying distributed personal data in combination with the blockchain technology to control access of personal data and to meet high Quality of Service requirements.

The rest of the paper is organized as follows. Section II analyzes suitability of the proposed solution from the perspective of the Privacy by Design principles and reviews related work. Section III describes design and implementation of the MyDataExperience personal data storage. Experimental evaluation of the proposed solution is reported in Section IV. Section V concludes.

II. BACKGROUND

The adoption of the distributed storage controlled using blockchain technologies is driven by the need to adhere to the

2473-3490/20/$31.00 ©2020 IEEE
DOI 10.1109/ICDEW49219.2020.00-13

Authorized licensed use limited to: University of Newcastle. Downloaded on June 02,2020 at 06:21:58 UTC from IEEE Xplore. Restrictions apply.

Privacy-by-design principles. The knowledge based information retrieval is introduced to address the efficiency concerns.

A. Privacy-by-design and blockchains

The privacy-by-design principles were to ensure that applications are developed having users' privacy and security needs as the most significant priority. These principles are [4]:

1. Proactive not Reactive and Preventative not Remedial
2. Privacy as the Default Setting
3. Privacy Embedded into Design
4. Full Functionality — Positive-Sum, not Zero-Sum
5. End-to-End Security — Full Lifecycle Protection
6. Visibility and Transparency — Keep it Open
7. Respect for User Privacy — Keep it User-Centric

If centralized and decentralized personal data storage solutions are compared then these principles are better met by decentralized and distributed storage solutions (Table I).

TABLE I. EVALUATION OF CENTRALIZED AND DECENTRALIZED STORAGE SOLUTIONS ACCORDING TO PRIVACY-BY-DESIGN

Principle	Centralized	Decentralized
Proactive not Reactive	Permissions are often imposed upon owner's request	Owner specifies permissions upfront and ownership is validated
Privacy as the Default Setting	Service provider defines permissions to select form	User specifies permissions
Privacy Embedded into Design	Assumption of trustworthiness; main emphasis on performance	Assumption of trustless environment
Full Functionality	Service provider benefits	Owner benefits
End-to-End Security	Points-of-failure	Security of assets, meta-data and tokens
Visibility and Transparency	Limited trace	Full trace
Respect for User Privacy	Courtesy of service provider	User requirements first

However, the decentralized and distributed approach causes its own challenges concerning trust and value exchange which can be addressed using blockchain technologies. Wust and Gervais [5] proposed an algorithm to determine suitability of blockchains for particular needs. The evaluation shows that distributed personal data storage has following features:

- Needs to store states;
- Have multiple writers;
- Does not have always online trusted third party;
- All writers are not known;
- All writers are not trusted;
- Public verifiability is required.

As a result the permissionless blockchain is suitable for distributed personal data storage and community based sharing. However, the blockchain alone does not address all aspects important distributed personal data storage and full functionality cannot be achieved. Smart contracts can be used for these purposes. They key features important to distributed personal data storage and sharing possibility to exchange assets, dispute-less and self-enforcement.

B. Related Work

Zyskin et al. [6] published one of pioneering woks on application of blockchain in personal data management. Similarly, blockchains can be used to control access to personal data [7,8]. However, there are couple of limitations to use blockchains for personal data management at scale. Singh and Lee [9] analyze non-functional characteristics and conclude that there are issues concerning flexibility, performance and cost efficiency. Kosteka et al. [10] add to this list latency, security and usability among others. Pongnumkul et al. [11] estimate that transaction execution latency increases exponentially with the increasing number of transactions and is several minutes long what is not acceptable for personal data management purposes. Ibanez et al. [12] analyze annotating of bitcoin transactions and indicate that limited memory is significant shortcoming. If blockchains are used for personal information then size of the blockchain becomes even more significant issue what will be resolved by using the blockchain for controlling access rather than for storing personal data themselves. The similar approach has been suggested in [13]. Blockchains enable development of GDPR compliant data exchange solutions [14]. Recognizing shortcoming of data retrieval from blockchains, recently, there has been and increasing interest in query languages for blockchains [15]. They add ability to perform range queries on top-k queries on the blockchain. Bartoletti et al. [16] develop analytical queries for analyzing cryptocurrency transactions.

Recently, the blockchain technology has been used jointly with distributed storage to provide unlimited storage capacity by combining on-chain and off-chain capabilities [17, 18]. These papers share similarity with the MyDataExperience approach, which mainly differs by introducing the knowledge base concept to facilitate data management processes.

III. DESIGN

A. Overview

The overview of the MyDataExperience personal data management solution is given in Fig. 1. A user who owns an asset (e.g., a picture) uses the solution to comprehend potential threats and benefits of sharing this asset with related entities (e.g., friends, companies willing to consume data). The asset is stored in a distributed storage, indexed in a knowledge base and a smart contract governing the asset usage is written into the blockchain. The asset is also indexed in knowledge bases owned by the related entities. The knowledge base contains the list of user and their addresses, access rights (relations among users and files), relations among the users (e.g. friendship relationships) and file usage data. The related entities or friends query the knowledge base to search for available assets. If conditions defined in the smart contract are validated, the blockchain provides access to the assets and the owner is compensated for data sharing using tokens.

Authorized licensed use limited to: University of Newcastle. Downloaded on June 02,2020 at 06:21:58 UTC from IEEE Xplore. Restrictions apply.

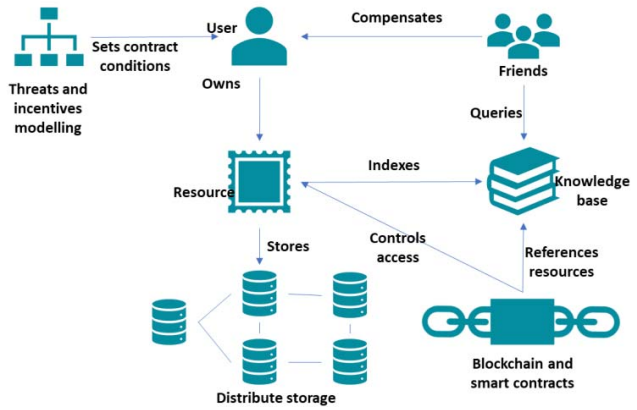


Fig. 1. Interactions among key elements of the `MyDataExperience` personal data management solution.

B. Data Management Processes

The MyDataExperience solution allows users to perform these core data management processes:

- Store personal data assets in distributed storage;
- Share personal data assets taking into account privacy preferences;
- Search of personal assets shared by related entities.

To store personal data assets, a user interacts with the MyDataExperience front-end and data files representing the assets are submitted to the knowledge base (Fig. 2). The knowledge based identifies appropriate storage (that could include selection of the most suitable storage according the user's preferences) and establishes a connection to the distributed storage service. The assets are stored in the distributed storage and a reference to the asset is obtained. This reference is stored in the blockchain, which requires a payment for the transaction. Both public and private blockchains could be used. The private blockchain could be used to reduce costs or to create a personal data sharing economy. Once the payment is made, the assets' storage is confirmed.

The sharing process allows users to share their personal data assets with friends as well as other parties subject to data access conditions. The friends are users with whom the data owner has established relations in the data sharing community and data access is governed for the group as a whole rather than on the case to case basis.

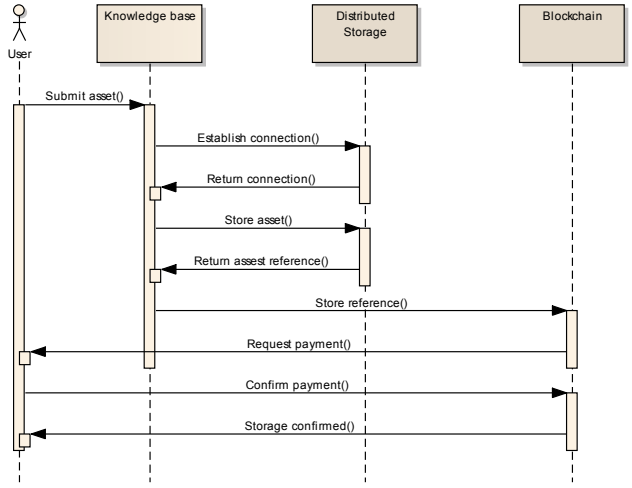


Fig. 2. Storing process.

In concert with her security preferences and expected sharing benefits, a data owner defines a smart contract describing conditions for sharing of resources. The smart contract is deployed in the blockchain (Fig. 3). A new blockchain transaction is executed with a change of the access rights. A friend can request access to the known asset (i.e., with a known reference without searching). The request is evaluated according to the smart contract stored in the blockchain. If it is a valid request, a new transaction including payment is created and the friend receives the access to the requested asset. The asset itself is retrieved from the distributed storage.

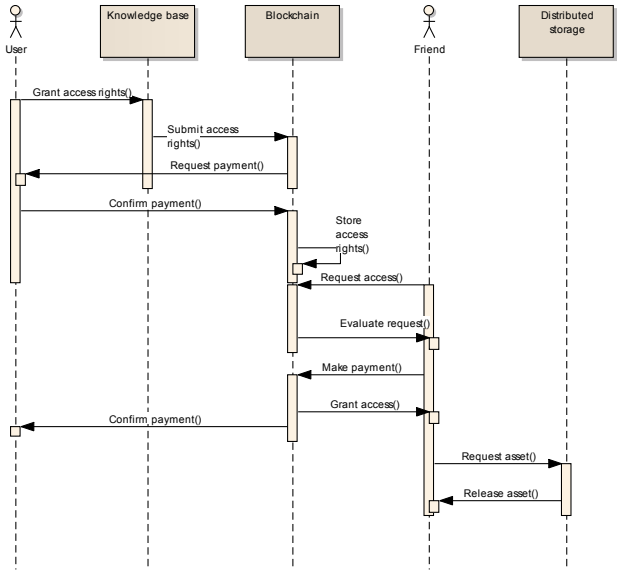


Fig. 3. Sharing process.

The search process (Fig. 4) is used by data consumers to find available personal data assets. The search can be performed by friends as well as other parties and using various search criteria such as assets' name, type and owner. Assets are searched in the knowledge base according to the search criteria

Authorized licensed use limited to: University of Newcastle. Downloaded on June 02,2020 at 06:21:58 UTC from IEEE Xplore. Restrictions apply.

provided. The access rights to the assets are validated against the blockchain. The knowledge base returns to the user a list of the relevant and accessible assets. The user indicates the assets she would like to retrieve what triggers delivery of these assets from the distributed storage.

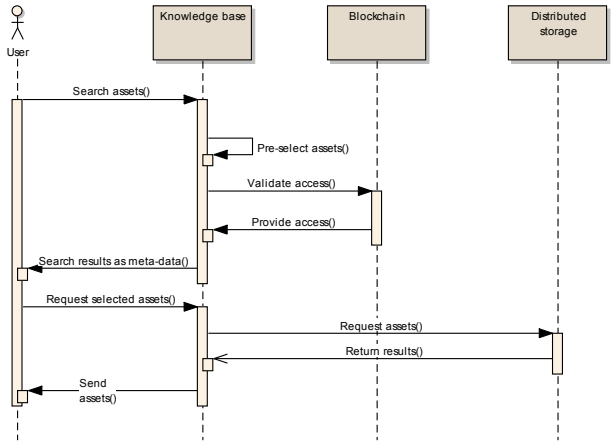


Fig. 4. Searching process.

The user searches the knowledge base for the resources in her neighborhood (e.g., shared by friends) and depth of the neighborhood can be controlled. The accesses right and sharing conditions for the requested resources are evaluated by retrieving this information from the blockchain for the pre-selected resources. This information is returned to the user who selects resources fitting her needs and means. This process could be fully automated. The MyDataExperience querying service mediates retrieval of the selected resources from the distributed knowledge and execution of the smart contracts is triggered on the blockchain.

C. Technology

The key components of the MyDataExperience prototype are front-end supporting user interactions, distributed storage, blockchain with smart contracts and knowledge base. The React [1] framework is used to develop the front-end as a standard web application although different types of front-ends could be used to access MyDataExperience functionality using API. Ethereum[2] blockchain is employed and the Ropsten[3] test network in particular, which is accessed with the help of Infura[4] API. Transactions are handled using Ethereum Wallet. A web application was also used to provide the knowledge base functionality. Smart contracts were developed using the Solidity programming language. InterPlanetary File System[5] (IPFS) provides distributed storage facilities.

The knowledge base currently is also implemented on-chain. This way it is also decentralized. However, that would reduce efficiency of the solution if there are many users and a large number of files.

[1] https://reactjs.org/
[2] https://ethereum.org/
[3] https://ropsten.etherscan.io/
[4] https://infura.io/
[5] https://ipfs.io/

Fig. 5 illustrates the results of data storage in the blockchain. The blockchain stores IPFS reference of the personal asset, its name (e.g., SSO.PDF) and the user reference. The assignment of the full access rights to the asset's owner is shown in Fig. 6, where the first value indicates the owner and the following array indicates the full access rights.

```
truffle(development)> contract.get().then((val) => console.log(val))
QmfBF4a7itUVLYcDKMZDhYnVhw3m5Rh1XnzCbg8SnzEyRS

truffle(development)> contract.ipfshashes(1).then((val) => console.log(val))
TypeError: contract.ipfshashes is not a function
truffle(development)> contract.ipfsHashes(1).then((val) => console.log(val))
[ BigNumber { s: 1, e: 0, c: [ 1 ] },
  'QmadvrtnpyVGKUBMNyX8kepwxsCy4wPPaDoqJY3UdjYRk7',
  'SSO.PDF',
  '0x0dD18429d91e4A91cE3Ec7b0dB89b6f8CE2D756B' ]

truffle(development)> contract.ipfsHashes(2).then((val) => console.log(val))
[ BigNumber { s: 1, e: 0, c: [ 2 ] },
  'Qmf8F4a7itUVLYcDKMZDhYnVhw3m5Rh1XnzCbg8SnzEyRS',
  'tesla-roadster-red-electricity-cars-luxury.jpg',
  '0x0dD18429d91e4A91cE3Ec7b0dB89b6f8CE2D756B' ]

truffle(development)>
```

Fig. 5. Sample files referenced in the blockchain.

```
truffle(development)> contract.getUser("0x0dD18429d91e4A91cE3Ec7b0dB89b6f8CE2D756B
[ '0x0dD18429d91e4A91cE3Ec7b0dB89b6f8CE2D756B',
  [ '0x0000000000000000000000000000000000000000',
    '0xd268ebc987f340616dac0c4e0cb5e42d1911e352' ],
  BigNumber { s: 1, e: 0, c: [ 0 ] } ]

truffle(development)>
```

Fig. 6. Access right assigned to the owner of the resource.

The MyDataExperience front-end provides functions for listing files, file upload, management of the community (i.e., friends of the assets' owner) and searching files (Fig. 7). The figure shows that assets are added to the distributed storage along with user identification. The MetaMask[6] interface from the browser to Ethereum Wallet shows the payment for storage of the asset.

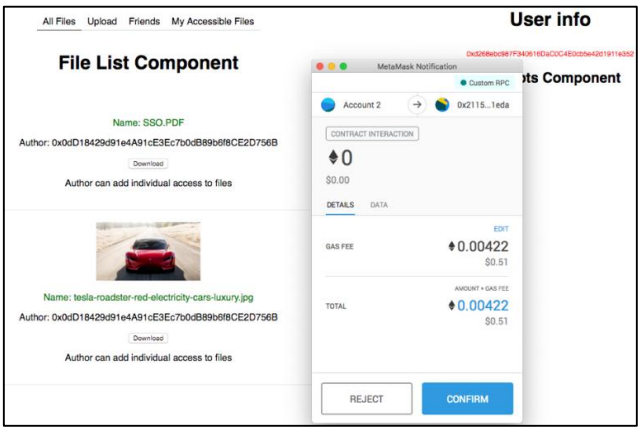


Fig. 7. The front-end of the MyDataExperience prototype.

IV. Experimental

Experimental studies are conducted to evaluate the proposed personal data storage and sharing solution. The MyDataExperience is compared with plain storage of data

[6] https://metamask.io/

Authorized licensed use limited to: University of Newcastle. Downloaded on June 02,2020 at 06:21:58 UTC from IEEE Xplore. Restrictions apply.

assets on-chain. The on-chain storage implies that files are stored directly in the blockchain rather in the distributed storage. The comparison is done according to the storage size and the data assets search efficiency.

A. *Storage size*

The first set of experiments evaluates the storage utilization what is measured by the required storage size and cost of storage transactions in the Ethereum blockchain.

The cost of storage transactions in the Ethereum network is depends on the quantity of gas required to perform these transactions. The quantity of gas is determined by data volume and complexity of processing [26]. In the case of on-chain storage, the files are converted in hexadecimal format and stored in the blockchain. Three different files are used in the experiments (Table II). The files differ by their size and type.

TABLE II. FILES USED IN THE EXPERIMENTAL STUDIES

File number	File type	Size, b
1	Text	1024
2	Image	3219
3	Image	4720

During the experiment these three files were uploaded in the directly in the blockchain and MyDataExperience data storage. Table III shows the results of the experiment including:

- The size of the hexadecimal data string representing the file and stored in the blockchain in bytes;
- The total size of data stored in the blockchain including the file and access rights;
- Ethereum gas consumption to execute the transaction;
- Transaction costs in the *Ether* crypto-currency.

The results are independent of the file size in the case of the MyDataExperience solution, while the blockchain size and transaction cost increases linearly depending on the file size. This is illustrated in Fig. 8 showing the relative storage requirements between the on-chain and MyDataExperience cases. Obviously, the on-chain storage is significantly more expensive and more importantly it would lead to rapid increase of the blockchain size making it unusable.

TABLE III. THE RESULTS OF DATA STORAGE USAGE EXPERIMENTS

File number	Hex file size, b	Total size, b	*Gas*, unit	Cost, Eth	BC size increase, b
On-chain					
1	2048	2436	1622719	0,041	2436
2	6438	6852	4691848	0,117	9288
3	9440	9828	6762981	0,160	19116
MyDataExperience					
1	46	356	215089	0,005	356
2	46	356	215089	0,005	712
3	46	356	215089	0,005	1068

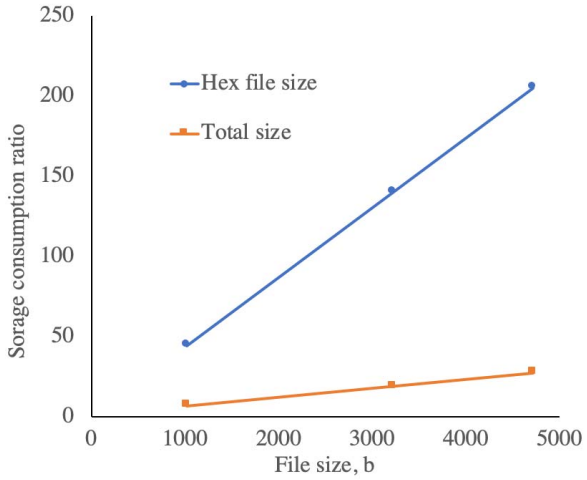


Fig. 8. Storage consumption ratio between the on-chain and MyDataExperience storage depending on the file size.

B. *File search time*

The search time is also evaluated. Two search approaches are compared:

- On-chain retrieval – files are searched in the blockchain without the help of the knowledge base;
- MyDataExperience – files are searched using meta-data in the knowledge base and then retrieved from the blockchain.

There are several types of users in the MyDataExperience data sharing community. These types are distinguished from the data assets owner perspective. That includes: 1) data owner; 2) data owner's friend; 3) unrelated user who shares data with the owner; 4) unrelated user. The following users having the aforementioned types are created:

- 0xa7DefDF3BE5B556B0d7f3fB3de5aF05a4f6E1DF2 – owner;
- 0xd335B51d960061cC1FA145e6d4a1Fcf1aB55677C – first friend of the owner;
- 0xFAa18b9a1348eF8f9D66C77e1B19eD43D7912D3c – second friend of the owner;
- 0xA345c16b69bC7817FEDE586e460acbD7EC9aDAd5 – unrelated user who shares data with the owner;
- 0x4CDFF8CdeDeDe2b4117Ff0e416C44566dbbF3b97 – unrelated user.

Relations among these users are stored in the knowledge base. At the same time twelve different files were stored. The appropriate references and access rights were written in the blockchain. The file ownership referenced in the blockchain is shown in Table IV. Two files are uploaded by the owner herself. Six files are uploaded by the owner's friends (files 3 to 8). Three files are shared directly with the owner without establishing friendship. One file is uploaded by an independent user not related to the owner.

Authorized licensed use limited to: University of Newcastle. Downloaded on June 02,2020 at 06:21:58 UTC from IEEE Xplore. Restrictions apply.

TABLE IV. FILE OWNERSHIP

Nr	File	Owner
1	Object.json	0xa7DefDF3BE5B556B0d7f3fB3de5aF05a4f6E1DF2
2	SUSe.pdf	
3	CV.docx	0xd335B51d960061cC1FA145e6d4a1Fcf1aB55677C
4	Twitter.jpeg	
5	Blockchain_Whitepaper.pdf	
6	Bird.jpeg	0xFAa18b9a1348eF8f9D66C77e1B19eD43D7912D3c
7	Vote.pdf	
8	BC_diagram.xml	
9	Kvalifikacijas_prasibas.docx	0xA345c16b69bC7817FEDE586e460acbD7EC9aDAd5
10	Google_app.png	
11	Object_2.json	
12	Location.jpg	0x4CDFF8CdeDeDe2b4117Ff0e416C44566dbbF3b97

These files are used to test several file search scenarios (Table V). The scenarios used different search conditions including file sharing conditions, friendship conditions, file type and file name.

TABLE V. FILE SEARCH SCENARIOS

Nr	Scenario
1	Find all files owned by friends
2	Find all files owned by the user herself
3	Find all files shared with the user
4	Find all PDF files owned by friends
5	Find all files titled "Blockchain_Whitepaper" and owned by friends
6	Find all files owned by friend 0xd335B51d960061cC1FA145e6d4a1Fcf1aB55677C

The scenarios are executed by searching files directly in the blockchain (files themselves are stored in the distributed storage) without using the knowledge based and by searching files in using the knowledge base and retrieval of the relevant files only. The search execution time including the data retrieval time is reported in Table VI and it is graphically illustrated in Fig. 9. The usage of the knowledge base in the MyDataExperience solution significantly reduces the search results. The search results depend only on the number of files returned and does not depend on the search criteria. In the case of on-chain search, the execution time depends on the search criteria because multiple conditions should be checked in the blockchain data.

TABLE VI. EXECUTION TIME OF THE FILE SEARCH SCENARIOS

Search type	Scenario	Execution time, s
On-chain	1	9
	2	10
	3	8
	4	14
	5	13
	6	19
MyDataExperience	1	3
	2	2
	3	3
	4	3
	5	3
	6	4

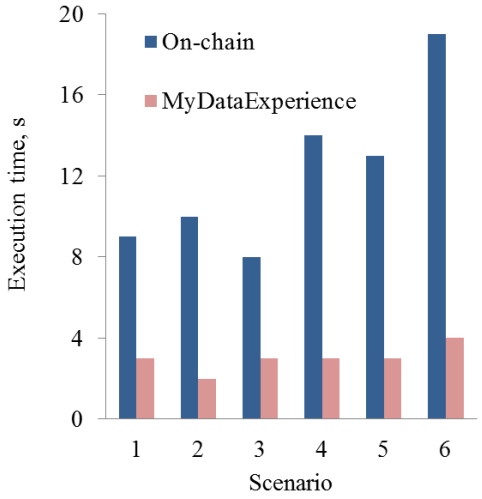


Fig. 9. Execution time of the File Search Scenarios.

V. CONCLUSION

The distributed personal data storage and sharing solution has been developed in the paper. It is show to adhere to the principles Privacy-by-design and provides its users a full complete control on the way personal data assets are stored and shared in the community of users. The knowledge base referencing the data assets is used to facilitate their search process.

The proposed solution has been compared with on-chain storage and search. It has been shown that the MyDataExperience solution requires less storage space and is less expensive than the on-chain storage. The storage space is not dependent on the size of files stored. Additionally, utilization of the knowledge base expedites the search process.

The experiments were performed using small number data assets what was sufficient to demonstrate differences between the on-chain and MyDataExperience solutions but not sufficient to evaluate performance of the MyDataExperience solutions in high load situations. That will be addressed in further studies. The cost of maintaining the knowledge base in the distributed mode is also not investigated in this paper and is subject of further research. The knowledge base is currently implemented on-chain what would become a bottleneck as the size of the network grows. The knowledge base itself could be redeveloped by using the MyDataExperience approach by combining on-chain and off-chain solutions.

It is also envisioned that the solution should be made more user friendly. In particular, definition of smart contracts will be done in a model-driven manner. That will allow any user to specify her privacy considerations and willingness to benefit from personal data sharing. These preferences would be used to create smart contracts balancing privacy concerns and potential monetary gains from data sharing.

Authorized licensed use limited to: University of Newcastle. Downloaded on June 02,2020 at 06:21:58 UTC from IEEE Xplore. Restrictions apply.

REFERENCES

[1] S. Sakr, A. Liu, D.M. Batista, and M. Alomari, "A survey of large scale data management approaches in cloud environments," IEEE Communications Surveys and Tutorials, vol. 13, no. 3, 2011, pp. 311-336.

[2] N. Fabiano, "Blockchain and data protection: The value of personal data," IMCIC 2018 - 9th International Multi-Conference on Complexity, Informatics and Cybernetics, Proceedings, 2018, pp. 112-115.

[3] V. Stankovski and R. Prodan, "Guest Editors' Introduction: Special Issue on Storage for the Big Data Era," Journal of Grid Computing, 16, 2018, pp. 161-163.

[4] A. Cavoukian, and M. Dixon, "Privacy and Security by Design: An Enterprise Architecture Approach," 2013, https://www.ipc.on.ca/wp-content/uploads/Resources/pbd-privacy-and-security-by-design-oracle.pdf

[5] K. Wust and A. Gervais, "Do you need a Blockchain?" Crypto Valley Conference on Blockchain Technology (CVCBT), 2018, pp. 45-54.

[6] G. Zyskind, Nathan, O. and Pentland, A.S., "Decentralizing privacy: Using blockchain to protect personal data", Proceedings - 2015 IEEE Security and Privacy Workshops, SPW 2015, pp. 180-184.

[7] A. Kalra, Hasnain, S.S., Bodorik, P. and Jutla, D., "Access control mechanism using ethereum blockchain," 26th International Conference on Software Engineering and Data Engineering, SEDE 2017, pp. 107.

[8] D. Di Francesco Maesa, Mori, P. and Ricci, L. "Blockchain based access control", IFIP International Conference on Distributed Applications and Interoperable Systems, 2017, pp. 206-220.

[9] I. Singh, and Lee, S.-. "Comparative requirements analysis for the feasibility of blockchain for secure cloud," Communications in Computer and Information Science. Volume 809, 2018, pp. 57-72

[10] B. Koteska, Karafiloski, E. and Mishev, A., "Blockchain implementation quality challenges: A literature review", CEUR Workshop Proceedings, 2017.

[11] S. Pongnumkul, Siripanpornchana, C. and Thajchayapong, S. "Performance analysis of private blockchain platforms in varying workloads," 2017 26th International Conference on Computer Communications and Networks, ICCCN 2017.

[12] L.D. Ibanez, Fryer, H, et al. "Attaching Semantic Metadata to Cryptocurrency Transactions," Proceedings of the Workshop on Decentralizing the Semantic Web 2017 co-located with 16th International Semantic Web Conference, 2017, pp. 1-18.

[13] X. Xu, C., Pautasso, L. Zhu et al. "The Blockchain as a Software Connector". Software Architecture (WICSA), 2016 13th Working IEEE/IFIP Conference, 2016, pp. 182-191 .

[14] R. Neisse, Steri, G. and Nai-Fovino, I., "A blockchain-based approach for data accountability and provenance tracking," ACM International Conference Proceeding Series, 2017, pp. 1-10.

[15] Y. Li, Zheng K. et al., "EtherQL: A Query Layer for Blockchain System," International Conference on Database Systems for Advanced Applications, DASFAA 2017: Database Systems for Advanced Applications, 2017, pp 556-567.

[16] M. Bartoletti, Lande, S., Pompianu, L. and Bracciali, A. "A general framework for blockchain analytics," SERIAL 2017 - 1st Workshop on Scalable and Resilient Infrastructures for Distributed Ledgers, Colocated with ACM/IFIP/USENIX Middleware 2017 Conference, 2017, pp. 1-6.

[17] M. Alessi, A. Camillò, E. Giangreco, M. Matera, S. Pino, and D. Storelli, "A decentralized personal data store based on ethereum: Towards GDPR compliance," Journal of Communications Software and Systems, vol. 15, no. 2, 2019, pp. 79-88.

[18] S. Wang, Y. Zhang, and Y. Zhang, "A blockchain-based framework for data sharing with fine-grained access control in decentralized storage systems," IEEE Access, vol. 6, 2018, pp. 38437-38450.

Authorized licensed use limited to: University of Newcastle. Downloaded on June 02,2020 at 06:21:58 UTC from IEEE Xplore. Restrictions apply.

Blockchain-based Ownership Management for Medical IoT (MIoT) Devices

M. Alblooshi, K. Salah, Y. Alhammadi
Department of Electrical and Computer Engineering
Khalifa University, UAE
{mansoor.albloushi, khaled.salah, yousof.alhammadi}@ku.ac.ae

***Abstract*—Recently, blockchain has been foreseen by industry and research community as a transformational and disruptive technology that is poised to play a major role in transforming the way we transact, trade, bank, track and register assets, and do business. Blockchain can potentially be used to provide trusted authenticity, origin, and ownership for IoT devices, in a way that is secure and decentralized without the involvement of a Trusted Third Parties (TTP) or centralized authorities and services. A TTP can be a subject to a single point of failure, and it can be disrupted, compromised or hacked. A TTP can potentially misbehave and become corrupt in the future, even if it is trustworthy now. This paper shows how Ethereum blockchain (with the powerful feature of programmability through smart contracts) can provide a trusted ownership management for medical IoT (MIoT) devices. Tracking the true ownership of MIoT devices is of a paramount importance as using counterfeited MIoT devices can be life-threatening to patients. The paper presents a general framework and solution to manage and trace back the true origin of ownership for an MIoT; whereby an MIoT device can be owned by multiple parties during its life cycle. The paper presents a detailed overview of our proposed system architecture, design, entity relationship, and interactions among all involved parties. The source code of the Ethereum smart contracts is made publicly available. Key aspects related to the algorithms, interactions, implementation and testing are discussed in the paper. Furthermore, we provide security analysis of our proposed solution in terms of satisfying the general security goals and also resiliency against popular attacks as those of DDoS, eavesdropping and replay attacks.**

***Keywords*—IoT, IoT Security, Blockchain, Authentication, Trust, IoT Device Management**

I. INTRODUCTION

During the past few years, Internet of Things (IoT) had experienced wide-spread development and had gained fast-growing popularity worldwide [1]. Nowadays, IoT is widely used in smart cars, smart homes, wearables, smart cities and healthcare. The market of IoT devices is growing quickly, covering a wide range of areas, such as traffic surveillance, wildlife monitoring, and gas leakage detection.

Today's massive adoption and deployment of IoT in many fields and industries introduce many key challenges which include security and privacy issues and management of IoT devices, as well as the data being generated from these IoT devices [2]. Unauthorized surveillance and uncontrolled data use are some of the key privacy challenges facing IoT devices. Management of IoT devices and their ownership, in a way that is authentic and trusted, is a key challenge these days, especially if the IoT device is sold and re-sold by multiple parties. There is a need to know and trace back the origin of the IoT device, and determine if it is original or counterfeited, specifically in healthcare industry. This is complicated by having multiple stakeholders, individuals, organizations, and third parties, being involved in IoT device ownership and management.

In the area of healthcare, authenticity and originality of medical IoT (MIoT) devices being used on patients are of paramount importance. MIoT devices can come today in many variety and forms of wearables, implantables, or injectables. A counterfeited MIoT device can be life threatening to patients. Therefore, it is important to have the ability to trace back the history of MIoT device in a credible and trusted manner with no centralized management or the use of a Trusted Third Party (TTP). Third parties and organizations take role in providing enhanced services for the individual and for the society at large [3]. Multiple solutions have been proposed in the literature to overcome these IoT challenges including privacy and ownership. Most of these solutions propose involvement of trusted third party acting as a centralized management. TTP can be a single point of failure, hacking, compromise, and potential of corruption. Therefore, a decentralized ownership management of MIoT devices becomes key.

In this paper, we show how the newly emerging technology of blockchain can play a major role in providing a decentralized trust and management of MIoT devices. The main contributions of this paper can be summarized as follows:

- We propose blockchain-approach using Ethereum smart contracts to manage the ownership of IoT devices in a decentralize manner with no trusted third party.
- We present a detailed overview of our proposed system architecture, design, entity relationship, and interactions between all parties interacting with the smart contracts which are uploaded on the Ethereum blockchain platform.
- We provide the full source code of the Ethereum smart contracts and discuss key aspects related to its algorithms, interactions, and logical flow. The full code is made avalable at Github[1]. In addition, we show how we implemented and tested the smart contract, and ascertain the correctness of the overall system functionality and behavior.

[1]https://github.com/MansoorUAE/IoTOwnership/blob/master/Manufacturer.sol

978-1-5386-6673-9/18/$31.00 ©2018 IEEE

- We analyze and discuss the security of our proposed solution in terms of satisfying the general security goals and also resiliency against popular attacks as those of DDoS, eavesdropping and replay attacks.

The rest of the paper is organized as follows. Section II gives a brief overview and background on blockchain, and Ethereum Smart Contracts. Section III highlights related work. Section IV details out the proposed solution and its implementation. Section V presents security analysis of our blockchain-based framework and solution. Finally, Section VI presents our conclusions and future work.

II. BACKGROUND

Our proposed solution is based on using blockchain and Ethereum smart contracts. This section provides a brief introduction on blockchain and Ethereum smart contracts.

A. Blockchain

Blockchain is a distributed database or ledger that can store records and data which can be globally accessed and shared among a network of independent parties [4] in a decentralized and trusted manner. Creation of immutable records is a key feature of blockchain. It gives the ability for any participant to manage the ledger securely without the need for a trusted intermediary or centralized trusted third party [4]. Removal of centralized authority while maintaining data integrity was the main motivation of blockchain [4]. Blockchain consists of a number of chained blocks, where each block contains a list of transactions recorded into a ledger, and they are chained by a hash, linking the new block to the previous one [4–6]. Any recorded data on blockchain is tamper resistant. Any new preformed record (block) that need to be added to an existing chain must be through a consensus mechanism, where special nodes (called miners) have to validate such transactions [4, 7]. The miners on blockchain network maintain their own replicated, shared, and synchronized database digital information [7]. All transactions on blockchain network are available to all of its users. It provides the ability of automating transactions, real time settlement with respect of providing protection against fraud [7]. Each user on the blockchain has the ability to read and write from the database. Each stored information and state of the database are validated, and then confirmed though cryptographic consensus mechanism. The added transaction is tamper resistant and it cannot be updated or deleted [7].

B. Ethereum Smart Contracts

The first type of blockchain network was Bitcoin. The second type is called Ethereum blockchain which has the same features and traits of the Bitcoin blocchain, but can be programmable using the notion of smart contracts. Smart contracts concept is the key feature of Ethereum blockchain. It provides users ability to design their own business code and programs to be executed by all miners. The execution outcome is validated by all mining nodes. This concept has been developed by Vitalik Buterin the founder of Ethereum in 2014 [8].

A smart contract is basically a program that validates a transaction along with immediate effect of the contract, without involvement of trusted third party [8]. Any deployed smart contract on a blockchain has a unique Ethereum public address. Users on the network access the smart contract using this address. In a smart contract, there are two types of functions: getters and setters. A getter function is used for retrieving information and state variables; whereas, a setter function is used for setting values to variables. Using qualifier called ”only” can restrict access to these functions. Smart contract provides user an ability of logging any changes on any concerned value by using events. Whenever the user executes a function, he/she would have to pay an amount of gas. The amount of gas depends on computational steps. The concept of smart contracts can be used in managing ownership of the IoT device among two parties as a decentralized manner.

III. RELATED WORK

This section briefly surveys and summarizes existing solutions in the literature addressing the IoT ownership problem. The solutions can be categorized as trusted third party ownership management, item level access control framework, and authentication and access control for the entire IoT device life cycle. First, the proposed ownership management system by [9] is based on a trusted third party (TTP) model. It covers both the ownership establishment and transfer modes as shown in Figures 1 and 2, as presented in [9]. In the figures, K denotes a secrete key and ID(p), ID(o), and ID(n) denote the identity of the IoT device, current owner, and new owner, respectively.

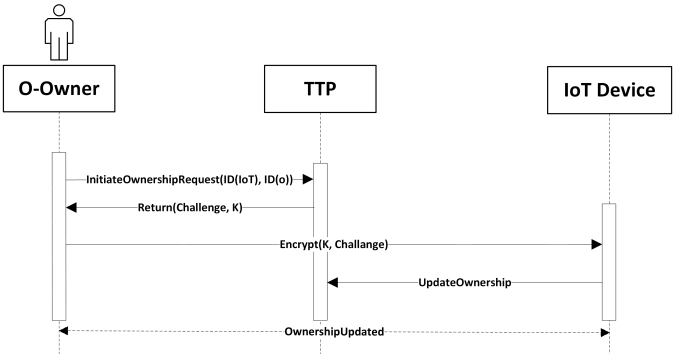


Fig. 1: Ownership creation model as proposed in [9]

According to [9], owner registration and the establishment of the security credentials with the particular IoT device is required. Moreover, TTP is responsible for generating ownership challenges as shown in Figure 1 and 2. Furthermore, the IoT device has a pre-installed secret shared with TTP and has a one-way function h() to generate the secret key for ownership. An ownership creation model is used when the IoT device ownership first is registered. The transfer procedure requires both the current owner and the new owner to be registered with the TTP as shown in Figures 1 and 2.

The authors in [10] argued that, in case of ownership

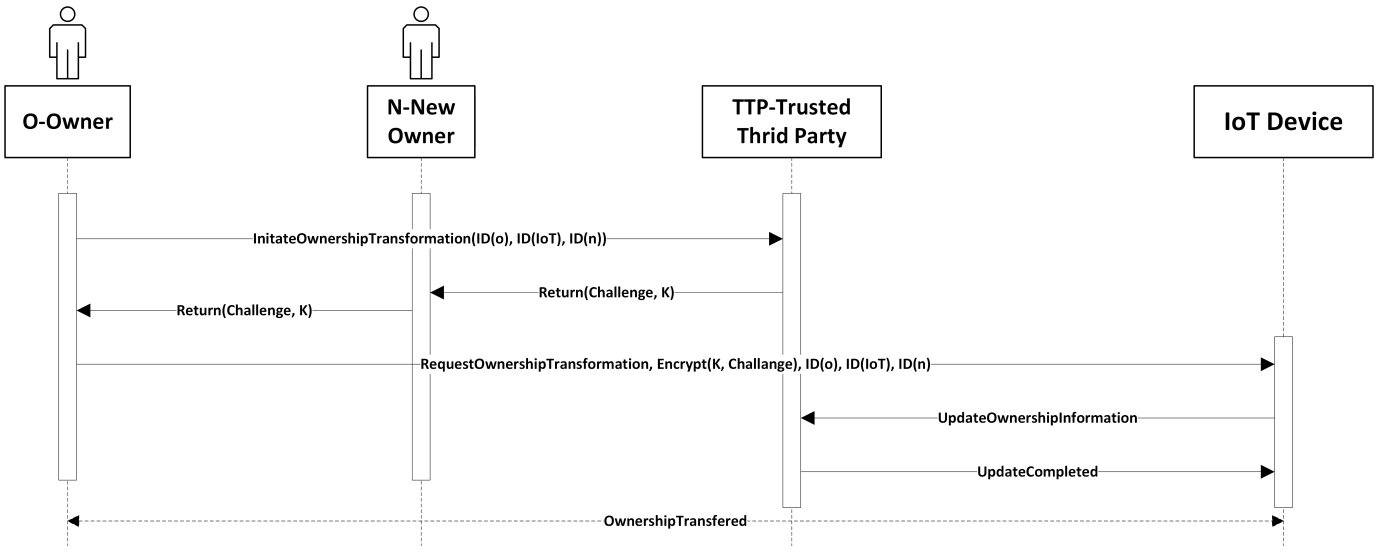


Fig. 2: Ownership transfer model as proposed in [9]

transfer for an IoT device, a trust should be established between both owners: current and new owner. This includes permissions and access control. A framework based on item level access control through mutual trust has been presented in [10] for IoT devices. In short, the newly manufactured device is given a key by a trusted third party. This key will be used to provide the current owner device permission control. For transfer of ownership, a token will be created by current owner of IoT device. This token is combined with device RFID identity to grant the new owner the permission control along with IoT device ownership.

The authors in [11] presented an IoT device life cycle, which is essentially divided into five stages, pre-deployment, ordering, deployment, functioning, and retirement. The cryptographic keys of the server are loaded into IoT device via the manufacturer at the pre-deployment stage. In the ordering stage, the new owner of the IoT device receives an IoT access PIN. The development stage establishes a trust relationship between the device and a TTP which handles the key management and the access control.

All of the previous solutions have clear limitations stemming from the use of centralized management entity or TTP for the ownership management and the key management. The solution has clear limitations. The solution depends on TTP (centralized management) which is subject to corruption, single point of failure, and being comprised or hacked. In addition, according to [9], the solution has performance issues. Also, the solution does not keep ownership history that is open and trusted.

