IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 20 November 2022, accepted 2 December 2022, date of publication 9 December 2022,
date of current version 20 December 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3227969

RESEARCH ARTICLE

SPChain: A Smart and Private Blockchain-Enabled Framework for Combining GDPR-Compliant Digital Assets Management With AI Models

WEI-SHAN LEE[1], JOHN A[1], (Member, IEEE), HSIU-CHUN HSU[2], AND PAO-ANN HSIUNG[1], (Senior Member, IEEE)
[1]Department of Computer Science and Information Engineering, National Chung Cheng University, Minxiong 62102, Taiwan
[2]Department of Information Management, National Chung Cheng University, Minxiong 62102, Taiwan

Corresponding author: Pao-Ann Hsiung (pahsiung@cs.ccu.edu.tw)

This work was supported in part by the Research Projects from the Ministry of Science and Technology, Taiwan, under Grant MOST 110-2643-F-194-006, Grant MOST 110-2420-H-194-002-TH, Grant MOST 110-2927-I-194-001, and Grant MOST 108-2221-E-194-033-MY3.

ABSTRACT In the traditional approach to a digital asset management system, the data processing mechanism is not transparent or visible to the data owners since the data is managed solely by the service provider. With the rapid development of blockchain technology, the above issues can be resolved by leveraging the tamper-resistance and decentralization characteristics of blockchain. However, post the implementation of the EU General Data Protection Rules (GDPR) in 2018, the protection of data owners has taken center stage. This has led to several principles of personal data deletion, such as Storage Limit and the Right to Be Forgotten to conflict with the blockchain. It is also observed that, out of the various smart contracts deployed to manage digital assets, often only specific smart contracts are invoked, while the rest of the deployed smart contracts are rarely invoked, leading to smart contract designs exhibiting similar patterns with very little creativity. This current scenario has motivated us to propose SPChain, a smarter and private GDPR-compliant digital asset management framework enabled by blockchain. In this approach, a decentralized InterPlanetary File System has been adopted to solve the problem of SPOF. In addition, the combination of digital assets with artificial intelligence models has been proposed so as to make digital assets accessible to a larger number of applications and to enable better creativity. In this design, artificial intelligence models have been run in independent, virtualized containers and invoked through smart contracts. The proposed SPChain can be applied to the field of digital art management to provide a complete implementation based on the Hyperledger Fabric. Using this proposed framework, model developers, digital art creators, collectors, service providers, as well as third parties can not only benefit from securely managing digital assets and combining them with AI models, but also from simultaneously complying with the rights stipulated in the GDPR. During the course of the experiments conducted, the latency, throughput, and resource consumption of different functions in the smart contracts have been measured. After adjusting the batch timeout of the block and the maximum number of transactions in a block, the throughputs were observed to be about 500 TPS, with 10 to 15 TPS for reading and writing operations, respectively. The latency ranges were found to range from 0 to 7 seconds, with 2.5 to 5 seconds for reading and writing operations, respectively.

INDEX TERMS Blockchain, AI smart contracts, GDPR, digital asset management.

I. INTRODUCTION

The present age has witnessed a more extensive number of items digitized around our lives than ever before. Everyone has their own digital data in the form of music, images, e-books, and even social media profiles. The number of digital assets generated or transferred are also at an all-time high. Especially in the field of Arts, innumerable artworks have been created by artists through computer software, scanners or tablets that are incorporated into digital art. According to

The associate editor coordinating the review of this manuscript and approving it for publication was Chun-Wei Tsai.

statistics, the value of output of digital media in the United States alone is close to \$63.9 billion [1].