IV. BLOCKCHAIN-BASED SYSTEM FOR DEVICE OWNERSHIP MANAGEMENT

This section gives a system overview and architectural design for solving the ownership problem for medical IoT devices (MIoTs) in a decentralized, trusted manner by using Ethereum smart contracts. Figure 3 illustrates the main participants of the system which include the medical IoT device, the potential owners of the device (patient), manufacturer, physicians, clinic and hospital. The system consists

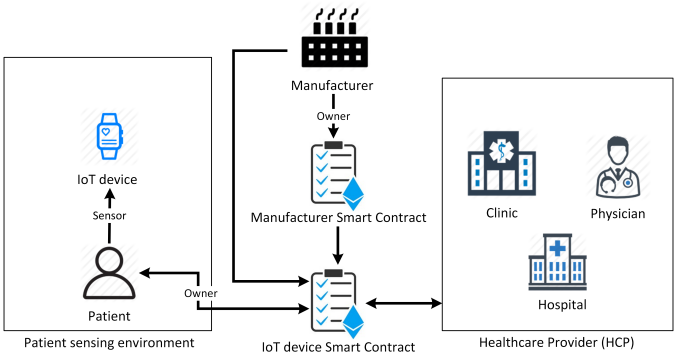


Fig. 3: System overview

of two smart contracts code, manufacturer and IoT device smart contract, which get deployed on Ethereum blockchain network. Manufacturer smart contract is used by manufacturer in order to deploy IoT device smart contract whenever IoT device is manufactured.The other smart contract gives the IoT device owner the ability to set rules and conditions for access and modifying the records pertaining to the MIoT device ownership records stored on the Ethereum blockchain network. The smart contract provides the owner with the mechanism for transferring the MIoT ownership without involving a TTP or centralized entity. In a way, any participant can interact with the contract to get, set, or modify its variables based on the rules that been set by the owner without involving a central authority, as the execution of the smart contract is carried out and the execution outcome is validated and agreed on by the thousands of miner nodes.

The following subsections discuss in more detail design aspects, smart contract logic and code, implementation and

testing of the overall system functionality.

A. Design Aspects

The proposed system consists primarily of five key components which all have Ethereum Addresses (EA): Patient, manufacturer, IoT device smart contract which gets created by the manufacturer, and the healthcare provider.

Patient. The owner of the IoT Device who can set the rules and conditions on the MIoT device smart contract for device management.

Manufacturer. The MIoT device manufacturer creates the original smart contract once the device is manufactured.

Smart Contract (SC). The smart contract holds all the code and logic for the rules to manage device ownership. The smart contract actually contains the getter, setter, and modifier functions to be accessed by participants. Moreover, it has the ability to log events. In our proposed solution, we use two smart contracts. One SC is used by the manufacturer for every MIoT device, and one SC is used by the owner.

Healthcare provider (HCP). HCP may include users who are interested in knowing, owning, or transferring the ownership of the MIoT device.

Figure 4 illustrates the entity relationship of the proposed system among the different participants in the context of Ethereum environment. As shown, each entity can have multiple relationship with the smart contract, and every IoT device is associated with a single owner and the owner can have ownership of multiple IoT Devices. Moreover, in order to deploy a successful system, each entity should be registered at Ethereum network with an Ethereum public address which is derived from the key pairs.

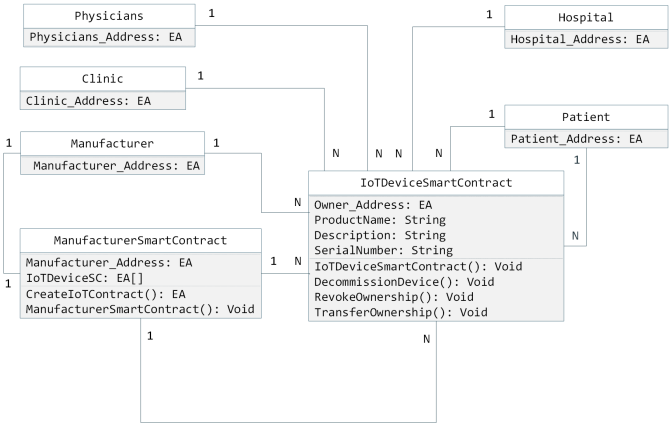


Fig. 4: Entity relationship of the proposed system

B. Logic Flow and Smart Contracts

This subsection illustrate briefly the logical flow of the developed smart contract. The manufacturer of the IoT device creates a contract through the constructor of the IoTCreation and specifies the owners public address along with the default IoT device details. Then the contract logs an event. Figure 5 shows the code for the IoTCreation constructor.

```
function IotCreation(address _owner, string _productName,
string _description, string _specification,
string _serialNumber){
// store IoT device owner record
owner = _owner;
productName = _productName;
description = _description;
specification.push(_specification);
serialNumber = _serialNumber;
// keep record of all IoT device owners
addNewOwner ('Device has been created', owner);
}
```

Fig. 5: IoT Smart Contract constructor

Figure 8 shows the sequence diagram of ownership transfer for the MIoT device, in which the new owner first needs to send his/her public address to the manufacturer. Once received, the manufacturer initiate the ownership transformation by calling TransferOwnership function, which records the new owner EA address and gives the new owner the full control of IoT device smart contract. Then, the new owner(Patient) can have control for modifying the records of the SC. The Solidity code for the ownership transfer is shown in Figure 6.

```
//function that allows owner to transfer device ownership
function ownershipTransfer(address newOwner) ifOwner {
owner = newOwner;
}
```

Fig. 6: Ownership transfer function code

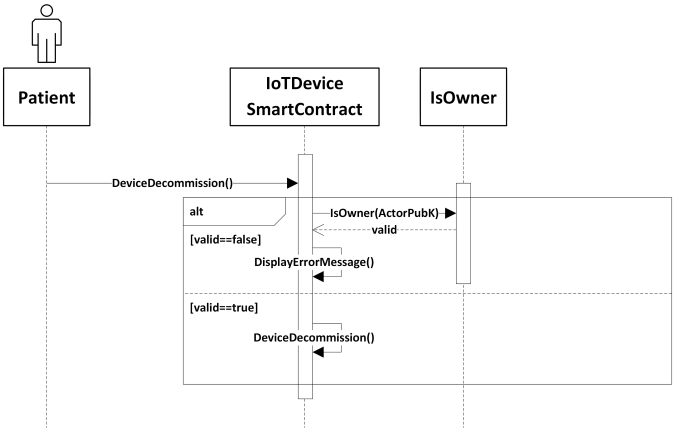


Fig. 7: Sequence diagram for decommissioning an MIoT device

If the Patient (or any owner) wishes to decommission the MIoT device, the Patient calls deviceDecommissioning function of the smart contract. The contract will verify the true ownership before proceeding with decommissioning using modifier only restriction as shown in Figure 7. The code for the deviceDecommissioning function is shown in Figure 9.

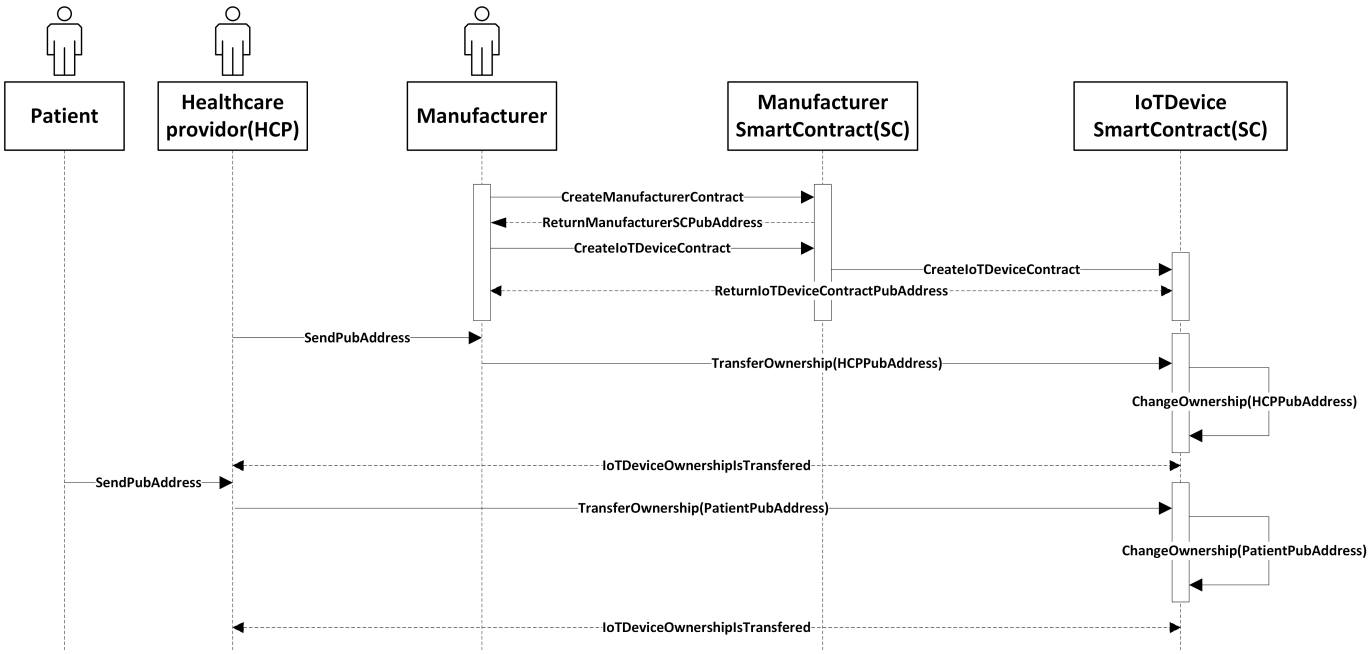


Fig. 8: Ownership management sequence diagram

```
//function controlled by the owner to decommission IoT device
function deviceDecommissioning() ifOwner{ //no revert back
    owner = 0;
    // keep record of IoT device decommissioning action
    addNewOwner ('Device has been decommissioned', owner);
}
```

Fig. 9: Function code for decommissioning MIoT device

status	0x1 Transaction mined and execution succeed
from	0xca35b7d915458ef540ade6068dfe2f44e8fa733c
to	IotCreation.ownershipTransfer(address) 0x692a70d2e424a56d2c6c27aa97d1a86395877b3a
gas	3000000 gas
transaction cost	28464 gas
execution cost	5784 gas
input	0xd3a417a2000000000000000000000000014723a09acff6d2a60dcdf7aa4aff308fddc160c
decoded input	{ "address newOwner": "0x14723A09ACff6D2A60DcdF7aA4AFf308FDDC160C" }

Fig. 10: Remix logs for successful ownership transfer

status	0x0 Transaction mined but execution failed
from	0xca35b7d915458ef540ade6068dfe2f44e8fa733c
to	IotCreation.ownershipTransfer(address) 0x692a70d2e424a56d2c6c27aa97d1a86395877b3a
gas	3000000 gas
transaction cost	23191 gas
execution cost	511 gas
input	0xd3a417a2000000000000000000000000014723a09acff6d2a60dcdf7aa4aff308fddc160c
decoded input	{ "address newOwner": "0x14723A09ACff6D2A60DcdF7aA4AFf308FDDC160C" }

Fig. 11: Remix logs for failed ownership transformation

C. Implementation and Testing

The smart contract has been developed, tested and implemented using the popular Ethereum Remix IDE[2]. This section covers key aspects related to testing smart contract functionality among different system participants. The testing includes all the aspects of the system such as ownership management and IoT device decommissioning. Ethereum Remix is a powerful IDE and tool that provides the ability of testing the smart contract on test network, prior to deploying it to the real network. The tool offers multiple Ethereum wallets, which allows for testing different scenarios with multiple parties. In order to verify and ensure the expected functionality and execution from all the functions (setters, getters, and restrict modifiers), various scenarios with multiple parties have been tested. In order to test the first scenario, an ownership transfer using Transfer ownership function has been initiated as shown in Figure 10. The ownership information of MIoT device has been successfully changed as shown in the logs. If the previous owner tries to initiate ownership transfer for the same device it will fail since he/she is no longer the owner of the device as as shown in Figure 11.

[2]https://remix.ethereum.org

V. SECURITY ANALYSIS

This section discusses and analyzes briefly the overall system security of our proposed solution. We address the fundamental objectives of security, and how our proposed system is resilient against known attacks and security vulnerabilities. Except for **confidentiality**, almost all security objectives of integrity, availability, and accountability are satisfied. Our solution uses Ethereum which is public or permissionless blockchain network in which all transactions are sent in the

clear so that the public minder nodes can verify and validate their content. Nevertheless, confidentiality can be achieved using private or permissioned blockchain Ethereum network, if confidentiality of records and transactions are required.

Non-repudiation is one of main security requirement for IoT device ownership management. This will verify the true message origination of participants on the blockchain network. This is achieved by providing any participant on blockchain network a 20-byte Ethereum Addresses (EA) along with asymmetric key pairs which will be used for signing messages and events. By design and as a feature of blockcahin transactions, every transaction or event issued to or from the smart contract among participants is cryptographically signed and verified, and therefore guarding against any attempt for **eavesdropping or Man-In-The-Middle attacks**. Moreover, every single transaction is embedded with nonce value and timestamps. This will provide the system with a protection mechanism against **replay attacks**.

Ownership information as well as interactions and execution outcomes are all stored, verified and validated by the tens of thousands of the public Ethereum miner nodes in a distributed and decentralized fashion. The outcome of the execution of the smart contract logic is agreed-on by consensus, and thus providing tamper-proof data storage with **high integrity, trust, resiliency, and availability** of records and data across the miner nodes. This is a powerful feature of blockchain platforms which is the resiliency against **Denial-of-Service(DoS)** attacks, as it would be practically impossible to DDoS attack all the globally distributed miner nodes, or alter their contents knowing such contents are hashed, and this contents are all duplicated in all miner nodes. Moreover, any attempt to tamper with or invalidate one block of data will invalidate all the blocks in the chain. By design, the records of IoT devices ownership and details of the IoT devices are all stored on Ethereum ledger in a decentralized manner. Such design is not subject to single point of failure, hacking, or compromise.

Lastly, we have performed **Smart contract vulnerability assessment** using Oyente tool[3] to ensure that our smart contract code presented in this paper is free of bugs and software vulnerabilities which can be exploited by attackers. For this, we used the popular Oyente which is an open source tool developed for scanning smart contracts. Oyente has the ability to detect and report if a smart contract code has known security vulnerabilities. As shown in Figure 12, no vulnerabilities have been detected in our smart contract.

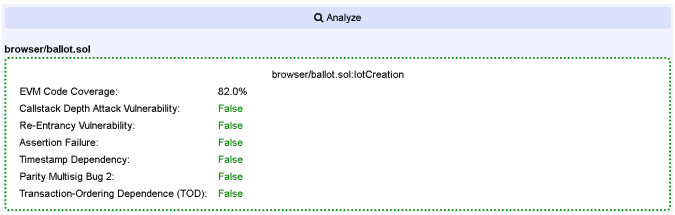


Fig. 12: Vulnerability assessment result

[3]https://oyente.melon.fund

VI. CONCLUSION

In this paper, we have presented a general framework and solution for ownership management and of MIoT devices to solve the counterfeiting problem. Our solution is based on using Ethereum blockchain smart contracts which offer a tamper-proof, trusted, secure, and credible tractability and tracking of origin history and true ownership in a way that is decentralized and resilient to common cybersecurity attacks. The paper provided security analysis of our approach, and discussed that except for the security goal of confidentiality, all security goals are satisfied. We also discussed and demonstrated that the smart contract code is free of bugs and security vulnerabilities. As a future work, we are in the process of implementing our solution on the real Ethereum network, and also developing individualized DApps (Distributed Applications) that will be utilized by the different system actors involved in managing the MIoT devices. Also we are investigating how blockchain can solve issues related to user authentication and access control for MIoT devices, without the involvement of trusted intermediaries or centralized entities.

REFERENCES

[1] Y. Zhang and J. Wen, "The IoT Electric Business Model: Using Blockchain Technology for the Internet of Things," *Peer-to-Peer Networking and Applications*, vol. 10, no. 4, pp. 983–994, 2017.

[2] X. Carron, R. Bosua, S. Maynard, and A. Ahmad, "The Internet of Things and Its Impact on Individual Privacy: An Australian Privacy Principle Perspective," *Computer Law & Security Review*, vol. 21, no. 1, pp. 4–15, 2016.

[3] D. McDermid, *Ethics in ICT: an Australian Perspective*. Pearson Higher Education AU, 2015.

[4] T. Laurence, *Blockchain for Dummies*. John Wiley & Sons, 2017.

[5] A. Bogner, M. Chanson, and A. Meeuw, "A Decentralised Sharing App Running a Smart Contract on the Ethereum Blockchain," in *Proceedings of the 6th International Conference on the Internet of Things*. ACM, 2016, pp. 177–178.

[6] R. Beck, J. S. Czepluch, N. Lollike, and S. Malone, "Blockchain-the Gateway to Trust-Free Cryptographic Transactions." in *ECIS*, 2016, p. ResearchPaper153.

[7] R. Lewis, J. McPartland, R. Ranjan *et al.*, "Blockchain and Financial Market Innovation," *Economic Perspectives*, no. 7, pp. 2–12, 2017.

[8] M. A. Khan and K. Salah, "IoT Security: Review, Blockchain Solutions, and Open Challenges," *Future Generation Computer Systems*, 2017.

[9] X. Leng, K. Mayes, and Y. Lien, "Ownership Management in the Context of the Internet of Things," in *Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC), 2014 International Conference on*. IEEE, 2014, pp. 150–153.

[10] Y. Xie and D. Wang, "An Item-Level Access Control Framework for Inter-System Security in the Internet of Things," in *Applied Mechanics and Materials*, vol. 548. Trans Tech Publ, 2014, pp. 1430–1432.

[11] A. L. M. Neto, A. L. Souza, I. Cunha, M. Nogueira, I. O. Nunes, L. Cotta, N. Gentille, A. A. Loureiro, D. F. Aranha, H. K. Patil *et al.*, "AoT: Authentication and Access Control for the Entire IoT Device Life-Cycle," in *Proceedings of the 14th ACM Conference on Embedded Network Sensor Systems CD-ROM*. ACM, 2016, pp. 1–15.

Digital Asset Management with Distributed Permission over Blockchain and Attribute-based Access Control

Yan Zhu[1*], Yao Qin[1], Zhiyuan Zhou[1], Xiaoxu Song[1], Guowei Liu[2], William Cheng-Chung Chu[3]
[1]School of Computer and Communication Engineering,
Univerisity of Science and Technology Beijing, Beijing 100083, China
[2]Beijing Municipal Commission of Economy and Informatization, Beijing 100101, China
[3]Department of Computer Science,Tunghai University Taichung, Taiwan
[*]Email: zhuyan@ustb.edu.cn

***Abstract*—Digital asset management (DAM) has increasing benefits in booming global Internet economy, but it is still a great challenge for providing an effective way to manage, store, ingest, organize and retrieve digital asset. To do it, we present a new digital asset management platform, called DAM-Chain, with Transaction-based Access Control (TBAC) which integrates the distribution ABAC model and the blockchain technology. In this platform, the ABAC provides flexible and diverse authorization mechanisms for digital asset escrowed into blockchain while the blockchain's transactions serve as verifiable and traceable medium of access request procedure. We also present four types of transactions to describe the TBAC access control procedure, and provide the algorithms of these transactions corresponding to subject registration, object escrowing and publication, access request and grant. By maximizing the strengths of both ABAC and blockchain, this platform can support flexible and diverse permission management, as well as verifiable and transparent access authorization process in an open decentralized environment.**

***Index Terms*—digital asset management; blockchain; access control; transaction; attribute-based**

I. Introduction

Digital asset management (DAM) provides a way to organise companys valuable files, called digital assets, in a way that makes them quick-to-find and easy to access. In an information economy, everyone can gain a huge advantage over their competitors by handling their digital assets more effectively than others. We all process digital assets, whether as documents, audible content, motion picture, and other relevant digital data that are currently in circulation, or files sent to us via email, all of which bring us wealth. Therefore, effective digital asset management has taken on increased importance.

A DAM system represents an intertwined structure incorporating both software and hardware and/or other services in order to manage, store, ingest, organize and retrieve digital assets. Moreover, it provides the unbroken maintenance of the ownership of a digitized object while permitting access to those who have obtained rights to that access. The digital asset management offers many advantages and benefits, which consist of

1) the ability to dynamically distribute assets to internal and external teams;
2) a place to quickly find and retrieve assets;
3) the ability for the owner to control who else may view, use, or modify the assets; and
4) a mechanism to keep track of assets' history, so as to reuse them for maximizing their value.

DAM has increasing benefits in booming global Internet economy, but it is still a great challenge for constructing an effective DAM. Fortunately, blockchain technology may be one of the effective ways to solve this problem. Exactly, blockchain is a digital decentralized ledger that keeps a record of all transactions that take place across a peer-to-peer network. The features of blockchain, including decentralization, tamper-proof, and traceability, stem from its cryptographically secure data structure [1] that takes a number of records and puts them in a block (like collating them on to a sheet). As shown in Fig. 1, each block is then chained to the next block, using a cryptographic hash. In addition, the Merkle hash tree, as a type of binary tree, is introduced to guarantee that the transaction details are validated, and thus cannot be altered later on. In virtue of decentralized nature and above-mentioned properties, the blockchain allow mutually distrustful parties to transact securely without trusted third parties.

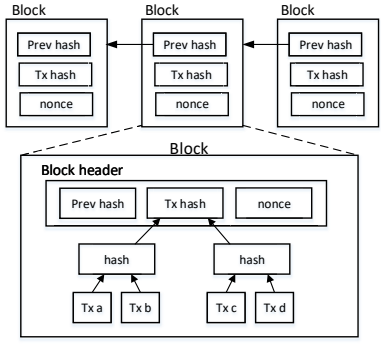


Fig. 1. Block structure of blockchain.

Blockchain technology should be a preeminent solution for DAM because it protects the ownership of digital assets and protects the information from being misused. Firstly, it is not easy to modify digital assets stored on blockchain because the assets are stored with a hash. In fact, it is distributed to a large number of validators which makes it extremely hard to tamper any data in blockchain. Secondly, the validity of a

2474-2473/18/$31.00 ©2018 IEEE
DOI 10.1109/SCC.2018.00032

IEEE computer society

DAM transaction is determined by consensus of the various computers or nodes in the network instead of having a central trusted or official record. Moreover, blockchain can be used to verify the transfer of ownership of digital assets or even just the right to access and use such assets. In summary, digital asset management could leverage security properties of blockchains, which include:

- Impossibility of counterfeit,
- Immutability,
- Disintermediation and ease of transfer, and
- Transparency and ease of auditing.

Therefore, there is no denying in the fact that Blockchain has immense potential to become an important component of any DAM strategy.

Although blockchain provides some unique properties for distributing and managing digital asset, it is still unable to meet the requirements of flexibility, diversity, and dynamicity for permission management. For example, if requesters want to get access to a certain digital asset, they must have the permission of whoever manages it and provide some declarations (or at least hints about how to transact on the required digital asset). More concretely, in the healthcare system the patient's medical records must follow strict access constraints to prevent the leakage of sensitive information. Or, the distribution of copyrighted media must be consistent with copyright requirements, such as, some movies can only be played in a specific region or a certain device. To overcome this problem, it is a direct and effective method to introduce access control technology into DAM.

In the last two years, several new researches have addressed the problem of integrating access control and blockchain technology. For example, Ouaddah *et al.* proposed a framework, called FairAccess, for access control in Internet-of-Things (IoT) on the blockchain [2]. This framework used organization-based access control (OrBAC) to enable users to own and control their data. However, the granularity of OrBAC is not enough to realize digital asset management considering that its organization structure may remain relatively fixed over time. Xu *et al.* also proposed a distributed ledger based access control (DL-BAC) scheme for Web applications [3]. This scheme adapt access control list (ACL) to realize a lightweight decision-making and granting access. For the future work, they plan to extend the privacy enhanced DL-BAC to support more complex access policies including role-based access control (RBAC) and so on. To address the same problem, Azaria *et al.* used the blockchain to implement a decentralized medical record access and permission management [4] with authentication, confidentiality and accountability. Xia *et al.* also proposed a blockchain-based data sharing framework [5] which allows access to only invited, and hence verified users. As a result of this design, further accountability is guaranteed as all users are already known and a log of their actions is kept by the blockchain. In general, these researches have enlightening significance for our research.

To implement a flexible, diverse and dynamic access control in DAM, in this paper we intent to introduce attribute-based access control (ABAC) [6] into the blockchain. ABAC provides an efficient approach to accommodate a wide breadth of access control policies and simplify permission management. Recently, several practical frameworks, such as Extensible Access Control Markup Language (XACML) [7] and Next Generation Access Control (NGAC), have been proposed to provide a standardized way for expressing and enforcing vastly diverse access control policies on various types of data services. These frameworks possess two major features:

- **Diversification:** the ability to extract attribute values from multi-source, various places, different types, e.g., GPS location, time, visitor traffic, or threat level;
- **Dynamicity:** the ability to provide runtime support for acquiring the policy and attributes associated with a subject, object, action, or environment in which access requests occur.

In other word, access control logic of ABAC engine can determine who should have access to what resources under what circumstances and by taking what actions. Hence, the ABAC is particularly useful for an open environment, such as blockchain. However, we should note that there exist still several actual challenges that need to be solved.

One of the core challenges to be addressed is how to integrate access control mechanisms in ABAC into blockchain. An effective solution for this problem is to construct various blockchain's transactions corresponding to the steps of access control procedure, e.g., store, ingest, organize and retrieve, for digital asset managed by DAM. By means of this approach, a registered user is allowed to publish his/her own digital assets to all members in blockchain, and then the platform can ensure the member's access abided by common ABAC's rules. Furthermore, these transactions recorded in the blockchain achieve the goal of ownership claiming, asset escrowing, transparency and verification of all steps.

Our Contribution. In this paper we focus on a new generation of digital asset management platform in a decentralized organization or alliance. This platform can support flexible and diverse permission management, as well as verifiable and transparent access process. Exactly, we not only provide a complete new access control framework and procedure, but also conduct researches on concrete implementation techniques. Our contributions are listed as follows:

- We present a new digital asset management platform, called DAM-Chain, with *Transaction-based Access Control (TBAC)* which integrates the distribution ABAC model and the blockchain technology. In this platform, the ABAC provides flexible and diverse authorization mechanisms for digital asset escrowed into blockchain, and the transactions in blockchain serve as verifiable and traceable medium of access request procedure.
- We present four types of transactions to describe the TBAC access control procedure, and provide the algorithms of these transactions corresponding to subject registration, object escrowing and publication, access request and grant. Moreover, the Bitcoin-type crypto-

graphic scripts are designed to guarantee authorization relationship among these transactions.

Organization. The rest of the paper is organized as follows. We describe our TBAC framework and various transactions in Section II and IV. The paper concludes in Section VI.

II. System Framework

Digital asset management (DAM) has increasing benefits in booming global Internet economy, but it is still a great challenge for providing an effective way to manage, store, ingest, organize and retrieve digital assets. To achieve our goal, we introduce the distributed ABAC model into the exiting blockchain for utilizing the policies to restrict user's access. Moreover, we intend to design new transaction-enabled mechanisms to guarantee the enforcement of these access policies correctly. These mechanisms would provide a good solution for enhancing the creditability of policy decision-making, as well as attribute diversification and dynamicity. This new solution should be sufficiently general to provide the following functions:

- **Asset security**: by using transactions, it needs to implement adequate authorization protocols to identify ownership and permit transfer or issuance of assets.
- **Secure asset issuance**: the shared objects are escrowed into blockchain by using a verifiable transaction, and then the access rules and policies will be used to prevent unauthorized entities from accessing to the object in the open environment.
- **Distributed Permission**: multiple authorization centers and all relevant parts can make a comprehensiveness decision of access request by providing the different attributes and certifications.

Blockchain and ABAC model also provide some good properties to implement our goal, e.g., the blockchain's transactions inherently provide authenticity, integrality, traceability and somewhat anonymous, and the ABAC model is featured as flexibility, dynamicity and diversity for resource management.

A. Standard ABAC Model

We illustrate the standard ABAC model given by NIST, which is the foundation of subsequent research. Within the NIST's ABAC model [6], there exist four kinds of entities, i.e., *subject, object, action*, and *environment*. The characteristics of these entities are defined as attributes. According to these attributes, access policy extracted from common rules is used to determine whether a subject requests to perform operations on objects should be allowed under a specific environment.

The ABAC's reference architecture is shown in Fig. 2. This architecture includes four service nodes such as policy enforcement point (PEP), policy decision point (PDP), policy information point (PIP), and policy administration point (PAP). Also, PDP and PEP functionality can be distributed or centralized, and they constitute so-called authorization service (AS). For an access request, the workflow of this architecture is described as follows:

1) The PEP intercepts the access request from an authenticated subject and sends the request to the PDP.
2) The PDP makes access decision according to access policy generated by PAP and the attributes of subject, object, and environment obtained by querying the PIP.
3) The final decision result given by the PDP is sent to the PEP, and then the PEP fulfills (either permits or denies) the access request according to the decision of PDP.

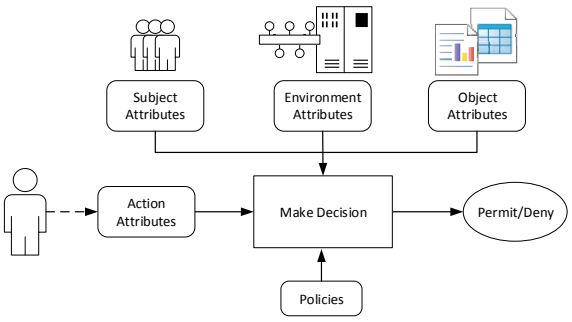


Fig. 2. The NIST's ABAC model.

The above architecture is also composed of two repositories and many environment perception modules. Two repositories store and manage common access rules and entity attributes, respectively. The environment perception module can acquire the environment information at the current time relevant to a request, in which these information may include the current time, location, threat level, device's type, etc.

B. Threat Model

In this work, we mainly consider the threats from the semi-trusted cloud server and malicious client in an open and large-scale resource sharing. Here, we first assume cloud itself is semi-trusted, which means it honestly follows protocols and does not pollute data integrity actively as a malicious adversary, but it may try to find out as much secret information of stored data as possible. Malicious clients may try to access the data without permission by data owner. The goal of our system is to guarantee that only authorized client can access the data and conversely unauthorized clients or cloud will learn nothing by using access control and resource encryption.

In addition, we assume that our TBAC is deployed in an open and untrusted environment. All components are decentralized and distributed across the complete platform. Moreover, the attacker can eavesdrop and counterfeit the communication between any two components in platform. However, we do not assume that the attacker can corrupt any functional components. To these assumptions, the security of TBAC focuses mainly on three aspects: *transaction security, authorization security,* and *decision-making security*. We dive into the details as follows.

III. Our TBAC Platform

We present a new digital asset management platform, called DAM-Chain, with *transaction-based access control* (TBAC) platform on blockchain and cloud storage to realize our design

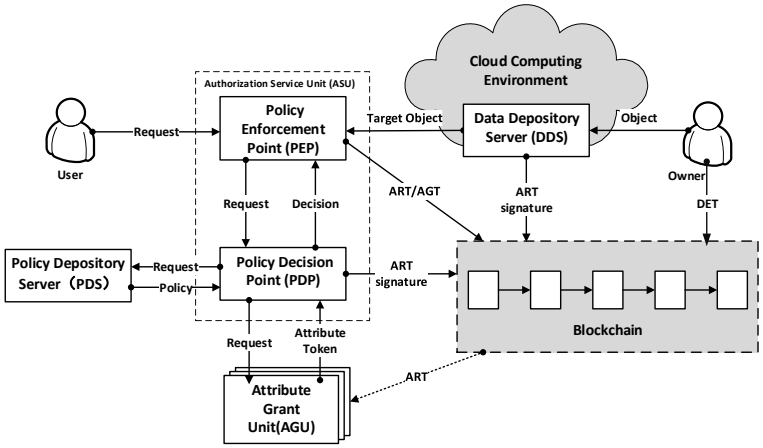


Fig. 3. The framework of TBAC model.

goal. This platform uses the blockchain as the core of resource distribution and sharing while introducing ABAC model to implement flexible resource access authorization.

A. Our TBAC Construction

The TBAC platform is perfectly engaged in "off-chain" storage, where blockchain as object directory is adopted for the retrieval service of resources, and the DDSs are actually used to deposit them by using cloud computing environment. The advantage of this storage way is to reduce the overhead of user's management data. Considering the privacy of shared objects, they will be stored into the DDSs in an encrypted way. The encryption process is implemented by the object's owner before submitting into the DDS.

As shown in the Fig. 3, the new TBAC consists of a blockchain and some additional modules under ABAC model. These additional modules are described as follows.

- **Data Depository Server (DDS)**: stores the owner's escrow objects, and provides access service for the shared objects.
- **Authorization Service Unit (ASU)**: evaluates and enforces access decisions in response to the request from a subject requesting access to a protected object.
- **Policy Depository Server (PDS)**: manages and retrieves common access rules, and produces the access policy from the rules according to the user's request.
- **Attribute Grant Unit (AGU)**: serves as the acquisition source of attribute values required for policy evaluation.

The presented TBAC platform is consistent with the NIST's ABAC model [6], so that each module contains a certain functional point in the ABAC model. Specifically, the AGUs, as function extension of the PIPs, are dispersed into the whole blockchain network, e.g., environment attribute values could come from a client-side equipment where the user sends an access request. Moreover, the attribute, acquired from the AGU, would be issued in a way of verifiable security Token, called **attribute Token**, in order to prevent a forgery of attribute.

The PDS can be considered as a specific implementation of the PAP in the ABAC model. It is responsible for managing the common access rules which can be combined together to express policy according to the user's access request.

The ASU, including PEP and PDP, is main execution agent to monitor, evaluate, and enforce the decision in terms of the user's access require. PEP and PDP functionality may be put together or separated from each other. For example, the overall ASU can be deployed on the client side. However, this setting, which requires high communication and computation overheads, is not suitable for lightweight devices. Instead, we would prefer to put the PDP into the blockchain's node, but to lay the PEP into the client's application in order to improve the performance.

B. The Workflow of TBAC

The TBAC platform provides the entire process management of resource distribution and sharing, including subject register, resource publish, permission management of access acquire. In TBAC, any user must register to the platform by a certain AGU. For a registered user, the AGU manages the user's subject attributes and submits the user's information (encapsulated into **subject registration transaction, SRT**) into the blockchain.

While intending to share resource, the owner sends the encrypted object and its attribute information to a DDS, and then the DDS submits the object's information (encapsulated into **object escrow transaction, OET**) into the blockchain. By means of this approach, the DDS implements the object escrow and the ownership claiming.

We next turn our attention to the process of access request. After a subject applies for access to an encrypted object, the PEP of ASU interprets and transfers this request (called **access request transaction, ART**) to the PDP and all blockchain nodes, and then waits for the decision result. The workflow of policy decision-making is described as follows.

1) **Generating Policy**: upon receiving the request, the PDP inquires of PDS about its access policy. In response, the PDS produces the policy relevant to the request from

policy repository, and then generates and sends it and the corresponding cryptographic policy to the PDP.

2) **Acquainting Tokens**: upon receiving the responsive policy, the PDP selects the necessary attributes that satisfy the policy, and then queries these attributes' tokens to the relevant AGUs. In response, the AGU generates and returns the attribute Tokens if the queried attributes are valid at the current time.
3) **Making Decision**: the PDP makes the access control decision based on the received cryptographic policy and the attribute Tokens. If the decision result is True, the PDP confirms this result to blockchain by sending its ART signature to all nodes. It finally sends the Token (called *Authorized Token*) of decision result to the PEP.
4) **Decrypting Object**: for an authorized decision, the PEP firstly requests the DDS to pass the target object back. The DDS returns the encrypted object and appends its signature into the corresponding ART Transaction. The PEP next uses the authorized Token to decrypt the returned object, and then confirms this access to blockchain by sending its ART signature to all nodes.
5) **Confirming Transaction**: when all blockchain nodes receive the confirmation signature from the PEP, the blockchain platform starts the consensus protocol to confirm that the ART transaction is complete, and records the complete ART (called **access granted transaction, AGT**) into the blockchain. Finally, the PEP fulfills the operation requested by the subject.

The only thing that is necessary for the subject is to authenticate his identity before entering the platform. Moreover, it is unnecessary for subject to retain any key corresponding to resource decryption. This means there is no need for the user to perform any cryptographic operation in addition to applying for an access request.

In the TBAC platform, an attribute Token is a one-time security proof for the authorized attribute issued by the AGU. We consider the Token as a public verifiable "ticket" to prove that the attribute is what they claim to be. This ticket usually contains the information of the issuer, attributes, issue time, as well as a tag, where the tag is a simple signature that verifies the authenticity of all other information by using the issuer's public key.

C. Access Rules and Policies in ABAC

In TBAC, we construct the PDS by using XACML that defines a policy specification language and reference architecture for ABAC implementation. Let $S = \{s, o, a, e\}$ denote a set that consists of subject, resource, action, and environment attribute. Here, attribute instances are specified as name-value pairs, where attribute name denotes the property or characteristic associated with certain attribute in S. For example, in a medical setting, $Role(s) = doctor$ denotes the attribute name $Role$ associated with a subject s is $doctor$, similarly, $Ward(o) = pediatrics$, $ResourceID(o) = medical - records$, $Time(e) = 12:11$, and so on.

In TBAC, an access policy is composed of a target and a set of rules. A target defines a simple Boolean condition that, if satisfied by the attributes, establishes the need for subsequent evaluation by a PDP. We use the form of "attribute category : attribute name : attribute value" to express attribute instance, where symbol ':' is used as separator. For example, "subject : role : doctor" denotes $Role(s) = doctor$. Considering a target may contain multiple attribute instance, we provides a way of reconciling these individual decisions, called "one-and-only", to restrict that each attribute must satisfy one condition. The target of this example applies to "All read or write accesses to medical records by a doctor or intern", that is, $target(s, o, a) := role(s) \in \{doctor, intern\} \wedge resouceID(o) = mdecial - records \wedge actionID(a) \in \{read, write\}$.

In addition to a target, a rule includes a series of Boolean conditions that if evaluated True have an effect of either "Permit" or "Deny". The conditions of a rule are typically more complex and may include functions involving logical operators (e.g., and, or) and relation operations (e.g., $\leq, \geq, =$) for the comparison of attribute values. For example, the first rule denotes that any access request will be denied if the ward assigned by subject is not the same ward where the patient is located, i.e., $rule1(s, o) := WardAssignment(s) \neq WardLocation(o)$. Similarly, the second rule denotes $rule2(s, a) := role(s) = intern \wedge actionID(a) = write$, and the third rule also is $rule3(s, o) := role(s) = doctor \wedge patientstatus(o) = critical$. Finally, we discuss the several standard combining methods that combine multiple rules into a single policy, including the following:

- Permit-overrides: if any decision evaluates to Permit, then the result is Permit, otherwise the result is Deny.
- Deny-overrides: if any decision evaluates to Deny, or no decision evaluates to Permit, then the result is deny. If all decisions evaluate to Permit, the result is Permit.

In this example, we apply for the permit-overrides to integrate rule1, rule2 and rule3 together, that is,

$$policy(s, o, a, e) := target(s, o, a) \wedge (\neg(rule1(s, o) \vee rule2(s, a)) \vee rule3(s, o))$$

denotes the access is denied if only the conditions stated in rule1 or rule2 apply except for the access under a critical situation.

IV. TBAC Transactions

The TBAC platform employs transactions to ensure that only transactions conforming to the requisite policies are authorized and registered into the blockchain. Basically, a transaction is a data structure that encodes a transfer of access request and authorization process among participants in the platform. Everything in the blockchain is designed to ensure that transactions can be created, propagated on the network, validated, and finally added to the blockchain (as the global ledger). In the proposed TBAC platform, there exists four main

transactions to enforce fine-grained access policies. These four transactions are described as blow.

A. Subject Registration Transaction

The SRT is used to record the information of one or more subjects, and must be signed and issued by the AGU who manages the subject's attribute information. In Algorithm 1 we show a high-level example of how a SRT transaction is constructed. The public keys of subjects are converted into Pay-to-Public-Key scriptPubKey scripts (explained below), while the output scripts are accompanied by signature of the corresponding AGU's private key.

Algorithm 1 Procedure for creating SRT.

Input: Subject information; Subject attributes; Authentication information; AGU information;
Output: Subject registration transaction;

/*List all subjects and their information, attributes and authentication information*/
for all subject S **do**
    public-attr ← get S's attributes information;
    pub, priv ← get S's keypair;
    addr ← get pub's hash;
    scriptPK ← (OP_PUSHDATA(33),pub,OP_CHECKSIG);
    Add (addr, tx_index, public-attr, scriptPK) to list of subject;
end for;
/*Specify AGU's information*/
AGU ← AGU's network address;
pub,priv ← get AGU's key pair;
sig ← $sign_{priv}$(All except scriptSig in SRT);
scriptSig ← (OP_PUSHDATA(72), sig);
Transaction ← (version, AGU, subject, time, tx_index, scriptSig);
return Transaction;

An instance of the executing result of Algorithm 1, where the SRT contains four types of information: 1) subject information, e.g., subject's wallet address (called *addr*) and this transaction's index and sequence number (expressed as *tx_index#sequence*); 2) subject attributes, e.g., the list of public attributes (*public-attribute*); 3) authentication information, e.g., the script of public key (*scriptPubKey*); and 4) AGU information, e.g., AGU's network address (*AGU*) and the script of its signature (*scriptSig*).

Similar to Bitcoin, the wallet is used to store the user's public/private key pair. When the user is registered into the platform, the wallet software is installed and a public/private key pair with an elliptic curve digital signature algorithm (ECDSA) is generated. The public key is then hashed and this hash value serves as the wallet address ("addr") that is transfered from/to in TBAC.