NFT (Non-Fungible Token) trading has gained popularity in the real-world market. It is a non-interchangeable unit of data stored on the blockchain as a digital ledger, and data units include digital files such as photos, videos, and audio. Because each token is uniquely identifiable. Though there are several platforms for NFT trading, there is still a lack of one that can allow creators of NFTs or creative digital artworks to proactively use the platform for trading, creation, and management. One of the biggest applications of blockchain is Bitcoin, the origin of trending cryptocurrency. In the real world, popular NFT-based platforms such as Dapper [2] and Pixura [3] are built on ERC-721 smart contracts, entitling users to create and collect digital assets. Although these NFT-based platforms are mature and popular, more relevant topics of compliance with GDPR and combination with AI models need to be made secure integrated platforms for creation, management, and trading of NFTs. In our work, secure and GDPR-compliant digital asset management framework by leveraging the security and non-tampering features of the blockchain. At the same time, digital assets are more accessible to many applications and include more creativity. The similar application fields of Digital Assets Management are financial institutions, healthcare, real estate, logistics, etc. Most of these digital assets are managed by Service Providers (SPs) or digital asset management systems (DAMS), so that Data Subjects (DSs) whose personal data is collected can easily and conveniently access them without considering the underlying operating mechanism. DAMs provide an organized way to quickly query and access digital assets. However, there are two shortcomings that arise from this: (1) The DS cannot transparently track the process of digital assets and therefore has less control over them. For example, the analyzing process behind the data is not revealed to the DS (2) Data is served by a centralized management that is prone to instances of single point of failure (SPOF). For example, Cambridge Analytica revealed that Facebook abused their control over users' personal data to affect the U.S. general elections [4]. Consequently, it has become a crucial area of research to explore how to effectively and safely manage digital assets while also restricting access rights and possession of Proof of Delivery (PoD), which ensures that the requested digital media has reached its users successfully [5], [6].

In May 2018, the General Data Protection Regulation (GDPR) [7] whose jurisdiction covers all residents of EU as well as those who provide services and goods to them, came into effect in all European countries.Apart from providing the legal framework for the collection and processing of personal data concerning individuals in the EU, it was also designed to cover important components of EU human rights law and privacy law. The GDPR defines and regulates the handling of personal data of the DSs. Violators stand to be fined up to 4% of their global turnover or \$20 million Euros, whichever is higher. Therefore, compliance with the GDPR is an urgent matter for companies. This work focuses on following the principles laid out for the processing of personal data and for preserving the rights of the DSs specified in the GDPR under secure DAM, where digital art is the primary asset.

A. BACKGROUND

Blockchain technology is all the rage today. Due to the tamper-resistant design of the ledger maintained by the nodes in a blockchain network, all changes or transactions made can be detected [8]. One of the biggest applications of blockchain is Bitcoin, which is also the precursor of all trending cryptocurrencies. Since its inception, blockchain has been applied and combined with different fields (e.g., the Internet of Things, financial institutions, healthcare and DAM [9], [10], [11]). Compared with original digital art, "rare" digital art (also known as crypto art) makes the artwork collectible in limited quantities by attaching a unique token generated by blockchain [12]. This kind of combination not only ensures that the provenance of artworks cannot be tampered with while eliminating the problem of SPOF, thanks to the transparency and decentralization of blockchain. There exists several crypto art gallery platforms currently such as SuperRare [13] and Cryptograffiti [14]. Although both platforms have drawn art creators' and collectors' attention successfully and have integrated with Ethereum well, while keeping the functionality of the Smart Contracts (SCs) relatively simple including uploading, updating, and transferring of digital assets, which restricts the art creators from generating more artwork applications. Kiffer et al. [15] investigated Ethereum's SC topology and revealed that most contracts are in fact direct or mere copies of the others. Therefore, improved designs and better compliance with the GDPR is the need of the hour.

Many outstanding works have proposed different methods for resolving this conflict. In [16], three layers of blockchains with different functions were combined to ensure the Right to be Forgotten and the Right to Portability. The off-chain mechanism has been used in [17] and [18] to achieve GDPR compliance in cooperation with the SCs. Bayle et al. [17] proposed a model based on MyHeathMyData, which stores the data in Off-Chain Storage of Data Controllers (DCs). Truong et al. [18] improved the traditional OAuth authorization process by transferring the process of authentication and authorization to the blockchain network, and applied the scheme in Personal Data Management (PDM). While most of these approaches assayed improvements, most of them: (1) Proposed conceptual frameworks only and lacked practical implementation, as well as experimental results, (2) Did not emphasize the articles of the GDPR enough (i.e., against the Right to be Forgotten and the Right to Portability). This void, therefore, prompted the researchers of this work to use additionally designed SCs and adapt the design concept of the GDPR-compliant PDM proposed by [18] to the field of digital art management.

B. MOTIVATION

To summarize, there are three main pain points in a mature GDPR-compliant digital art management system: (1) Lack of diversity in the SCs (2) Lack of practical implementation as well as performance evaluations and (3) Lack of more comprehensive GDPR analysis. This has led the authors to propose SPChain, a secure and private blockchain-enabled framework combined with AI Models to tackle the above issues. Most of the commonly used topology of traditional SCs are limited to a few functions such as creating, transferring, etc., resulting in very low rates of invoking other SCs.

Traditional authentication mechanisms may allow SPs to hand out personal data beyond the users' commitment [18]. The blockchain-based access token is therefore introduced in SPChain, where all logs of interactions with the assets are appended to an immutable ledger. Further compliance with the GDPR principles, metadata of artworks and privacy related data are stored into off-chain distributed file systems. In this context, it may be stated that the GDPR principles include (1) Lawfulness, fairness and transparency, (2) Purpose limitation, (3) Data minimization, (4) Accuracy, (5) Storage limitation, and (6) Integrity and confidentiality, where rights toward the DSs include (1) Right to rectification, (2) Right to erasure, (3) Right to data portability.

1) CONTRIBUTION

In this manuscript we proposed SPChain to achieving a secure and GDPR-compliant digital asset management framework by leveraging the security and non-tampering features of blockchain. The SPChain can be credited with three main contributions such as

i. A thorough design for securely and privately managing the digital assets with a digital artwork used for a complete illustration.

ii. The adoption of decentralized Off-Chain Storage (i.e., IPFS) to replace traditional centralized storage for avoiding the problem of single point of failure.

iii. The combination of digital assets and AI models via virtualized container technology.

II. RELATED WORK

This section presents relavant knowledge on combining GDPR complaints digital management with AI models. Table 1 shows different notations and their descriptions for better understanding.

Compared to physical art or fine art, crypto art is entirely connected with specific and unique tokens on the blockchain where the tokens represent provenance for a piece of digital art [12]. More specifically, such tokens are represented by Non Fungible Tokens (NFTs) in Ethereum [23], which guarantee that the digital assets or files will not be replaced. Non Fungible Tokens are based on the ERC-721 smart contract, making it easier for Ethereum developers to construct related applications. In another vein, since AlexNet [24] and ResNet [25] have come out on top of the image classification tasks held by the ImageNet project in 2012 and 2015 respectively, the amount of research efforts made with respect to deep learning has increased significantly. Therefore, the application of AI in the Arts can be quite abundant and can be generally divided into synthesis and generating, style transferring, and AI as assistant.

TABLE 1. Terminologies and description.

Terminologies	Description
AG	Art Gallery
CA	Certificate Authority
CL	Collector
CR	Creator
DAMS	digital asset management systems
DCs	Data Controllers
DPs	Data Processors
DSs	Data Subjects
GDPR	General Data Protection Regulation
HLF	Hyperledger Fabric
IPFS	InterPlanetary File System
MD	Model Developer
PoD	Proof of Delivery
SCs	Smart Contracts
SPOF	Single Point of Failure
SPOF	single point of failure
SPs	Service Providers
TP	Third Party

Relatively few studies have been conducted on DAM with blockchain. Most of the current work, in fact, has applied blockchain as a notarization service to PDM [16], [18], [26], [27], [28]. Even so, personal data could be considered a kind of digital asset. The GDPR has standardized data protection principles and put together a most stringent personal data protection policy. It follows that the mechanism of personal data must conform to the principle of "Personal Data Processing Principles" defined in Article 5 of the GDPR. DPs must not violate the rights of DSs. It is noteworthy that none of the three types of blockchain (i.e., public, private, consortium) can guarantee "Storage Limitation" and "Accuracy" [29]. Furthermore, Bernabe et al. [30] conducted a detailed investigation on the blockchain privacy challenges and observed that most research proposals mainly revolve around transaction likability, private key management, confidentiality and control, privacy regulations, and so forth. The authors of Paul Ryan [31] proposed an Accountability semantic model for ROPA (Registers of Processing Activities) using GDPR and its concept of accountability. This semantic model provides the solution to the compliance problem faced by an organization. The authors Maria Koutli et al. [32] proposed the VICINITY IoT Framework, which included a GDPR-compliant mechanism for providing secure health services to the middle-aged and the elderly. However, privacy, security availability, and data integrity have not been covered in the framework. The authors of [33] provided an overview of the functional varieties of different types of AI explanations from a legal point of view using the GDPR framework, and it has been extended further to cover the protective and technical aspects of banking law.