The above-mentioned scripts are sequences of instructions called *opcodes* that get executed by all entities in our platform. In particular, our TBAC platform makes use of Bitcoin's scripting language, that is stack-based and without loops. Moreover, TBAC employs five kinds of scripts. We here give all the types as follows:

- **Type I: pay-to-public-key** contains two components:
  scriptPubKey: <pubKey>, OP_CHECKSIG
  scriptSig: <sig>
- **Type II: pay-to-public-key-hash** contains two components:
  scriptPubKey: OP_DUP, OP_HASH160, <pubKeyHash>, OP_EQUALVERIFY, OP_CHECKSIG
  scriptSig: <sig>, <pubkey>
- **Type III: pay-to-script-hash** contains two components:
  scriptPubKey: OP_HASH160, <scriptHash>, OP_EQUAL
  scriptSig: <sig>, <script>
- **Type IV: multiple signature** contains two components:
  scriptPubKey: M, <pubKey A>, <pubKey B>, <pubKey C>, N, OP_CHECKMULTISIG
  scriptSig: OP_0, <sig B>, <sig C>
- **Type V: OP_Return** contains two components:
  scriptPubKey: OP_RETURN, <data>
  scriptSig: NULL

For example, the subject's script takes the structure of Type I to store the user's public-key into "scriptPubKey", in which OP_PUSHDATA(33) is to push the subsequent 33-byte public key into the stack, and "OP_CHECKSIG" expects two values on the stack to be verified. Similarly, the AGU's script "scriptSig" takes the structure of Type I to push the 72-byte AGU's signature into the stack by using OP_PUSHDATA(72). The execution process of scripts refers to [8].

B. Object Escrow Transaction

The OET, as escrow credential and ownership claim, is used to record various information of protected objects. Algorithm 2 describes the procedure of an OET transaction, where the OET requires the signatures from both the owner and the DDS who is the actual object's depository. Thanks to openness of blockchain, the OETs are publicly accessible to all members in the whole platform, so that any member can retrieve the required objects conveniently.

The structure of OET produced by Algorithm 2 can be divided into two parts: one is the owner's profile and the other is the escrowed objects' profile. This kind of structure describes the "*one-to-many*" relationships between the owner and the escrowed objects.

The owner's profile contains two types of information: 1) the owner's information, i.e., the owner's SRT index and sequence number (*owner_tx_index*), which is used to find owner's information stored in the corresponding SRT; and 2) ownership claim, i.e., the script of the owner's signature (*scriptSig*). As mentioned above, the script of TBAC currently utilizes two different *scriptSig/scriptPubKey* pairs which can be cryptographically linked together to enforce the agreement. This kind of agreement can be assessed by using the public-key (*scriptPubKey*) to verify its holder's signature (*scriptSig*).

The script $scriptSig$ stores the owner's ECDSA signature over the transaction itself. This signature, verified by the *scriptPubKey* in SRT related to *owner_tx_index*, proves that the objects in this transaction were possessed by the subject in the SRT.

Algorithm 2 Procedure for creating OET.

Input: Profile of owner; Profile of escrowed objects;
Output: Object escrow transaction;

```
/*Specify owner*/
owner_tx_index ← the owner's SRT index#sequence
for all escrowed object O do
    name ← get O's identity;
    attribute ← get O's attributes;
    DDS ← get DDS's URL address;
    pub,priv ← get DDS's keypair;
    sig ← sign_priv(O's information and attributes);
    scriptSig ← OP_PUSHDATA(72) sig;
    Add (name, DDS, tx_index, attribute, scriptSig) to list of escrow;
end for;
pub, priv ← get owner's key pair by owner's SRT;
sig ← sign_priv(All except scriptSig in OET);
scriptSig ← (OP_PUSHDATA(72), sig);
Transaction ← (version, owner_tx_index, escrow, time, tx_index, scriptSig);
return Transaction;
```

The profile of escrowed object consists of three parts: 1) object information, e.g., the object's name (*name*), the DDS's address (expressed by URL), and its transaction index (*tx_index*); 2) object attributes, e.g., the list of attributes (*attribute*); and 3) escrow credential, i.e., the DDS's signature (*scriptSig*). This signature, authenticated by the PKI public-key certificate obtained from the URL of *DDS*[1], proves the object was escrowed by the owner of the certificate.

C. Access Request Transaction

The ART is a credible ticket to take note about object's access procedure in the form of transaction logs. It contains all necessary information of access request generated by PEP, and these information will be used by the subsequent decision-making for the access request. As shown in Fig. 4, we describes the procedure of an ART transaction. Considering that the ART contains several unauthorized signatures (expressed by *empty*), it is not allowed to append into the blockchain as a valid access permission. In ABAC, access request is generally used to specify *what subject wants to have access to what object with what kinds of actions*. For this reason, the ART ought to contain three segments *(subject, object, action)* of an access request. As shown in Fig. 4, the *subject* segment consists of the subject's SRT index (*tx_index*) and the subject's signature (*scriptSig*), which can be verified by the *scriptPubKey* in the indexed SRT. In the same way, the *object* segment consists of the object's OET index (*tx_index*) and an empty DDS's signature (*DDS_Sig*), which will be verified by the PKI certificate issued by the DDS. The *action* segment is the list of all authorized operations, each of which is expressed as "*company.technology#action*".

```
{
   "ver": "ART",
   "PEP": pep-server.healthgrades.com,
   "PDP": pdp-server.beaumont.org,
   "subject":{
     "tx_index": 322810234#0,
     "scriptSig": "OP_PUSHDATA(22) 00141b0f7c787311455b
       3c2272d3869d76176db3d9fe",
    },
   "object":{
     "tx_index": 322786399#0,
     "DDS_Sig": "OP_PUSHDATA(72) 2044022049bde80a68
        3a220a1a2754dd...b2b63cce11",
   },
   "action":[
     "org.apache.pdfbox.pdfctrl#read",
     "com.sun.java.drawgraph#append"
   ]
   "time": 1515658108,
   "declaretimeouts": 10m,
   "tx_index": 323175628,
   "PDP_Sig": "OP_PUSHDATA(72) 30141b0f7c787311455b
     3c2272d3869d76...176db3d9fe",
   "PEP_Sig": "OP_PUSHDATA(72) 304502210d12a190c59f2a1
     d58f3205060...d8f81601"
   "hash":"ef4e4e679ea137def17821ba98129...6be9ac8d2"
}
```

Fig. 4. Example of access request transaction.

In addition, the ART contains the information that is used to provide dynamic authentication from relevant authorized entities in TBAC platform. These authentication information consists of the PEP/PDP address (expressed by URL) and their signatures (*PEP_Sig* and *PDP_Sig*), as well as access request time and period of validity. As mentioned in Section III-B, the initial signatures of DDS, PEP, and PDP are empty, but they will be signed according to the order of PDP, DDS and PEP for a valid request.

D. Access Granted Transaction

The AGT is the final form of ART after all empty signatures are fulfilled, and then it will be stored into blockchain as an access log once all of signatures are validated by the block generator. For the sake of clarity, we next describe the authentication relationship between the above-mentioned transactions (including SRT, OET and AGT) and different entities (PDP, PEP, DDS, AGU, etc.) from an AGT-centered viewpoint.

The AGT is submitted to the blockchain until the four signatures in AGT are checked. Among subject's scriptSig, DDS_Sig, PEP_Sig and PEP_Sig, only the scriptSig is written in scripting language. So that, the signature of subject is verified by performing function $executeScript$ which sequentially executes the scriptSig and scriptPubKey written in scripting language, and then the sigVerify function is called respectively to verify the signature of the DDS, the PDP, and the PEP through using public-key certificates issued by the CA.

[1]Several entities, including DDS, PDP, PEP and AGU, utilize PKI certificate to issue their public keys. The SRT may be another option for issuing their public key.

V. System Analysis

A complete authentication system can be generated from complex authorization or permission relationship among transactions and platform's entities. Algorithm 3 is used to implement this procedure of AGT verification.

Algorithm 3 Procedure to verify AGT

Input: Access granted transaction;
Output: valid or invalid;

```
/*Verify each signature in the transaction*/
scriptPubKey ← extract from SRT according to subject's tx_index;
if executeScript(scriptSig, scriptPubKey)==false then
    return invalid;
end if
DDS_Pub ← get DDS's public key from CA;
if sigVerify(DDS_Pub, DDS_Sig)==false then
    return invalid;
end if
PDP_Pub ← get PDP's public key from CA;
if sigVerify(PDP_Pub, DDS_Sig)==false then
    return invalid;
end if
PEP_Pub ← get PEP's public key from CA;
if sigVerify(PEP_Pub, DDS_Sig)==false then
    return invalid;
end if
return valid;
```

For an access request, the *subject* segment of AGT is used to point to a requester's SRT that contains the public key of the requester. Such that, the requester's signature in the AGT can be verified by using this public key. For the accessed object, the *object* segment in the AGT points to the OET which contains the address of the corresponding DDS. The public key certificate of DDS can be obtained according to this address. Therefore, any one can use the DDS signatures, stored in both the *object* in AGT and the *eacrow* in OET, to validate the DDS authorization. The *owner* segment in OET can also help us to find out the SRT of object's owner through the *tx_index*, and the owner claims ownership of the object using the signature signed by his private key. Finally, the *PDP_Sig* and *PEP_Sig* in AGT are considered as access permissions issued by the PDP and the PEP. In addition, the *AGU* in SRT stores its own address which is convenient for acquire the subject's attributes.

According to the above description, our TBAC platform implements these functions as follows:

- **Access Openness**: the platform supports data sharing and exchanging with unlimited number of users in the public network environment.
- **Authorized Trusteeship**: after signing the object escrow transaction, the owners do not need to participate in subsequent access authorization.
- **Rule Generality**: the rules in TBAC can support the general security restrictions (such as read-up/write-down), and they can change as required.
- **Policy Dynamics**: for the current request, the policy is produced dynamically from the security rules according to the current state of the request.
- **Access Traceability**: the process of digital asset sharing and exchanging is traceable via transactions, and their access authorizations are cryptographically verifiable.

VI. Conclusions

In this paper we take transaction as a bridge to integrate ABAC and blockchain into a new platform for resource distribution and sharing. Our proposed platform supports flexible and diverse permission management, as well as verifiable and transparent access authorization process. We hope our research could provide useful reference for globe resource sharing.

Acknowledgment

The work is supported by Beijing Municipal Commission of Economy and Information Technology (The project entitled "Research on network security of big data in E-government"), the National Natural Science Foundation of China (Grant No. 61472032), NSFC-Genertec Joint Fund For Basic Research (Grant No. U1636104), Joint Research Fund for Overseas Chinese Scholars and Scholars in Hong Kong and Macao (Grant No. 61628201) and National Key R&D Program of China (Grant No. 2017YFB0802503).

References

[1] I.-C. Lin and T.-C. Liao, "A survey of blockchain security issues and challenges." *IJ Network Security*, vol. 19, no. 5, pp. 653–659, 2017.

[2] A. Ouaddah, A. Abou Elkalam, and A. Ait Ouahman, "Fairaccess: a new blockchain-based access control framework for the internet of things," *Security and Communication Networks*, vol. 9, no. 18, pp. 5943–5964, 2016.

[3] L. Xu, L. Chen, N. Shah, Z. Gao, Y. Lu, and W. Shi, "Dl-bac: Distributed ledger based access control for web applications," in *Proceedings of the 26th International Conference on World Wide Web Companion*. International World Wide Web Conferences Steering Committee, 2017, pp. 1445–1450.

[4] A. Azaria, A. Ekblaw, T. Vieira, and A. Lippman, "Medrec: Using blockchain for medical data access and permission management," in *Open and Big Data (OBD), International Conference on*. IEEE, 2016, pp. 25–30.

[5] Q. Xia, E. B. Sifah, A. Smahi, S. Amofa, and X. Zhang, "Bbds: Blockchain-based data sharing for electronic medical records in cloud environments," *Information*, vol. 8, no. 2, p. 44, 2017.

[6] V. Hu, D. Ferraiolo, D. Kuhn, A. Schnitzer, K. Sandlin, R. Miller, and K. Scarfone, "Guide to attribute based access control (abac) definition and considerations," pp. 162–800, 01 2014.

[7] R. Nasim and S. Buchegger, "Xacml-based access control for decentralized online social networks," in *Proceedings of the 2014 IEEE/ACM 7th International Conference on Utility and Cloud Computing*. IEEE Computer Society, 2014, pp. 671–676.

[8] I. Giechaskiel, C. Cremers, and K. B. Rasmussen, "On bitcoin security in the presence of broken crypto primitives." *IACR Cryptology ePrint Archive*, vol. 2016, p. 167, 2016.

A Blockchain Based Decentralized Computing And NFT Infrastructure For Game Networks

Koushik Bhargav Muthe, Khushboo Sharma, Karthik Epperla Nagendra Sri
CSE Department
SRM University AP
Amaravati, India
Email: koushik_bhargav@srmap.edu.in, khushboo_sharma@srmap.edu.in, epperla_nagendra@srmap.edu.in

***Abstract*—The market value of the Gaming industry was said to be over 138 Billion USD in 2019. Competitive Gaming or eSports was already included in the Asian Games 2020, and the Olympic committee is considering including eSports into Olympics 2024. The online gaming segment amounts up to 7 quintillion bytes of internet traffic every month. Most of this data is controlled by centralized gatekeepers like cloud agencies and game creators. This is causing many issues ranging from privacy concerns to latency. The game makers have complete rights over the game to arbitrarily change the rules, set prices for the assets, and control over game servers. Even the reward mechanisms in most of the games, including eSports, are controlled by the game producers, and these rewards have no real-world value. The motivation behind this research is to create a decentralized computation and token management infrastructure for game networks. This paper focuses on using Ethereum Blockchain, IPFS and ERC - Ethereum Request for Comment 1155 architecture to build a gaming-oriented public decentralized network.**

***Index Terms*—Cloud Gaming, Ethereum Blockchain, Decentralized Computation, IPFS - Inter Planetary File System, ERC 1155 - Ethereum Request for Comment.**

I. Introduction

Gaming is one of the largest spaces in the Entertainment Industry. It is massive, with annual revenue of over 138 Billion USD in 2019 [1]. It also has an enormous audience, with over 2.47 Billion gamers around the world by 2019 [2]. Competitive Gaming or eSports is another primary market in Gaming with projected revenue of 2.96 Billion USD by the end of 2022 [3]. It was included in the Asian Games 2018 as a demonstrational sport and the Olympic committee has considered to include it in the 2020 Summer Olympics in Tokyo. Provided with these rapid changes in the gaming industry game makers are spending billions of dollars into building and improving the games in the market. The increase in the number of people with access to the internet has increased the amount of online Gaming which accounts for seven quintillion bytes of internet traffic every month. Very few significant entities control the gaming industry. Even the large volume of internet traffic is controlled by some large corporations who act as gatekeepers for the gaming network. Gaming is no more an act of leisure as the scope of the gaming industry has increased a lot in the past few years. This applies to the contribution made by players across various gaming platforms which includes their profiles and assets. Gamers spend a lot of money in the form of fiat for purchasing in-game assets, and these assets have no real-world value. The game producers control the prices of the assets, and the gaming community has no stake in it. There are also no storage spaces and real-world market places that can support secure storage and trading of these in-game assets. Gamers often face latency issues because of the centralized gatekeeper architecture of the gaming network [4]. The gaming community should have more stake in the decisions made by the game creators extending from prices of in-game assets to game features.

The primary focus of this paper is to propose a gaming network where the game players are given a stake in the gaming industry. A decentralized computing architecture would provide a solution to this by eliminating centralized game servers. The Ethereum blockchain would provide gamers a real-world value for their intangible assets by converting them into NFTs - Non Fungible Assets, which are rare and unique blockchain managed digital assets. These assets have a real-world value and can be traded in markets outside the game network. IPFS is a peer to peer hypermedia protocol which is a reliable Web3.0 based protocol. IPFS removes the need for centralized entities such as game servers and connects the nodes in the network directly. This can potentially reduce the latency and also provide a secure channel for gaming. Ethereum 2.0 aims to introduce proof of stake architecture into the leading network, enabling gamers to have a stake in the network, which can be utilized to improve their game features. This can reduce the complete control of games by game makers and also helps game developers to add better features which are in consensus with the gaming community.

II. Background and Related Works

Cloud gaming is aimed to be one of the most revolutionizing technologies in the gaming industry. However, the first few glimpses from major players such as Google and Microsoft were not playable. Issues such as latency and low quality are consistently seen across all services. The work by Mark Claypool and David Finkel clearly stated the effects of latency on the performance of the players [5]. The work done by Bryce Mariano and Simon G. M. Koo clearly stated that Cloud Gaming is nearly impossible if it depends on the current internet architecture [6]. The paper by Daniel Uribe and Gisele

978-1-7281-8370-1/20/$31.00 ©2020 IEEE

Authorized licensed use limited to: Univ of Calif Santa Barbara. Downloaded on May 17,2021 at 18:05:47 UTC from IEEE Xplore. Restrictions apply.

Waters presented the privacy and decentralized advantages of using NFTs but is limited to Genome Research [7]. The limitations of the previous works include dependency on centralized server-based communication for gaming and limited inclusion of Blockchain-based NFTs for game objects or assets. The proposed protocol has an advantage over the previous works as it utilizes IPFS and Proxy Computation, which eliminates the need for a centralized game server. This considerably reduces the latency issues. It also integrates Ethereum ERC-1155 for NFT based game objects to create and distribute game objects in a decentralized infrastructure.

A. Ethereum Blockchain

A Blockchain is an immutable distributed ledger with multiple blocks enchained together, and every block stores transactions in such a way that it's unacceptable to change these transactions. It's an enormous step forward in terms of decentralized and distributed applications. Blockchain technology guarantees advantages in trustability, collaboration, organization, identification, quality, and transparency. Decentralization in easy terms implies that the service or application is deployed on a network in such a way that it provides no comprehensive management over data and execution to any server. No one inside this cluster can vary or delete the previous transactions; instead, every server includes a current copy of data and execution logic. Distributed implies that any server or node on a network is connected to each alternative node directly or indirectly. Ledger is associated with an accounting term, and you can consider it as specialized storage and retrieval of data.

Fig. 1. Blockchain - Block Diagram

Ethereum is the implementation of Blockchain and permits extending its practicality with the assistance of smart contracts. Vitalik Buterin first planned the kernel of this work in Nov 2013 [8]. The state in Ethereum denotes the balances of the account and potential additional data. The main aim of Ethereum is to validate transactions from statements, update their status (state), and continue to maintain that state as the current state until another transaction is approved. One key goal is to facilitate transactions between willing people who would otherwise do not have any means to trust each other. This could be due to geographical separation, interfacing problems, or maybe the incompatibility, censorship, disposition, privacy, expense, uncertainty, or inconvenience.

B. Smart Contracts

A Smart Contract is a digital set of rules and regulations that expedite, verify, or implement the performance of a contract. Smart contracts are used to validate the credibility of transactions without any third party. Nick Szabo is the pioneer in developing the concept of Smart Contracts [9]. Smart Contacts made Ethereum robust and scalable to multiple dimensions. Solidity and Vyper are two prominent languages to develop Smart Contracts.

C. EVM - Ethereum Virtual Machine

Ether is the cryptocurrency used in the Ethereum blockchain. Any transaction between accounts is expressed as a distributed currency by Ether. The Ether is the fuel for operation in Ethereum. This permits many applications, starting from the exchange of cryptocurrencies to financial applications, storing and managing tokens and digital assets, conventional systems, identity management, and ballot systems, up to those applications that need traceability resources and assets. EVM - Ethereum Virtual Machine, is the runtime surroundings for smart contracts in Ethereum [10]. It is not solely sandboxed; however, really fully isolated, which suggests that code running within the EVM has no access to the network, filesystem, or alternative processes. Smart contracts even have restricted access to alternative smart contracts. Ethereum Virtual Machine guarantees security by preventing Denial-of-service attacks, that are somewhat associated with rising challenges within the crypto business. Secondly, EVM interprets and executes Ethereum programming language and make sure that communication may be achieved with no interference.

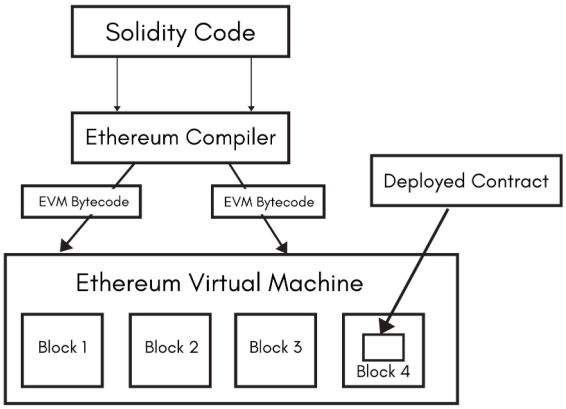


Fig. 2. Ethereum Virtual Machine

D. IPFS - Inter Planetary File System

Blockchain is a secure medium for data storage, but because of its computational and network limitations, it is not ideal for handling large amounts of data. For the economical storage of enormous data and content, a different file system known as IPFS can be used. IPFS stands for Inter Planetary File System, a distributed, decentralized system and a platform to store data and files with high integrity and resiliency. It synthesizes thriving concepts from previous peer-to-peer networks, as well as DHTs, BitTorrent, Git, and SFS. The contribution of IPFS is connecting evidenced techniques into one significant system than the sum of its components. IPFS presents a brand new

Authorized licensed use limited to: Univ of Calif Santa Barbara. Downloaded on May 17,2021 at 18:05:47 UTC from IEEE Xplore. Restrictions apply.

platform for writing and deploying large scale applications. It is a new system for distributing and versioning massive data which can be used to spread computation and storage across multiple nodes in the network.

Fundamentally, IPFS is a peer-to-peer, open-sourced globally distributed classification system that can be used for storing and sharing massive volumes of files with high throughput [11]. Inter Planetary File System or IPFS is a protocol and a network which is designed to form content-addressable, peer-to-peer hypermedia storing and sharing system in a distributed network. Since IPFS is peer-to-peer, no nodes are privileged. IPFS nodes store IPFS objects in native storage. Nodes connect and transfer these objects, which represent files and data structures. IPFS uses the hash of the content to identify and share it among the nodes. It integrates alternative technologies like GIT for version management and MerkelDAG data structure for storing data. IPFS integrates each of the advanced Merkle-DAG structure with the data-addressability of P2P file-sharing systems. The content is distributed over a peer-to-peer network IPFS seeks to connect all computing devices with each other directly. IPFS can even evolve the web itself. In some ways, IPFS is comparable to the World Wide Web (www); however, IPFS can be seen as one BitTorrent swarm, exchanging objects inside one git repository. In alternative words, IPFS provides a high-throughput, content-addressed block storage model, with content-addressed hyperlinks. IPFS combines a distributed hash table with incentivized block exchange and a self-certifying namespace. IPFS has no single point of failure, and hence data transit can not be tampered by anyone. Distributed Content Delivery saves bandwidth and prevents DDoS attacks that Hypertext Transfer Protocol (HTTP) struggles with. The filesystem can be accessed in many kinds of ways like FUSE or ”File system in Userspace” over HTTP. A local file will be added to the IPFS file system, making it available to the globe. Files are known by their hashes; thus, it’s cache-friendly. Any user who downloads the file additionally serves the data to the other users of the network.

The IPFS Protocol is split into a stack of sub-protocols which are assigned totally different functions such as identities managers, network managers, content-based routing, exchange protocols (BitSwap), a content-addressed data structure (Merkle DAG) [12], version control (Git), naming service, etc. IPFS is primarily aimed at replacing HTTPS, but it can become a universal file transfer protocol [13]. IPFS can become compatible with gaming networks when combined with Quick UDP Internet Connections (QUIC) network protocol at the transport layer.

E. NFT - Non Fungible Tokens

Fungible is anything that is transferable. Crypto-currencies such as Ether are fungible tokens as every single token is identical in use and value. On the contrary, a Non-Fungible Token has a unique value and identity. Every NFT is non-dividable and non-mergeable [14]. The first token standard adapted by the Ethereum Blockchain is ERC-20 (Ethereum Request for Comment) which supports Fungible Tokens only [15]. ERC-20 token standard promoted many ICOs which are based on Ethereum Blockchain. ERCs are predefined rules developed using Smart Contracts for implementing token measures in Ethereum Blockchain. ERC-20 can define by providing the contract address and availability of tokens. Lack of support for Non Fungible Tokens is one of the significant drawbacks of ERC-20, which led to another token standard ERC-721.

```
contract ERC721 {
    function name ( ) constant returns (string name);
    function symbol ( ) constant returns (string symbol);
    function toeaISuppIy() constant returns (uint256 toeaISuppIy);
    function balanceof(address _owner) constant returns (uint balance);
    function ownerOE (uint256 tokenId) constant returns (address owner);
    function approve (address _ to, uint256 _tokenId);
    function takeOwnership(unit256 _tokenId);
    function transfer (address _Eo, uint256 tokenId);
    function tokenOEOwnerByIndex( address _owner, uint256 index) constant returns (uint tokenId);
    function tokenMeeadata(uint256 _tokenId) constant returns (string infourl);
    event Transfer(address indexed _from, address indexed _ to, uint256 _tokenId);
    event Approval(address indexed _owner, address indexed _ approved, uint256 _tokenId);
}
```

Fig. 3. ERC-20 - Soildity Functions

ERC-721 token standard supports Non Fungible Tokens. It makes these tokens have a unique value and identity [16]. Tokens are attached to digital objects using metadata to help off-chain rendering or storage. Limitations of ERC-721 include lack of support for multiple tokens in a single, smart contract. Games have various types of unique assets and to support numerous assets, several quick contacts have to be implemented, which increases the gas fee and latency.

F. ERC 1155

ERC-1155 is the new final token standard on the Ethereum Blockchain [17]. It is a universal standard as it supports the features of ERC-20 (Fungible) and ERC-721 (Non-Fungible). It enables game objects to possess a real-world value as it can help multiple tokens in a single contract. Trades and minting of tokens are more comfortable with the introduction of ERC-1155. ERC 1155 also supports converting existing tokens and minting new tokens out of them. Though the focus of this paper is limited to gaming, ERC-1155 has applications in several areas such as documentation and artwork.

```
contract ERC1155 is IERC1155, ERC165, CommonConstants
{
    function safeTransferFrom(address _from, address _to, uint256 _id, uint256 _value, bytes calldata _data);
    function safeBatchTransferFrom(address _from, address _to, uint256[] calldata _ids, uint256[] calldata _values, bytes calldata _data);
    function balanceOf(address _owner, uint256 _id);
    function balanceOfBatch(address[] calldata _owners, uint256[] calldata _ids);
    function setApprovalForAll(address _operator, bool _approved);
    function isApprovedForAll(address _owner, address _operator);
    function _doSTACheck(address _operator, address _from, address _to, uint256 _id, uint256 _value, bytes memory _data);
    function _doSBACheck(address _operator, address _from, address _to, uint256[] memory _ids, uint256[] memory _values, bytes memory _data);
}
```

Fig. 4. ERC-1155 - Soildity Functions

III. Proposed Architecture

The proposed architecture enables decentralized computation and token infrastructure for gaming networks. It integrated IPFS and Ethereum blockchain to distribute the computation requirements in the game networks without any centralized agency. Independent nodes can provide computational activities such as rendering by utilizing their local machines in exchange for rewards which are supported

Authorized licensed use limited to: Univ of Calif Santa Barbara. Downloaded on May 17,2021 at 18:05:47 UTC from IEEE Xplore. Restrictions apply.

by Ethereum. This incentivizes more nodes to participate in the network. The decentralized token infrastructure integrates the ERC 1155 protocol to generate unique game objects which have a real-world value. These objects can be traded for other objects or currency and can also be minted into new objects. The proposed infrastructure is divided into following four elements:

1) Proxy Computation: Decentralized computing is an essential aspect of the proposed architecture. Computing is distributed among several nodes in the network instead of relying on a centralized game server. This completely eliminates the need for central gatekeepers in the gaming networks. Proxies are any nodes willing to participate in the network by utilizing their computational resources. In return for the computational work performed, proxies receive rewards that are based on Ethereum. This incentivizes proxies to participate in the network. In the case of multiplayer games, the computational activities include real-time transmission of data about the game state to the player nodes. In the case of single-player games, proxies act as distributed rendering nodes by effectively replacing the slow and inefficient cloud gaming servers.

2) Data Transfer (IPFS + QUIC): The consensus is an important aspect when it comes to game networks. All the players should have a consensus regarding the state of the game. The proposed architecture utilizes IPFS for transferring data related to states of the nodes in the network. IPFS eliminates the need for any centralized game server as the nodes can directly communicate with each other. The data related to the state of the nodes is exchanged using the Bitswap protocol. IPFS supports a wide range of networking protocols such as aa QUIC and UDP, which are being actively used in the existing gaming networks.

The steps involved in the computation are:

1) Game Networks have a pool of players producing data related to their states. In a multiplayer game, these states have to be synced with each other off-chain, and in a single-player, game rendering has to be performed off-chain. Hence gamers directly send the data related to their states onto a pool of proxies through IPFS.

2) Proxies receive the data related to the states and communicate with other proxies in the network to sync all the players' states. In the case of single-player games, proxies receive the data related to the states and render the game content on their machines.

3) Proxies transfer the synced and rendered data back to the players using the IPFS.

4) Players can also perform in-game communication among themselves as they are directly connected.

5) All the transactions between the nodes are recorded on the Ethereum blockchain after being validated by game smart-contracts.

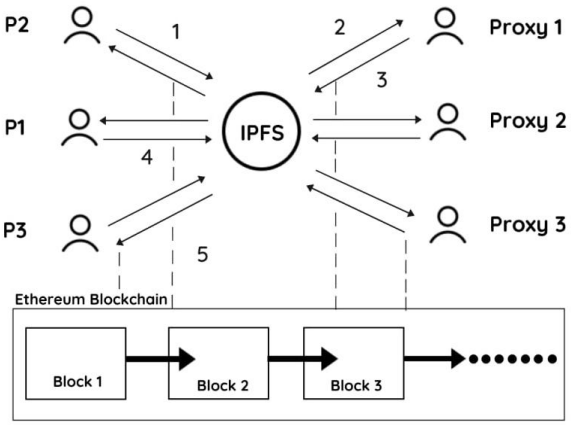


Fig. 5. Architecture - Computation

3) Decentralized NFT Management: The proposed architecture utilizes ERC-1155 extensively as the universal token standard. Each and every token is based on ERC-1155 as this will give consistency across all players and games.

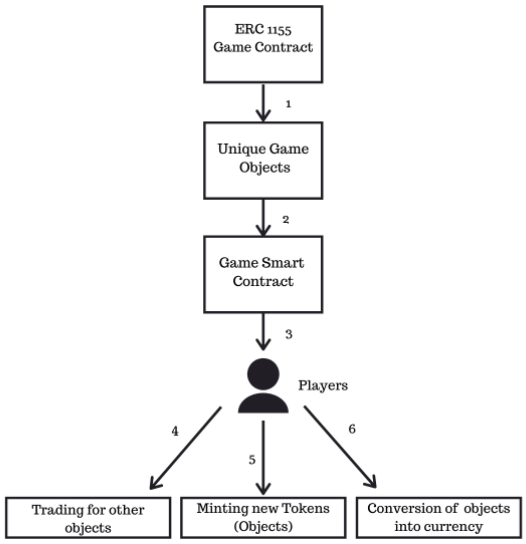


Fig. 6. Architecture - NFT Distribution

These steps involved in the NFT distribution are:

1) The creators of the game will frame the ERC-1155 contract concerning the game and attach them to the game objects present in the game using metadata.

2) The objects are rendered in the game based on the rules set in the game smart-contract.

3) Players receive objects in the game, and objects are added to their inventory, based on the Ethereum wallet.

Authorized licensed use limited to: Univ of Calif Santa Barbara. Downloaded on May 17,2021 at 18:05:47 UTC from IEEE Xplore. Restrictions apply.

4) Tokens received by the player can be traded with other players for other objects by exchanging them as ERC-1155 supports NFT transfers. The game object attached to the NFT is also transferred as the metadata is attached.

5) Players can mint the NFT by removing the metadata attached to the token and attach metadata related to another game object to it.

6) Players can also convert their NFTs to cryptocurrency directly without depending on any third party market places.

4) Fraud Detection: Gaming networks are prone to attacks and cheats every time. In order to protect these networks from attacks, a fail-proof monitoring mechanism can be deployed. In the proposed architecture, the nodes communicate using the IPFS protocol integrated with Ethereum Blockchain. Every network transaction performed by the player is recorded over the blockchain, as it maintains an immutable distributed ledger among all of its participants. Anomalies and attacks can be detected easily, as they are validated and seen by the entire network. In this way, fraudulent nodes can be eliminated from the network, which enables fair play. A private blockchain network can be used in order to reduce the latency in the network.

IV. Experimental Result

This section focuses on the evaluation of the proposed decentralized gaming architecture and its efficiency. Relying on the distributed proxy computation architecture reduced the latency and improved reliability in the gaming network. The below metric shows a comparison between Game Server and Proxy Computation.

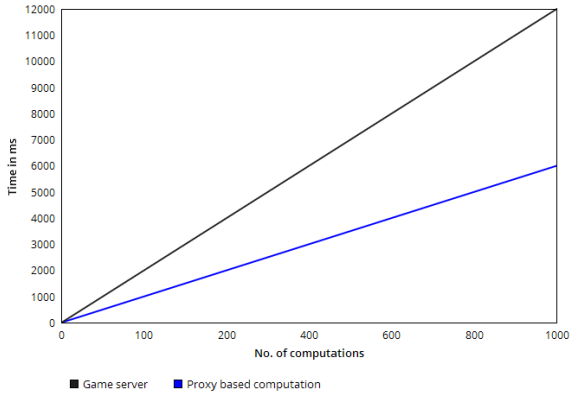


Fig. 7. Game Server vs Proxy Computation

V. Conclusion

The proposed architecture utilizes the decentralized network of Etherereum Blockchain which ensures the availability of nodes. In a centralized scenario, the availability of the game depends on the availability of the server. Whereas in a decentralized scenario there are independent proxy nodes that are available throughout. Although for it to be scalable the network should be large to an extent where there are enough populated proxy nodes.

The vision of this paper is to propose a fully decentralized gaming infrastructure. This paper also discussed the difficulties with the current centralized gaming networks and proposed a protocol for its complete decentralization. As the proposed protocol is a proof of concept, it is advised not to use it for production. Reliability on Ethereum 1.0 can slow down the network as it has a consensus mechanism based on proof of work. But this can be improved by integrating other proof of stake based blockchains. The smart contract programming language solidity is a fully functional programming language which can cause security issues with the NFTs, usage of other languages such as Vyper is preferred.

References

[1] K. Anderton, "The business of video games: Market share for gaming platforms in 2019 [infographic]," Jun 2019. [Online]. Available: https://www.forbes.com/sites/kevinanderton/2019/06/26/the-business-of-video-games-market-share-for-gaming-platforms-in-2019-infographic/6f39edfe7b25

[2] C. Gough, "Number of gamers worldwide 2021," Aug 2019. [Online]. Available: https://www.statista.com/statistics/748044/number-video-gamers-world/

[3] C. D. Merwin, "esports joins the big leagues," 2018. [Online]. Available: https://www.goldmansachs.com/insights/pages/infographics/e-sports/

[4] B. Ward, Y. Khmelevsky, G. Hains, R. Bartlett, A. Needham, and T. Sutherland, "Gaming network delays investigation and collection of very large-scale data sets," *2017 Annual IEEE International Systems Conference (SysCon)*, 2017.

[5] M. Claypool and D. Finkel, "The effects of latency on player performance in cloud-based games," *2014 13th Annual Workshop on Network and Systems Support for Games*, 2014.

[6] B. Mariano and S. G. M. Koo, "Is cloud gaming the future of the gaming industry?" *2015 Seventh International Conference on Ubiquitous and Future Networks*, 2015.

[7] D. Uribe, "Privacy laws, non-fungible tokens, and genomics," *The Journal of The British Blockchain Association*, vol. 3, no. 2, p. 1–10, 2020.

[8] V. Buterin, "Ethereum: A next-generation smart contract and decentralized application platform," Dec 2014.

[9] Nick, "Smart contracts: Building blocks for digital markets," *Organization of Phonetic Sciences, Amsterdam*, 1996. [Online]. Available: https://www.fon.hum.uva.nl/rob/Courses/InformationInSpeech/CDROM-/Literature/LOTwinterschool2006/szabo.best.vwh.net/-smartcontracts2.html

[10] R. Norvill, B. B. F. Pontiveros, R. State, and A. Cullen, "Visual emulation for ethereums virtual machine," *NOMS 2018 - 2018 IEEE/IFIP Network Operations and Management Symposium*, 2018.

[11] J. Benet, "Ipfs - content addressed, versioned, p2p file system," Jul 2014. [Online]. Available: https://ipfs.io/ipfs/QmR7GSQM93Cx5eAg6a6yRzNde1FQv7uL6X1o4k-7zrJa3LX/ipfs.draft3.pdf

[12] A. Auvolat and F. Taiani, "Merkle search trees: Efficient state-based crdts in open networks," *2019 38th Symposium on Reliable Distributed Systems (SRDS)*, 2019.

[13] Kingma, F. H., Abbeel, Pieter, and Jonathan, "Bit-swap: Recursive bits-back coding for lossless compression with hierarchical latent variables," Oct 2019. [Online]. Available: https://arxiv.org/abs/1905.06845

[14] S. Chevet, "Blockchain technology and non-fungible tokens: Reshaping value chains in creative industries," *SSRN Electronic Journal*, 2018.

[15] V. Buterin and F. Vogelsteller, "Eip 20: Erc-20 token standard," Nov 2015. [Online]. Available: https://eips.ethereum.org/EIPS/eip-20

[16] W. Entriken and D. Shirley, "Eip 721: Erc-721 non-fungible token standard," Jan 2018. [Online]. Available: https://eips.ethereum.org/EIPS/eip-721

[17] W. Radomski, A. Cooke, and P. Castongua, "Eip 1155: Erc-1155 multi token standard," Jun 2018. [Online]. Available: https://eips.ethereum.org/EIPS/eip-1155

Authorized licensed use limited to: Univ of Calif Santa Barbara. Downloaded on May 17,2021 at 18:05:47 UTC from IEEE Xplore. Restrictions apply.

2023 IEEE IAS Global Conference on Emerging Technologies (GlobConET)
Loughborough University, London, United Kingdom. May 19-21, 2023

2023 IEEE IAS Global Conference on Emerging Technologies (GlobConET) | 979-8-3503-3179-0/23/$31.00 ©2023 IEEE | DOI: 10.1109/GlobConET56651.2023.10150013

A Blockchain based framework for Lending Digital Assets implemented using NFT

Darshan M[1], S.R Raswanth[1], Priyanka Kumar[1,2], and Gautam Srivastava[3]

[1]Dept. of Computer Science and Engineering, Amrita School of Computing, Coimbatore, India.
Email: {cb.en.u4cse19126, cb.en.u4cse19648}@cb.students.amrita.edu

[2]Department of Computer Science, University of Texas at San Antonio, San Antonio, USA
Email: priyanka.kumar@utsa.edu

[3]Department of Mathematics and Computer Science, Brandon University, Brandon, Canada.
Email: srivastavag@brandonu.ca

***Abstract*—In recent times, digital assets have been on the rise. Crypto-enthusiasts have been investing in this market segment for quite some time. The returns from these investments for the investors cannot be fathomed. Blockchain Technology is one of the main pillars of the Industry 4.0 revolution. As of now, the entities can buy or sell digital assets but the ability to lend/rent a digital asset is still a question that needs to be addressed. With this motivation, we have proposed a framework as a solution that will allow owners to lend their digital assets to buyers by determining the ownership rights of the asset dynamically. This research work presents the comprehensible approach to the buyer-seller protocol using NFT smart contracts and blockchain technology, to provide proof of ownership and secure transferability in the marketplace.**

***Index Terms*—Non-fungible tokens (NFTs),Proof of Ownership, Blockchain, Decentralized Autonomous Organization (DAO).**

I. Introduction

Non-Fungible Tokens (NFTs) became a sensation in the changing market, with many wealthy celebrities purchasing them as a trend. At first, the general public was uncertain about the use of NFTs, but over time, the most beneficial application emerged: using NFTs as a pass to exclusive events. These events typically included concerts, luxurious parties, and access to yacht and golf clubs. Participating in such events offered ample networking opportunities, which resulted in significant business profits. Gradually, people recognized the true potential of these events and, more crucially, comprehended the value of the ”ticket.”(i.e. NFT) for such events [1]. One of the limitations of this “ticket” is that the owner has to be physically present in the region of these events provided it happens in the real world. With global destinations for the organizers to choose from it becomes very difficult for the owner to join every such event. These days, not everyone can afford to buy the NFT, yet many people aspire to have access to such events. There is a huge untapped market that is interested in leasing out their NFTs and an even bigger market that is interested in renting the same. With the recent global trends, one can easily understand the potential of blockchain and NFTs. Many people regret not investing in NFTs and this solution brings them closer to owning one. The Decentralized Autonomous Organization (DAO) we propose would comprise intuitive smart contracts which would automate the entire lending process making it user-friendly for both the lender and the renter. The blockchain-powered smart contract would take in the date and time of leasing and when it expires the ownership of the NFT would be reverted to the lender. This way the renter could own the ticket for a few days and attend exclusive events. The lender gets equivalent profits based on the NFT price and market demand. The proposed framework takes both the lender’s and renter’s requirements into consideration and provides a solution with exactly what they require. This framework could be utilized to lend/rent any form of digital assets, making the whole system transparent and accessible to the entities involved.

The paper is organized as follows. In Section II, we cover some related works about Non-Fungible Tokens and Blockchain Smart Contracts. Section III provides a brief explanation of the proposed methodology. Section IV deals with the implementation of blockchain-powered smart contracts. Section V describes the results obtained from the proposed research implementation. Finally, Section VI concludes the research work and considers the future aspects of the proposed framework.

II. Related Works

In [2], the authors have drawn an extensive view of non-fungible tokens and how they are being used in this day and age. The underlying technology Blockchain is also discussed in the same. With the wide variety of features that blockchain has to offer such as transparency and authenticity, NFTs make wise use of them. NFTs are fundamentally different from cryptocurrencies in terms of their purpose but it is still being treated the same way for trading. NFTs play a major role in promoting digital art and brands like CryptoPunks, CryptoKitties etc. were able to capitalize on the same. The authors give a brief insight into some of the technical components and terminologies used in the entire research work. The entire workflow for an NFT system is simplified with

Authorized licensed use limited to: Zhejiang University. Downloaded on July 09,2025 at 07:25:54 UTC from IEEE Xplore. Restrictions apply.

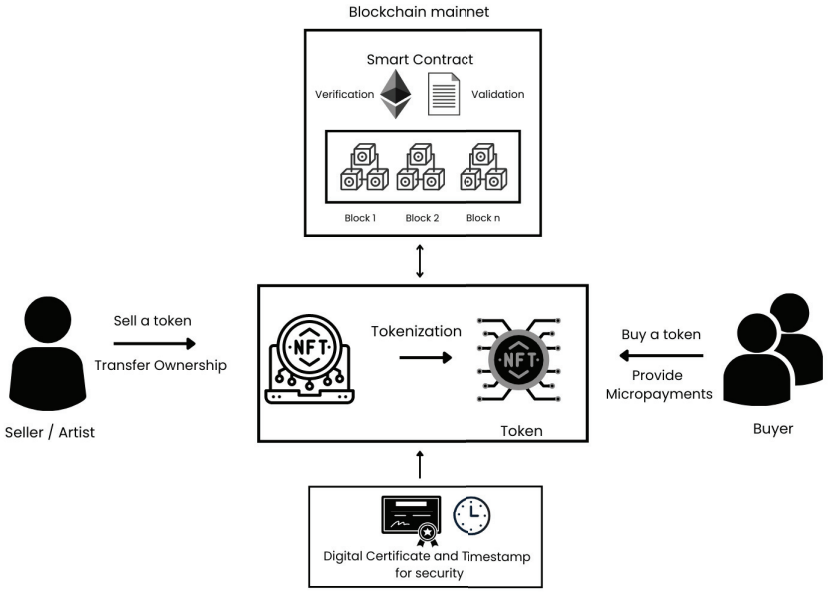


Fig. 1. Workflow of the proposed methodology

the NFT owner/ creator and NFT buyer being the only two stakeholders. This research works to provide a deep insight into the existing standards of the NFT market. Some of the desired properties of a non-fungible token are Verifiability, Transparent Execution, Availability, Tamper-resistance, Usability, Atomicity, and Tradability. Taking all this into account the proposed research work tries to expand this concept by adding a new stakeholder to the workflow. The proposed entity, "NFT Lender" would act as a temporary sale based on timestamp. The proposed solution ensures that all the desired properties are retained and verified. When focusing on security concerns, all security aspects such as authenticity, integrity, non-reputability, availability, and access control need to be taken care of. The proposed solution ensures that all of the above-discussed security concerns are addressed. Some of the other sections of the research paper include future potential and opportunities, challenges, and Security & Privacy Issues. This research paper dwells deep into the entire existing NFT marketplace procedure and helps anyone with little to no knowledge about NFT to get some perspective.

In [3], NFTs are treated as an investment option, especially for people possessing cryptocurrencies. The research work starts with the introduction of non-fungible tokens and then gradually shifts its focus toward the monetization of the ecosystem. The author presents an interesting take on how NFTs transformed into an existing asset class. The paper describes the mentality of generic investors and how they viewed NFTs as just another investment opportunity. The research work gives an insider view as to what goes into making an NFT and getting it up in the marketplace. Right from minting, and trading to auctioning, all the steps are discussed in detail. In the NFT marketplace, two more closely related concepts are used namely NFT liquidity mining and NFT farming. NFT liquidity mining is a methodology with two primary goals, creating NFT deposits (providing liquidity) on the NFT platform, Two generating a profit for the NFT investor for making the NFT deposit. The two methodologies are similar to the buy-and-hold strategy. A brief insight is also given on NFT collateral-based loans and Fractional NFT. Keeping all these factors in mind the dataset is collected and analyzed. From the research work, a lot of insights can be inferred related to the risk and returns attributes. The extensive research work helps in understanding the dataset from a better perspective and the proposed research takes a more optimistic view of the future of NFTs.

In [4], the authors provide a few great insights. The research work starts deals primarily with the latest in the NFT marketplace. The article elaborates on the historic rise of NFTs and the newer use case scenarios where NFTs fit in as the lost puzzle piece. The article mentions a few use cases of NFTs ranging from art and collectibles to gaming and metaverse. Though the article draws a broad concept as to what might happen, readers need to be aware that those are concepts, and turning them into a reality would require a lot of development and open-minded technology enthusiasts to accept the new products. The later part of the paper deals with the financial aspects of NFT trading. The curve (graph) explains the entire NFT boom with accurate timelines. To understand the profitability of an NFT one has to analyze the risk versus the reward domain and make sure that they fall in any category before the "late majority". Expanded De-Fi, Dematerializing Real World Assets and Supply Chain Management are some of the other use cases where smart contracts would substitute for legal or official paperwork.

Authorized licensed use limited to: Zhejiang University. Downloaded on July 09,2025 at 07:25:54 UTC from IEEE Xplore. Restrictions apply.

III. Proposed Methodology

With the detailed literature survey, we propose the following methodology that includes an NFT smart contract for digitalizing the digital assets lending system [5]. A smart contract is a computer-based protocol that allows the entity involved in the transaction to interact with each other [6]. NFT is a unique proof of ownership for a particular digital asset that enables the entities involved to collect and trade the digital versions of collectibles [7]. The creation of an NFT involves the creation of a smart contract and storing it on the blockchain through a process called minting. To ensure transparency and immutability of the NFT transaction, the information is stored on the blockchain using smart contracts. The primary objective of the NFT smart contract is to verify the proof of ownership and to handle the transferability of the particular trading asset.

The conventional way of lending using an asset as collateral, suffers from a lack of transparency and data breaching which negatively impacts the credit score of the user [8]. To encounter the above-stated complications, a digital contract such as a smart contract would make the entire system traceable and highly secure due to the underlying blockchain layer. The proposed system emphasizes the idea of providing the financial needs to the decentralized autonomous organizations using NFT as assets, thereby introducing the Decentralized Finance (DeFi) protocol using smart contracts. Fig. 1 depicts the overall workflow of the proposed methodology between a buyer and a seller. This Proposed Methodology is broadly classified into three subsections namely: Tokenization of NFTs, Proof of Ownership, and Token Security.

A. Tokenization of NFTs

The term token corresponds to the concept of a privately issued asset that has a value associated with it. The methodology of tokenizing the real-world asset into its digital form makes it easier to transfer, mint, and trace the entire flow with full transparency. This enables the entities to transfer or lend a particular digital asset such as an NFT across borders securely. This establishes a self-sustainable micro-economics known as Tokenomics that helps decentralized autonomous organizations manage their financial requirements through self-determined stakeholders around the world.

```
struct Token {
        address owner;
        uint256 balance;
        string certificate;
        string name;
        uint32 price;
        uint256 timestamp;
        bool isSold;
    }

    mapping(address => Token)
    public tokens;
```

Listing 1. Structure of an NFT after tokenization.

The above code snippet describes the structure of a particular NFT after undergoing the process of tokenization. Solidity is a high-level contract-oriented language that is used to create executable smart contracts on EVM. Structs are custom-defined types, formed by grouping several variables with multiple properties. Mappings can be visualized as hash tables that comprise key-value pairs.

The next process in the workflow is to perform basic operations such as adding, updating, and selling a particular token to a potential buyer which is illustrated in Fig. 2. The Buyer-Seller protocol implemented in the smart contract is developed to such a degree that the proposed system would cover all the edge cases in the process workflow.

```
function sellToken(address _owner,
uint256 _amount) public {
        Token memory token = tokens[_owner];
        token.balance -= _amount;
        token.isSold = true;
        tokens[_owner] = token;
    }
```

Listing 2. Code Snippet for Selling a Token.

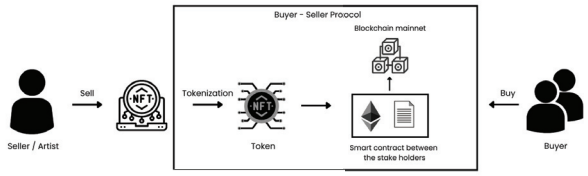


Fig. 2. Buyer-Seller Protocol

B. Proof of Ownership

Ownership of a token is a primary aspect of the trading industry as it determines the possession of a particular asset at that specific moment in time. This regulates the rights of the owner, which narrows down to recording the activities to provide proof of ownership with the help of smart contracts. The smart contract is designed in such a way that verification of the proof of ownership is registered on the blockchain. This improves the traceability of the NFT that is being borrowed or owned by the individuals in the distributed network of entities to facilitate the smooth flow of business. Another important aspect is transparency in the chain of the NFT due to the shared data which is accessible by the legitimate buyers/sellers present in the network. There are mathematically enforced rights enforced in the smart contract developed in the proposed system that guarantees the right to proof of ownership.

To determine and prove the ownership of the particular, we defined a protocol that aggregates the tokenized data using a hash structure called Merkle Tree which allows secure and efficient verification of the content in a huge body of data [1]. Merkle Tree is constructed by repeated hashing pairs of nodes until only one node (Merkle Root) is left through a Bottom-Up approach. Each leaf node is a hash of transactional data

Authorized licensed use limited to: Zhejiang University. Downloaded on July 09,2025 at 07:25:54 UTC from IEEE Xplore. Restrictions apply.

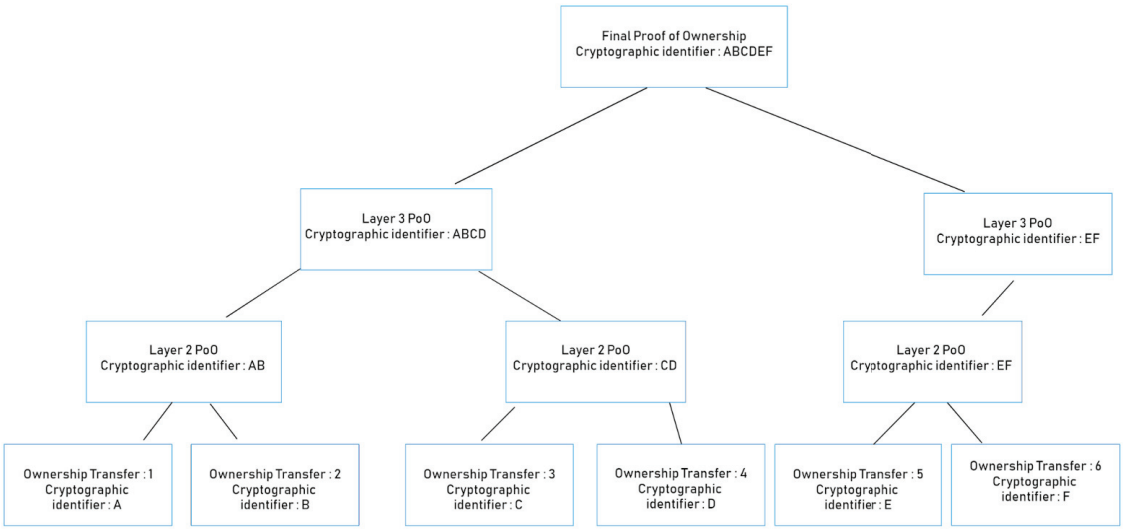


Fig. 3. Proof of Ownership using Merkle Tree

(Transfer of ownership) and each non-leaf data is a hash of previous values. This hierarchical structure has been visualized in Fig. 3, which helps us to understand the provided proof of ownership of all hashed tokens within the tree by storing the Merkle Root on the Blockchain. Integrity and Validation of data are maintained through this structure in the proposed solution.

```
function transferToken(address _owner,
address _newOwner) public {
        Token memory token = tokens[_owner];
        token.owner = _newOwner;
        tokens[_owner] = token;
    }
```

Listing 3. Code Snippet for transferring token.

The above code snippet describes the use case of transferring the ownership of the token. To achieve this a public function named transfer token has been defined. This particular function could be called inside the contract and inside contracts that inherit the main contract. Since the process of transferring ownership is dynamic, the data is handled by memory variables to provide temporary data storage.

C. Token Security

The structure of the token has been designed in such a way that there is a digital certificate and timestamp to protect the NFT from cyber attacks and preserve the security measures of the proposed system. The metadata of the NFT transaction is signed as a digital certificate by the seller/buyer's wallet address, which is further hashed using the SHA-256 hashing algorithm. The output is assigned to a recipient. The timestamp is an important factor in the token which determines the proof of the existence of the asset that is being traded. This prevents fraudulent transactions and minimizes the risk factor in the proposed methodology. The inclusion of timestamps in the tokens also ensures the removal of copyright claim issues.

```
function copyright(address _newowner,
uint256 _amount) public {
        Token memory token =
        tokens[_newowner];
        token.balance -= _amount;
        tokens[_newowner] = token;
    }

function verifyCertificate(address _owner,
string memory _certificate)
public view returns (bool){
        Token memory token =
        tokens[_owner];
        return
            keccak256(abi.encodePacked
            (token.certificate)) ==
            keccak256(abi.encodePacked
            (_certificate));
}
```

Listing 4. Code Snippet for Copyright & Verification for the token certificate.

The above code snippet describes the provision of copyright and verification of the certificate of the token. The verifyCertificate function returns a boolean value after the verification of the digital certificate. This authentication mechanism ensures the security of the proposed methodology from disputes and vulnerabilities.

```
function pay(address _owner, uint256 _amount)
public {
        Token memory token = tokens[_owner];
        token.balance += _amount;
        tokens[_owner] = token;
    }
```

Listing 5. Code Snippet for Buying a token.

Another important use case is the temporary usage of the token while preserving the copyright to ensure transferrable ownership between the entities in a trustable manner. To achieve this, a public function called pay has been defined in the proposed system, which enables the person who wants to use the particular to send

Authorized licensed use limited to: Zhejiang University. Downloaded on July 09,2025 at 07:25:54 UTC from IEEE Xplore. Restrictions apply.

a micropayment to the present owner. This prevents copyright claim issues through the use of smart contracts by achieving a consensus between the entities involved.

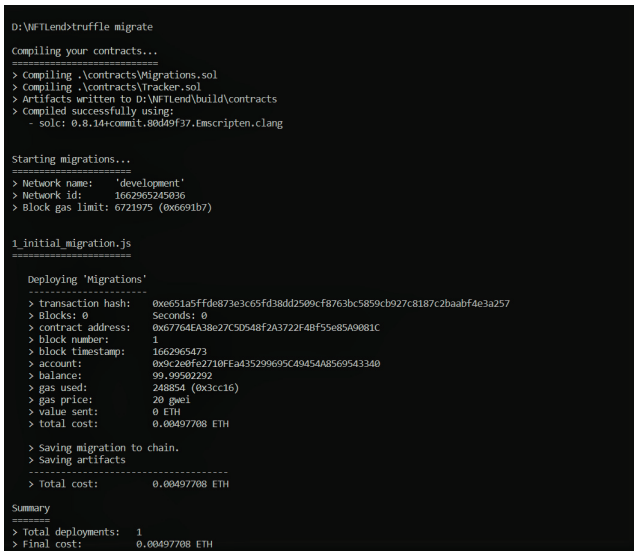


Fig. 4. Compilation and Migration of Contracts

IV. Implementation and Results

A. Ethereum Virtual Machine

A smart contract is a digitalized ledger and contractual clauses that execute through a programmable code. The code and agreement are enforced between the entities which exist across a distributed, decentralized blockchain network. These self-executing programs are triggered when the defined pre-composed conditions are met. The buyer-seller protocol of the proposed NFT lending system is embedded in the form of automated code logically through smart contracts. A high-level programming language called solidity is used for smart contract development which runs on Ethereum Virtual Machine (EVM) [9]. EVM is essential to provide a level of abstraction between the executable smart contract and executing machine. An EVM compilers the smart contract into a low-level machine instruction called opcodes. To avoid computationally expensive operations which lead to a slowdown of the Ethereum network, every opcode has its own gas cost. The Contract calls require a piece of data that records all functions (along with the arguments) and events called "ABI" (Application Binary Interface).

B. Test Environment

To compile the developed smart contract in our system, tools such as Truffle and Ganache CLI were utilized. Truffle is a node.js framework that is used to create, compile and deploy smart contracts to the Ethereum network [10]. Ganache CLI is an Ethereum client service that enables a developer to simulate a local blockchain for testing decentralized applications. Ganache CLI automatically creates 10 accounts each having its private keys and 100 ethers for testing purposes (See Fig. 5).

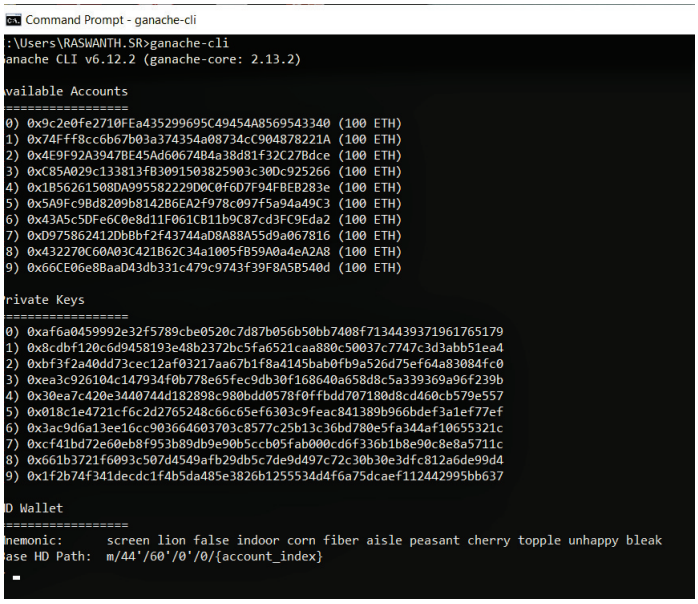


Fig. 5. Ganache's local Blockchain network

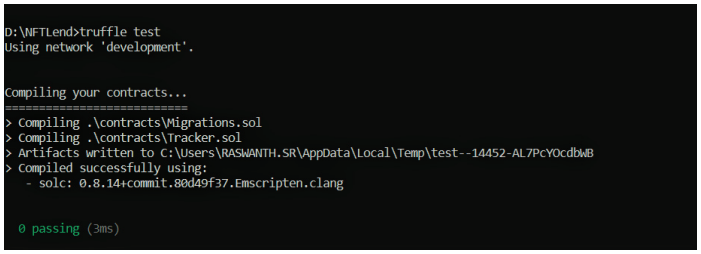


Fig. 6. Testing of Smart Contract

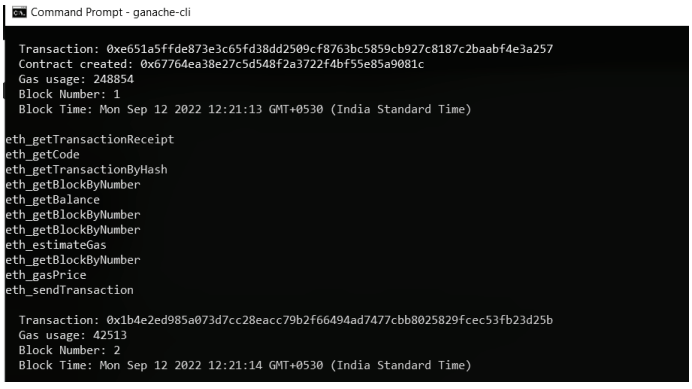


Fig. 7. Generation of new blocks in the Ganache CLI

V. Results Obtained

After setting up the ganache client service, the address of the localhost is set to 127.0.0.1 and port as 8545. This completes the configurational setup of our local blockchain

Authorized licensed use limited to: Zhejiang University. Downloaded on July 09,2025 at 07:25:54 UTC from IEEE Xplore. Restrictions apply.

network and the smart contracts developed are compiled using the command 'truffle compile' command [11]. The next step is to deploy our contract onto the network using the command 'truffle migrate' (See Fig. 4). On successful migration, new blocks would be created in the ganache console (See Fig. 7). The testing of the instances of the contract with the right set of parameters would return true in the truffle console indicating a successful development of the contract which is depicted in Fig. 6. The Ethereum network has been extensively utilized to develop and migrate the smart contract for the NFT lending system. The immutable and traceable nature of the blockchain enhances the proposed methodology to provide proof of ownership through tokenization and digitization through smart contracts.

VI. Conclusion

The objective of this research work is to provide a cumulative solution for the NFT lending environment through digitization using distributed ledger-based technology. This proposed methodology has been implemented and tested in an Ethereum-based environment to provide rapid deployment service and high performance. Verification of ownership and transferability of NFT is handled by the defined functions and buyer-seller protocol implemented in the smart contract. With the implication of digital certificates and cryptographic algorithms, the security aspects of the system are preserved from any malicious means. The decentralized architecture of the system enables every stakeholder involved in the lending system to instill trust and confidence to create NFT tokens and unlock new revenue streams by lending them in these marketplaces. The future aspect of this research study is to analyze the adaptation of the protocol in different forms of digital assets and risk management measures to be taken by utilizing the necessary security and machine learning mechanisms.

References

[1] M. Mazur, "Non-fungible tokens (nft). the analysis of risk and return," *Available at SSRN 3953535*, 2021.

[2] U. W. Chohan, "Non-fungible tokens: Blockchains, scarcity, and value," *Critical Blockchain Research Initiative (CBRI) Working Papers*, 2021.

[3] A. Park, J. Kietzmann, L. Pitt, and A. Dabirian, "The evolution of nonfungible tokens: Complexity and novelty of nft use-cases," *IT Professional*, vol. 24, no. 1, pp. 9–14, 2022.

[4] F. Valeonti, A. Bikakis, M. Terras, C. Speed, A. Hudson-Smith, and K. Chalkias, "Crypto collectibles, museum funding and openglam: challenges, opportunities and the potential of non-fungible tokens (nfts)," *Applied Sciences*, vol. 11, no. 21, p. 9931, 2021.

[5] M. Amet, D. M, G. Srivastava, and J. Crichigno, "A cross-chain interoperability architecture for smart city environments," in *2022 IEEE Globecom Workshops (GC Wkshps)*, 2022, pp. 263–268.

[6] L. S. Sankar, M. Sindhu, and M. Sethumadhavan, "Survey of consensus protocols on blockchain applications," in *2017 4th international conference on advanced computing and communication systems (ICACCS)*. IEEE, 2017, pp. 1–5.

[7] V. Buterin, "A next generation smart contract & decentralized application platform," 2015.

[8] Q. Wang, R. Li, Q. Wang, and S. Chen, "Non-fungible token (nft): Overview, evaluation, opportunities and challenges," 2021.

[9] A. M. Thomas, R. Ramaguru, and M. Sethumadhavan, "Distributed identity and verifiable claims using ethereum standards," in *Inventive Communication and Computational Technologies: Proceedings of ICICCT 2021*. Springer, 2022, pp. 621–636.

[10] M. Darshan, S. Raswanth, S. Skandan, S. Shakthi Saravanan, R. Chandramohanan, and P. Kumar, "A secured blockchain based facial recognition system for two factor authentication process," in *Innovations in Electrical and Electronic Engineering: Proceedings of ICEEE 2022, Volume 2*. Springer, 2022, pp. 492–502.

[11] D. M, S. Raswanth, S. V. V. S. Akella, and P. Kumar, "A secured distributed ledger based fundraising framework using smart contracts," in *2021 IEEE 4th International Conference on Computing, Power and Communication Technologies (GUCON)*, 2021, pp. 1–5.

Authorized licensed use limited to: Zhejiang University. Downloaded on July 09,2025 at 07:25:54 UTC from IEEE Xplore. Restrictions apply.

Demo: Non-Fungible Tokens in Asset-Backed Securitization

Vyacheslav Davydov
HSE University
Moscow, Russia
v.davydov@hse.ru
Quicktoken Tech FZ-LLC
Dubai, UAE

Alexander Krymov
HSE University
Moscow, Russia
apkrymov@edu.hse.ru

Deniz Ozmaden
HSE University
Moscow, Russia
dozmaden@edu.hse.ru

Yaroslav Pashchenko
HSE University
Moscow, Russia
yavpaschenko_1@edu.hse.ru

Alexander Tenyaev
HSE University
Moscow, Russia
aitenyaev@edu.hse.ru

Yury Yanovich
Skolkovo Institute of Science and Technology
Moscow, Russia
Quicktoken Tech FZ-LLC
Dubai, UAE

2023 IEEE 43rd International Conference on Distributed Computing Systems (ICDCS) | 979-8-3503-3986-4/23/$31.00 ©2023 IEEE | DOI: 10.1109/ICDCS57875.2023.00114

***Abstract*—This paper demonstrates the use of blockchain technology in asset-backed securitization (ABS) and presents Quicktoken, a blockchain platform for ABS. Financial institutions can use Quicktoken to assign a correspondence between initial assets and securities, deploy smart contracts for securities issuance, and store the correspondence between assets and non-fungible tokens (NFTs) on the blockchain. Investors can buy, sell, and get dividends upon securities redeem via an Android application, while all transactions are secured by the blockchain. The paper highlights the advantages of blockchain usage in ABS, such as a diversification without a loss of auditability.**

***Index Terms*—Blockchain; Smart contract; NFT; Securization; Asset-Backed Securitization**

I. Introduction

The crypto industry is gradually becoming closer to real areas of activity and is more applied in such areas as banking sector and investment companies. Banking technologies are conservative and still use outdated solutions that are reliable but have numerous drawbacks [1]–[3]. Financial institutions own balanced portfolios with different types of assets, resulting in a low-risk investment [4]. At the same time, assets cannot be split into smaller parts with comparable risk for resale owing to regulators' restrictions caused by a lack of auditability [5]. Blockchain is a backbone technology for crypto [6], [7]. It provides data storage with an unchangeable transaction log and audit instruments [8]–[10]. Blockchain allows to split assets into smaller parts in an auditable manner with smart contracts–to issue securities [11], [12]. The popular format of securities in smart contracts is a non-fungible token (NFT) [13]. NFTs do not mix and are easy to trace back to the initial assets.

The general asset-backed securitization (ABS) workflow with blockchain is as follows [14], [15]. A financial institution has a portfolio. The portfolio consists of assets with certain economic parameters, including volume, expected income and risk. Based on ABS objectives, the institution assigns a correspondence between initial assets and securities. The correspondence can be viewed as a matrix with assets in rows, securities in columns, and the amount of a given asset in a given security at the intersection of the row and the column [16]. The institution deploys a smart contract for the securities issue to the blockchain. A single NFT represents a security. The collection stores the correspondence between initial assets and NFTs. Investors can buy securities as NFTs, sell them on a secondary market or wait until the institution redeems them.

In the demo, we present Quicktoken–blockchain platform for ABS. The platform consists of backend and Android application both interacting with a blockchain. Financial institutions can perform ABS via an application programming interface (API). Investors can buy, sell and redeem securities via Android application. All the transactions are secured by a blockchain. A backend administrates the access in smart contracts and cashes the data to guarantee good user experience when reading from blockchain. The source code is available on Github [17].

II. Project Description

A. User Flow

Quicktoken platform redeems a portfolio from a financial institution and issues securities as QTK NFT tokens. QTK tokenized portfolios are available for purchase on the Quicktoken platform and on the secondary market. It is possible to purchase a tokenized portfolio for the first time only for the native token of the platform–QTKX. The purchase of already issued portfolios is possible on the secondary market for fiat currencies and cryptocurrencies. Upon expiration of the lifetime of securities, the financial institution buys them back in accordance with predetermined conditions. At the

The article was prepared within the framework of the Basic Research Program at HSE University.

2575-8411/23/$31.00 ©2023 IEEE
DOI 10.1109/ICDCS57875.2023.00114

Authorized licensed use limited to: Zhejiang University. Downloaded on July 09,2025 at 07:22:53 UTC from IEEE Xplore. Restrictions apply.

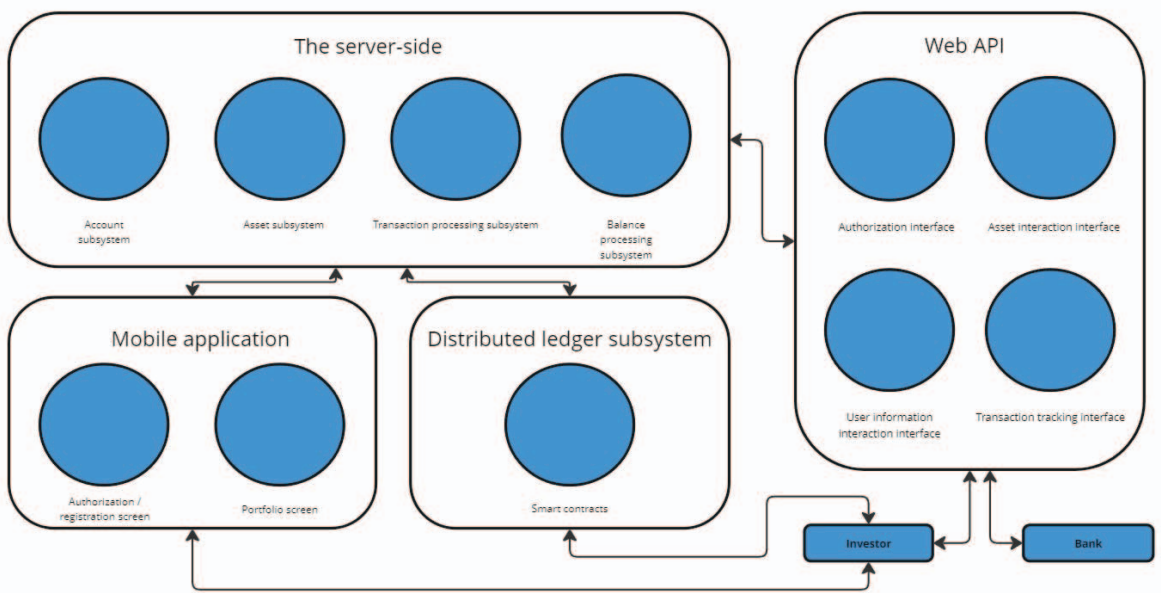


Fig. 1. Demo architecture

same time, NFTs are burned, and investors receive profit in a currency.

B. Architecture

The demo comprises four parts: a mobile application, an application server, Web API and a distributed ledger (see Figure 1).

The mobile application is an Android operating system application through which an investor accesses the system. It consists of an authorization/registration module and a portfolio screen. In the authorization/registration module, the user can register or log in as the owner of the crypto wallet. The portfolio screen allows the investor user to buy and sell securities, and get information about securities.

The application server is a set of modules responsible for the off-chain part of the project. It consists of an authentication processing module responsible for creating user accounts, a balance processing module responsible for presenting to the user his list of assets, an asset system responsible for interacting (purchasing and creating) with QTK and QTKE, and a transaction processing subsystem responsible for processing state of transactions in the blockchain network.

Web API enables users to access system functionality through a Swagger facade in a browser. It includes methods for authorization/registration, getting account info and balance history, creating and transferring QTKs, and checking transaction status.

The distributed ledger subsystem is a part of the project located in the blockchain network. It includes executable smart contracts, interaction with which, namely, access to their methods, allows the emission and circulation of QTK and QTKE. The blockchain is maintaned by third-parties, and Quicktoken is a client application for it.

C. Technology Stack

We use Remix IDE to develop smart contracts in the Solidity language with the help of the OpenZeppelin library Contracts Wizard.

For the backend, we use the C# programming language with JetBrains Rider and incorporate libraries such as MVC, Nethereum, EntityFrameworkCore, and Autofac to ensure seamless functionality.

Our Android mobile application is built using Kotlin and utilizes Jetpack Compose, WalletConnect SDK, OkHttp, and Retrofit to provide users with a smooth and reliable experience. User registration supports MetaMask and Wallet Connect.

The implemented platform is compatible with any Ethereum Virtual Machine (EVM) blockchain, and our demo specifically works with the Ethereum public testnet known as Sepolia.

We host our project repository on the Azure DevOps platform to ensure efficient collaboration and development.

III. Demo Overview

The demo workflow consists of four main steps.

A. Securitization

A financial institution decides to securitise its portfolio. Each asset in the portfolio has an individual set of economic parameters, such as interest rate, expiration date, quantity, and value in QTKX. The platform operator generates the issue of securities (see Figure 2).

B. Buy Security

An investor logs into the QuickToken mobile application on the Android operating system and completes authorization using Wallet Connect technology, which grants them access to the platform (see Figure 3). After successful authorization, the

Authorized licensed use limited to: Zhejiang University. Downloaded on July 09,2025 at 07:22:53 UTC from IEEE Xplore. Restrictions apply.

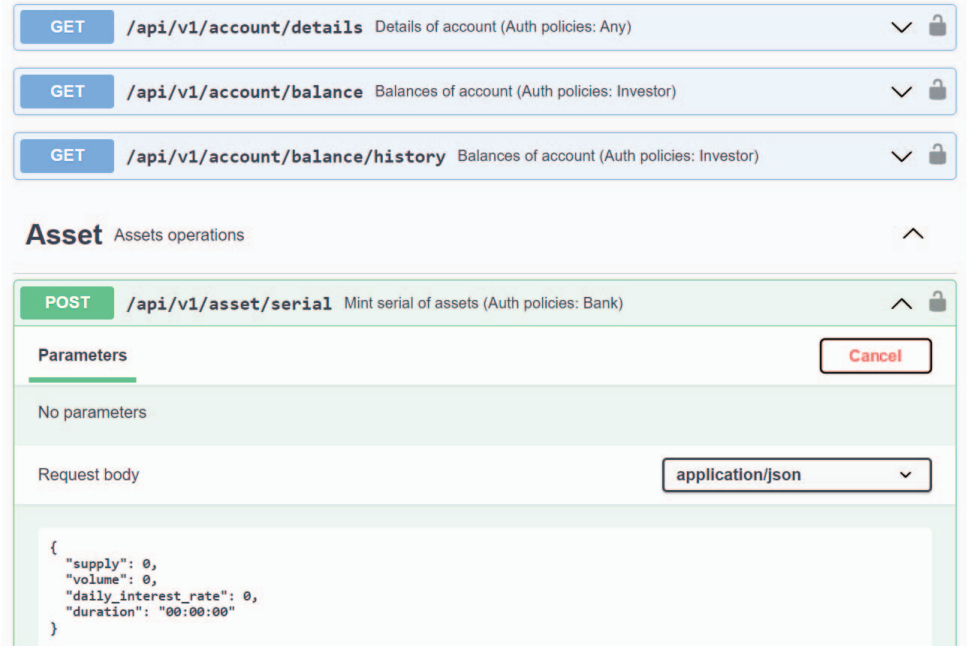


Fig. 2. Issue of securities via Web API

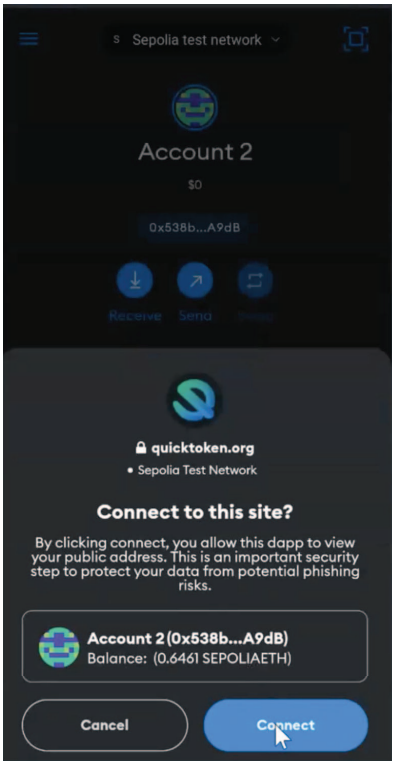


Fig. 3. Authorization using WalletConnect

Fig. 4. Main screen of the mobile application

investor enters her personal account, where she have access to all information regarding her interactions with the platform. In this section, the investor can view her wallet address, balance in QTKX and ETH, see the changes in the total current profit,

Authorized licensed use limited to: Zhejiang University. Downloaded on July 09,2025 at 07:22:53 UTC from IEEE Xplore. Restrictions apply.

Fig. 5. Trading screen

and receive information on current assets and expected profits (see Figure 4).

C. Get Revenue

The investor also has the ability to sell her assets before their expiration date. In the Trading screen, the investor can review a list of all available tokens, on which a daily interest rate is charged (see Figure 5). Using the search function, the investor can choose the most optimal asset, taking into account her needs, capabilities, and token characteristics. After the security expiration date, the investor claims revenue and burns expired NFT.

IV. Conclusion

In conclusion, the integration of blockchain technology in asset-backed securitization (ABS) has significant benefits for financial institutions and investors alike. Quicktoken, a blockchain platform for ABS, allows for the assignment of correspondence between initial assets and securities, deployment of smart contracts for securities issuance, and storage of the correspondence between assets and non-fungible tokens (NFTs) on the blockchain. With the ability to split assets into smaller parts in an auditable manner with smart contracts and the ease of tracing NFTs back to the initial assets, Quicktoken provides a more efficient and secure way for financial institutions to manage their portfolios and for investors to buy and sell securities. As blockchain technology continues to advance and become more widely adopted, we can expect to see further innovation and development in the ABS space.

References

[1] V. A. Davydov, S. A. Kruglik, and Y. A. Yanovich, "Comparison of Banking and Peer-to-Peer Lending Risks," *Automation and Remote Control*, vol. 82, no. 12, pp. 2155–2168, 12 2021. [Online]. Available: https://link.springer.com/article/10.1134/S0005117921120079

[2] R. Patel, M. Migliavacca, and M. E. Oriani, "Blockchain in banking and finance: A bibliometric review," *Research in International Business and Finance*, vol. 62, p. 101718, 12 2022.

[3] H. Benedetti and G. Rodríguez-Garnica, "Tokenized Assets and Securities," *The Emerald Handbook on Cryptoassets: Investment Opportunities and Challenges*, pp. 107–121, 1 2023.

[4] H. Markowitz, "Portfolio Selection," *The Journal of Finance*, vol. 7, no. 1, pp. 77–91, 1952.

[5] J. Putnis, Ed., *Banking Regulation Review*, 13th ed. London, UK: Law Business Research Ltd, 2022. [Online]. Available: www.TheLawReviews.co.uk

[6] S. Nakamoto, "Bitcoin: A Peer-to-Peer Electronic Cash System," *www.bitcoin.org*, pp. 1–9, 2008. [Online]. Available: https://bitcoin.org/bitcoin.pdf

[7] V. Buterin, "On Public and Private Blockchains - Ethereum Blog," 2015. [Online]. Available: https://blog.ethereum.org/2015/08/07/on-public-and-private-blockchains/

[8] Bitfury Group, "On Blockchain Auditability," *bitfury.com*, pp. 1–40, 2016. [Online]. Available: https://bitfury.com/content/downloads/bitfury_white_paper_on_blockchain_auditability.pdf

[9] S. Kruglik, K. Nazirkhanova, and Y. Yanovich, "Challenges beyond blockchain: scaling, oracles and privacy preserving," in *2019 XVI International Symposium "Problems of Redundancy in Information and Control Systems" (REDUNDANCY)*. IEEE, 10 2019, pp. 155–158. [Online]. Available: https://ieeexplore.ieee.org/document/9003331/

[10] V. Ermolaev, I. Klangberg, Y. Madhwal, S. Vapper, S. Wels, and Y. Yanovich, "Incorruptible Auditing: Blockchain-Powered Graph Database Management," in *2020 IEEE International Conference on Blockchain and Cryptocurrency (ICBC)*, 2020, pp. 101–103.

[11] V. Davydov and Y. Yanovich, "Financial Instruments Generation via Tokenization into Commodity," in *2020 2nd Conference on Blockchain Research & Applications for Innovative Networks and Services (BRAINS)*. IEEE, 9 2020, pp. 25–29. [Online]. Available: https://ieeexplore.ieee.org/document/9223295/

[12] W. Pan and M. Qiu, "Application of Blockchain in Asset-Backed Securitization," *Proceedings - 2020 IEEE 6th Intl Conference on Big Data Security on Cloud, BigDataSecurity 2020, 2020 IEEE Intl Conference on High Performance and Smart Computing, HPSC 2020 and 2020 IEEE Intl Conference on Intelligent Data and Security, IDS 2020*, pp. 71–76, 5 2020.

[13] N. Wang, S. Chi-Kin Chau, and Y. Zhou, "Privacy-Preserving Energy Storage Sharing with Blockchain," in *Proceedings of the Twelfth ACM International Conference on Future Energy Systems*. New York, NY, USA: ACM, 2021. [Online]. Available: https://doi.org/10.1145/3447555.3464869

[14] V. Davydov, A. Gazaryan, Y. Madhwal, and Y. Yanovich, "Token Standard for Heterogeneous Assets Digitization into Commodity," in *Proceedings of the 2019 2nd International Conference on Blockchain Technology and Applications*. New York, NY, USA: ACM, 12 2019, pp. 43–47. [Online]. Available: https://dl.acm.org/doi/10.1145/3376044.3376053

[15] Y. Chaleenutthawut, V. Davydov, A. Kuzmin, and Y. Yanovich, "Practical Blockchain-Based Financial Assets Tokenization," *ACM International Conference Proceeding Series*, pp. 51–57, 12 2021. [Online]. Available: https://dl.acm.org/doi/10.1145/3510487.3510495

[16] V. Davydov and Y. Yanovich, "Optimal Portfolio Sold-Out via Blockchain Tokenization," in *Proceedings of the 2020 2nd International Electronics Communication Conference*. New York, NY, USA: ACM, 7 2020, pp. 129–136. [Online]. Available: https://dl.acm.org/doi/10.1145/3409934.3409950

[17] A. Krymov, D. Ozmaden, and Y. Pashchenko, "QuickToken: Blockchain platform for the tokenization of financial assets," 2023. [Online]. Available: https://github.com/apkrymov/QuickToken

Authorized licensed use limited to: Zhejiang University. Downloaded on July 09,2025 at 07:22:53 UTC from IEEE Xplore. Restrictions apply.