It is the authors' belief that while several works have endeavoured to solve relevant conflicts around the scenario of a secure DAM (or PDM) in compliance with the GDPR principles, they have either not addressed the GDPR principles completely, or been limited to conceptual designs or incomplete experimental work as well as lacked extendibility to other scenarios. The authors of this work, therefore, have proposed SPChain, a secure and private blockchain-enabled framework combined with AI Models as a novel way of extending existing designs to include additionally designed SCs for better creativity, incorporating a more complete adherence to the GDPR principles, along with an implementation that goes beyond the conceptual to a complete experimental evaluation and with the combined AI models, the ability to readily extend its functionality to more intelligent applications.

Leda kamal et al. [41] review the challenges, privacy and privacy mechanism in blockchain based on the different realms of blockchain, such as IoT and smart agriculture. Mobile and ubiquitous computing provide transparent and stream-less services to the environment [42]. In this context, the GDPR constraints are used to implement the privacy regulations. The author of Vinden Wylde [43] reviews the organization's and business's key challenges in public and utilizes using blockchain technology. The legal framework and types of security policy and outcomes are generated using the GDPR. The authors of [44] proposed a blockchain-cloud-based architecture for hybrid data marketing. In this work, content-based secure data trading and achieved privacy of data owner in distributed credential issuance. Using this architecture, unfair marketing operations are detected. The authors of [45] presented blockchain technology and GDPR to provide adequate healthcare records in the health sector. The combination of blockchain and GDPR provides a secure decentralized network with the public ledger. The authors of [18] proposed a platform to develop a GDPR-based personal data management system using blockchain and smart contract technologies. The main goal of this platform is to create a decentralized mechanism for service providers. Similarly, the authors of [47] proposed a scheme for processing services made in the smart grid using the blockchain and supported framework to trust free data computations and tracking. The operations and services in the different existing proposed frameworks [18], [31], [32], [33], and [47] are not managed securely, and the main rights of GDPR are not set out and checked. The detailed state-of-the-art work of different frameworks is presented in Table 2.

TABLE 2. State-of-the-art of different GDPR framework's.

S.No	Framework and References	Description
1	BPDIMS [16]	Using GDPR and blockchains architecture different functions were combined to ensure the Right to be Forgotten and the Right to Portability.
2	ROPA using GDPR [31]	Provided accountability using GDPR and semantic model provide the solution to the compliance problem faced by an organization.
3	VICINITY Framework [32]	GDPR compliant mechanism for providing secure health services to the middle-aged and the elderly.
4	Funtional Legal view using GDPR Framework [33]	Provided different types of AI explanations from a legal point of view using the GDPR framework.
5	GDPR for ubiquitous Computing [42]	Using GDPR managed the Privacy policies and provided seamless and transparent services.
6	Block-DM [44]	This framework, content-based secure data trading and achieved privacy of data owner in distributed credential issuance. Using this framework, unfair marketing operations are detected.
7	GDPR and Blockchain for healthcare [45]	Presented blockchain technology and GDPR to provide adequate healthcare records in the health sector. The health data controlled and owned by particular patients instead of transferring different healthcare service provider.
8	GDPR-Personal Data Management [18]	Developed a platform to develop a GDPR-based personal data management system using blockchain and smart contract technologies. The main goal of this work is to create a decentralized mechanism for service providers.

III. SPChain DESIGN

The design of SPChain is explained in details this section. First of all, the roles of participants of the SPChain are described in Section III-A. Then, the high-level system architecture including Docker Registry, DAMS, off-chain storage, and HLF blockchain, along with a description on the AI models is given in Section III-B. In Section III-C, we describe the blockchain used for SPChain and the raft consensus in particular. In Section III-D, we discuss the design of how artworks are managed securely including initialization, uploading, and transferring of digital artworks. Section III-E, reveals how the creator of artworks can use smart contracts to invoke AI models that are uploaded by model developers. Finally, in Section III-F, the process of requesting data with the consent of the service providers and data subjects has been described including request consent, request data and revoke consent.