2020 IEEE 40th International Conference on Distributed Computing Systems (ICDCS) | 978-1-7281-7002-2/20/$31.00 ©2020 IEEE | DOI: 10.1109/ICDCS47774.2020.00163

FabAsset: Unique Digital Asset Management System for Hyperledger Fabric

Sangwon Hong*, Yoongdoo Noh*, Jeyoung Hwang, and Chanik Park
Department of Computer Science and Engineering, POSTECH
Pohang, South Korea
{sangwonhong, yoongdoo0819, capricorn116, cipark}@postech.ac.kr

***Abstract*—Business is innovating with the advent of blockchain that tokenizes digital assets. To expand the blockchain's potential, Ethereum, a representative permissionless blockchain platform, supports the fungible token (FT) standard ERC-20 and the non-fungible token (NFT) standard ERC-721. Hyperledger Fabric (Fabric), a representative permissioned blockchain platform, proposed FabToken to support tokens in version 2.0.0 alpha. But FabToken contains only FTs, not NFTs. Given the market share in the enterprise blockchains, Fabric needs to support NFTs as soon as possible. This paper presents a unique digital asset management system called FabAsset so that Fabric can run decentralized applications that require NFTs. This paper describes the design of FabAsset, consisting of chaincode and SDK (Software Development Kit), and the prototype of a decentralized signature service leveraging FabAsset to validate its usefulness.**

***Index Terms*—permissioned blockchain; decentralized application; smart contract; non-fungible token;**

I. Introduction

Blockchain is a distributed ledger system that immutably records transactions maintained by nodes participating in a P2P network without a trusted authority [1]. Blockchain was used primarily as a technology for cryptocurrencies such as Bitcoin [2]. Ethereum [3], a representative permissionless blockchain platform, introduced smart contracts for the first time among blockchains. Smart contracts enable decentralized applications (dApps) to run on the blockchain and issue tokens for specific purposes.

Tokens represent digital assets governed by smart contracts. Tokens are classified into fungible tokens (FTs) and non-fungible tokens (NFTs) in terms of fungibility, which means whether one token can be replaced with another token of the same type and quantity. FTs are interchangeable with other FTs because they have the same value as others, and can be divided into smaller units. NFTs are not interchangeable with other NFTs because every NFT is a unique and indivisible unit itself. Ethereum adapts token standards such as ERC-20 [4] for FTs and ERC-721 [5] for NFTs to guarantee interoperability between dApps.

Tokens have revolutionized business. Initial coin offerings (ICOs) [6] introduced a new way of fundraising using ERC-20 tokens [7]. ICOs are referred to as crowd sales where startups trade their ERC-20 tokens with ethers. ICOs democratize the fundraising for startups, expanding the startup investment focused on venture capitals to a wider range of investors. CryptoKitties [8] pioneered the ERC-721 protocol for the first time by representing digital cats as ERC-721 tokens. Unique digital assets such as digital cats can be globally traded on NFT exchanges such as OpenSea[1]. Due to the utilization of tokens, Ethereum runs the most dApps among permissionless blockchain platforms [9].

Despite the potential of tokens in Ethereum, Hyperledger Fabric (Fabric) [10], a representative permissioned blockchain platform, did not support tokens until version 1.4. But Fabric version 2.0.0 alpha introduced FabToken to try to support tokens for the first time [11].[2] FabToken [13] is a token management system that enables clients to issue, transfer, and redeem tokens on Fabric. However, this system contains only FTs, not NFTs.

Fabric is dominating nearly half of protocol frameworks for deployed enterprise blockchain networks [14]. Given the market share in the enterprise blockchains, Fabric needs to support NFTs as soon as possible to take advantage of the potential of NFTs. The extensible NFT model for Fabric (XNFT) [15] addresses this issue. XNFT provides standard structure and interface with reference to ERC-721, and extensible structure and interface to accommodate various requirements of dApps.

In this paper, we present a unique digital asset management system for Fabric called FabAsset. XNFT focused only on the design of the NFT. FabAsset not only advances XNFT, but also modularizes its components into chaincode, i.e. smart contract for Fabric, and SDK (Software Development Kit) to build a complete system for NFTs. Then we demonstrate a prototype of a decentralized signature service to validate the usefulness of FabAsset.

Suppose that tokens indicate only NFTs, and digital assets indicate only unique digital assets in Section II and Section III.

II. The FabAsset Design

FabAsset[3] (Fig. 1) is a digital asset management system that provides NFTs for dApps leveraging digital assets in Fabric. By default, all tokens in FabAsset follow ERC-721 [5]; that is, FabAsset reflects the ERC-721 specification configured for the Fabric environment. In addition, this system provides different configurations based on the dApp requirements within the

* indicates co-first authors.

[1] https://opensea.io/assets
[2] Note that FabToken was excluded in Fabric version 2.0.0 beta [12].
[3] https://github.com/FabAsset

2575-8411/20/$31.00 ©2020 IEEE
DOI 10.1109/ICDCS47774.2020.00163

Authorized licensed use limited to: Queens University Belfast. Downloaded on May 17,2021 at 07:14:00 UTC from IEEE Xplore. Restrictions apply.

same specification. The nature of this system, which makes token specifications uniform, fosters interoperability.

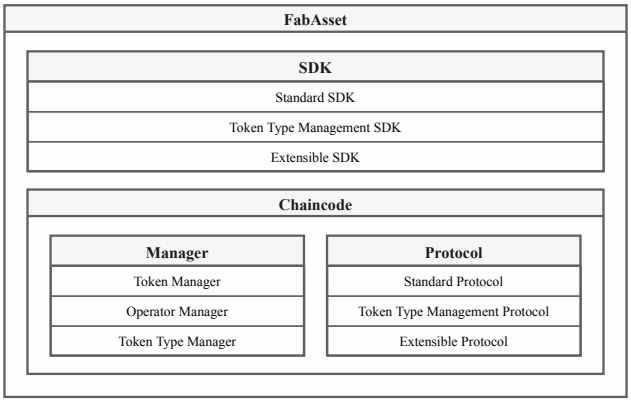


Fig. 1. FabAsset Overview

FabAsset consists of chaincode and SDK, which aim to provide token libraries for dApp implementations.

A. FabAsset Chaincode

FabAsset chaincode controls tokens on the ledger. This chaincode has two components: *manager* and *protocol*. The *manager* is a set of data structures to organize states associated with tokens. The *protocol* is a set of functions to retrieve or update the states of the *manager*, and guarantees the interoperable interface.

1) Manager: The *manager* falls into three classes: *token manager*, *operator manager*, and *token type manager*. These classes have attributes and methods. The attributes represent states. The methods access the world state, which holds the current ledger state, to set or get the attributes.

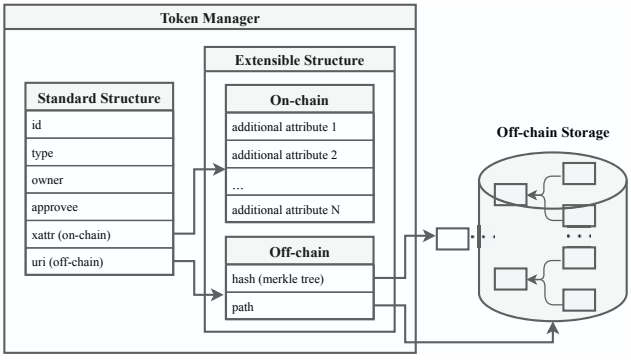


Fig. 2. Token Manager

The *token manager* (Fig. 2) is a class that manages token objects. Identical parts of all token objects are organized by standard structure while different parts of objects are organized by extensible structure. The standard structure contains standard and extensible attributes, and the extensible structure contains additional attributes that are stored as sub-attributes of extensible attributes. The standard attributes are *id*, *type*, *owner* and *approvee*. Attribute *id* represents the unique identifier of the token on the ledger. Attribute *type* represents the token type or the nature of the token. Token types define dApp requirements, and separate some requirements from others. If some tokens have the same token type, then they have the equal extensible structure. The default token type is *base* that does not require the extensible structure. Attribute *type* can be assigned by only one of the predefined token types on the ledger except for *base*. Attributes *owner* and *approvee* represent client roles. FabAsset accommodates client roles referenced in ERC-721 [5]. The client roles, such as owner, approvee and operator, have different permissions on tokens. The owner is a client who has the ownership of the token. Each token must have only one owner. The approvee is an approved client who has the permission to transfer the ownership of the token. Each token can have at most one approvee. The operator is a client who manages all tokens of a client who has authorized the operator. Each client can have multiple operators. The operator is separated from the *token manager* and managed by the *operator manager* because operators depend on clients whereas the other client roles depend on tokens.

The extensible attributes are *xattr* (extended attribute) and *uri* (uniform resource identifier). If a token is the base type, then the extensible attributes are unused because the token does not require the extensible structure. Attribute *xattr* indicates an on-chain extensible attribute that manages on-chain additional attributes. The token types determine which on-chain additional attributes the token has. Attribute *uri* indicates an off-chain extensible attribute that connects with an off-chain storage to store metadata for the token. Unlike on-chain additional attributes, every token has the same off-chain additional attributes, i.e. *hash* and *path*, regardless of token types. Attribute *hash* indicates the merkle root originated from the merkle tree of which the leaves are the hash of metadata stored in the storage. This attribute can prove whether off-chain metadata has been manipulated. Attribute *path* indicates the path of the storage.

The *token manager* stores tokens with key as the token ID and value as the JSON for all attributes and their values of the token in the world state.

The *operator manager* (Fig. 3) is a class that manages a table to store operator relationships for each client. The operator relationship table is an attribute for the *operator manager*. If a client has operators, the table stores the operators mapped to the client and marks them as true that means the operators are enabled by the client. If the client disables an operator, then the operator is marked as false. To sum up, client A is not an operator for client B if client A is marked as false or not mapped to client B in the table.

The *operator manager* stores the table with key as *OPERATORS_APPROVAL* and value as the JSON for the operator relationships between clients in the world state.

The *token type manager* (Fig. 4) is a class that manages a table to store token types enrolled on the ledger. The token

Authorized licensed use limited to: Queens University Belfast. Downloaded on May 17,2021 at 07:14:00 UTC from IEEE Xplore. Restrictions apply.

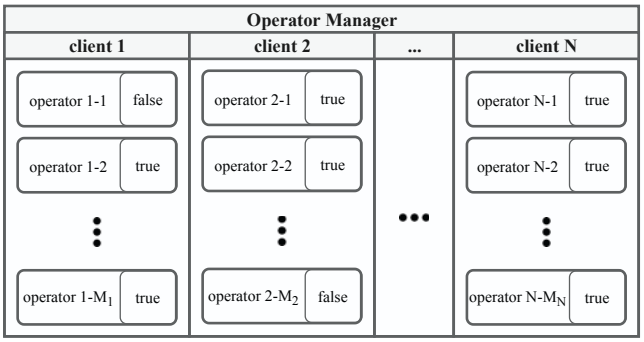


Fig. 3. Operator Manager

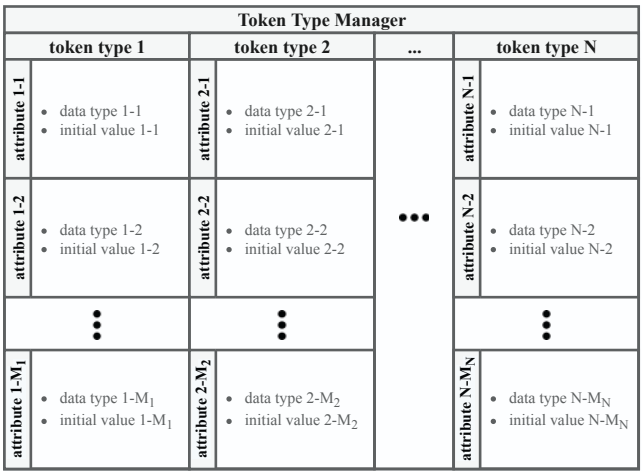


Fig. 4. Token Type Manager

type table is an attribute for the *token type manager*. Only tokens whose token type is already enrolled on the ledger can be issued except for *base*. Tokens that belong to the identical token type must have the same on-chain additional attributes. To ensure that the on-chain additional attributes remain the same, they must be enrolled with the token type. In the table, each token type is mapped to a set of the on-chain additional attributes; each on-chain additional attribute has its information that describes its data type and its initial value. When issuing a token, if some of the on-chain additional attributes are not initialized by clients, then the uninitialized attributes are initialized to the initial values considering the data types.

The *token type manager* stores the table with key as *TOKEN_TYPES* and value as the JSON of the enrolled token types in the world state.

2) Protocol: The *protocol* (Fig. 5) provides the uniform interface to be interoperable. The *protocol* cannot directly access attributes of the *manager*, but it can indirectly access them through the methods of the *manager*. For the read operation, anyone maintained by a membership service provider (MSP) can call read functions in the *protocol*. For the write operation, on the other hand, only the clients who have the permission can call write functions in the *protocol*. The conditions of the permission are different depending on the write functions.

The *protocol* consists of *standard protocol*, *token type management protocol*, and *extensible protocol*. The *standard protocol* performs the operations that are common to all tokens regardless of token types. This protocol is classified into *ERC-721 protocol* and *default protocol*. The *ERC-721 protocol* reflects some of ERC-721 functions that are appropriate for the Fabric environment. This protocol performs the operations for client roles related to attributes *owner* and *approvee* of the *token manager* and the operator relationship table of the *operator manager*. For the read operation, function *balanceOf* counts tokens owned by a client. Function *ownerOf* queries who is the owner of the token. Function *getApproved* queries who is the approvee of the token. Function *isApprovedForAll* queries whether client A is an operator for client B. For the write operations, function *transferFrom* transfers the ownership of the token from a sender to a receiver. The sender should be equal to the current owner. Only the current owner of the token, the approvee of the token, and the current owner's operators can call this function. Function *approve* sets the approvee of the token. If this approvee is called when the approvee of the token is already set, then the approvee is reset to a new approvee. Only the owner of the token and the owner's operators can call this function. Function *setApprovalForAll* enables or disables the caller's operator.

The *default protocol* is not included in ERC-721, but performs the operations related to the *token manager*, which are required to support ERC-721. For the read operation, function *getType* queries what the token type of the token is. Function *tokenIdsOf* queries the list of token IDs owned by a client. Function *query* queries the JSON for all attributes and their values of the token. Function *history* queries the list of modification histories of the attributes of the token. For the write operation, function *mint* issues a standard token that is the base type. The owner of the token is assigned to the caller of this function. Function *burn* removes the token. Only the owner of the token can call this function.

The *token type management protocol* performs the operations related to the *token type manager*. For the read operation, function *tokenTypesOf* queries the list of token types enrolled on the ledger. Function *retrieveTokenType* queries the on-chain additional attributes, including their information i.e. the data type and the initial value, associated with the token type. Function *retrieveAttributeOfTokenType* queries the information associated with the on-chain additional attribute of the token type. For the write operation, function *enrollTokenType* enrolls a token type on the ledger. The caller of this function becomes an administrator for the token type. Function *dropTokenType* drops the token type in the world state. Only the client that enrolled the token type, i.e. the administrator, can call this function.

The *extensible protocol* performs the operations related to extensible tokens that have the additional attributes in the *token manager*. For the read operation, function *balanceOf*

Authorized licensed use limited to: Queens University Belfast. Downloaded on May 17,2021 at 07:14:00 UTC from IEEE Xplore. Restrictions apply.

redefines the function of the same name in the *ERC-721 protocol* to count tokens of the specific token type that are owned by a client. Function *tokenIdsOf* redefines the function of the same name in the *default protocol* to query the list of token IDs of the specific token type that are owned by a client. Function *getURI* queries one of the off-chain additional attributes by accessing attribute *uri*. Function *getXAttr* queries one of the on-chain additional attributes by accessing attribute *xattr*. Functions *getURI* and *getXAttr* have *index* as one of the input parameters. Parameter *index* indicates the additional attribute name. These getter functions query the value of the additional attribute whose name matches with parameter *index*. For the write operation, function *mint* redefines the function of the same name in the *default protocol* to issue an extensible token for initializing the additional attributes. The owner of the token is assigned to the caller of this function. Function *setURI* updates one of the off-chain additional attributes by accessing attribute *uri*. Function *setXAttr* updates one of the on-chain additional attributes by accessing attribute *xattr*. Functions *setURI* and *setXAttr* have *index* and *value* as the input parameters. Parameter *index* indicates the additional attribute name, and parameter *value* indicates a new value to update. These setter functions update the value of the additional attribute, whose name matches with parameter *index*, to parameter *value*. The setter functions do not require any permissions when clients call these functions. To restrict the permissions for each additional attribute, developers should customize a function for each attribute by wrapping the setter functions.

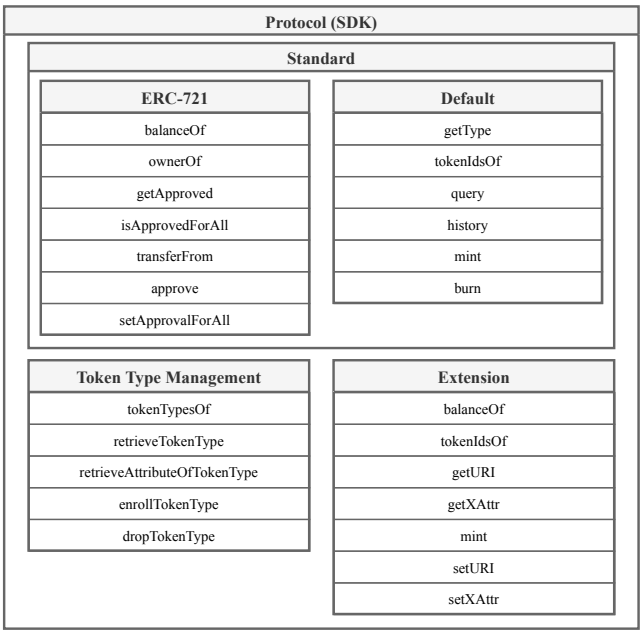


Fig. 5. Protocol (SDK)

B. FabAsset SDK

The FabAsset SDK provides APIs that allow clients to access the FabAsset chaincode. The FabAsset SDK is a set of functions that wrap the protocol functions. Each SDK function handles the protocol function of the same name (Fig. 5). The SDK also has the same classification as the protocol of the chaincode; i.e. it is classified into *standard SDK*, *token type management SDK*, and *extensible SDK*; and the *standard SDK* consists of *ERC-721 SDK* and *default SDK*.

III. Application

To validate the usefulness of FabAsset, we implemented a prototype of a decentralized signature service using FabAsset. Our service allows the digital signing process to proceed digital contracts without a trusted third party. Our service uses a signature type and a digital contract type, so administrator *admin* enrolls both types on the ledger by calling SDK function *enrollTokenType* (Fig. 6). Administrator *admin* is automatically stored in attribute *_admin* of both types to identify who is the administrator of these types. The signature type includes attribute *hash* representing the hash of the signature image. The data type of attribute *hash* is string, and the initial value of it is an empty string. The digital contract type includes attributes *hash* representing the hash of the contract document, *signers* representing the list of signers for the digital contract, *signatures* representing the list of signers' signatures and *finalized* representing whether the signing process of the digital contract is complete. The data type of attribute *hash* is string, and the initial value of it is an empty string. The data type of attribute *signers* is string list, and the initial value of it is an empty list. The data type of attribute *signatures* is string list, and the initial value of it is an empty list. The data type of attribute *finalized* is boolean, and the initial value of it is false.

```
"TOKEN_TYPES": {
    "signature": {
        "_admin": ["String", "admin"],
        "hash": ["String", ""]
    },
    "digital contract": {
        "_admin": ["String", "admin"],
        "hash": ["String", ""],
        "signers": ["[String]", "[]"],
        "signatures": ["[String]", "[]"
            ],
        "finalized": ["Boolean", "false
            "]
    }
}
```

Fig. 6. Example of Token Types Stored in the World State

When clients sign a digital contract, they add their own signature token ID to attribute *signatures* of the digital contract

Authorized licensed use limited to: Queens University Belfast. Downloaded on May 17,2021 at 07:14:00 UTC from IEEE Xplore. Restrictions apply.

token. The signing operation ensures that the client signed himself because this operation proves whether the signature token is owned by the client before the token ID is inserted into attribute *signatures*. If all the signers in attribute *signers* of the digital contract token complete signing, then attribute *finalized* of the token is changed from false to true. This means that all the signers correctly conclude the digital contract. If contractual disputes happen, the digital contact token can be effective as an evidence.

We implemented custom functions such as functions *sign* and *finalize* to construct our service using FabAsset. The participants in the contract must sign the digital contract token, so our service needs function *sign*. Protocol function *sign* is implemented in the chaincode using the protocol functions. This function checks whether its caller is the owner of the digital contract token because only the owner can sign the digital contract token, whether he is included in the list of the signers read by calling function *getXAttr* that takes "signers" as the value of parameter *index*, and whether he is a correct order to sign. If all conditions pass, this function goes through the process for inserting the signature token ID owned by its caller into the list of the signatures. First, function *getXAttr* that takes "signatures" as the value of parameter *index* is called to read the list of the signatures. Second, the signature token ID of the caller is inserted into the list of the signatures. Finally, function *setXAttr* that takes "signatures" as the value of parameter *index* and the updated list of the signatures as the value of parameter *value* updates attribute *signatures* in the world state. Our service also needs function *sign* as an SDK function because function *sign* performs its operation when clients call it. With the same name as the protocol function, we implemented SDK function *sign* by wrapping protocol function *sign*. When the signing process is over, the digital contract token should be finalized not to modify the states of the token after completion of the contract. So protocol function *finalize* is implemented in the chaincode using the protocol functions. When issuing the digital contract token, the initial value for attribute *finalized* is assigned as false. Function *finalize* performs the operation to update attribute *finalized* to true when attribute *signatures* is full by calling function *setXAttr* that takes "finalized" as the value of parameter *index* and true as the value of parameter *value*. Our service also needs function *finalize* as an SDK function to conclude the contract. So we implemented SDK function *finalize* by wrapping protocol function *finalize*.

We describe a scenario utilizing our service. The scenario simulates on Fabric version 1.4. It is conducted on three clients (companies 0, 1, and 2), three peers (peers 0, 1, and 2) and a solo orderer in one channel (Fig. 7). Organizations group peers and clients; org 0 manages peer 0 and company 0; org 1 manages peer 1 and company 1; and org 2 manages peer 2 and company 2. Chaincode that utilizes the FabAsset chaincode as a library is installed in all peers.

Fig. 8 shows a process of signing a digital contract where company 0 has agreed to provide a down payment, and companies 1 and 2 have agreed to fulfill company 0's requirements. Clients, i.e. companies 0, 1 and 2, must issue their own

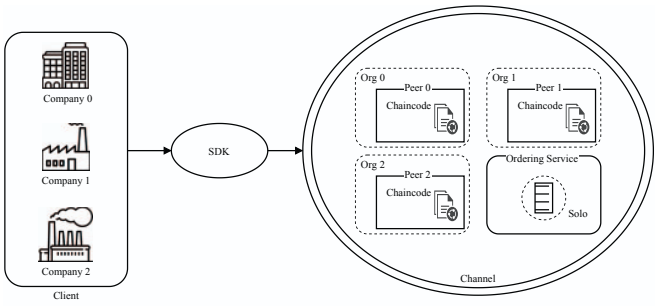


Fig. 7. Fabric Environment for the Decentralized Signature Service

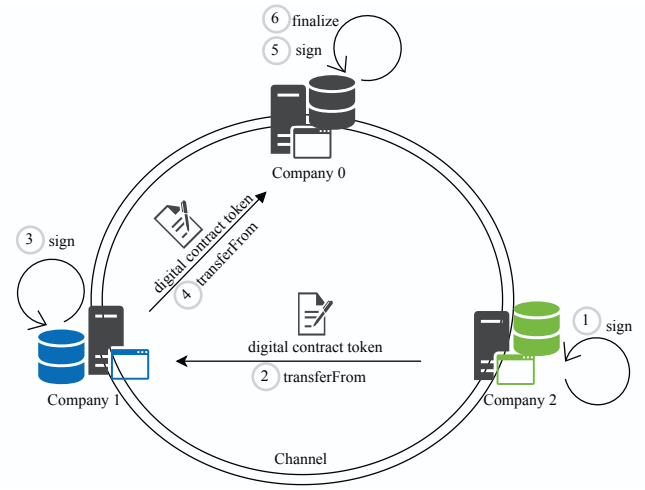


Fig. 8. Scenario for the Decentralized Signature Service

signature tokens before signing the digital contract. So they issue their own signature tokens based on their own signature images uploaded in the off-chain storage. The order of signing the contract is companies 2, 1 and 0. First, company 2 issues a digital contract token using function *mint* based on the agreement among companies 0, 1, and 2. In the digital contract token, function *mint* initializes standard attributes, such as assigning attribute *owner* to company 2 and assigning attribute *type* to digital contract, on-chain additional attributes, such as assigning attribute *hash* to the hash of the contract document and assigning attribute *signers* to companies 2, 1 and 0, and off-chain additional attributes, such as assigning attribute *hash* to the merkle root coming from some metadata, e.g. the contract document and token creation time, stored in the off-chain storage and assigning attribute *path* to the path of the off-chain storage. Some on-chain additional attributes uninitialized by clients are initialized to the initial values defined in the the token type such as assigning attribute *finalized* to false.

Using function *sign*, ① company 2 signs the digital contract. Attribute *signatures* stores the signature token ID of company 2. Using function *transferFrom*, ② company 2 transfers the ownership to the next signer, i.e. company 1. Company 1 becomes a new owner of the digital contract token and verifies

Authorized licensed use limited to: Queens University Belfast. Downloaded on May 17,2021 at 07:14:00 UTC from IEEE Xplore. Restrictions apply.

the token. After verification, ③ company 1 signs the token using function *sign*. Attribute *signatures* stores the signature token IDs of companies 2 and 1. After signing, ④ company 1 transfers the digital contract token to company 0 using function *transferFrom*. Company 0 also verifies the token and ⑤ proceeds signing. Finally, attribute *signatures* stores the signature token IDs of companies 2, 1 and 0. Once all participants have signed, ⑥ company 0 finalizes the digital contract by calling function *finalize*. The digital contract token ensures that the contract is successfully concluded. Fig. 9 shows the final states of the digital token contract stored in the world state.


```
"3": {
  "id": "3",
  "type": "digital contract",
  "owner": "company 0",
  "approvee": "",
  "xattr": {
    "hash": "8decc8571946d4cd70a024949e
       033a2a2a54377fe9f1c1b944c20f9ee1
       1a9e51",
    "signers": ["company 2", "company 1
       ", "company 0"],
    "signatures": ["2", "1", "0"],
    "finalized": true
  },
  "uri": {
    "hash": "e1cee4f587e56d4ef9b03b44b8
       c8bcc89bb59e1abdf1d715e538502f01
       7cde81",
    "path": "jdbc:log4jdbc:mysql://
       localhost:3306/hyperledger"
  }
}
```


Fig. 9. Example of the Digital Contract Token Stored in the World State

IV. Conclusion

This paper presents FabAsset, a unique digital asset management system to provide NFTs for dApps. FabAsset supports chaincode and SDK. The FabAsset chaincode is composed of *manager* and *protocol*. The *manager* manages the NFT-related states. The *protocol* controls the *manager* and provides the interoperable interface. The FabAsset SDK performs the operations that allow clients to access the FabAsset chaincode in the application. We validated the usefulness of FabAsset by implementing the decentralized signature service and demonstrating the scenario.

In the permissioned blockchains, applications that maintain different ledgers need to communicate with each other for a collaborative workflow [16]. If the applications communicate with each other via NFTs, FabAsset can exert its potential. To realize communication between different ledgers or channels, research on cross-channels such as [17] should be conducted.

Acknowledgement

This research was supported by the MSIT (Ministry of Science and ICT), Korea, under the ITRC (Information Technology Research Center) support program (IITP-2020-2018-0-01441) supervised by the IITP (Institute for Information & Communications Technology Planning & Evaluation)

References

[1] C. Cachin and M. Vukolić, "Blockchain Consensus Protocols in the Wild," in *31st International Symposium on Distributed Computing (DISC 2017)*, Vienna, Austria, Oct. 16–20, 2017, pp. 1:1–1:16.

[2] S. Nakamoto, "Bitcoin: A Peer-to-Peer Electronic Cash System," 2008. Accessed: Feb. 26, 2020. [Online]. Available at https://bitcoin.org/bitcoin.pdf.

[3] V. Buterin, "A Next-Generation Smart Contract and Decentralized Application Platform," 2013. Accessed: Feb. 19, 2020. [Online]. Available at https://github.com/ethereum/wiki/wiki/White-Paper.

[4] F. Vogelsteller and V. Buterin, "ERC-20 Token Standard," Nov. 19, 2015. Accessed: Feb. 19, 2020. [Online]. Available at https://eips.ethereum.org/EIPS/eip-20.

[5] W. Entriken, D. Shirley, J. Evans, and N. Sachs, "ERC-721 Non-Fungible Token Standard," Jan. 24, 2018. Accessed: Feb. 19, 2020. [Online]. Available at https://eips.ethereum.org/EIPS/eip-721.

[6] N. Lipusch, "Initial Coin Offerings – A Paradigm Shift in Funding Disruptive Innovation," Mar. 23, 2018. Available at *SSRN*: https://ssrn.com/abstract=3148181.

[7] G. Fenu, L. Marchesi, M. Marchesi, and R. Tonelli, "The ICO Phenomenon and Its Relationships with Ethereum Smart Contract Environment," in *2018 International Workshop on Blockchain Oriented Software Engineering (IWBOSE)*, Campobasso, Italy, Mar. 20, 2018, pp. 26–32.

[8] Dapper Labs, "CryptoKitties: Collectible and Breedable Cats Empowered by Blockchain Technology," Version 2.0, 2018. Accessed at Feb. 19, 2020. [Online]. Available at https://drive.google.com/file/d/1soo-eAaJHzhw_XhFGMJp3VNcQoM43byS/.

[9] Dapp Review, "2019 Dapp Market Report," 2019. Accessed at Feb. 26, 2020. [Online]. Available at https://dapp.review/article/238.

[10] E. Androulaki, A. Barger, V. Bortnikov, C. Cachin, K. Christidis, A. De Caro, D. Enyeart, C. Ferris, G. Laventman, Y. Manevich, and S. Muralidharan, "Hyperledger fabric: a distributed operating system for permissioned blockchains," in *Proceedings of the Thirteenth EuroSys Conference (EuroSys 2018)*, Porto, Portugal, Apr. 23–26, 2018, pp. 1–15.

[11] https://github.com/hyperledger/fabric/releases/tag/v2.0.0-alpha. (accessed Feb. 19, 2020).

[12] https://github.com/hyperledger/fabric/releases/tag/v2.0.0-beta. (accessed Feb. 19, 2020).

[13] https://github.com/hyperledger-labs/fabric-block-archiving/blob/master/docs/source/token/FabToken.md. (accessed Feb. 19, 2020).

[14] M. Rauchs, A. Blandin, K. Bear, and S. B. McKeon, "2nd Global Enterprise Blockchain Benchmarking Study," Sep. 18, 2019. Available at *SSRN*: https://ssrn.com/abstract=3461765.

[15] S. Hong, Y. Noh, C. Park, "Design of Extensible Non-Fungible Token Model in Hyperledger Fabric," in *Proceedings of the 3rd Workshop on Scalable and Resilient Infrastructures for Distributed Ledgers (SERIAL 2019)*, Colocated with Middleware 2019, UC Davis, CA, USA, Dec. 9, 2019, pp. 1–2.

[16] M. J. Amiri, D. Agrawal, and A. E. Abbadi, "CAPER: A Cross-Application Permissioned Blockchain," in *45th International Conference on Very Large Data Bases (VLDB 2019)*, LA, CA, USA, Aug. 26–30, 2019, pp. 1385–1398.

[17] E. Androulaki, C. Cachin, A. De Caro, E. Kokoris-Kogias, "Channels: Horizontal Scaling and Confidentiality on Permissioned Blockchains," In *23rd European Symposium on Research in Computer Security (ESORICS 2018)*, Barcelona, Spain, Sep. 3–7, 2018, pp. 111–131.

Authorized licensed use limited to: Queens University Belfast. Downloaded on May 17,2021 at 07:14:00 UTC from IEEE Xplore. Restrictions apply.

2020 International Conference on Computer Science and Management Technology (ICCSMT) | 978-1-7281-8668-9/20/$31.00 ©2020 IEEE | DOI: 10.1109/ICCSMT51754.2020.00049

Blockchain-based Power Grid Data Asset Management Architecture

SHEN Liang
Big Data Center of State Grid Corporation of China
Beijing, China

HAO Baozhong
Big Data Center of State Grid Corporation of China
Beijing, China

LI Yang
Big Data Center of State Grid Corporation of China
Beijing, China

YU Han
Big Data Center of State Grid Corporation of China
Beijing, China

ZHANG Song
Big Data Center of State Grid Corporation of China
Beijing, China

MEN Hao
Beijing China-Power Information Technology Co., Ltd.
Beijing, China

FAN Jingang*
Beijing Quantum View Network Technology Co., Ltd.
Beijing, China
453511248@qq.com

Abstract: **Although the continuous construction of smart grid and power Internet of Things has brought massive data to grid enterprises, the cost of data storage, operation and maintenance is gradually increasing, on the contrary, the value of data has not been effectively mined and utilized, the main reason is that the current data asset management of State Grid Corporation of China is still an infancy stage, and still faces the problems of data aggregation quality, safety and compliance control, shared application scope, and management mechanism efficiency and so on. In order to solve the above problems, Big Data Center of State Grid Corporation of China takes advantage of the blockchain with distributed consensus autonomy and data storage, non-tampering and traceability, and business intelligence contract script, explores the application of blockchain in power grid data asset management, and then, tries to construct the architecture of power grid data asset management based on blockchain, which is to promote security compliance and open sharing of power grid data operation, and realize the efficient management and operation of power grid data assets.**

Key words: blockchain; power grid data asset management; solution architecture; security compliance; open sharing

I. INTRODUCTION

In recent years, in order to cope with global energy problems, countries around the world have comprehensively established smart grids covering the entire production process of the power system, carried out to research on smart grids, and generated and accumulated massive amounts of multi-source heterogeneous grid data [1, 2]. Therefore, the problems of centralized management and application are becoming more and more prominent, how to manage efficiently, mine and analyze power grid data has become an urgent requirement for power grid development. Blockchain provides a new solution to the credible aggregation, safe storage, efficient management and application of power grid data.

State Grid Corporation of China (SGCC) established Big Data Center of SGCC (BDC of SGCC, or BDC) in May 2019 to actively carry out unified management and operation of power grid data. SGCC explores to apply blockchain technology in power grid data asset management, to promote the fusion application of blockchain and data governance, application, operation and security, etc., so as to support the value-added realization of power grid data, and facilitate the construction of new technological infrastructure in the "New Infrastructure".

Based on an in-depth analysis of the current problems of SGCC's data management, this paper proposes a blockchain-based solution which combines the business processes and the existing technology platforms of the BDC to construct a blockchain-based power grid data asset management architecture, to realize the binding of recording of data, business process, rights and responsibilities, and to provide support for the credible management of power grid data.

II. GENERAL BACKGROUND INFORMATION

A. Status of SGCC's data development

After years of informatization construction and application, SGCC has accumulated a large amount of data resources, and lays a solid foundation for further promoting the innovative operation and value realization of power grid data [3].

In terms of data infrastructure, a unified data management and control platform has been built with data integration technologies, such as data access, storage, and calculation , etc., to lay the foundation of data application; in terms of unified

978-1-7281-8668-9/20/$31.00 ©2020 IEEE
DOI 10.1109/ICCSMT51754.2020.00049

Authorized licensed use limited to: National University of Singapore. Downloaded on July 03,2021 at 11:09:41 UTC from IEEE Xplore. Restrictions apply.

data management, SGCC has carried out data inventory, governance verification and application mining, etc., to improve data quality and application effectiveness; in terms of data aggregation and sharing, with relying on the data middle platform to gather massive business data, SGCC has already opened up external data channels to strengthen data acquisition capabilities and provides internal and external sharing services. With regards to data mining applications, the value of data is explored deeply, and multiple professional report management applications have been realized, such as marketing, customer service, electricity safety and metering. And when it comes to data security system, a relatively complete network security management and technical defense system has also been established to ensure the security of data use.

In general, SGCC has the basis for unified data management. However, there are still some problems that have not been effectively solved during the implementation process [4].

With reference to data aggregation, unified storage and management have not yet been achieved, and the phenomenon of "one data with multiple sources" is widespread. As regards to data management, SGCC's unified data directory management tool has not yet been formed, and there is a lack of flexible and reliable data resource directory dynamic update mechanism, which is prone to cause problems such as repeated storage, multi-head entry, and data inconsistency etc. In terms of data application, standards and regulations on data right confirmation, usage, and open circulation are not yet complete, and it is insufficiently supported for lean management, smart operation and innovative development of the company. Meanwhile, when facing data sharing requirements, such as data desensitization, watermark traceability, and compliance control are relatively weak.

Generally speaking, due to constrained by power grid data aggregation management of involving multiple departments and systems, cross-regions and cross-levels, data collaborative management applications and data asset value mining are greatly restricted. The blockchain technology of key features such as multi-center, collaborative consensus and storage, orderly recording and traceability has great potential in the current situation. Therefore, SGCC is actively exploring the research and application of blockchain in order to better solve many problems encountered in power grid data management.

B. SGCC's blockchain construction status

With the development trend of digital economy and energy transition, SGCC attaches great importance to blockchain research and application verification. As early as 2017, it released the "SGCC's Blockchain Technology Application White Paper", covering the active exploration carried out in the blockchain technology platform architecture and data governance etc.

In 2019, under the background of comprehensive analysis of blockchain technology and development trends, SGCC further increased the research and application of blockchain technology, and subsequently launched the construction of blockchain public service capabilities, combined with the company's business development needs. SGCC creatively proposed a "one master and two sides" blockchain architecture system, namely one master chain and two side chains, in which the two side chains are transaction side chain and data side chain respectively. Thence, BDC of SGCC is responsible for the construction of the data side chain, and conducts research work on the application of the data side chain in supporting data security, data value-added, and data sharing services.

In 2020, in order to further promote business application needs of provincial company, SGCC will efficiently support the scalability of blockchain applications in various pilot units, and expand the "one master and two sides" architecture to "one master and two sides with multiple slaves" to build a company-level blockchain platform-State Grid Chain, which enhances the cross-chain mutual trust capability with external industry chains.

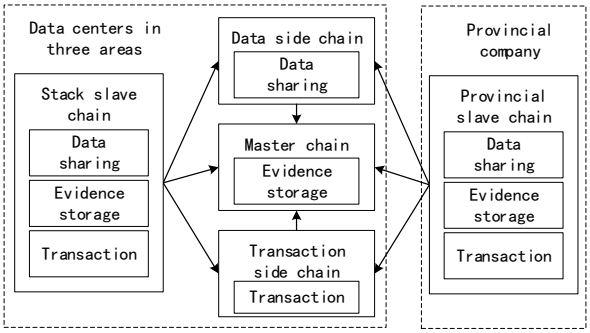


Figure 1. Public service architecture of State Grid chain

According to the different support units and business types, State Grid Chain can be divided into five parts: master chain, data side chain, transaction side chain, provincial slave chain and stack slave chain. The master chain, side chains and slave chains respectively form a complete blockchain system.

III. SOLUTIONS BASED ON BLOCKCHAIN

Although SGCC possesses massive amounts of data, it has not yet been used effectively, partly due to the problem of overall and centralized management and application of data is serious. The blockchain integrates a variety of technologies such as distributed consensus mechanisms, peer-to-peer networks, asymmetric encryption algorithms, and smart contracts, and has the characteristics of multi-concentration, disintermediation, traceability and difficulty in tampering with information [5, 6], which provides new solutions for the full aggregation of power grid data, security compliance, trusted sharing and efficient management.

A. Data resource directory on-chain storage

The main problem of data aggregation lies in multi-party data, data quality caused by data inconsistency, etc. [7], and it also brings management difficulties.

Therefore, when using the data middle platform [8] as the basis of data aggregation from various business information systems, the blockchain can be used to organize and manage each department in a unified manner and to build a distributed and autonomous data upload, aggregation, and storage management system. Meanwhile, using the data on the

Authorized licensed use limited to: National University of Singapore. Downloaded on July 03,2021 at 11:09:41 UTC from IEEE Xplore. Restrictions apply.

blockchain that has consensus features and is difficult to tamper with, the data including resource catalog, operation data, and event records are stored on the blockchain to ensure that can be traced and cannot be tampered with. Due to the ownership is clear [9], and the unified power grid data resource directory system is build, it can further enhance the credibility and realize the autonomy of the distribution of data from various business sources.

B. Data control process based smart contracts

The main problem of data security compliance and control management lies in that it frequently requires manual review by multiple departments. As a result, the business process is long and the data utilization efficiency is low.

Based on blockchain technologies such as asymmetric encryption and automatic execution of smart contracts, an online synchronization review mechanism for multiple departments is built. Data application, authorization, access, transmission and other processes are all recorded on the chain, to ensure that the data life cycle is legal compliance and traceability [10]. Beyond that, smart contracts automatically trigger business processes such as security compliance applications, approvals, and registrations in the sharing, which can improve business efficiency and the ability to ensure power grid data security.

C. Data safety and trusted sharing

The sharing in this article mainly refers to the data circulation outside the system. The current external sharing of power grid data mainly has the problems of data privacy leakage and data credibility [11].

In response to the original data leakage caused by data sharing, a data security sandbox with homomorphic encryption, secure multi-party computing, or a trusted execution environment [12] is constructed to achieve access control with blockchain. In this process, each data party that request, processing and other operation records of the data are stored on the blockchain can be queried and verified. The data sandbox guarantees the credibility and transparency of data calculations. Source data are not obtained by the outside users to ensure "available but not accessible".

Furthermore, we combine data aggregation (A) in which the blockchain is used to record the data ownership and relationship between business departments, with the internal compliance control (B) in which the blockchain is used to record the entire process of internal circulation of data. A more complete data credibility chain is formed from the convergence of the source to the shared circulation that supports the safe interaction and credible sharing of power grid data.

D. Flattening data management architecture

Currently, the circulation of power grid data is mostly in the form of work orders with a level-by-level manual approval and authorization that it is hierarchical data management method, this not only leads to lengthy business processes, but also information distortion easily due to multi-level data flow.

Blockchain, as a distributed ledger technology, can build a credible data sharing environment among multiple participants[13], and can resolve ownership disputes, losing data, malicious tampering, single point of failure and privacy leakage and other issues in the current power grid data asset management. it ensures the trustworthiness and verifiability of data on the chain among multiple subjects, and realizes the flattening of the power grid data asset management organization structure, and improves the transparency and efficiency of management.

In short, the application of blockchain technology in data production and aggregation, compliance and control, shared circulation and management mechanisms can ensure the authenticity, integrity, consistency and credibility of data, and avoid the dilemma of "want to use data to solve problems but fear using data to generate problems", so as to solve the problems of "storage, management and usage" [14], and facilitate the capitalization and value-added realization of power grid data.

IV. BLOCKCHAIN-BASED POWER GRID DATA ASSET MANAGEMENT ARCHITECTURE

The application of blockchain to develop power grid data asset management requires a top-down comprehensive dimensional system architecture design. The existing information systems such as data side chain and data middle platform are used for coordination in blockchain way to realize the unified management and control of power grid data confirmation, privacy protection, sharing and traceability, etc. A blockchain-based power grid data asset management architecture is formed to reduce the cost of cross-system data interaction and improve data credibility.

1) Basic resource layer

The basic resource layer mainly includes the existing power systems of the national grid system at all levels and the corresponding storage and network environment. The power system represented by smart grid technology is the source and carrier of power big data, including panoramic real-time data of multiple links such as power generation, transmission, transformation, distribution, power consumption and dispatching [15].

The basic resource layer solves the problems of power grid data generation, collection and storage through the construction of smart grid.

2) Unified data exchange platform

The unified data exchange platform is the data channel between the data middle platform and the power system, providing unified data access, exchange, and dispatch monitoring services.

In order to support the access and transmission of structured data, unstructured data, and collected measured data, the unified data exchange platform provides methods of online ETL (Extract-Transform-Load) and offline batch data import/export to extract regularly various source-end business system's data to the source layer of the data middle platform.

3) Data middle platform

The data middle platform is a data resource convergence center, a data asset conversion center, and a data value discovery center, serving users in various departments within SGCC and external energy, government, and finance.

Authorized licensed use limited to: National University of Singapore. Downloaded on July 03,2021 at 11:09:41 UTC from IEEE Xplore. Restrictions apply.

Specifically, the data middle platform integrates data asset aggregation, application and operation. It is divided into three layers: the source layer, the sharing layer and the analysis layer. The source layer realizes data aggregation of all levels of business systems, the sharing layer integrates the source layer data according to unified data standards and models, and the analysis layer constructs a multi-dimensional data analysis theme according to the needs. In this way, it internally supports the integration of traditional power grid business, and externally supports the expansion of emerging business [16].

To sum up, the data middle platform solves the problems of integration, processing and application of power grid data through big data processing technology.

4) Data side chain

The data side chain (SG-DSC) not only provides blockchain support services with the underlying core technology of blockchain to realize the data on-chain management, data contract management, and data privacy authority management, etc., but also provides basic functional component support for various blockchain applications and services.

5) Directory blockchain

The directory blockchain is a trusted sharing system based on data side chain and data resource directories. By linking data directories, sharing rules, business processes, and operating behaviors to the chain, the entire process of using power grid data can be traced back to audit and compliance control, when simultaneously providing trusted data services internally and externally.

The directory blockchain solves the problems of power grid data verification and authentication, security, privacy, and credible sharing [17], and then realizes the open sharing and circulation of power grid data.

6) Standard specification system and safety guarantee system

These systems provide institutional guarantee and technical support for the application of blockchain to carry out data asset management.

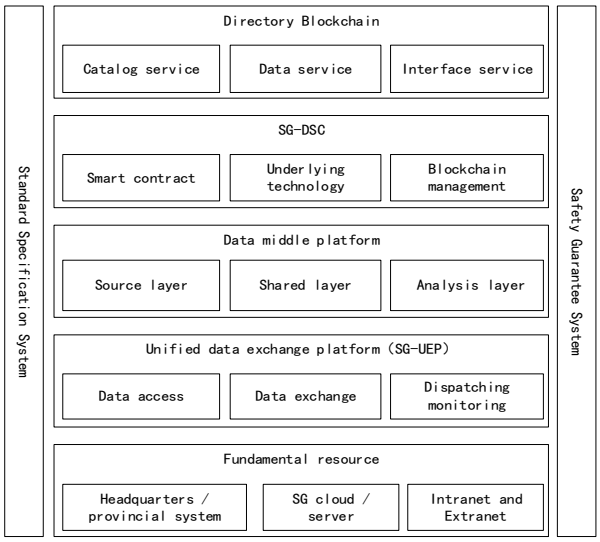


Figure 2. System architecture of grid data asset management based on blockchain

V. CONCLUSION

Aiming at some shortcomings of the company's current data asset management, SGCC conducts research on the application of blockchain charactered by high credibility and synergy in power grid data asset management, to optimize technology and business design, to construct a blockchain-based power grid data asset management architecture, to solve the problems of registration and confirmation of data collection and access, authorization verification of data use, traceability of data circulation, and privacy protection of data sharing, which is to promote the open sharing of power grid data, and assist the company's digital transformation and the construction of "new infrastructure".

The power grid data asset management architecture based on blockchain proposed in this paper is based on a certain degree of informatization, and data have been initially aggregated, so as to the exploration of blockchain multi-node collaborative management, full-link monitoring and traceability is carried out. In order to verify the effectiveness of the architecture, we will combine the business processes and scenarios with specific departments, and fully utilize blockchain to manage the data resource directory which is made up of the meta data in the data middle platform in the way of business collaboration and consensus records among multiple nodes. On this basis, it will promote record locking and collaborative interaction of data, business and authority, to achieve real-time perception of data change, the whole process record of data access, and coordinate data application, finally provide technical and platform support for power grid data asset operation.

Acknowledgement

Fund projects: Big Data Center Project of State Grid Corporation of China (Research on Key Technologies of power grid data traceability and high frequency interaction based on data side chain)

Authorized licensed use limited to: National University of Singapore. Downloaded on July 03,2021 at 11:09:41 UTC from IEEE Xplore. Restrictions apply.

REFERENCES

[1] ZHANG Jun, Wang FeiYue. "Blockchain based Digital Asset Management System Architecture for Power Grid Big Data," Electric Power Information and Communication Technology, vol. 16, pp. 1-7, August 2018.

[2] SONG Yaqi, ZHOU Guoliang, ZHU Yongli. "Present Status and Challenges of Big Data Processing in Smart Grid," Power System Technology, vol. 37, pp. 927-935, April 2013.

[3] Cui Jifeng, Yan Bin, Zhang Peng. "Research and exploration of data asset management," China Power Enterprise Management, pp 94-95, December 2014

[4] LI Guohe, FENG Zheng, WANG Zhuoyu, SUN Yong, GUO Yang, SAN Qiguo. "Practical exploration on the construction of data asset management system," Telecommunications Science, vol. 35, pp. 111-118, February 2019.

[5] Sun Guomao. "A Study of the Essential Characteristics of the Block Chaining Technique and Its Application in the Financial Field," Theory Journal, pp. 58-67, March 2017.

[6] Zheng X, Jia H, Wang J. "Energy Internet Development Based on Blockchain Technology," ICCREM 2019: Innovative Construction Project Management and Construction Industrialization. Banfu Tong, Canada, pp. 1-12, August 2019.

[7] DU Xiaoyong, CHEN Yueguo, FAN Ju, LU Wei. "Data wrangling: a key technique of data governance," Big Data Research, vol. 5, pp. 16-25, March 2019.

[8] LI Bingsen, HU Quangui, CHEN Xiaofeng, GAO Bingqiang. "Research and Design of Data Platform for Power Grid Enterprise," Electric Power Information and Communication Technology, vol. 17, pp. 29-34, July 2019.

[9] PENG Yun. "Research on authenticating data rights in Big Data environment," Modern Science & Technology of Telecommunications, vol. 46, pp. 17-20, May 2016.

[10] WANG Yingzi, HOU Jue, ZHANG Yue. "Data management based on block chain technology," Electronic Design Engineering, vol. 27, pp. 87-90+95, June 2019.

[11] LI Xu, WANG Hejian. "Power data sharing mechanism based on blockchain," Ecological Interconnection & Digital Power: Proceedings of 2019 Annual Conference on power industry informatization. Beijing: Telecommunications Science Editorial Department of people's Posts and Telecommunications Press, pp. 180-183, 2019.

[12] LI Yang, MEN Jinbao, YU Han, WANG Sining, MEN Jinbao, FAN Jingang. "Overview of Blockchain Capacity Expansion Technology," Electric Power Information and Communication Technology, vol. 18, pp. 1-9, June 2020.

[13] YANG Qingfeng. "Data Sharing and Privacy Protection: Philosophical Demonstration on the Technological Scheme," Studies in Dialectics of Nature, vol. 34, pp. 111-116, May 2018.

[14] Xia Junjie, Sun Ye, Yang Haitao, Chen Chang. "Research and Application of Data Asset Protection and Trading Platform Based on Blockchain," Designing Techniques of Posts and Telecommunications, pp. 5-9, September 2019.

[15] ZHANG Dongxia, MIAO Xin, LIU Liping, ZHANG Yan, LIU Keyan. "Research on Development Strategy for Smart Grid Big Data," Proceedings of the CSEE, vol. 35, pp. 2-12, January 2015.

[16] LI Bingsen, HU Quangui, CHEN Xiaofeng, GAO Bingqiang. "Research and Design of Data Platform for Power Grid Enterprise," Electric Power Information and Communication Technology, vol. 17, pp. 29-34, July 2019.

[17] ZHU Hongbin, AN Long, YANG Mingchen. "Research and practice on power big data security governance system," Telecommunication Science, vol. 35, pp. 140-145, November 2019.

Authorized licensed use limited to: National University of Singapore. Downloaded on July 03,2021 at 11:09:41 UTC from IEEE Xplore. Restrictions apply.

2019 IEEE International Conference on Blockchain (Blockchain)

Token-based Sharing Control for IPFS

Shigenori Ohashi
NTT Service Evolution Laboratories
NTT Corporation
Kanagawa, Japan
shigenori.ohashi.ur@hco.ntt.co.jp

Hiroki Watanabe
NTT Service Evolution Laboratories
NTT Corporation
Kanagawa, Japan
hiroki.watanabe.eh@hco.ntt.co.jp

Tatsuro Ishida
NTT Service Evolution Laboratories
NTT Corporation
Kanagawa, Japan
tatsurou.ishida.xa@hco.ntt.co.jp

Shigeru Fujimura
NTT Service Evolution Laboratories
NTT Corporation
Kanagawa, Japan
shigeru.fujimura.wg@hco.ntt.co.jp

Atsushi Nakadaira
NTT Service Evolution Laboratories
NTT Corporation
Kanagawa, Japan
atsushi.nakadaira.hy@hco.ntt.co.jp

Junichi Kishigami
Muroran Institute of Technology
Hokkaido, Japan
jay@kishigami.net

***Abstract*—Blockchains can now be used to distribute digital assets independent of specific organizations. One way to handle digital assets on a blockchain is through tokens. Tokens define chunks of data on a blockchain, allowing it to be distributed like virtual currency. However, storing all data related to digital assets in a token is difficult. Data is currently managed separately because storing a large amount of data in a blockchain can cause the ledger to grow large. When the management of data depends on a particular organization, the stability of the token's value depends on that particular organization. In this paper, we propose an organization-independent management method that combines a distributed content-addressable file system with a blockchain. Our proposed method places the data associated with the token and the access information to the token into the distributed content-addressable file system. As a result, the token associated with the data has the sharing control of the data stored in the distributed content-addressable file system, and a configuration independent of the specific contract is realized.**

Keywords—IPFS, blockchain, ethereum, token, access control, sharing control, p2p, content-addressable file system

I. INTRODUCTION

Almost all blockchain platforms have a default cryptocurrency. Bitcoin [1] has “bitcoin” and Ethereum [2] has “ether”. In addition to remittances, ether also pays fees for executing programs called “contracts” on Ethereum. Contracts can be for a variety of processes, and various proof-of-concept cases have been proposed; for example, digital rights management [3], supply chain management [4][5], and self-sovereign identity [6].

Users have recently used contracts to define chunks of data called “tokens” to generate a new cryptocurrency [7] or to trade video game characters [8]. The expectation is that tokens allow circulation of even more diverse types of data like cryptocurrency [9]. However, presently, only simple and small amounts of data are tokenized; large amounts of data such as image data and complicated data are managed outside the blockchain network. The reason is in the formation mechanism of a blockchain ledger. Blockchains are designed to allow participants to validate the ledger without relying on the other party, and the data contained in the transaction is replicated to all participants for validation. Such a formation mechanism of a reliable distributed ledger is inefficient when looking at a blockchain network as distributed storage because it excessively consumes the storage of the entire system. Unlike regular file systems, a blockchain is not suited to handle large amounts of data. Data management outside a blockchain network could be done on a server managed by an operator, however the operator could have a single point of failure and could not guarantee that tokens would continue to circulate. For example, if the visual image data of a digital asset stored on an external server is lost, the token may lose its value. If we can use a blockchain to tie more data to a token and treat it as part of a token, we can expand the use cases for tokens.

A distributed content-addressable file system like the InterPlanetary File System [10] is one of the best distributed systems for managing and sharing files. In an example IPFS implementation, only a person who wants to own a file owns the file's data. This enables more efficient storage utilization compared to a blockchain that replicates its ledger to all participants. A file stored in a content-addressable file system has a unique ID created from a hash or the like. A user can verify that the data obtained through the system is the desired data. Therefore, distributed content-addressable systems should complement the file-management function of a blockchain [11].

Currently a distributed content-addressable file system and a blockchain network operate independently, therefore it is proposed various framework to use the file system with blockchain [12][13]. A distributed content-addressable file system manages a file itself, and a blockchain network manages its metadata such as the ID of the file. It is expected to manage large files in a distributed system like a blockchain using the proposed framework. However, it is still difficult to control file distribution and to limit the nodes that hold the file. If a blockchain and a distributed content-addressable file system can control distribution of a file together, the blockchain can be used for more use cases.

At the time of this writing, a method has been proposed in which file control information is aggregated in a specific contract and IPFS uses this information to control file sharing [15]. In this paper, we propose a method of controlling file sharing on a distributed content-addressable file system according to the token associated with each file. Our method features a data structure in a distributed content-addressable file system that stores access information to a token. Although our method is not limited to a specific blockchain platform, the following description assumes Ethereum as a platform for distributing tokens.

This paper is structured as follows. Section II discusses related works and the focus of our study. Section III describes our method. Section IV describes the sequence of file registration and sharing controls and its implementation. Section V describes our experiments and illustrates their results. Section VI describes the security considerations of

978-1-7281-4693-5/19/$31.00 ©2019 IEEE
DOI 10.1109/Blockchain.2019.00056

our method. Section VII describes the use cases using our method. Section VIII concludes this work.

II. Related work

This section reviews related work and explains the focus of our research.

A. Cost of registering data in the blockchain

A blockchain is a technique of forming a reliable ledger by all participants verifying a new block independently. Data contained in a transaction is included in a new block and is replicated to all participants for validation. Therefore, viewing a blockchain network as mere data storage is redundant and irrational. For example, if we store 1 MB of data using a transaction in a blockchain system that consists of 100 nodes, a 100-MB piece of data is consumed as a system in 1 transaction.

In the rationale of using a blockchain network as data storage using public Ethereum, data storage fees are a measure of rationality. In Ethereum, the transaction fee is called "Gas." We created a contract based on the *SimpleStorage* contract described in the Solidity Reference [17]. Our contract is different from the *SimpleStorage* contract in the data type of *storedData*, which is bytes32 in our contract. We examined the Gas involved in data registration using the contract. When we entered 32 bytes of data into this contract, 41,775 Gas was consumed. Using the median GasPrice of 3 Gwei at the time of this writing, we got 0.000125325 ETH, or about $0.013 when 1 ETH equals $102. This results in it costing about $426 to store 1 MB of data. The cost of storing data using Bitcoin is also discussed in [16]. Given the cost of storing data, using a blockchain system as a distributed storage would not be reasonable.

B. Data distribution using tokens

Various token specifications are discussed in the Ethereum community. For example, there is ERC-20 [7], which allows the user to define a cryptocurrency and ERC-721 [18], which allows the user to define a non-fungible token (NFT). A NFT is suitable for tokenizing something unique. By using this NFT, we expect to capitalize and distribute various data.

The tokens based on ERC-721 are actually used. One of them is CryptoKitties [8]. CryptoKitties is a cat-raising game in which each cat is represented as an ERC-721 token. Each token contains genetic information about a cat's characteristics, however the information is not complete in a blockchain. The CryptoKitties contract does not include cat images or introductory text, and the tokens are designed to make sense only with CryptoKitties servers.

If various data can be associated with tokens and distributed without a specific server, the use cases of data distribution by tokens will expand, such as the use of game items across games or the distribution of digital content by rights holders. Currently, you can use IPFS as a platform to store data if controlling the data distribution is not necessary. However, if there is value on the data side, you may need to control the distribution and limit the users to retain those authorized by the token. Use cases that control the distribution of data are currently difficult to realize.

Data distribution by tokens will expand in the future. In such an era, a number of contracts for generating and managing tokens with various distribution policies can be created by an unspecified number of users. Our goal is to create a distributed data management system for such an era.

C. Managing Files with Blockchains

The conventional practices of managing files distributed outside a blockchain system can be divided into 3 categories according to the fluidity of the files managed by each method.

The first is a simple way of registering a file in a distributed content-addressable file system and recording its file ID in a blockchain [12][13]. In this case, the user may register the encrypted file. In the case of IPFS, the files are freely shared and distributed when requested by others. It is also proposed to register a file in IPFS and use the file on IPFS from the blockchain using Oraclize [14]. In these methods, the blockchain and the distributed content-addressable file system operate independently of each other and then files on IPFS are freely shared, therefore the fluidity of the file is high. With IPFS, data that participants perceive as valuable is naturally replicated and less likely to be lost. Since the blockchain guarantees the occurrence of data, this method is suitable for managing highly public data such as scientific research data or open data.

The second is distributed cloud storage such as Storj[1], Sia[2], and Filecoin[3]. Distributed cloud storage is a system that connects storage and file providers. File retention agreements are recorded in a blockchain, and the storage provider stores the files in its own storage. Provided files are encrypted and not read by the storage provider. Because distributed cloud storage is intended to store files, file movement is limited and file fluidity is low.

The third is allowing the distribution of files under certain conditions. Steichen et al. [15] describe the file sharing conditions in a specific contract on the blockchain and control the file sharing on the IPFS according to the conditions.

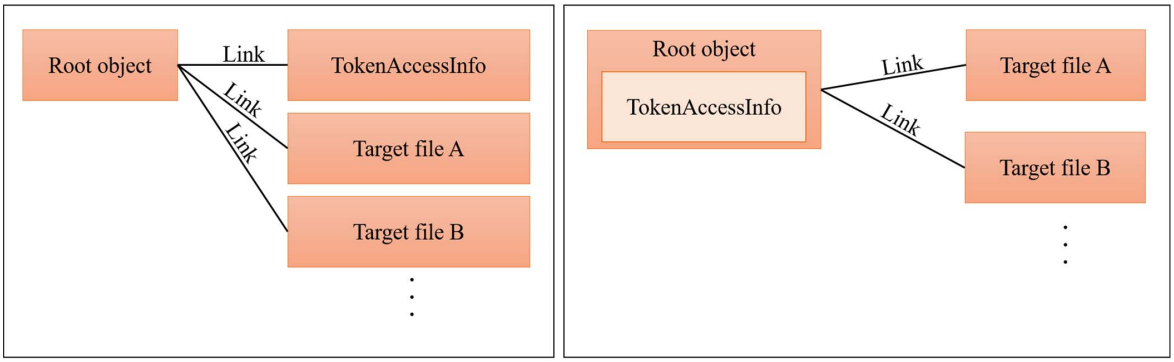
Our research is similar to Steichen et al. in that it aims at control that permits the distribution of a file under some conditions. Our unique feature is that we focus on tokens and aim to enable distributed content-addressable file systems to perform correct file sharing control based on the token associated with the file. When a file is managed by a specific contract, we may assume that the contract controlling the file is known to everyone, however a method of tying tokens and files is necessary in order to control the file according to the information in each token associated with the file.

As Steichen et al. created an IPFS-based accl-IPFS as a distributed file system, we also created an IPFS-based system as a distributed file system linked with a blockchain system. The reasons are as follows: an IPFS is a distributed system with no single point of failure, an IPFS is a content addressable and the retrieved files from IPFS can be verified, and the files are only shared when other peers request them. The advantage of the third feature is that the owner of the file can be clearly defined, and the modification to the IPFS to realize a control the file can be reduced.

[1] https://storj.io/

[2] https://sia.tech/

[3] https://filecoin.io/



(a) Type 1 (b) Type 2

Fig. 1: Data structure in IPFS

III. Proposed Method

This chapter explains our proposal method. Since an IPFS is the implementation closest to our method, we will explain our method based on an IPFS, starting with the structure of data to be registered in an IPFS, then the method of registering a file based on the data structure, and finally the method of controlling file sharing using the data structure.

A. Structure of data to be registered in IPFS

Figure 1 shows a data structure to be stored in the IPFS. In our proposed method, the IPFS holds a target file and a token access information file (tokenAccessInfo). The token access information file has the information to refer the token. The target file and the token access information file are linked to another IPFS object that organizes them. This IPFS object is referred to as a root object. Type 2 in figure 1 shows a data structure where token access information is described in the root object. The root object shown in Fig. 1 is associated with a token on the blockchain. When a data structure is like type 1 in Fig. 1, the root object should be obtained first and then the token access information should be obtained using the ID written in the root object. When a data structure is like type 2 in Fig. 1, you can directly obtain the token access information from the root object. In both cases, you can obtain the information you need using a root object ID. By storing the target file ID as a key and the root object ID as a value in the distributed hash table (DHT), you can locate the token access information using the target file ID efficiently. In the case of Ethereum, the token access information is the contract address and application binary interface (ABI) information.

When the IPFS is used in conjunction with a blockchain in a conventional way, the ID of the file in the IPFS is usually registered in the blockchain and the user who wants the file gets the ID from the blockchain and gets the file from the IPFS. Conventionally, the contract on the blockchain is known to the user. By storing the token access information in the IPFS in association with the file (as in our method), we do not need to know the contract in advance and we can know the contract information when necessary. If the file associated with the token is worthwhile, the file ID or the ID of the object corresponding to the file may be known to users and the contract of the token may be unknown. Our method can cope with the age when tokens managed by various contracts are distributed.

It is also conceivable to store a token access information for each target file only in the DHT without using the data structure as shown in Fig. 1. However, since you cannot verify the association between the target file and the token access information or its authenticity only by the information stored in the DHT, we bind the token access information and the target file as IPFS objects.

B. Method of registering a file based on the data structure

We describe the procedure up to the file registration based on the type 1 data structure shown in Fig. 1. The similar procedure is also performed based on the type 2 structure shown in Fig. 1. The flow is 1) acquiring the ID of the file to be registered, 2) generating the token, and 3) registering the file in the IPFS, assuming that the contract to manage the token was created in advance.

A user who wants to register a target file collects the address and ABI of the contract and creates a file that includes access information to the token. For example, we name this file tokenAccessInfo. The user uses the functionality of IPFS to make the tokenAccessInfo and the target file into IPFS objects, creates the root object to hold them together, and gets the ID. The user generates a token containing the ID on the blockchain system. For example, the ID of the root object may be a token ID. When generating the token, the user registers the blockchain address of the generator and the address of the IPFS peer in the token as the Owner. The owner confirms the registration to the blockchain is completed, and then registers the tokenAccessInfo, the target file, and the root object in the IPFS.

C. Method of controlling file sharing using the data structure.

File sharing is when a user requests a file through a distributed file system and the file owner responds to the request. Our method has a breakpoint before the file owner responds. We assume that a user who wants to obtain a target file (file requester) knows the ID of the root object of the file.

The file requester peer makes a request to obtain the root object and the tokenAccessInfo file using the root object ID. Using standard IPFS commands, you can request a tokenAccessInfo file from the root object and link name

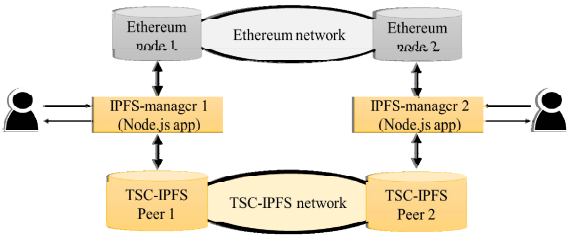


Fig. 2: Implementation system

tokenAccessInfo. The owner peer that receives the request specifies the object requested from the ID, confirms that the object is not the target file but the root object and the tokenAccessInfo file, and returns their IPFS objects. The file requester who obtains the tokenAccessInfo issues a transaction to request the acquisition of usage rights to the blockchain using the access information to the token described in the tokenAccessInfo file. At this time, the file requester registers the ID of his own IPFS peer. After confirming the issuance of the usage rights, the file requester specifies the root object and requests the owner peer described in the token.

The owner peer that receives the file request specifies the tokenAccessInfo from the ID included in the request and refers to the ledger. The owner peer confirms that the requesting peer has the usage rights and transmits the IPFS objects of the target file. The file requester peer that receives the target file confirms that the received objects are the desired object by the ID of the objects.

The index DB can be constructed to efficiently find the root object and tokenAccessInfo from the ID of the object. The index DB can be reconstructed by periodically verifying the objects held in each IPFS peer.

IV. IMPLEMENTATION

Our method differs from ordinary IPFS in that there is a breakpoint when a file acquisition request is received, and the continuation of processing is determined by referring to the blockchain. We implement the function to determine the continuation of processing as an application outside of IPFS in order to minimize the modification to IPFS. A modified version of IPFS that adds a breakpoint when a file request is received is hereinafter referred to as a TSC-IPFS. In addition, an application that determines whether to continue processing is referred to as the IPFS-manager. We implemented the IPFS-manager as an application of node.js. Figure 2 shows an overall view of our implementation system. We used Ethereum as a blockchain platform. The software versions used in our system are summarized in Tables 1–3.

Table 1: Software for blockchain

Software name	Version
geth	1.8.18-stable

Table 2: Software for IPFS-manager

Software name	Version
js-ipfs-api	18.2.1
Node.js	8.11.2

Table 3: Based software of TSC-IPFS

Software name	Version
go-ipfs	0.4.17

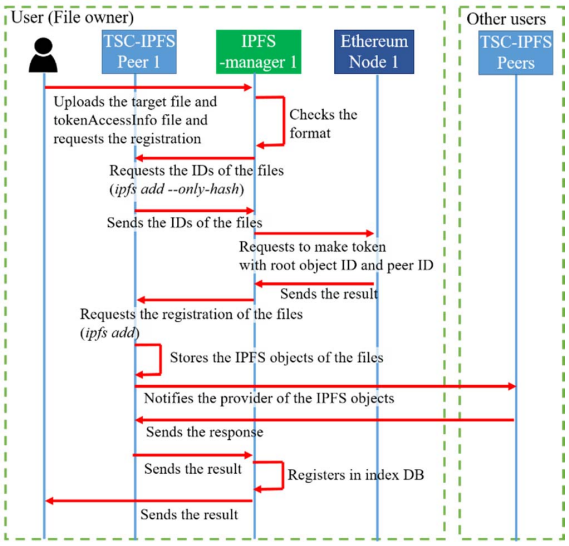


Fig. 3: Sequence of adding file

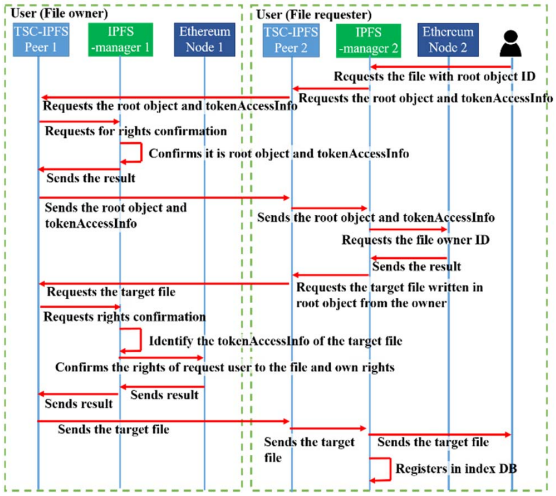


Fig. 4: Sequence of sharing file

A. *Add file sequence.*

Figure 3 shows the sequence of adding files implemented in the data structure in Fig. 1. The file owner uploads the tokenAccessInfo file and the target file to the IPFS-manager 1, then creates a folder that contains these files. IPFS-manager 1 creates the objects of the data structure in Fig. 1 by using the existing IPFS command (*ipfs add --only-hash*) for that folder and obtains the IDs. IPFS-manager 1 registers the acquired IDs in the Ethereum using the user's account and generates a token. Upon successful token generation, the IPFS-manager 1 executes the *ipfs add* command for that folder and registers them in TSC-IPFS peer 1. TSC-IPFS peer 1 performs the same processing as ordinary IPFS and

completes file registration processing. After receiving notification of the completion of the registration from TSC-IPFS peer 1, IPFS-manager 1 creates an index DB that efficiently finds the root object ID from the target file ID.

As you can see from the sequence, these file registration processes can be realized using the existing API of IPFS. In addition to normal IPFS file add processing, our sequence has ID acquisition of the files and registration processing to the Ethereum.

B. *Get file sequence.*

Figure 4 shows an example file sharing sequence for a user who was already granted usage rights by the file owner. The file requester issues a file request that includes the root object ID to IPFS-manager 2. IPFS-manager 2 requests acquisition of the root object and the tokenAccessInfo file to TSC-IPFS peer 2. TSC-IPFS peer 2 finds the peer that has the root object and the tokenAccessInfo file. Now, the peer is TSC-IPFS peer 1. TSC-IPFS peer 2 requests the files to TSC-IPFS peer 1. The TSC-IPFS peer 1 that received the request of the root object and the tokenAccessInfo file asks IPFS-manager 1 whether to send the files to TSC-IPFS peer 2. IPFS-manager 1 confirms that the object to be requested is not controlled by the token and permits TSC-IPFS peer 1 to share the files. In our implementation, the index DB is utilized in the processing here. IPFS-manager 2 obtained the root object and tokenAccessInfo and uses tokenAccessInfo to obtain the owner's TSC-IPFS peer ID from the Ethereum. IPFS-manager 2 gets the file ID from the root object. IPFS-manager 2 asks TSC-IPFS peer 2 to request owner peer to send the target file. Now, the owner peer is TSC-IPFS peer 1. Upon receiving the request, TSC-IPFS peer 1 (owner peer) asks IPFS-manager 1 whether to send the file to TSC-IPFS peer 2 or not. IPFS-manager 1 identifies the root object from the ID described in the request and acquires the associated tokenAccessInfo. The index DB is utilized in the processing here. IPFS-manager 1 refers Ethereum Node 1 using tokenAccessInfo, confirms that the requester has a permission, and gives TSC-IPFS peer 1 permission to respond to the request. TSC-IPFS peer 1 sends the target file to TSC-IPFS peer 2.

Compared with the conventional IPFS, the implemented system has the process to acquire the root object and the tokenAccessInfo, and the process to refer to Ethereum. When the user submits an application for usage rights to the Ethereum and the owner approves it, the root object and the tokenAccessInfo file are shared with the user before the user requests the target file. In this case, the sequence that shares the root object and the tokenAccessInfo file in Fig. 4 is not executed.

V. EXPERIMENT & THE RESULT

In adding and sharing files, the implemented system based on our method has additional processes: generation and transmission of root objects and tokenAccessInfo. Furthermore, our system has a process that depends on the Ethereum. Therefore, this implemented system has a longer processing time than IPFS. In this chapter, we measure the time required for each process to investigate the influence of additional processes.

A. *Environment of experiment*

We built two virtual machines on the same PC and measured the processing time. Specifications of the PC, virtual machines, and software used for building virtual machines are listed in the Tables 4 and 5. Ethereum was operated with Proof of Authority (PoA), and block generation was set at intervals of 3 seconds on average. The contract was created by inheriting from ERC 721. The management token having the ID of the root object was generated on the contract, and a user who can obtain the target file was managed by the management token. The configuration file was identical for IPFS and TSC-IPFS. The configuration file was modified from the default only the Bootstrap and Addresses items necessary for network formation.

Table 4: Host PC specification

Host PC	
OS	Windows 10
CPU	Intel Core i7-6700 3.4 Ghz
RAM	16 GB
ROM	SSD 512 GB
Virtualization software	VirtualBox 6.0

Table 5: Virtual PC specification

Virtual PC	
OS	Ubuntu 16.04
CPU	1 core
RAM	4 GB
ROM	30 GB

B. *Method of measurment and result*

We prepared two image files of different sizes (2.39 MB, 179 KB). We registered and shared each image file with IPFS and TSC-IPFS, and we measured the time required for each processing. For TSC-IPFS, we also measured the time required for the IPFS-manager and Ethereum processing. We initialized IPFS, TSC-IPFS, IPFS-manager every time enforcement, and measured both image files 10 times.

We summarize the average time required for file registration processing in Fig. 5. Figure 6 summarize the average time required for the file sharing process.

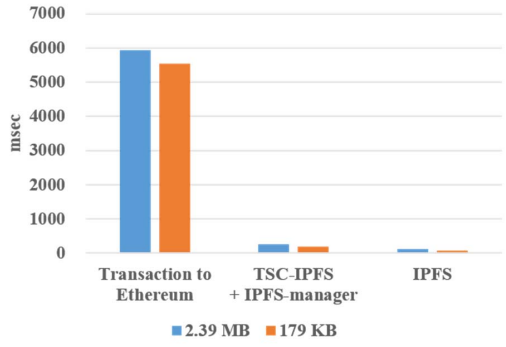


Fig. 5: Average time required for adding file

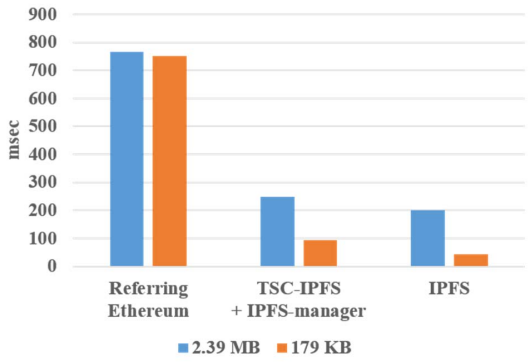


Fig. 6: Average time required for sharing file

C. *Discussion*

In the sequence of file registration, there is transaction issuance to Ethereum. In the experimental environment, by setting the block generation interval to about 3 seconds, the time required for transaction processing was shortened compared with the public Ethereum. Comparing the time required for file registration by IPFS and the time required for transaction processing, the time required for IPFS file registration was sufficiently short. Even if the time required for processing TSC-IPFS + IPFS-manager is compared with the time required for transaction processing, the time required for transaction processing is large, and the transaction processing required for file registration is dominant. When comparing TSC-IPFS + IPFS-manager and IPFS, TSC-IPFS + IPFS-manager takes more processing time because they have additional processing, however the system works in conjunction with a blockchain, therefore the range is acceptable. We estimate that the reason the processing time of the 2.39 MB file is longer than that of the 179 KB file is that the 2.39 MB file needs to be divided into more IPFS objects and pin these objects than the 179 KB file does.

Because there is no change in the ledger in file sharing, the file sharing sequence is not affected by the block generation time of Ethereum. The time required for Ethereum is small compared to the file registration sequence. However, the time required for processing Ethereum is visibly larger than IPFS or TSC-IPFS + IPFS-manager even in the file sharing sequence. Comparing TSC-IPFS + IPFS-manager and IPFS, the difference of time required for processing is almost the same even if the file size increases. This can be determined as the influence of the additional process in our method is almost the same even if the size of the target file increases. The difference of time required for processing is due to the influence of the time required for data transfer of the target file. When we form an actual system, the influence of the additional process becomes relatively smaller because the network latency is added. The time due to the additional process in our method is also small compared with the time required for processing for Ethereum. The time due to the additional process can be considered as an acceptable range considering it is a system that operates in cooperation with the blockchain.

VI. SECURITY

Our proposal method controls the sharing of the target file by referring to the token tied to the file. The correct tokenAccessInfo must be associated with the target file in order to reference the correct token. Our proposal method has the flexibility to generate a new root object by changing the tokenAccessInfo associated with the same target file or by changing the configuration of the target files. However, there is a concern that this feature may cause an unauthorized tokenAccessInfo to be bound to the target file. Current IPFS and TSC-IPFS cannot control the creation of new objects with links to existing objects. Our proposed scheme needs a mechanism to properly control token generation on the blockchain. If unauthorized information can be registered on the blockchain, an unauthorized tokenAccessInfo may be generated. If a malicious user attempts to create an unauthorized token on the same contract as the original user, it would be easy to control the creation of a new token since it is a token for the same target file. If a token is generated on a contract different from the contract on which the original user has registered his/her token, cooperation between the contracts is required in order to detect an unauthorized token registration. One of our next tasks is to study the design of contracts to prevent the generation of unauthorized tokens and the system design to detect the use of unauthorized tokens in the operation.

In our implemented system, index DB is constructed in order to efficiently find the root object and tokenAccessInfo from the requested ID, and determine whether the requested file is the target file or not. If the index DB is cracked, correct control may not be possible. In our method, since each peer can reconstruct the index DB from the data it holds, the risk of unauthorized file sharing can be reduced by periodically reconstructing the index DB.

VII. USE CASE

With our method, data can be associated with tokens for distribution. Therefore, our method enables the distribution of digital assets that are worth collecting. Examples include in-game items and autographs of celebrities. Celebrity autographs are often not digitized, however by tokenizing them, we can find out when they were created, which will increase the value of their collection.

Our method could also be applied to Decentralized Identifier (DID) [19] management. Personally-Identifiable Information (PII) associated with a DID should not be freely distributed, even if encrypted. Our method could be used to distribute the encrypted PII. Since the user requesting the PII can obtain reference information of the contract from TSC-IPFS, there is no need to know the contract in advance.

VIII. CONCLUSION

In this paper, we proposed a method of controlling file distribution based on tokens linked to the file. Our method makes it possible to link large data to tokens and distribute them according to tokens and to eliminate the conditions for which the contract is known to all users. Our method enables the users to distribute their data by themselves. Compared to regular IPFS, our method takes a lot of time to register and share files but is acceptable assuming it operates in conjunction with the blockchain system. Information management on the blockchain is necessary to control the correct distribution of files using our method. In the future,

we will study the token specification and monitoring method in addition to TSC-IPFS and IPFS-manager function enhancement.

REFERENCES

[1] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," 2008.

[2] G. Wood, "Ethereum: A secure decentralised generalised transaction ledger," *Ethereum Proj. Yellow Pap.*, vol. 151, pp. 1–32, 2014.

[3] Z. Ma, M. Jiang, H. Gao, and Z. Wang, "Blockchain for digital rights management," *Futur. Gener. Comput. Syst.*, vol. 89, pp. 746–764, 2018.

[4] T. Bocek, B. B. Rodrigues, T. Strasser, and B. Stiller, "Blockchains everywhere - a use-case of blockchains in the pharma supply-chain," in *2017 IFIP/IEEE Symposium on Integrated Network and Service Management (IM)*, 2017, pp. 772–777.

[5] K. Toyoda, P. T. Mathiopoulos, I. Sasase, and T. Ohtsuki, "A novel blockchain-based product ownership management system (POMS) for anti-counterfeits in the post supply chain," *IEEE Access*, vol. 5, pp. 17465–17477, 2017.

[6] C. Lundkvist, R. Heck, J. Torstensson, Z. Mitton, and M. Sena, "UPORT: A PLATFORM FOR SELF-SOVEREIGN IDENTITY."

[7] "ERC-20 Token Standard | Ethereum Improvement Proposals." [Online]. Available: http://eips.ethereum.org/EIPS/eip-20. [Accessed: 18-Mar-2019].

[8] "CryptoKitties | Collect and breed digital cats!" [Online]. Available: https://www.cryptokitties.co/. [Accessed: 18-Mar-2019].

[9] V. Sanghavi, R. Doshi, D. Shah, and P. Kanani, "Blockchain Based Asset Tokenization." Nov. 2018.

[10] J. Benet, "Ipfs-content addressed, versioned, p2p file system," *arXiv Prepr. arXiv1407.3561*, 2014.

[11] J. Eberhardt and S. Tai, "On or off the blockchain? Insights on off-chaining computation and data," in *European Conference on Service-Oriented and Cloud Computing*, 2017, pp. 3–15.

[12] S. Wang, Y. Zhang, and Y. Zhang, "A blockchain-based framework for data sharing with fine-grained access control in decentralized storage systems," *IEEE Access*, vol. 6, pp. 38437–38450, 2018.

[13] "Build a simple Ethereum + InterPlanetary File System (IPFS)+ React.js DApp." [Online]. Available: https://itnext.io/build-a-simple-ethereum-interplanetary-file-system-ipfs-react-js-dapp-23ff4914ce4e?gi=f6004f641c74. [Accessed: 29-May-2019]

[14] "Using IPFS with Ethereum for Data Storage | Tooploox" [Online]. Available: https://www.tooploox.com/blog/using-ipfs-with-ethereum-for-data-storage. [Accessed: 29-May-2019].

[15] M. Steichen, B. Fiz Pontiveros, R. Norvill, W. Shbair, and R. State, "Blockchain-Based, Decentralized Access Control for IPFS," 2018.

[16] X. Xu *et al.*, "A taxonomy of blockchain-based systems for architecture design," in *2017 IEEE International Conference on Software Architecture (ICSA)*, 2017, pp. 243–252.

[17] "Solidity — Solidity 0.5.7 documentation." [Online]. Available: https://solidity.readthedocs.io/en/latest/. [Accessed: 18-Mar-2019].