A. PARTICIPANTS ASSUMPTIONS

The major roles played by the participants engaged in the SPChain have been described as follows:

- *Creator* (*CR*) :
  The *Creator* is the role played by the creative artists who establish the digital works in the first place and publish the artworks to the art gallery for sale. Additionally, the *CR* can invoke the models developed by the *MD* to make more creative artificial artworks. With respect to the GDPR, the *CR* is regarded as the DS.
- *Collector* (*CL*) :
  After the *CR* uploads the masterpieces to the IPFS, the *CL* can purchase the artworks they are fond of through the smart contracts. Besides, the *CL* also has the ability to resell collected artworks whereupon the original *CR* gets specific rates of royalty (e.g., Artist's Resale Royalty (ARR) in Australia's Art market). As with the *CR*, the *CL* also plays the role of the DS with respect to the GDPR.

- *Model Developer* (*MD*) :
  The *MD* can upload diverse machine learning models to the Docker Registry so that the creators can make use of these models. As a reward, when the model is adopted by one of the *CR*s, a specific bonus is deposited into the corresponding *MD*'s wallet.
- *Art Gallery* (*AG*) :
  The *Art Gallery* is in charge of the platform or the interface for the different participants to interact with each other. As a legal SP, the *AG* first obtains the consent from the DS and thereafter owns various rights such as reading the information of artworks, acting as the delegate to carry out transactions or triggering the entry point for invoking models in both the IPFS and the blockchain network.
- *Third Party*(*TP*) :
  Under the circumstances of transferring the DSs' data to another agency or entity, the *TP* is a DP (Data Processor) as per the GDPR terminology. The *TP* aims at developing extra services with those with the desired personal data. It is noteworthy that the *TP* is not involved in the blockchain network.

B. HIGH LEVEL SYSTEM ARCHITECTURE

As shown in Figure 1, there are four main modules involved in the architecture, namely the Docker Registry, the DAMS, the Off-Chain Storage, and the HLF Blockchain. The DAMS module is involved during the interaction between the participants. More specifically here, the *Art Gallery* as an SP provides the end-users with functionality such as uploading of artworks, uploading of models, transferring the artwork, and the ability to view the collections. The data generated by these interactions are stored into a tamper-free ledger in the HLF and the Off-Chain Storage (i.e., IPFS). Out of the two, the IPFS is the one that stores large files that are not easy to store in the blockchain (e.g., images or music of artworks), as well as personal private data (e.g., profile, information of artworks created by *CR*, and collections collected by the *CL*). Conversely, the ledger in the blockchain stores the hash pointer to the IPFS and the data that is not required for the validation the provenance (e.g., the first-hand creators or the life cycle of the digital assets). A third party can request consent from the end-users to access their personal data for extra services or applications (which is not discussed in this work). In contrast, as a containerized model registry platform, the Docker Registry is in charge of managing the models and avoids directly storing up models containing large parameters in the blockchain and IPFS so that better management of these models and reduced overhead for consensus agreement in distributed systems can be achieved. In SPChain, we have implemented five smart contracts to ensure confidentiality, integrity, and regulatory compliance in the DAMS as described below. It is imperative for all the blockchain participants to be issued certificates from Fabric CA before invoking the smart contracts. More details on the smart contract algorithms inside the SCs shall be introduced in the following sections.

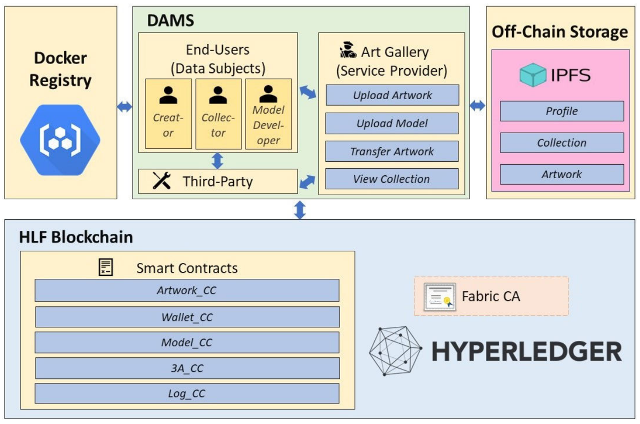

FIGURE 1. High-level system architecture of SPChain.

- Artwork_CC
  Defines or Creates, Reads, Updates, or Deletes (CRUD) the artworks.
- Wallet_CC
  Manages wallets that store a custom virtual token used to represent the medium of exchange.
- Model_CC
  Stores the address of the models in the Docker Registry and offers interfaces for invoking the model API.
- 3A_CC
  3A_CC is used to manage authentication, authorization, and access-control when different participants want to access the data [18]. Specifically, the 3A_CC lets data subjects define their consents.
- Log_CC
  Log_CC logs the operations related to the data and acts as a second layer of protection to prevent illegal roles from accessing personal data.

1) AI MODELS

In contrast to traditional SCs, AI SCs are combined with AI models to increase creativity and an inherent orientation towards diverse applications by invoking model inference APIs inside the contracts. In particular, using neural networks such as Convolutional Neural Networks (CNNs) [19] or Generative Adversarial Networks (GANs) [20] enables one to transfer the style of artworks or even create new artifacts. SPChain also integrates the idea of exchanging machine learning models online [21] with digital art management platforms [22]. As far as combination with AI models is concerned, the proposed SPChain is a novel one and combines DAM (or PDM) with AI models so as to not only eliminate the maintenance costs for individual platforms of blockchain and AI model marketplace, but also to orient towards more intelligent applications. combination of digital assets and AI models via virtualized container technology.

C. DESIGN OF BLOCKCHAIN NETWORK

In the HLF, a group of roles with similar interests or common values can be formed into an organization. Therefore, the

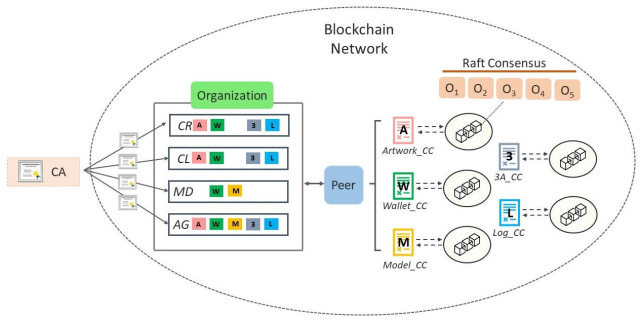

FIGURE 2. **Blockchain network architecture.**

participants described in Section III-A can be represented by four different organizations including ***CR***, ***CL***, ***MD***, and ***AG***. The transaction proposal sent by the client application is handled by peers in the organizations for the purpose of invoking the SCs. The blockchain network architecture in the HLF is shown in Figure 2. The four organizations are installed with different SCs whose color scheme displayed after the name of the organization corresponds to the color of the SCs they own. The reason for the usage of blockchain and raft consensus is as follows.

1) BLOCKCHAIN

We used blockchain to not only secure the digital assets but also to ensure that their provenance is traceable. This is not easily attainable without blockchain or if achievable needs a lot of intricate workings; however, blockchain provides a very intuitive way of meeting these requirements for digital asset management, namely asset security as well as provenance tracing. Furthermore, another major reason for using blockchain is because several actions can be automatically and securely conducted via smart contracts. The digital asset platform that we developed requires automatic processing of digital asset management actions, including creation, delivery (trading), privacy checking, and AI-based transformation of digital assets (such as NFTs).

2) RAFT CONSENSUS

This consensus algorithm not only guarantees safety but also performance (500 ms for leader election). Since the proposed digital asset management system needs both safety and performance, thus we chose Raft Consensus. Compared to other state-of-the-art distributed consensus algorithms such as Paxos or EPaxos, we chose Raft because it is more intuitive and also more implementable. The log continuity feature of Raft is also required in our platform due to the possibility of platform failure and needs for recovery. Thus, in summary, Raft was a much better choice for our platform design requirements.

D. DESIGN OF MANAGING ARTWORKS

The management of the artworks can be divided into three stages. The first is the initialization phase, which means that ***Art Gallery*** needs to obtain the consent for manipulating the personal data from the End-User before providing any other functionalities. Technically, this kind of off-chain data is stored in orbitdb, which is a distributed database based on the IPFS protocol. The ***Creator*** and the ***Collector*** initially have their own orbitdb databases to store their creations or collections. The other two stages involve uploading (registering) and transferring of the artworks. In all the three phases, the *3A_CC* and log_CC is used for authentication, authorization and logging, the artwork_CC is used for the proof of the artwork provenance while the wallet_CC is invoked when monetary transactions are involved.