[18] "ERC-721." [Online]. Available: http://erc721.org/. [Accessed: 18-Mar-2019].

[19] "Decentralized Identifiers (DIDs) v0.11" [Online]. Available: https://w3c-ccg.github.io/did-spec/. [Accessed: 18-Mar-2019].

Decentralized Digital-Asset Exchanges: Issues and Evaluation

Wei-Tek Tsai
Digital Society & Blockchain Laboratory
Beihang University
Beijing, China
Arizona State University
Tempe, AZ 85287, USA
Beijing Tiande Technologies
Beijing, China
Andrew International Sandbox Institute
Qingdao, China
IOB Laboratory
National Big Data Comprehensive Experimental Area
Guizhou, China
tsai@tiandetech.com

Juan He
Digital Society & Blockchain Laboratory
Beihang University
Beijing, China
xiongbao@buaa.edu.cn

Rong Wang
Digital Society & Blockchain Laboratory
Beihang University
Beijing, China
wangrong@buaa.edu.cn

Enyan Deng
Beijing Tiande Technologies
Beijing, China
enyandeng@yahoo.com

2020 3rd International Conference on Smart BlockChain (SmartBlock) | 978-1-6654-4073-8/20/$31.00 ©2020 IEEE | DOI: 10.1109/SMARTBLOCK52591.2020.00024

***Abstract*—Recently, Decentralized Digital-Asset Exchanges (DDAE) have received significant attention. This paper proposes six criteria to evaluate DDAE based on the Principles of Financial Market Infrastructure (PFMI), evaluate current DDAE systems with respect to two main protocols: inter-chain and exchange protocols. This paper introduces four inter-chain protocols and six exchange protocols, summarizes these protocols into several technical models and analyzes their characteristics. These analyses provide technical reference for the DDAE design in the future.**

Keywords—Decentralized Exchange, Blockchain, Inter-chain Technology, Exchange Protocol

I. Introduction

Decentralized Digital-Asset Exchanges (DDAE) received significant attention recently. However, they still have low trading volume and they are often designed to resist regulation. The transactions made in current DDAE will not be visible to regulators, but this may be changed in the future as regulators may impose rules on them. It is likely that regulators may ask DDAE systems to make their transactions visible, and impose regulations such as KYC/AML rules. In this case, these regulated DDAE can be used to supplement centralized exchanges to reduce their workload.

This paper analyzes DDAE technologies according to six evaluation criteria based on Principles of Financial Market Infrastructure (PFMI). This paper is organized as follows: Section 2 presents six DDAE evaluation criteria based on PFMI; Section 3 discusses inter-chain protocols; Section 4 analyzes its characteristics and performance of these protocols; finally, section 5 compares these different DDAE models.

II. Six Evaluation Criteria

PFMI was proposed to prevent the recurrence of the 2008 Global Financial Crisis (GFC), as one reason that the GFC happened is that then financial infrastructures were not properly designed for the stressed market condition. During the GFC, as one system in a country breaks down, it may cause partnering systems in other countries to break down too, and this propagates the financial crisis from one country to another.

Committee on Payment and Settlement Systems (CPSS) and International Organization of Securities Commissions (IOSCO) jointly issued PFMI to address this problem [1]. PFMI requires any financial systems to consider service compliance, risk resistance, and scalability. According to PFMI, any exchanges including DDAE need to satisfy the following six principles [3, 4]:

Reliability and fault tolerance: Reliability is reflected in two aspects: reliability of the trading platform system [4], and how the system reacts when parts of the system failed.

Scalability: When the workload increases, the system can still perform with similar performance as before [5].

Regulation compliance: the system design and operation process should follow local regulation rules.

Data transparency: Regulatory authorities can query account and transaction information at any time and efficiently without incurring excessive delays.

Operational efficiency: The system should be efficient in performing tasks including computation, communication, business processes [4].

Rollback: Any regulated financial system should support transaction rollbacks after trading. For example, if a transaction has been determined to be irregular, the involved parties have the option to roll back the transaction within two business days.

III. Inter-chain Protocols

Inter-chain protocols are used for data communication between different blockchain (BC) systems, this process has three sub-processes:

1) Transaction disclosing: During inter-chain protocol process, transactions information will be made available. For regulated transactions, almost all the related information must be available, but for other transactions, only the necessary information needed for transactions will be made available.

2) Data consensus: Consensus algorithms must be used in inter-chain protocols to ensure that all the parties involved have a consistent view.

978-1-6654-4073-8/20/$31.00 ©2020 IEEE
DOI 10.1109/SmartBlock52591.2020.00024

Authorized licensed use limited to: University of Prince Edward Island. Downloaded on June 08,2021 at 11:30:26 UTC from IEEE Xplore. Restrictions apply.

3) Results on the chain: When the transaction is completed, all the results should be stored in BCs to ensure regulation compliance and record keeping.

A. Inter-Chain Atomic Swap Protocol

The inter-chain atomic swap protocol is designed to reduce the communication workload between two transaction parties while they do not need to trust each other, and to ensure transaction atomicity. This protocol ensures transaction atomicity, and can be combined with other BC technologies, such as smart contracts (SCs). There-fore, the performance of the protocol largely depends on the underlying system. The characteristics of the protocol are summarized in Table 1.

TABLE I. CHARACTERISTICS OF INTER-CHAIN ATOMIC SWAP PROTOCOL

Principles	Process		
	1	*2*	*3*
Reliability	Transactions broadcast	Decided by BC	Good
Scalability	Scalable	Decided by BC	Decided by BC
Regulation	Difficult	Difficult	Difficult
Transparency	Decided by BC	Good	Good
Efficiency	Decided by SC	Decided by SC	/
Rollback	Decided by BC	No	Yes

B. Notary Model

The notary model is used in the Corda system. Corda is similar to a permissioned BC system (but is not a BC system) that uses a model similar to Bitcoin's Unspent Transaction Object (UTXO), without the concept of account [6]. Corda introduced notary nodes to record transactions and consensus data. It triggers consensus protocols only when it involves transactions between different notary nodes, and it uses the PBFT protocol. In other cases, there is no need for consensus. Users need to send transaction data to the notary node only, the notary node will record the transaction and will not trigger consensus with other notary nodes. The characteristics of the protocol are summarized in Table 2.

TABLE II. CHARACTERISTICS OF NOTARY MODEL

Principles	Process		
	1	**2**	**3**
Reliability	Transactions broadcast	Decided by BC	Good
Scalability	Scalable	Decided by BC	Decided by BC
Regulation	Difficult	Difficult	Difficult
Transparency	Decided by BC	Good	Good
Efficiency	Decided by SC	BC Decided	/
Rollback	Decided by BC	No	Yes

C. Inter-BC Communication (IBC)

The inter-BC communication (IBC) protocol is a protocol for inter-chain transac-tions in the Cosmos system. Cosmos provides one approach, realizes multi-chain parallel operation, and makes up for the scalability needs of platforms such as Lightning Network [7, 8]. Cosmos is composed of multiple Zones and Cosmos Hub.

The Cosmos Hub uses the Tendermint consensus algorithm to manage the inter-chain ledger in the Cosmos network. The transactions between Zones are completed through the Cosmos Hub. These are centralized approaches, thus they may become the bottleneck of the system. If transactions are completed within a hub, the system works well; however, once they need to through the center, with central hub performing consensus. It does not consider the regulation compliance. The characteristics of IBC protocol are summarized in Table 3.

TABLE III. CHARACTERISTICS OF IBC PROTOCOL

Principles	Process		
	1	**2**	**3**
Reliability	Transactions broadcast	Tendermint algorithm	Zones and Hub has
Scalability	Issues	Issues	Issues
Regulation	Difficult	Difficult	Difficult
Transparency	Zones and Hub has	Good	Good
Efficiency	Decided by SC	Decided by SC	Decided by SC
Rollback	No	No	No

D. Golden Monkey Model

The golden monkey model is an inter-chain transaction model. It is a distributed protocol and allow concurrent transactions to occur. The transactions done can be regulated and support scalability as the system does not have central controller that may become the bottleneck [9]. The golden monkey model consists of participation BCs and intermediate BCs. The intermediate BC is designed to provide the communication between the participating BCs. Supervisory nodes participate in the intermediate chain and the participation chain to ensure the regulation compliance. Also the model allows additional BCs to be added into the model without changing the overall structure. The characteristics of Golden Monkey Model are summarized in Table 4.

TABLE IV. CHARACTERISTICS OF GOLDEN MONKEY MODEL

Principles	Process
	3
Reliability	Good
Scalability	Scalable
Regulation	Yes
Transparency	Good
Efficiency	Consensus efficiency
Rollback	Yes

E. Analysis

Summarizing the above inter-chain communication schemes, it can be classified into three models: atomic swap model, notary model and intermediate BC model. The characteristics of these three patterns are analyzed below.

Authorized licensed use limited to: University of Prince Edward Island. Downloaded on June 08,2021 at 11:30:26 UTC from IEEE Xplore. Restrictions apply.

The atomic swap model is based on hash lock, and its main function is to realize the atomic operation of transactions. The characteristics of atomic swap model are summarized in Table 5.

TABLE V. CHARACTERISTICS OF ATOMIC SWAP MODEL

Principles	Characteristics
Reliability	Depended on BC
Scalability	Scalable
Regulation	No
Transparency	Depended on BC
Efficiency	Depended on BC
Rollback	No

The notary model is to set up notary nodes in the system, the nodes are independent of each other, and each maintains its own ledger. The characteristics of notary model are summarized in Table 6.

TABLE VI. CHARACTERISTICS OF NOTARY MODE

Principles	Characteristics
Reliability	Depended on consensus protocols
Scalability	Scalable
Regulation	Difficult
Transparency	Consensus nodes are visible
Efficiency	Depended on consensus protocol
Rollback	/

The intermediate BC model is currently a commonly used model. This model adds intermediate BCs between various BCs to maintain transaction consistency between BCs. The characteristics of intermediate BC model are summarized in Table 7.

TABLE VII. CHARACTERISTICS OF INTERMEDIATE BC MODEL

Principles	Characteristics
Reliability	Depended on design
Scalability	Depended on design
Regulation	Depend on design
Transparency	System visible
Efficiency	Storage
Rollback	/

IV. DDAE TRANSACTION PROTOCOLS

This section focuses on typical transaction protocols such as Atomic Swap, Uniswap, Bancor, 0x, Kyber, Airswap protocol, and Panda model. DDAE systems have four main functionalities:

1) Fund custody: Both trading parties need to deposit their assets in the exchange to facilitate trading.

2) Transaction matching: The matching transaction refers to the trading market determines the transaction price and generates an electronic transaction contract.

3) Transaction settlement: When the two parties have confirmed the transaction order, the exchange will enter the transaction assets of the two parties into the other's account respectively.

4) Funds withdrawal: Users can withdraw funds from the account through the exchange.

A. Atomic Swap Protocol

Atomic swap protocol is a cross-ledger asset exchange solution [10]. This protocol is used as the basic protocol of decentralized transaction protocols, and can be called by other protocols to ensure the transaction atomicity. This protocol is mainly used for the fund settlement after the trade. The characteristics of this protocol are summarized in Table 8.

TABLE VIII. CHARACTERISTICS OF ATOMIC SWAP PROTOCOL

Principles	Process
	3
Reliability	Good atomicity and security
Scalability	Decided by BC
Regulation	Difficult to regulate
Transparency	Good
Efficiency	Decided by BC
Rollback	Yes

B. Uniswap Protocol and Bancor Protocol

Uniswap [11] and Bancor protocols [12] address the market liquidity problem in DDAE systems, and realize the automatic adjustment of token price. Uniswap uses the constant product automatic market making model, Bancor protocol introduces a smart token BNT, and maintains BNT and other tokens at a fixed ratio through the Bancor formula. Smart tokens and other tokens are connected using connectors. The characteristics of these two protocols are summarized in Table 9.

TABLE IX. CHARACTERISTICS OF UNISWAP PROTOCOL AND BANCOR PROTOCOL

Principles	Process	
	1	3
Reliability	Use SC	Decided by consensus
Scalability	Decided by BC	Decided by BC
Regulation	Decided by BC	Difficult
Transparency	Decided by BC	Decided by BC
Efficiency	/	Consensus efficiency
Rollback	/	Decided by BC

C. 0x protocol

0x is an open exchange protocol used on Ethereum [13]. The 0x protocol integrates the automated market maker and the status channel, overcomes the shortcomings of the two plans, and proposes an "off-chain matching, on-chain settlement" operation plan. 0x runs on the Ethereum, so the performance depends on the Ethereum performance. Digital assets can be safely transferred, transaction information is

Authorized licensed use limited to: University of Prince Edward Island. Downloaded on June 08,2021 at 11:30:26 UTC from IEEE Xplore. Restrictions apply.

public, but user information is confidential. The system has good privacy, and is difficult to regulate. The characteristics of 0x protocols are summarized in Table 10.

TABLE X. CHARACTERISTICS OF 0X PROTOCOL

Principles	Process
	2/3
Reliability	Good security
Scalability	Decided by SC
Regulation	Difficult to regulate
Transparency	Good
Efficiency	Decided by BC
Rollback	No

D. Kyber Protocol

Kyber protocol is an exchange protocol on the Ethereum, providing users with appli-cations for capital conversion between multiple assets, providing an asset exchange interface for both parties to transactions, reducing transaction risks and improving transaction efficiency [14]. In principle, the Kyber protocol uses the same method as the 0x protocol. The characteristics of Kyber protocol are summarized in Table 11.

TABLE XI. CHARACTERISTICS OF KYBER PROTOCOL

Principles	Process	
	1	2/3
Reliability	Use Reserve	Decided by SC
Scalability	Decided by Reserve manager	Decided by SC
Regulation	Difficult	Difficult
Transparency	Good	Good
Efficiency	/	Decided by BC
Rollback	/	Yes

E. Airswap Protocol

The Airswap platform is built on the Ethereum and uses ERC20 tokens [15]. The solutions of off-chain negotiation and on-chain settlement provide point-to-point transactions and support free price negotiation, commission orders, transaction matching, transaction settlement, and other services. Currently, Airswap allows users to transact without considering their risk of anonymity. Thus, AirSwap needs to be strengthened in supporting supervision. The characteristics of the Airswap protocol are summarized in Table 12.

TABLE XII. CHARACTERISTICS OF AIRSWAP PROTOCOL

Principles	Process		
	1	2/3	4
Reliability	Use SC	Decided by SC	Decided by SC
Scalability	Decided by SC	Scalable	Decided by SC
Regulation	Difficult	Difficult	Decided by SC
Transparency	Good	Good	Decided by SC
Efficiency	/	Decided by SC	/
Rollback	/	No	/

F. Panda Model

The panda model consists of three types of components, ABCs (Account BCs), TBC (Transaction BCs), and supervisory nodes (also are ABCs). An ABC maintains account information, including basic user information and asset balances [16]. All information modification to accounts will be recorded by the ABC. A TBC is responsible for processing all transactions, including inter-chain transactions. In this way, ABCs and TBCs keep same information but data are organized differently. This design follows a software engineering principle where one module or data structure performs only one function [2], and this will significantly reduce the system complexity. Supervisory nodes monitor all system operations and can quickly locate violating accounts once they find potential transaction violations. The double-chain structure separates user information from trading information so to maintain the consistency of accounts and transaction records at ABCs and TBCs, and to protect data privacy. The characteristics of the panda model are summarized in Table 13.

TABLE XIII. CHARACTERISTICS OF PANDA MODEL

Principles	Process
	1/2/3
Reliability	Decided by SC
Scalability	Scalable
Regulation	Yes
Transparency	Good
Efficiency	/
Rollback	Yes

G. Analysis

These protocols can be classified into atomic swap model, automatic pricing model, relay model, peer-to-peer model, and double BC model.

Atomic Swap protocol is a typical application of the atomic swap model. The characteristics are summarized in Table 14.

TABLE XIV. CHARACTERISTICS OF ATOMIC SWAP MODEL

Principles	Characteristics
Reliability	Good
Scalability	Decided by BC
Regulation	Decided by BC
Transparency	Decided by BC
Efficiency	Decided by BC
Rollback	Yes

Automatic pricing model is often in decentralized finance. Uniswap and Bancor protocols are two examples. The characteristics of this model are shown in Table 15.

TABLE XV. CHARACTERISTICS OF AUTOMATIC PRICING MODEL

Authorized licensed use limited to: University of Prince Edward Island. Downloaded on June 08,2021 at 11:30:26 UTC from IEEE Xplore. Restrictions apply.

Principles	Characteristics
Reliability	Executing pricing algorithms
Scalability	Decided by SC
Regulation	Decided by BC
Transparency	Decided by BC
Efficiency	Algorithm efficiency
Rollback	Yes

Relay model adopts the off-chain matching but on-chain settlement. The transaction matching process is handed over to the relay to improve the efficiency. The characteristics of this model are shown in Table 16.

TABLE XVI. CHARACTERISTICS OF RELAY MODEL

Principles	Characteristics
Reliability	Good atomicity and security
Scalability	Decided by BC
Regulation	Difficult to regulate
Transparency	Good
Efficiency	Decided by BC
Rollback	Decided by SC

Airswap protocol is a point-to-point model, the characteristics are shown in Table 17.

TABLE XVII. CHARACTERISTICS OF POINT-TO-POINT MODEL

Principles	Characteristics
Reliability	Use SC
Scalability	Decided by BC
Regulation	Decided by BC
Transparency	Decided by BC
Efficiency	Decided by BC
Rollback	No

Double BC model is aimed at addressing operational efficiency, regulation compliance, and scalability. The characteristics of this model are shown in Table 18.

TABLE XVIII. CHARACTERISTICS OF DOUBLE BC MODEL

Principles	Characteristics
Reliability	SC execution
Scalability	Scalable
Regulation	Yes
Transparency	Good
Efficiency	Decided by SC
Rollback	Yes

V. CONCLUSION

DDAE is an important application of BCs, and various factors affect the functionality and performance of these protocols, including inter-chain protocols, consensus protocols, ID management. This paper classifies inter-chain protocols into three types of models. Table 19 compares these three models.

TABLE XIX. CHARACTERISTICS OF CROSS-CHAIN MODEL

Principles	Process		
	Atomic Trading	Notary	Decided by consensus
Reliability	Use hash lock	Notary confirms transaction	Decided by consensus
Scalability	Decided by BC	Yes	Good
Regulation	Decided by BC	Yes	Partially regulated
Transparency	Decided by BC	Good	Good
Efficiency	Decided by BC	General	General
Rollback	Decided by BC	No	Yes

Transaction protocols can be classified into five models. Among them, the automatic pricing model comes from the DeFi projects. Relay model and point-to-point model mainly address transaction matching and settlement issues. The double-BCs architecture is aimed at supervision and scalability. Table 20 summarizes the characteristics of these five modes.

TABLE XX. CHARACTERISTICS OF EXCHANGE MODEL

Principles	Process		
	Atomic Trading	Automatic Pricing	Relay
Reliability	Use hash lock	Decided by SC	Decided by notary and SC
Scalability	Decided by BC	Decided by BC	Yes
Regulation	Decided by BC	Decided by BC	Decided by BC
Transparency	Decided by BC	Transparent pricing	Decided by BC
Efficiency	Decided by BC	Decided by BC	Decided by BC
Rollback	Yes	Yes	Decided by BC

Principles	Process	
	Peer-to-Peer	Double BC
Reliability	Decided by consensus	Decided by TBC and ABC
Scalability	Good	Great
Regulation	Decided by BC	Yes
Transparency	Decided by BC	Guaranteed data privacy
Efficiency	Decided by BC	Fast
Rollback	Decided by BC	Yes

ACKNOWLEDGMENT

This work is supported by Chinese Ministry of Science and Technology (Grant No. 2018YFB1402700). This work was also supported by Beijing Municipal Natural Science Foundation (GrantNo.9142012 and 9152009), and National Natural Science Foundation of China (Grant No.61472032, No.M1450009, No.71271013 and No.61462003). This is also supported by LaoShan government. This is also supported by Major Science and Technology Innovation Projects in Shandong Province (Grant No. 2018CXGC0703).

Authorized licensed use limited to: University of Prince Edward Island. Downloaded on June 08,2021 at 11:30:26 UTC from IEEE Xplore. Restrictions apply.

REFERENCES

[1] Bai, X., Tsai, W. T., & Jiang, X. (2019, April). "Blockchain Design-A PFMI Viewpoint". In 2019 IEEE International Conference on Service-Oriented System Engineering (SOSE) (pp. 146-14609). IEEE.

[2] Ramamoorthy, C. V., & Tsai, W. T. (1996). "Advances in software engineering". Computer, 29(10), 47-58.

[3] Wang, R., Tsai, W. T., He, J., Liu, C., & Deng, E. (2018, December). A Distributed Digital Asset-Trading Platform Based on Permissioned Blockchains. In International Conference on Smart Blockchain (pp. 55-65). Springer, Cham.

[4] Tsai, W. T., Blower, R., Zhu, Y., & Yu, L. (2016, March). A system view of financial blockchains. In 2016 IEEE Symposium on Service-Oriented System Engineering (SOSE) (pp. 450-457). IEEE.

[5] Gao, J., Manjula, K., Roopa, P., Sumalatha, E., Bai, X., Tsai, W. T., & Uehara, T. (2012, December). A cloud-based TaaS infrastructure with tools for SaaS validation, performance and scalability evaluation. In 4th IEEE International Conference on Cloud Computing Technology and Science Proceedings (pp. 464-471). IEEE

[6] Hearn, M. (2016). Corda: A distributed ledger. Corda Technical White Paper, 2016.

[7] Kwon, J., & Buchman, E. (2016). Cosmos: A network of distributed ledgers. URL https://cosmos. network/whitepaper.

[8] Bai, X., Wang, Y., Dai, G., Tsai, W. T., & Chen, Y. (2007, July). A framework for contract-based collaborative verification and validation of web services. In International Symposium on Component-Based Software Engineering (pp. 258-273). Springer, Berlin, Heidelberg.

[9] Deng, E. Core algorithm of blockchain Internet Model of Inter-Chain Transaction: CN 107301600A.

[10] DarterDEX - Atomic Swap Decentralized Exchange of Native Coins, https://github.com/KomodoPlatform/KomodoPlatform/wiki/barterDEX-Whitepaper-v2.

[11] Adams, H., & Robinson, D. (2020). Uniswap v2 Core. URl: https://uniswap. org/whitepaper. pdf.

[12] Hertzog, E., Benartzi, G., & Benartzi, G. (2017). Bancor protocol: continuous liquidity for cryptographic tokens through their smart contracts. White paper.

[13] Warren, W., & Bandeali, A. (2017). 0x: An open protocol for decentralized exchange on the Ethereum blockchain. URl: https://github. com/0xProject/whitepaper.

[14] Luu, Y. V. L. Kybernetwork: A trustless decentralized exchange and payment service. URl: https://home. kyber. network/assets/KyberNetworkWhitepaper. pdf.

[15] Swap: A Peer-to-Peer Protocol for Trading Ethereum Tokens, https://swap.tech/whitepaper/.

[16] Tsai, W. T., Zhao, Z., Zhang, C., Yu, L., & Deng, E. (2018, September). A Multi-Chain Model for CBDC. In 2018 5th International Conference on Dependable Systems and Their Applications (DSA) (pp. 25-34). IEEE.

[17] FIRA, Anti-Money Laundering, https://www.finra.org/rules-guidance/guidance/reports/2018-report-exam-findings/anti-money-laundering.

[18] KYC3, Your Guide to KYC and AML Compliance, https://www.kyc3.com/quick-guide-to-kyc-and-aml-compliance/#what-is-am.

[19] Tsai, W. T., Automatic Real-Time Supervision Reporting System based on Double-Chain Architecture Blockchain: CN, 2018.04.18.

[20] Y. Song, Y. Li, L. Jia and M. Qiu, "Retraining Strategy-Based Domain Adaption Network for Intelligent Fault Diagnosis, " IEEE Transactions on Industrial Informatics, vol. 16, no. 9, pp. 6163-6171, Sep. 2020.

[21] Y. Li, Y. Song, L. Jia, S. Gao, Q. Li and M. Qiu, "Intelligent Fault Diagnosis by Fusing Domain Adversarial Training and Maximum Mean Discrepancy via Ensemble Learning", IEEE Transactions on Industrial Informatics, to be published. DOI: 10.1109/TII.2020.3008010.

Authorized licensed use limited to: University of Prince Edward Island. Downloaded on June 08,2021 at 11:30:26 UTC from IEEE Xplore. Restrictions apply.

Unveiling Digital Asset Movement and Market Characteristics: A Study of Non-Fungible Token Dynamics Ownerships

2024 7th International Conference on Informatics and Computational Sciences (ICICoS) | 979-8-3503-7588-6/24/$31.00 ©2024 IEEE | DOI: 10.1109/ICICoS62600.2024.10636907

Naufal Rizky Radea
School of Economic and Business
Telkom University
Bandung, Indonesia
naufalrizkyradea@student.telkomuniversity.ac.id

Andry Alamsyah
School of Economic and Business
Telkom University
Bandung, Indonesia
andrya@telkomuniversity.ac.id

***Abstract*—This study aims to understand the characteristics of the digital asset market by analyzing the dynamics of Non-Fungible Token (NFT) movements and ownerships that shape market concentration and current market interest. We explored the distribution of NFT across various wallets to determine whether collections form randomly or tend to cluster. We examined how NFT interacts within wallets by utilizing network analysis of 2.8 million transactions from the top five NFT marketplaces between November 2017 and April 2021. Employing centrality metrics and modularity analysis uncovered the most influential NFT and the community structure, including the five prominent communities within the network. Our findings demonstrate how diversification and ownership patterns in the 'Games' and 'Art' categories provide fresh insights into preferences and behaviors of market characteristics, highlighting the dynamics of the NFT interconnectedness.**

Keywords—non-fungible token, digital asset, network analysis, market dynamics

I. Introduction

Non-Fungible Token (NFT) have become a significant part of the digital economy, facilitating the trade of diverse assets, including art, music, in-game items, and collectibles across digital platforms [1]. NFT is notable for their unique and valuable assets, enabling ownership, purchase, and exchange across various digital platforms [1]. The NFT ecosystem includes a linked network of various entities, such as artists, collectors, investors, developers, and technology service providers [2]. Understanding the Non-Fungible Token (NFT) ecosystem and its characteristics affords us new insights into the ownership of digital assets. This comprehensively analyses how such tokens are distributed across various owners.

The NFT ecosystem built upon blockchain technology, fostered transparent and secure ownership, thus simplifying the verification of the NFT ownership that can be traced and authenticated by owners [3]. Additionally, NFT marks a significant shift towards creator empowerment within the digital economy, underscored by Web 3.0's decentralized and trustless ledger technology [4]. This transition integrates seamlessly with the broader digital economy, enhancing control and transparency for creators in the evolving Web 3.0 landscape.

The rise of NFT has revolutionized the digital asset landscape, presenting novel opportunities for ownership and trading preference [5]. Motivated by the need for an in-depth understanding of market characteristics, this study addresses the existing gap by exploring the dynamics of NFT ownership movement. This exploration is crucial for investors, collectors, and buyers to grasp the market's complexities [6]. We focus on observing trends in community formations and networks analysis, shedding light on the intricate nature of the NFT ecosystem.

We collected data in selecting top five NFT marketplaces from November 2017 to April 2021, a period during public attention towards NFT significantly increased and trading volumes peaked [7][8]. Firstly, Cryptokitties a platform combining digital art and gaming elements, notably reached a trading volume of $8.1M in 2021, highlighting its impact on the digital collectibles. Secondly, OpenSea, the world's largest NFT marketplace is renowned for its wide array of digital assets including art, collectibles, and virtual real estates, achieving a trade volume of $234M in 2021. Thirdly, Decentraland has pioneered utilizing NFT for trading virtual land, underscoring the value of NFT in virtual real-estate. Next, Atomic known for its art and collectibles. Lastly, GodsUnchained has leveraged NFT within gaming industry. These marketplaces play crucial role in providing comprehensive data for analysing NFT ownership dynamics and market characteristics.

We conducted a network analysis of 2.8 million transactions from the top five NFT marketplaces to identify dynamic clustering and wallet patterns in ownership distribution. This methodological approach revealed that 'Games' and 'Art' sectors are current focal points of market interest, demonstrating distinct patterns of market concentration and diversification. Our findings underscore the complex structure of NFT ownership and its impact on the characteristics of the digital asset market, providing valuable insights into the evolving landscape of the digital economy.

II. Literature Review

A. Network Analysis of NFT Interconnectedness

Network analysis serves as an effective method for understanding the dynamics of intricate systems by examining the relationships between entities [9][10]. This approach allows us to observe and analyses how nodes, interconnected by lines known as edges, form various network structure [11]. By mapping these connections, researchers can identify significant patterns and trends in the interconnections among entities within the network. This methodology is crucial for exploring how these relationships influence the behaviour of the network as a whole.

Authorized licensed use limited to: Zhejiang University. Downloaded on July 09,2025 at 07:18:43 UTC from IEEE Xplore. Restrictions apply.

B. Centrality Measurement

The centrality metric measures the importance of nodes within a network [12]. We focused on three principal centrality metrics as follows.

Degree Centrality (DC) measures the most influenced nodes within the network [12]. DC is calculated by counting the number of edges connected to a node. A high DC score indicates the node has more than average connections within the network. The formula for DC can be expressed as (1).

$$C_D(v) = \frac{d(v)}{n-1} \quad (1)$$

Where C_D represents the degree centrality of node v, $d(v)$ denotes the degree of node v, defined as the number of connections, and n symbolizes the total count of nodes in the network. DC serves as a fundamental metric for evaluating the relative importance of nodes in the network structures [13].

Betweenness Centrality (BC) measures the degree to which nodes stand between each other based on the shortest path that acts as a bridge between networks [13]. Nodes with high BC can be considered to influence within the wallets. The formula for betweenness centrality can be expressed as (2).

$$C_b(v) = \sum_{s \neq v \neq t} \frac{\sigma(s,t|v)}{\sigma(s,t)} \quad (2)$$

Where $\sigma(s,t)$ is the total number of the shortest paths from node s to node t, $\sigma(s,t|v)$ is the number of paths that pass through node v, s, and t are distinct nodes from v. This formula quantifies the extent to which a node serves as a bridge along the shortest paths between pairs of nodes in the network.

Closeness Centrality (CC) measures the node's efficiency in accessing every other node within the network, thereby providing an estimate of the node's effectiveness and ease of connectivity [13]. The formula for the closeness centrality can be expressed as (3).

$$C_c(u) = \frac{1}{\sum_{v \neq u} d(u,v)} \quad (3)$$

In this context, $\sum_{(v \neq u)} d(u,v)$ represents the aggregate sum of the shortest paths from node u to all other nodes v within the network, where $d(u,v)$ denotes the minimum distance between node u and node v. CC helps identify well-connected nodes that can efficiently spread information within the network.

C. Network Modularity

Network Modularity quantifies the extent to which a network is portioned into distinct communities [14][15]. Network Modularity within the NFT ecosystem emphasizes the need to comprehend the interconnections among NFT ownership network components and their contribution to cohesive community formation [9]. The expression for Network Modularity measurements can be articulated as (4).

$$Q = \frac{1}{2m} \sum_{i,j} \left[A_{ij} - \frac{k_i k_j}{2m} \right] \delta(c_i, c_j) \quad (4)$$

In the given context, Q represents the modularity score that measures the formed community's strength within a network. The score ranges between -1 and 1, to evaluate the extent of network segmentation into distinct modules or communities. Subsequently, m denotes the number of edges in networks. A_{ij} refers to the adjacency matrix's entry, marking the link between nodes I and j; k_i specifies the degree of nodes I, c_i and c_j indicating the community classifications for nodes i and j. $\delta(c_i, c_j)$ is employed to determine community membership, assigning a value of 1 when nodes i and j are in the same community and 0 otherwise [15][16].

The Louvain Algorithm, known for its efficiency and scalability, is a widely used method for community detection in large networks [17]. It optimizes the modularity score to identify distinct clusters, significantly enhancing network modularity analysis. This algorithm plays a central role in our study, allowing us to detect and analyze community structures essential for understanding interaction patterns within NFT networks [18].

III. METHODOLOGY

We designed a methodology to analyze the NFT ownership network, as depicted in Fig. 1 and detailed in Table 1. This approach allowed us to achieve a comprehensive understanding of the dynamic's interconnectedness among various NFT tokens.

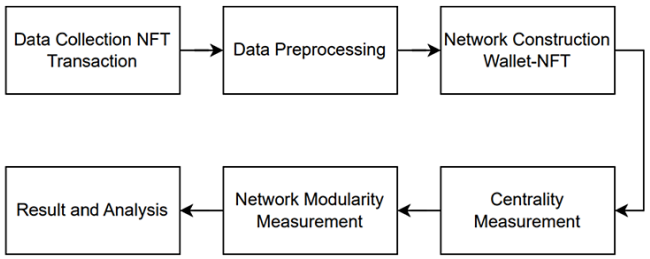


Fig. 1. Research methodology workflow

TABLE I. RESEARCH METHODOLOGY EXPLANATION

Phase	Explanation
Data Collection NFT Transaction	Our initial transaction dataset from the 5 biggest NFT markets, including Cryptokitties, OpenSea, Atomic, Decentraland, and Godsunchained about 6,071,027 raw data obtained via web scrapping.
Data Preprocessing	The dataset was reduced to a more manageable 2,877,452 records, particularly in the Buyer_address column, to enhance the quality of the data point.
Network Construction Wallet-NFT	We generated pairs from 'ID_token' entries. Each token pair was transformed into 'Source' and 'Target' to create a network.
Centrality Measurement	The construction of an undirected network interaction is done by embarking on three specific metrics: degree centrality (DC), betweenness centrality (BC), and closeness centrality (CC).
Network Modularity Measurement	We employed modularity to identify distinct communities within the network, examining the relational patterns of NFT interconnectedness
Result and Analysis	We have identified five prominent communities and the most influential node.

A. Data Collection NFT Transaction

We collected an NFT transaction dataset of 6,071,027 raw data entries from five prominent NFT marketplaces through web scrapping. Focusing on key variables, 'ID_token' and 'Buyer_address'—marked with an asterisk (*) in our dataset—enables precise tracking of NFT distribution and market dynamics. A detailed explanation, including the reasons for selecting these two variables, is provided below, along with a description of the data sources for these NFT transactions, as detailed in Table II.

Authorized licensed use limited to: Zhejiang University. Downloaded on July 09,2025 at 07:18:43 UTC from IEEE Xplore. Restrictions apply.

- ID_token allows us to trace each NFT's journey and accumulation across wallets, essential for understanding market dynamics.
- Buyer_address helps identify preferred NFT and their clustering among specific buyers.

TABLE II. RESEARCH METHODOLOGY EXPLANATION

Data Variable	Explanation
Smart_contract	The specific smart contract associated with NFT
ID_token*	The unique identifier of NFT within its respective smart contract
Transaction_hash	A transaction hash related to the sale of an NFT
Seller_address Seller_username	Seller address along with their marketplace username
Buyer_address* Buyer_username	Buyer address along with their marketplace username
Image_url_1, Image_url_2, Image_url_3, Image_url_4	Web link to the digital item linked to the NFT
Price_crypto Price_USD	Conversion to USD based on daily exchanges rates
Name	The title associated with the NFT offering
Description	A narrative description accompanying the NFT
Collection	The specific collection that includes the NFT
Datatime_update Datatime_update_seconds	Records the transaction time, precise to the day or second
Permanent_link	A verification link for the NFT's authenticity
Unique_id_collection	Marks the transaction time with the day or second precision
Collection_cleaned	Corrects the frequent typographical errors in the 'Collection' field and standardized the names
Category	The category of the NFT, such as Art, Games, Collectibles, etc

B. Data Preprocessing

We manage and handle missing values, particularly in 'Buyer_address' column. This stage includes eliminating duplicate entries and selectively filtering data to align with our analytical framework [19][6]. We ensure each address matches the standard Ethereum format (beginning with '0x'). We streamlined this to a more manageable 2,877,452 records. Next, we focused on ensuring the reliability and integrity of the 'ID_token' and 'Buyer_address' columns as a key variable to identify how tokens were spread across different wallet addresses. The data preprocessing phase can be shown in Table III.

C. Network Construction Wallet-NFT

We construct a network with NFT as nodes and the shared ownership among NFT as edges, forming an undirected graph that represents their mutual ownership [7][9]. We employ the two-tuple combination algorithm to facilitate these connections, highlighting the most significant connections of tokens by matching 'ID_token' with 'Buyer_address' from our preprocessed data, as shown in Table III. Each pair of tokens is categorized as 'Source' and 'Target', detailed in Table IV. The network features 16,419 nodes and 1,122,940 edges, illustrating the complex web of NFT ownership and reflecting the dense landscape of token interrelations [9] [20].

TABLE III. KEY VARIABLES PREPARATION

ID_token	Buyer_address
81144621...	0xbedc90d9d595e8fd88afdbefe79108dc4e1ef956
35675	0xf52393e120f918ffba50410b90a29b1f8250c879
11237526	0x77560848b891190965c3d295e09528587947424d
118932623	0x8b51c1ba09ee33e7649cac62ccb6d0f410f5647a
1727405	0xddc9c668f90bba52c4bdef773dae5224813393e1
…	…
51566644...	0x4d57f82ce4ca2514d19adc3f99046a83df13fe20
10778793...	0xf80b31b311c3faf500eefebfb9687785d87c7446
41844935...	0x182d7f604b37a5c5d7caeaa4bf94219e1d4cc0fc
40023	0xe7f6f446469352f9212e78dce4452c77448bc74c
1694534	0x5f4bcf5179aa248765af12bb37228b1be36b5f28

TABLE IV. SOURCE AND TARGET FOR NETWORK CONSTRUCTION

Source	Target
6834	6130
6834	5912
6834	1697
6834	2830
6834	3553
…	…
1576427	1711736
1576427	1808180
1711736	1808180
3096042	906338
8617988	3680301

D. Centrality Measurement

We employ centrality measurement with these specific metrics employed, as follows: degree centrality (DC) to identify well-connected NFT; betweenness centrality (BC) to asses NFT's influence in networks; and closeness centrality (CC) to evaluate the flow efficiency within the network. These metrics help understand the NFT interconnectedness, highlighting the most influential and accessible NFT within single wallets [21].

E. Network Modularity Measurement.

We quantitatively assessed network modularity to evaluate the structure of the NFT ownership network. Utilizing the Louvain algorithm, we identified specific clusters to understand how the NFT ownership network is formed. We then constructed a network graph to visually identify these clusters, highlighted by unique color schemes. This modular network analysis reveals significant insights into the ecosystem's nature and market dynamics of the digital asset movement.

Authorized licensed use limited to: Zhejiang University. Downloaded on July 09,2025 at 07:18:43 UTC from IEEE Xplore. Restrictions apply.

TABLE V. CENTRALITY METRICS RANK

Centrality Metrics								
Degree Centrality			***Betweenness Centrality***			***Closeness Centrality***		
Id Token	Score	Token Identification	Id Token	Score	Token Identification	Id Token	Score	Token Identification
39463074.0	0.1080	Games	39463074.0	0.00275438	Games	39463074.0	0.11024136	Games
3939463.0	0.1080	Games	3939463.0	0.00275438	Games	3939463.0	0.11024136	Games
15461284.0	0.0863	Games	3.0	0.00199094	Art	15461284.0	0.09552359	Games
8987200.0	0.0835	Games	15461284.0	0.00147190	Games	8987200.0	0.09390045	Games
14879623.0	0.0833	Games	2.0	0.00088838	Art	14879623.0	0.09383260	Games
40313706.0	0.0833	Games	8987200.0	0.00080578	Games	40313706.0	0.09383260	Games
42315143.0	0.0816	Games	19.0	0.00073375	Art	42315143.0	0.09285972	Games
126791354.0	0.0814	Games	7342079983.0	0.00070133	Games	126791354.0	0.09276023	Games
45894328.0	0.0811	Games	9.0	0.00055863	Art	45894328.0	0.09256188	Games
22908362.0	0.0804	Games	20.0	0.00052571	Art	22908362.0	0.09243012	Games

IV. RESULT AND ANALYSIS

A. Centrality Measurement

Our study applies centrality metrics to analyze relational dynamics in the NFT ownership network, highlighting the significant nodes and their roles within individual wallets. Upcoming tables will summarize key aspects of the network analysis. Table V shows the top 10 nodes with the highest DC, showing influential NFT tokens; the top 10 nodes with the highest BC, marking NFT tokens as network connectors, and the top 10 highest CC, showcasing NFT tokens that optimize information or value flow.

The token identification column introduces a new perspective to our network analysis by associating specific nodes (Id Token) with defined entities like games, art, and other categories in the NFT marketplace. As shown in Table V, the node identified by id token “39463074.0” ranks as the most popular in the NFT ownership network, encompassing attributes such as having the most connected (DC), controlling the flow of information (BC), and the most efficient in reaching other nodes (CC). These metrics enable us to derive critical insights into the prominence and dynamics of NFT within an individual wallet.

The most popular node, identified by id token “39463074.0”, is known as “Pandora the Curious” from the blockchain-based game GodsUnchained. This card is a part of the game’s vast collection of unique cards that players can collect, buy, sell, or use in the gameplay. A more in-depth examination of its GodsUnchained profile reveals its rarity, labeled as “Legendary”—a tier that typically represents the highest rarity in collectible card games. Legendary cards are often produced in limited quantities and have unique abilities, making them highly sought after by players and collectors. By exploring the behaviors and preferences of this prominent node, deepens our understanding of significant nodes and their roles within the dynamic structure of the network.

B. Identify the Headings

Table VI displays the network modularity scores and the number of formed communities. This insight is key to understanding the complex web relationships and dynamics present in these networks, with a particular focus on identifying and examining the structure of communities to unveil the patterns of interactions and growth among NFT.