– Initial Phase

The ***Art Gallery*** as an interface manages the interactions of participants and needs to request permission for the personal orbitdb from the end-users before providing any functionalities. The address of orbitdb has been asymmetrically encrypted by the end-users in advance (denoted with *enhash*) with the public key of orbitdb pk_{enc} generated by an asymmetric encryption algorithm $\mathscr{R}$. Additionally, the data stored into orbitdb is also encrypted with a public key pk_{data} to avoid being read as plaintext. In the meantime, the ***AG*** plays the role of data controller or data processor. As shown in the sequence in Figure 3, the ***AG*** first attaches its signature Sig_{DC} to end-users for the consent request. As soon as the consent is obtained, ***AG*** receives the returned end-users signature Sig_{DS}, *enhash*, sk_{data} and the key-pair (pk_{enc}, sk_{enc}) used to encrypt and decrypt the address of orbitdb (step 1-2). Thereafter, the ***AG*** updates the policy value in 3A_ledger, which ensures that the end-user's consent is written into the ledger (steps 3-4). It is worth noting that the snapshots of 3A_ledger can be seen by the ***CR***, the ***CL*** and the ***AG*** since all of them are installed with 3A_CC.

The data model of 3A_ledger is shown in Listing 1. 3A_ledger consists of a key-value pair where the key is the combination of pk_{DS} and pk_{DC}, and the values are consent policy $\mathbb{P}$, *timestamp*, *enhash*, and pk_{enc}. 3A_ledger is entitled to manage and determine who has the ability to access the DS's data which is escrowed by the DC. The policy defines the respective permissions for accessing the underlying database (i.e., CRUD). When log_CC is invoked, sk_{data} is stored into the log_ledger as the reference for providing the key to decrypt the ciphertext (step 5). log_CC logs the entire operation and status, which in this case is granting the consent. Listing 2 shows the key of log_ledger, which comprises of pk_{DS}, pk_{DC}, pk_{DP} while the values are sk_{data}, *timestamp*, *status*, and *operation*. *timestamp* represents the time that the *operation* is triggered, *status* reveals whether the transaction has been verified successfully, and the *operation* stands for the operation of this transaction. Lastly, the End-User dynamically adds the access control toward IPFS and then the ***AG*** gets the consents from the End-User successfully.

```
---
Key: pk_DS & pk_DC \\

Value:
# P for policy
P: {"C": "{"pk_x1","pk_x2",...}",
        "R": "{"pk_x1","pk_x2",...}",
        "U": "{"pk_x1","pk_x2",...}",
        "D": "{"pk_x1","pk_x2",...}"},
timestamp="1618909467",
enhash="d1ad9da01fa3...",
pk_enc="MIIBIDANBgkq..."
---
```

Listing 1. Data model of 3*A_ledger*.

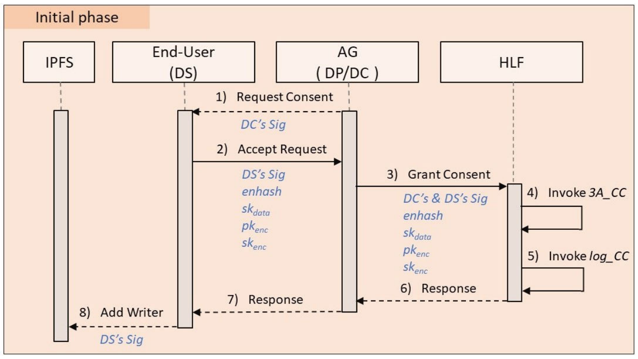

FIGURE 3. Initial phase of granting consent for AG.

```
---
\Key: pk_DS & pk_DC & pk_DP \\

Value:
sk_data="56KB4024..."
timestamp="1618909467",
status="approved",
operation="R"
---
```

Listing 2. Data model of *Log_ledger*.

– Uploading artworks

The conceptual process of uploading artworks from the ***CR*** has been illustrated in Figure 4. First of all, the artwork including the interrelated metadata is added to orbitdb which has already been established by the ***CR*** in the Initial Phase. Once the artwork is successfully added, the multi-hash of the adding entry is returned. artwork_CC then records the proof of provenance of the newly established artwork immediately. A more detailed process including the parameters can be found in Algorithm 1.