TABLE VI. NETWORK MODULARITY RESULT

Metrics	Values
Modularity	0.392
Number of Communities	2686

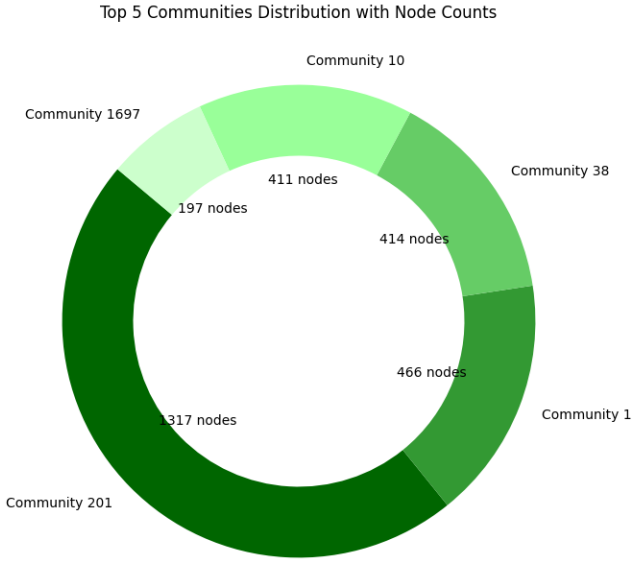


Fig. 2. NFT distribution with node counts

The initial modularity analysis presented in Table VI identified 2,686 communities within the network, featuring an overall modularity score of 0.392. This significant finding led to a focused analysis of the five largest clusters, as shown in Fig. 2, which illustrates the distribution of nodes across these communities. The analysis revealed the following distribution: Community 201 comprises 1317 nodes, Community 1 includes 466 nodes, Community 28 contains 414 nodes, Community 10 encompasses 411 nodes, and

Authorized licensed use limited to: Zhejiang University. Downloaded on July 09,2025 at 07:18:43 UTC from IEEE Xplore. Restrictions apply.

Community 1697 contains 197 nodes. This distribution showcases a network of highly interconnected groups, suggesting potential shared interests and characteristics among the nodes.

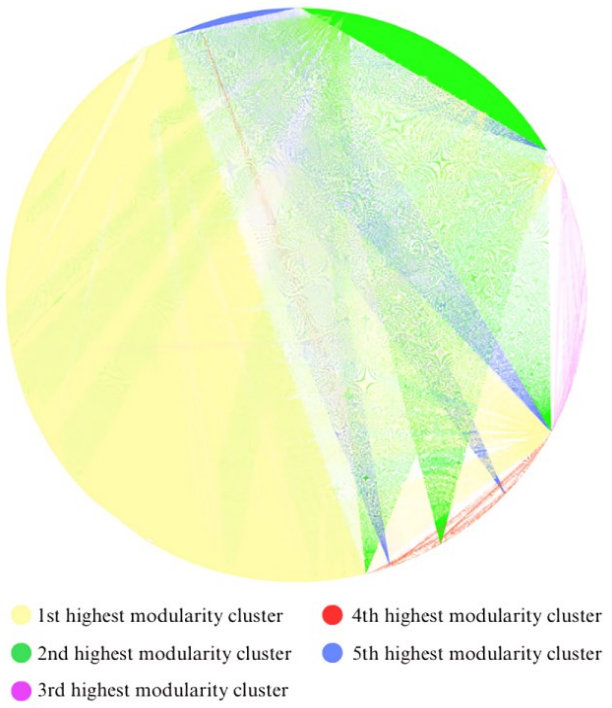


Fig. 3. NFT ownership network

TABLE VII. NFT OWNERSHIP NETWORK TOPOLOGY

Modularity Rank	Community	Percentages
1st	Community 201	46.92%
2nd	Community 1	16.6%
3rd	Community 38	15.43%
4th	Community 10	14.04%
5th	Community 1697	7.02%

We utilize the NFT Network Ownership, illustrated in Fig.3, to depict the modularity clustering formation of the network. This visual representation highlights the complex interconnections within the network. We have identified and color-coded the five largest communities to facilitate a clearer understanding of the network's topology. The detailed breakdown of these communities as depicted in Table VII:

- Community 201, represents the largest cluster, encompasses 46.92% of the network, indicated by the yellow color. This community's prominence is evident from the extensive interconnections among its nodes, illustrating a dense cluster of activity as detailed in Fig.2.
- Community 1, the second largest group, accounts for 16.6% of the network, marked by green. This community shows a significant concentration of nodes that are likely to engage in frequent transactions or shared ownership.
- Community 38, colored purple, includes 15.43% of nodes, denoting another active segment of the network.
- Community 10, shown in red, makes up 14.04% of the network, highlighting its substantial but slightly lesser engagement compared to the leading groups.
- Community 1697, depicted in blue, comprises 7.02% of the network, indicating its specialized but essential role within the network.

TABLE VIII. TOP 5 TOKEN DISTRIBUTION WITHIN COMMUNITIES BASED DEGREE CENTRALITY

Community	Id Token	Degree Centrality	Token Identification
Community 201	15461284.0	0.0863	Games
	8987200.0	0.0835	Games
	14879623.0	0.0833	Games
	40313706.0	0.0833	Games
	42315143.0	0.0816	Games
Community 1	39463074.0	0.1080	Games
	3939463.0	0.1080	Games
	14540592.0	0.0283	Games
	120011042.0	0.0278	Games
	60320792.0	0.0278	Games
Community 38	2.0	0.0035	Art
	1.0	0.0032	Art
	11.0	0.0021	Art
	561821.0	0.0020	Art
	607831.0	0.0020	Art
Community 10	3.0	0.0043	Art
	7.0	0.0041	Art
	10.0	0.0028	Art
	55.0	0.0028	Art
	68.0	0.0027	Art
Community 1697	24653457.0	0.0119	Games
	32463477.0	0.0119	Games
	40871490.0	0.0119	Games
	8995976.0	0.0119	Games
	121896283.0	0.0119	Games

Further analysis focuses on the distribution of tokens in the five biggest communities within the network, as shown in Table VIII, which aggregate into communities based on shared characteristics, categories, and, specifically, ownership. Tokens with the highest DC signify the most interconnected and influential within their communities, especially in the Games and Art sectors. A collection of high DC tokens, such as those in Community 201, serve as network hubs, suggesting their pivotal role in trade volume and significant ownership. The dominance of Community 201, representing nearly half of the network, may point to a concentrated interest in a particular subset of the NFT ecosystem.

Authorized licensed use limited to: Zhejiang University. Downloaded on July 09,2025 at 07:18:43 UTC from IEEE Xplore. Restrictions apply.

The presence of other significant communities, such as Community 1, Community 38, Community 10, and Community 1697, which are notable for their sizes, suggests diversity in the types of NFT being collected and traded. The variety in token categories across these communities, from Games to Art, underscores the multi-dimensional nature of NFT ownership and the different values that collectors and investors might find in various types of NFT. For instance, the predominance of the Games category indicates a current market interest or the utility of these tokens within popular blockchain games. It also raises questions about the role of Art tokens within the network, considering their lower DC scores might reflect a more niche and specialized market segment.

V. Conclusion

Our study leverages network analysis metrics, including centrality measurements such as degree centrality (DC), betweenness centrality (BC), and closeness centrality (CC). These metrics allowed us to understand the relational dynamics, emphasizing the importance of key nodes and their roles in the NFT ecosystem. Network modularity provides valuable insights into how communities form and the interactions among tokens within the NFT ownership network. We identified significant patterns by exploring the modularity score and community structure, especially within the five primary clusters. Our findings show that tokens tend to form clusters, following Barabasi's "the-rich-get-richer" concept, where nodes tend to connect and attach to well-known nodes. These insights enhance our understanding of the complex dynamics of the NFT ownership network and their implication for digital asset ownership.

The study is limited by its focus only on specific NFT genres, overlooking the broader category of digital assets beyond NFT. Further, the analysis could benefit from employing various data analytics models, including machine learning techniques. These constraints may impact our findings, suggesting the potential for future studies to build upon this work by addressing these gaps. Future research may explore a broader scope, such as identifying investment groups based on price ranges and understanding additional factors like market sentiment, real-world events, or technical aspects of NFT (e.g., rarity or uniqueness). These findings are expected to significantly contribute to the academic discourse on blockchain technology and the evolving digital economy landscape.

References

[1] A. Lennart, "The non-fungible token (NFT) market and its relationship with Bitcoin and Ethereum," pp. 1–9, Jun. 2021.

[2] The Open Platform, "NFT Landscape," Medium. Accessed: Jan. 15, 2024. [Online]. Available: https://topco.medium.com/ton-nft-landscape-f60d4b60bcde

[3] R. Conti, "What is an NFT? Non-Fungible Tokens explained," Forbes Advisor. Accessed: Jan. 15, 2024. [Online]. Available: https://www.forbes.com/advisor/investing/cryptocurrency/nft-non-fungible-token/

[4] T. Conlon and S. Corbet, "The problem with NFTs," SSRN Elect. Journ., Nov. 2023.

[5] M. Golomb, "Rise of a new disruptor: how NFTs are revolutionizing the art and entertainment worlds," Forbes Business Council. Accessed: Jan. 15, 2024. [Online]. Available: https://www.forbes.com/sites/forbesbusinesscouncil/2021/09/07/rise-of-a-new-disruptor-how-nfts-are-revolutionizing-the-art-and-entertainment-worlds/?sh=9fb17001a90f

[6] A. Alamsyah, M. K. Bratawisnu, and P. H. Sanjani, "Finding pattern in dynamic network analysis," Insti. of Elect. and Electr. Eng. Inc., Nov. 2018, pp. 141–146.

[7] M. Nadini, L. Alessandretti, F. Di Giacinto, M. Martino, L. M. Aiello, and A. Baronchelli, "Mapping the NFT revolution: market trends, trade networks, and visual features," Sci. Rep. Vol. 11, no. 1, Dec. 2021.

[8] A. Mekacher et al., "Heterogeneous rarity patterns drive price dynamics in NFT collections," Sci. Rep. Vol. 12, no. 1, Dec. 2022.

[9] A. Alamsyah and F. Adityawarman, "Hybrid sentiment and network analysis of social opinion polarization," Inst. of Elec. and Elec. Eng. Inc., Oct. 2017.

[10] A. Alamsyah, D. P. Ramadhani, and F. T. Kristanti, "Event-based dynamic banking network exploration for economic anomaly detection," Jour. of Theo. and Appl. Infor. Tech. Vol. 98, no. 7, pp. 1089–1100, 2020.

[11] B. Laszlo, "Network Science," in Network Science, Cambridge University Press, 2016, p. 783.

[12] Z. Wan, Y. Mahajan, B. W. Kang, T. J. Moore, and J. H. Cho, "A survey on centrality metrics and their network resilience analysis," IEEE Access. Vol. 9, pp. 104773–104819, 2021.

[13] J. Zhang and Y. Luo, "Degree centrality, betweenness centrality, and closeness centrality in social network," Atlantic Press, 2017, pp. 300–303.

[14] S. Alizadeh, A. Setayesh, A. Mohamadpour, and B. Bahrak, "A network analysis of the non-fungible token (NFT) market: structural characteristics, evolution, and interactions," Appl. Netw. Sci. Vol. 8, no. 1, Dec. 2023.

[15] A. P. Rabbani, A. Alamsyah, and S. Widiyanesti, "An effort to measure customer relationship performance in indonesia's fintech industry," Bandung, Feb. 2021.

[16] J. Kim and K. H. Cho, "Robustness analysis of network modularity," IEEE Trans. Control Netw. Syst. Vol. 3, no. 4, pp. 348–357, Dec. 2016.

[17] M. E. J. Newman, "Modularity and community structure in networks," PNAS. Vol. 103, no. 23, pp. 8577–8582, 2006.

[18] B. Yao, J. Zhu, P. Ma, K. Gao, and X. Ren, "A constrained louvain algorithm with a novel modularity," Appl. Sci. (Switzerland). Vol. 13, no. 6, Mar. 2023.

[19] A. A. Ali, A. Alamsyah, and M. Ariyanti, "The dynamics of non-fungible token marketplaces: a network analysis of opensea transactions," IEEE, pp. 83–88, Nov. 2023.

[20] A. Alamsyah and I. F. Muhammad, "Unraveling the crypto market: A journey into decentralized finance transaction network," Dig. Busin. Vol. 4, no. 1, pp. 100074, Jun. 2024.

[21] A. Alamsyah, D. P. Ramadhani, and L. S. Mulyani, "Rise or fall? discovering the global world trade network rise and fall under major situations," Jour. of Open Inno.: Tech. Mark. and Comp. Vol. 9, no. 1, Mar. 2023.

Authorized licensed use limited to: Zhejiang University. Downloaded on July 09,2025 at 07:18:43 UTC from IEEE Xplore. Restrictions apply.

Blockchain-based secure digital asset exchange scheme with QoS-aware incentive mechanism

Jiawei Zheng, Xuewen Dong, Wei Tong, Qihang Liu, Xinghui Zhu

***Abstract*—As the Internet of things (IoT) is increasingly popular, the number of IoT devices such as sensors and smart equipments are growing at an astonishing rate and data generated by these devices is exploding. However, these massive IoT data, stored in the form of isolated data centers, can not be shared by others who also need it. Moreover, data exchange is now needing a secure and fair mechanism to guarantee the data provider's rights and data security. Data providers also lack the motivation to share their data, as no effective mechanism exists to reward this behavior. To solve these problems, we propose a digital asset exchange mechanism based on blockchain technology, in which we record the behavior of data publishing and exchanging into the blockchain, which can ensure the reliability and transparency of data exchange without the restriction of trusted third-party payment institutions. Especially, to inspire the data providers to share their high-quality data, we design an incentive mechanism based on QoS, which gives higher rewards to those who provide high-quality data. Experimental results of this prototype demonstrate that this mechanism is appropriate to be applied in practice.**

***Index Terms*—digital asset exchange, Internet of Things, incentive mechanism, blockchain, smart contract**

I. Introduction

Nowadays, the Internet of Things is expanding at a fast speed, in which many IoT applications equipped with smart devices are helping people do basic things at home and changing people's lifestyle [1] [2]. Some reports predict that the amount of IoT devices will reach 26 billion by 2020, which is related to the popularity of a variety of IoT applications that include smart home, city, factory, transportation, etc. At the same time, these applications will generate a large amount of IoT data.

To realize such a huge vision of the IoT, it is worth considering data transmission and sharing [3]. Currently, most cloud-based IoT solutions depend on centralized paradigm, which means massive IoT data will be transferred through the Internet to centralized cloud servers. This approach may proper functioning nowadays, but the expected growth and expansion indicate that a new architecture is needed to address the upcoming problems. In addition, more and more IoT applications form a phenomenon of information isolation [4]. The IoT applications, as well as their equipped devices, such as various sensors, do not share data and interact with other smart terminals or applications. A new architecture needs to be proposed to guarantee high-level data sharing and mitigate the pressure of centralized servers, which will realize the exchange of data with value. However, data exchange currently lacks a secure enforcement mechanism to protect the rights of data providers [5]. Traditionally, the data marketplace acts as a trusted third party to ensure the interests of both parties in the process of data exchange. But in this data exchange procedure, there is often a single point of failure, default and other serious problems. At the same time, data providers must be provided with sufficient motivation to contribute their data. Some of the relevant proposals had put forward some incentive mechanisms to promote data quality and data providers' enthusiasm. However, before the arrival of blockchain technology, there is still something missing in terms of security and availability of the mechanism [6].

Blockchain techniques can prove to realize the goal of coordinating, tracking, executing transactions and storing information in a distributed way [7], enabling the connection between numerous devices in the form of P2P. It also makes it possible to develop applications without centralized cloud servers. Blockchain technology has developed at an amazing speed in the past two years. Blockchain technologies have been used as a distributed ledger in Bitcoin for the first time, where cryptocurrency transfer occurs, and the transaction information is recorded persistently by each blockchain node. The distributed ledger is stored in the form of head to tail connections of each block. Regarding smart contracts, their function is to act as a contract between two or more parties, which can be triggered automatically when certain conditions are met and execute specific operations. Smart contracts can be applied in many life and business scenarios to improve the present service process, such as logistics management, international transaction [8], mortgaging and supply chain finance. A couple of years after the emergence of Bitcoin, based on blockchain, there are over thousands of cryptocurrencies emerging successively. The use of cryptocurrencies like Bitcoin [9] is said to revolutionize the payment ways due to their advantages in contrast to traditional currencies. There is no need for middlemen like banks and third-party payment institutions. Multinational entrepreneurs can manage their finance more easily no longer demand complex procedure. Instead, they can receive funds immediately, through peer to peer network with consensus mechanism. As in the IoT scenarios, there

This work is supported by National key R&D Program of China (No.2017YFB1400700), Shaanxi Science & Technology Coordination & Innovation Project(No.2016KTZDGY05-07-01), National Natural Science Foundation of China(No.U1736216, U1536202, 61571352)

Jiawei Zheng, Xuewen Dong, Qihang Liu and Xinghui Zhu are with the School of Computer Science & Technology, Xidian University, Xi'an, and are with the Shaanxi Key Laboratory of Network and System Security, Xi'an, China (e-mail: jwzheng@stu.xidian.edu.cn, xwdong@xidian.edu.cn, liuqihang1016@foxmail.com, xinghui_zhu@163.com)

Wei Tong is with the School of Cyber Engineering, Xidian University, Xi'an, China (e-mail: wtong@stu.xidian.edu.cn.)

978-1-7281-1686-0/19/$31.00 ©2019 IEEE

are also many distrustful entities that need to process and execute transactions with each other, such as sensor nodes, network gateways, data centers, and end-users, we can apply the characteristics of blockchain into the digital asset exchange in IoT environment.

At the same time, a flexible incentive mechanism is needed to promote shared data quality and participatory initiative [10]. In order to guarantee high real-time requirements and user experience in IoT, we need to evaluate the data quality. Data providers must be encouraged to publish more economical service with high quality at low cost. In traditional payment systems, the data providers need to get services from the third payment parties to complete the asset exchange with value. In this case, both data consumers and providers need to establish trust and agreement between these middlemen and the other party. Most of these agreements are complex and difficult to establish for taking lots of time. These middlemen are also based on centralized architecture, and the transaction information and logs are not transparent to both sides. Considering these problems, we can use the blockchain to upgrade the existing system and realize secure data exchange, accompanying value exchange.

Based on the discussion above, this paper proposes a digital asset exchange mechanism based on blockchain, which realizes a fair and secure data exchange between each peer. And an incentive mechanism is proposed for motivating improve high-quality services. The main contributions are as follows:

1) To solve the data exchange problem mentioned above, we propose a digital asset exchange mechanism through smart contracts based on a distributed IoT architecture, realizing secure and tamper-proof transactions.
2) A QoS-based incentive mechanism is presented to motivate data providers to publish more cost-efficient data service.
3) The prototype is implemented in practice, and the experimental results are given.

The rest of the paper is structured as follows. We introduce the necessity of using the blockchain technology in distributed IoT environments in section II. Next, we describe the system model on which the data exchange based in section III. In section IV, we give the detail of the data exchange scheme. Our prototype is implemented to verify the incentive mechanism's availability in section V. Section VI gives the conclusion of this paper.

II. PRELIMINARIES

Before delving into how to use blockchain in IoT environments, it must be considered that the necessity of using blockchain. Specifically, to determine whether the use of blockchain is appropriate, we consider the following features:

- **Decentralization**: By using blockchain technology, IoT applications do not need to trust blindly a third-party agency or banks. IoT applications can be decentralized when there is no trusted centralized system.
- **P2P exchanges**: Most communications are established between peer to peer in IoT environments. Especially, in edge or fog computing paradigms, communications among peers are very common.
- **Payment system**: There are probably some financial transactions between entities through third-party payment institutions in some IoT applications. The payment processes can be built through cryptocurrencies by using the blockchain, enabling the transactions to be retrospect.
- **Common sequential transaction logs**: Blockchain can provide the common sequential transaction logs among all of the peers. It shares common missions that IoT data need to be stored time-serially. And the transaction can be provided sequentially.
- **Trusted distributed system**: The use of blockchain take trust into consideration in distributed systems. It can attain a trusted and automated distributed system through the consensus mechanism.
- **Persistent transaction record**: Transactions between IoT devices in IoT scenarios need to be recorded sequentially, aiming for tracing, audit supervising or because data mining and machine learning technology will be used in the future.

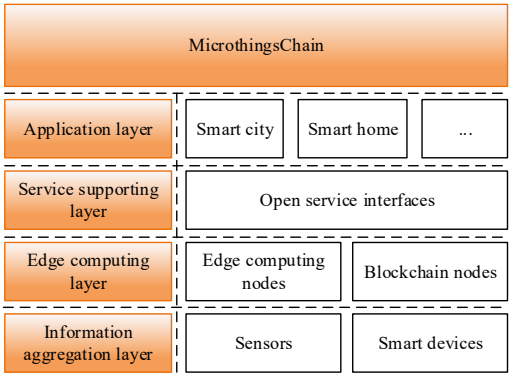


Fig. 1. Overview of the architecture

III. SYSTEM DESCRIPTION

After considering the necessity of introducing the blockchain technology into distributed IoT system, we have designed a distributed IoT architecture [7] based on blockchain and edge computing technology. By using a mixture of clouds and edge clouds to deploy edge computing nodes efficiently and securely, IoT devices can efficiently collect environmental data and response timely according to the scenarios. In this section, we give a brief description of this architecture and analyze the superiority of digital asset exchange based on this architecture.

We designed this architecture that includes **information aggregation layer** in the bottom for collecting information, **edge computing layer** at the top of the information layer for computing, analyzing and making decisions, **service supporting layer** and **application layer** at the top of the architecture to end-users. The function of each layer is given as follows:

- **Information aggregation layer**: There are a large number of heterogeneous devices, such as sensors, smart

equipments or any other things for collecting information in this layer. These devices are responsible for detecting and monitoring the real-time status of the environment and send it to the edge computing layer.

- **Edge computing layer**: Edge computing layer plays a vital role in storing, processing and analyzing data in the edge network, by which it can relieve the stress of core network and cloud servers.
- **Service supporting layer**: Edge computing layer opens some operation interfaces, which are used to access the data corresponding to the environments and control the underlying deployed devices by services.
- **Application layer**: Application developers can create IoT applications according to their demands through the service support layer's open service interfaces. These interfaces are given in the form of container and Kernel Virtual Machine (KVM), reaching the purpose of modularization and plug and play.

Based on blockchain technology, we present one cryptocurrency in this system in order to facilitate the exchange of digital asset between edge computing nodes, which form a P2P network and reach a common system state applying consensus mechanisms. Data sharing behaviors between edge computing nodes are summarized as transactions, in which the detail data of the sharing information and data exchange is included. These transactions are packed into a block, which is then added to the blockchain. These operations of data exchange are persistently recorded in the blockchain, enabling users and regulators to trace and audit them. On this basis, we propose a data exchange mechanism based on smart contracts to bridge the data sharing gap between data providers and data consumers. This mechanism is agreed and maintained upon by all the edge computing nodes, so data consumers can access data from other edge computing nodes without considering the trust problems. In other words, this mechanism supports access data cross-domain regardless of trust between each domain.

IV. DIGITAL ASSET EXCHANGE

After the data providers set prices of their service according to the rules, data consumers need to purchase services based on their demands. Data consumers then send the cryptocurrency to the data exchange contract to subscribe to the corresponding services. If the transaction is created, the data exchange contract packages the license according to the data consumers' will and send it to them, by which the data consumers can access the data they have subscribed to.

A. The Structure of License

The license contains the following elements: index, authorized account, rights, usage rules, etc. When the license is defined, it consists of a service index unit and one or more subsequent base units, which are given as follows:

1) License index: The version and ID information is contained in the license index, and the ID field is the unique key for the license.
2) Authorized account: Authorized account is the owner of this license. This means that the account has access to specific data.
3) Service ID: The ID of the service authorized by this license, by which the smart contract finds out the corresponded service to verify.
4) Rights: Rights means the different permission for using the data, such as requesting, downloading, etc.
5) Usage rules: Usage rules refer to the way that data consumers use the data, for instance, according to time, number of times or authorization period.
6) Access token: The token defines the information all above units based on a hash function, which is used for verifying the correctness and integrity of the license.

B. Data Exchange Scheme

By introducing the publish-subscribe model into our system, we design an elastic data exchange scheme, which is maintained by each edge computing node. As the P2P network is composed of edge computing nodes through a consensus mechanism, users access data cross-domain don't need to consider the trust problem. This mechanism makes data exchange more secure and breaks the data isolation. Additionally, we apply cryptocurrency into our ecology to improve the quality of shared data.

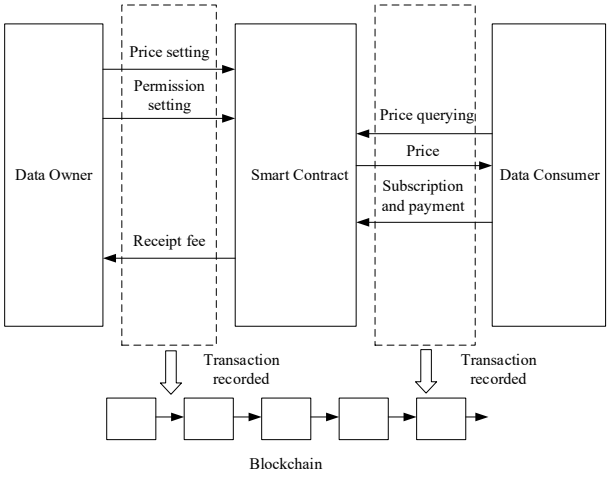


Fig. 2. Digital assets exchange transaction

Data providers publish their data and price their data. The pricing mechanism is implemented by cryptocurrency in the blockchain. The data providers can set prices for their data based on different permissions, for example, the cost of requesting data, the cost of downloading and storing data, etc. Additionally, data providers can also set different prices according to different usage rules, so data consumers can choose to use data based on time, the number of times and authorization period. By supplying multiple choices for data consumers, they can flexibly make their plan based on their demands. There is a complementary relationship between permission prices and usage rule prices. The price

setting and permission setting operations of data providers and subscription and payment operations of data consumers will all be included in transactions into block added in the blockchain. The digital assets exchange transaction is shown in Fig.2.

The process of digital asset exchange is shown in Fig.3. The data is shared in the form of various services.

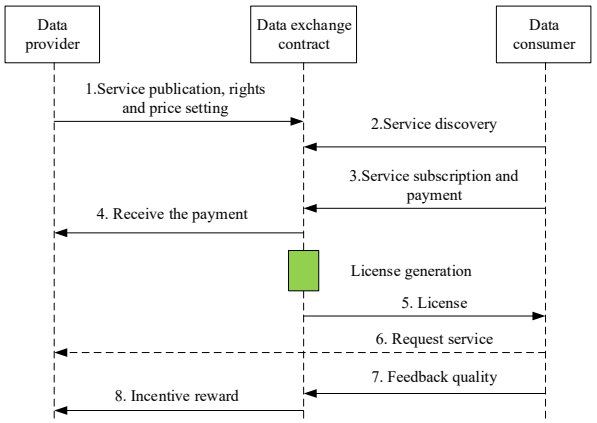


Fig. 3. The process of digital asset exchange

1. The data provider publishes the service through the data exchange contract, setting prices for different permissions and usage rules.

2. The data consumers interact with data exchange contract to explore the optional service provided. According to the demand, they can select the rights and usage rules that they would like to pay.

3. Then, the data consumers need to subscribe to the service they needed and pay for the corresponding fee to the data exchange contract.

4. The data exchange contract justifies the pre-setting trading model and transfers the service fee to the data provider. Next, the data exchange contract generates the license for data consumers to enable them to access the subscribed service.

5. The data exchange contract generates a license based on the permissions and usage rules selected by the data consumer and then send it to the data consumer.

6. After the data consumer received the access token, the consumer uses the access token to request the service they subscribed.

7. The data consumer feeds the quality of the service back to the data exchange contract. The feedback contains whether the service was successfully requested and some extra criteria for evaluating the service. This information contributes to the incentive mechanism.

8. Finally, the data exchange contract give reward to the data provider by the quality of service, which will be present in the incentive mechanism in sectionIV-C.

C. QoS-based Incentive Mechanism for Service Provider

We design an incentive mechanism for the service provider, giving rewards to those who provide high-quality services. To distinguish various services in the service delivery process, we consider both their functional and their non-functional characteristics. In order to reach this goal, we adopt a model of Web service quality based on a series of non-functional attributes, such as their pricing and reputation.

1) **Execution price**. The execution price $QoS_{price}(s)$ is the price a consumer must pay when requesting the given service s.
2) **Execution duration**. The execution duration $QoS_{du}(s)$ is the moment that evaluates the expected delay (in seconds) of a given service s between sending the request and receiving the result. The value of the execution duration is calculated by using the formula (1).

$$QoS_{du}(s) = T_{trans}(s) + T_{process}(s) \tag{1}$$

where for a given service s, $T_{trans}(s)$ is the transmission time and $T_{process}(s)$ is the processing time. Each service s will display its $T_{process}(s)$ directly or provide a query method for it. $T_{trans}(s)$ is a dynamic number calculated by the formula (2).

$$T_{trans}(s) = \frac{\sum_{i=1}^{n} T_i(s)}{n} \tag{2}$$

where $T_i(s)$ is past performance of the transmission time, and n is the number of execution times observed in the past. So, the transmission time $T_{trans}(s)$ is recalculated after the last request of the service s.
3) **Reputation**. Given a service s, the reputation $QoS_{req}(s)$ is a measure of its trustworthiness. It mainly depends on end users' experiences of using the service s. The value of the reputation is defined as the average ranking given to the service by end-users and calculated by the equation (3).

$$QoS_{rep}(s) = \frac{\sum_{i=1}^{n} R_i}{n} \tag{3}$$

where R_i is the reputation ranking of the service by the end-user and n is the number of times the service is rated. Typically, at the end of the process, a scope is given to the end-users to sort the Web services.
4) **Successful execution rate**. The successful execution rate $QoS_{rat}(s)$ represents the probability of successful execution for a given service s, which is calculated by the formula (4).

$$QoS_{rat}(s) = \frac{N_c(s)}{n} \tag{4}$$

where $N_c(s)$ is the number of successful executions of service s, and n is the total number of invocations.
5) **Availability**. For a given service s, the availability $QoS_{av}(s)$ is the possibility of the service is requested. The availability is calculated by the formula (5).

$$QoS_{av}(s) = \frac{T_a(s)}{\theta} \tag{5}$$

where T_a is the total time the service s is accessible within the last θ seconds. Constant θ is defined by the

administrator according to the service scenario.

Based on the given attributes of service quality, we classify these properties as positively correlated and negatively correlated. As the values of each element in QoS are very different, each attribute values of QoS need to be normalized. The values of positively correlated elements are easily converted to values between 0 and 1, so we just consider the transformation of negatively correlated elements values, which include execution price and execution duration. The value is calculated by the expression (6).

$$QoS_i = \begin{cases} \frac{Q_i - Q_i^{max}}{Q_i^{max} - Q_i^{min}}, & Q_i^{max} \neq Q_i^{min} \\ 1, & Q_i^{max} = Q_i^{min} \end{cases} \quad (6)$$

where Q_i represents the value of a negatively correlated element. Q_i^{max} and Q_i^{min} show the maximum and minimum values respectively of this element, which is dynamically changing with services usage and feedback.

Through calculating and normalizing the value of every service quality elements, comprehensive quality of service is given in expression (7).

$$QoS(s) = \sum_{i=1}^{n}(QoS_i(s) * W_i), W_i \in [0,1], \sum_{i=1}^{n} W_i = 1 \quad (7)$$

where $QoS_i(s)$ is the value of each service quality elements and W_i represents the weight of corresponding elements. According to the specific scenario and user' demand, the weight of each QoS element can be adjusted dynamically. In this paper, we set each weight to 0.2.

After evaluating the quality of all the services, we can give corresponding rewards to the data providers, who published these services. The reward for each service is calculated by the formula (8).

$$R_s = \frac{\sum_{i=1}^{n} QoS(s_i)}{n} * P \quad (8)$$

where $QoS(s_i)$ indicates the service quality of last i times during the last n times. P is the proportion of QoS to reward.

When data consumers explore their needed services, some high QoS services will have higher possibilities to be subscribed. Data consumers are more likely to make a choice from these presented services based on their expected QoS and the desired price (how much they are pleased to pay). In this way, the service providers who present cost-efficient services can earn much more profits. By evaluating the QoS and rewards at the completion of each request, this mechanism can guarantee the continuous QoS avoiding the problem of too many requestors of a service leading to poor quality of service. Those who provide economical service will be motivated, and they will be inclined to offer more high-quality services.

V. EXPERIMENT AND EVALUATION

We develop an implementation of this prototype based on blockchain and evaluate the performance of the incentive mechanism in this section. The prototype is deployed on the virtual machine in Ubuntu 16.04 64bit operation system using Intel(R) Xeon(R) CPU E5-2620 v4 @ 2.10Ghz, 4GB RAM. The main business processes are written in a specific language named Solidity [11], whose syntax is similar to the JavaScript. We use the official IDE (Integrated Development Environment) named Remix [12] to debug and test the smart contract and utilize *Truffle* [13] (the development framework) to compile and deploy the data exchange contract in the blockchain network. We implement a blockchain network based on private Ethereum [14] and smart contract. Invoking the smart contract requires Web3.js API [15], which includes web3.eth.getTransaction, web3.eth.accounts and web3.eth.contract. We also package these basic interfaces into some integrated functions, opening in the form of SDK (Software Development Kit). These SDKs is convenient for data consumers to develop new applications based on existing services. The data consumers and data providers use the SDK to query and invoke the data exchange contract to record the requests and responses to the blockchain.

Next, we show the experiment results. When data consumers want to subscribe to the data they need, they will request the data exchange contract with the corresponding fee. In this process, the data exchange contract will run the validation whether he/she has sent a sufficient fee. If the verification passes, the data exchange contract will transfer the fee to the data provider and generate permission for data consumer access the data they have subscribed to. At last, the notification shows the transaction hash of this process, which indicates that the digital asset exchange has been completed. We can also look up the detail information of transaction in the blockchain explorer. Fig.4 shows this process.

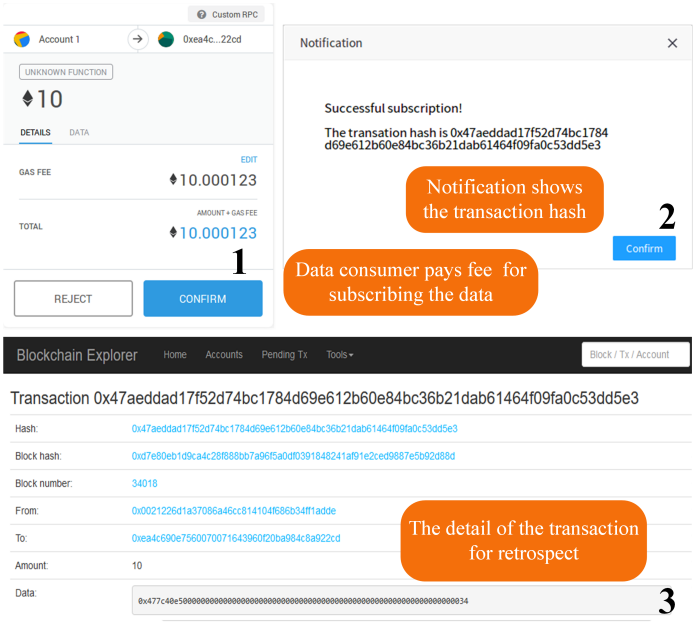


Fig. 4. The results of the digital asset exchange

Finally, we give the experiment of the QoS-based incentive

TABLE I
THE EVALUATION OF QUALITY OF 10 SERVICES

Services	Quality Services								Poor Services	
	A	B	C	D	E	F	G	H	I	J
Initial QoS	0.9500	**0.9000**	0.8500	0.8000	0.7500	0.7000	0.6500	**0.6000**	**0.9000**	**0.6000**
10 times	0.9593	**0.8715**	0.8831	0.8238	0.8037	0.7319	0.7403	**0.6697**	**0.7343**	**0.4811**
20 times	0.9281	**0.8794**	0.8699	0.8653	0.8507	0.8021	0.8023	**0.7330**	**0.5392**	**0.4101**
30 times	0.9052	**0.8603**	0.8699	0.8513	0.8780	0.8261	0.8510	**0.7934**	**0.4130**	**0.3179**
40 times	0.8814	**0.8951**	0.8645	0.8445	0.8868	0.8248	0.8247	**0.8059**	**0.4263**	**0.2848**
50 times	0.9006	**0.8667**	0.8843	0.8694	0.8781	0.8729	0.8720	**0.8193**	**0.3122**	**0.2189**
60 times	0.8874	**0.8611**	0.9033	0.8702	0.8565	0.9066	0.8932	**0.8241**	**0.3157**	**0.2130**
70 times	0.9155	**0.8443**	0.9204	0.8730	0.8782	0.8766	0.9106	**0.8266**	**0.3252**	**0.2102**
80 times	0.8851	**0.8406**	0.9175	0.8635	0.8810	0.8981	0.9090	**0.8697**	**0.2892**	**0.2409**
90 times	0.8819	**0.8358**	0.8974	0.8683	0.8744	0.8808	0.9003	**0.8959**	**0.2574**	**0.2391**
100 times	0.8910	**0.8818**	0.9091	0.8827	0.8656	0.9077	0.9001	**0.9210**	**0.2912**	**0.3009**

mechanism. We investigate 10 services, with 7 high-quality services and 3 poor services with low quality. TABLE I shows the evaluation data, from which we can see low-quality services' QoS will decline as the increasing number of requests, while high-quality services will not. And we select two high-quality services and two low-quality services to compare the reward. The result is shown in Fig.5. As the number of requests increases, we can see the low-quality services will attain less and less reward for providing data, but high-quality services will maintain high reward all the time. In this way, data providers will be motivated to publish more efficient services to earn more profits.

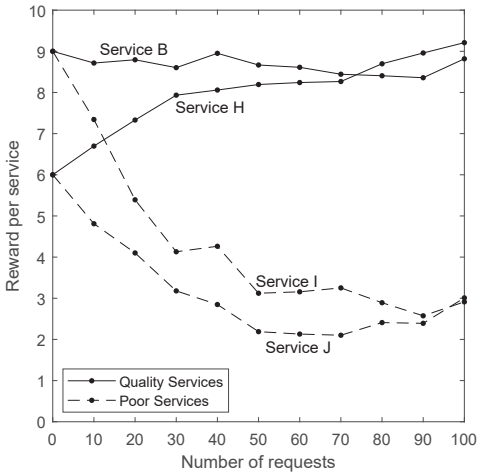


Fig. 5. The comparison of reward between different quality services

VI. CONCLUSION

To address the lack of a secure execution mechanism for data exchange and realize a future vision of the large-scale distributed IoT systems, we propose a secure digital asset exchange mechanism based on blockchain technique and QoS-based incentive mechanism in this paper. Data providers can fairly protect their data rights and easily retrospect transaction information. Especially, the incentive mechanism motivates the data providers to publish more cost-efficient services. After implementing the prototype in the private Ethereum network, it is concluded that the presented mechanism can protect the data providers' rights and automatically complete the process of digital asset exchange. By evaluating the performance of the incentive mechanism, it is concluded that the mechanism can effectively stimulate the provision of economical services.

REFERENCES

[1] Y. Shen, T. Zhang, Y. Wang, H. Wang, and X. Jiang, "Microthings: A generic iot architecture for flexible data aggregation and scalable service cooperation," *IEEE Communications Magazine*, vol. 55, no. 9, pp. 86–93, 2017.

[2] W. Tong, X. Dong, Y. Shen, and X. Jiang, "A hierarchical sharding protocol for multi-domain iot blockchains," *International Conference on Communications*, pp. 1–6, 2019.

[3] O. Salman, I. Elhajj, A. Kayssi, and A. Chehab, "Edge computing enabling the internet of things," in *Internet of Things (WF-IoT), 2015 IEEE 2nd World Forum on*. IEEE, 2015, pp. 603–608.

[4] R. Li, H. Asaeda, and J. Li, "A distributed publisher-driven secure data sharing scheme for information-centric iot," *IEEE Internet of Things Journal*, vol. 4, no. 3, pp. 791–803, 2017.

[5] J. Liang, W. Han, Z. Guo, Y. Chen, C. Cao, X. S. Wang, and F. Li, "Desc: enabling secure data exchange based on smart contracts," *Science China Information Sciences*, vol. 61, no. 4, p. 049102, 2018.

[6] Y. Zhang, S. Kasahara, Y. Shen, X. Jiang, and J. Wan, "Smart contract-based access control for the internet of things," *IEEE Internet of Things Journal*, 2018.

[7] J. Zheng, X. Dong, T. Zhang, J. Chen, W. Tong, and X. Yang, "Microthingschain: Edge computing and decentralized iot architecture based on blockchain for cross-domain data shareing," in *2018 International Conference on Networking and Network Applications (NaNA)*. IEEE, 2018, pp. 350–355.

[8] P. Wang, X. Liu, J. Chen, Y. Zhan, and Z. Jin, "Poster: Qos-aware service composition using blockchain-based smart contracts," in *2018 IEEE/ACM 40th International Conference on Software Engineering: Companion (ICSE-Companion)*. IEEE, 2018, pp. 296–297.

[9] S. Nakamoto *et al.*, "Bitcoin: A peer-to-peer electronic cash system," 2008.

[10] Y. He, H. Li, X. Cheng, Y. Liu, C. Yang, and L. Sun, "A blockchain based truthful incentive mechanism for distributed p2p applications," *IEEE Access*, vol. 6, pp. 27 324–27 335, 2018.

[11] The Documentation of Solidity[Online], Available:https://solidity.readthedocs.io/en/v0.5.6/.

[12] Remix - Solidity IDE [Online], Available:http://remix.ethereum.org.

[13] Truffle Suite — Sweet Tools for Smart Contracts[Online], Available:https://truffleframework.com/.

[14] G. Wood, "Ethereum: A secure decentralised generalised transaction ledger," *Ethereum project yellow paper*, vol. 151, pp. 1–32, 2014.

[15] Web3 javascript api to interact with blockchain nodes. [Online], Available:https://github.com/ethereum/wiki/wiki/JavaScript-API.