As stated in Section II, crypto art is embodied by a unique digital token, which represents the life cycle of the artwork. Consequently, a set of 16 digits Universally Unique Identifier (UUID) is randomly generated to represent the specific digital token, which is shown

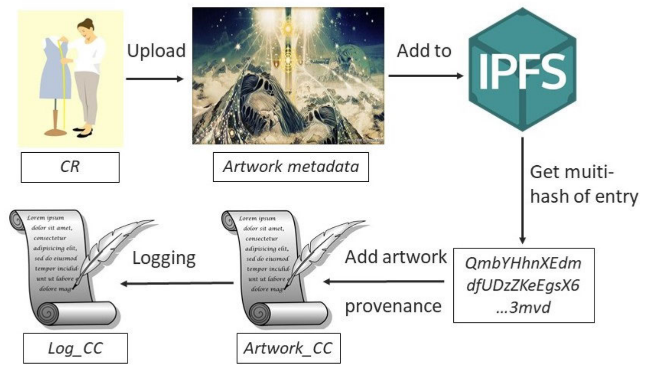

FIGURE 4. Flow of uploading artworks.

in Algorithm 1, line 4. After connecting with the ***CR***'s orbitdb, the digital token, together with the metadata of the artwork (e.g., image, name, related description, price, and the creator) are deposited into orbitdb. Thereafter, a multi-hash that has the type of string is generated as the output (line 3-5). This ensures that privacy-related and possibly changed data attributes are stored in an off-chain distributed storage to conform to the "Storage limitation" and "Right to erasure" in the GDPR. Next, the token, the multi-hash value of the entry, the collector (at this moment the collector has the role to create due to the firsthand transaction), the creator and the timestamp are written into the ledger via artwork_CC (line 8). Such an operation ensures the protection of the existing provenance of the artwork as well as any other changes made to the artwork by leveraging the blockchain's capability of maintaining evidence. Besides, the degree of transparency of the transition processes changes with the policies defined by the various organizations in the entire consortium of the blockchain. Last but not the least, the log_CC eventually writes the operation toward orbitdb into log_ledger (line 9).

– Transferring artworks

The workflow of transferring the artwork is shown in Figure 5. We assume the following two scenarios: (1) When the ***CL*** is interested in a favorite work and has an intent to buy it from the ***CR***. (2) When the artwork has been transferred at least once and is going to be transferred to the new ***Collector*** (***NCL***). In Figure 5, the ***AG*** first looks up the ***CL*** and the ***CR*** for transferring the artwork from the ledger, which aims at obtaining the roles related to the currency flow. After that, the ***AG*** tries to obtain the selling price of the artwork from orbitdb and pass it along with the roles involved in the monetary flow to wallet_CC. Then balances in the digital wallets are re-calculated and updated. Furthermore, the footprint of the artwork is also updated into the artwork_ledger while the metadata of the artwork is appended to the ***NCL***'s orbitdb. Lastly, the log_CC records this operation into the log_ledger.

Algorithm 1: Uploading Artworks Process

```
Input:
$\mathbb{D}_{CR}$: The orbitdb of $\boldsymbol{CR}$;
$PH_w$: The path of uploading artwork;
$N_w$: The name of uploading artwork;
$Desc_w$: The description of uploading artwork;
$P_w$: Current price of uploading artwork;
$CR_w$: $\boldsymbol{CR}$ of the uploading artwork;
$token_w$: A random unique UUID;
$pk_{CR}$: The public key of $\boldsymbol{CR}$;
$pk_{AG}$: The public key of $\boldsymbol{AG}$;
$pk_{DP}$: The public key of $\boldsymbol{DP}$;
$Sig_{CR}$: The signature of $\boldsymbol{CR}$;
$Sig_{DP}$: The signature of $\boldsymbol{DP}$;
$op$: Operation toward DS's data;
$\mathscr{V}$: The verifying signature algorithm;
Output:
$ret$: Returned result;
Init:
$ret$ = error;
Variable:
$mhash_w$: The multi-hash after uploading artwork to IPFS;
$t$: Current time;
$db$: A promise resolves to $\mathbb{D}_{CR}$;
$check$: Signature verify result;
1  $check = \mathscr{V}(pk_{CR}, Sig_{CR})$ && $\mathscr{V}(pk_{DP}, Sig_{DP})$;
2  if $check$ then
3      $db = orbitdb.open(\mathbb{D}_{CR})$;
4      $token_w = uuid()$;
5      $mhash_w = db.put(PH_w, N_w, Desc_w, token_w, P_w, CR_w)$;
6      if $mhash_w \neq NULL$ then
7          $t = Time.now(), op = \text{“}C\text{”}$;
           // $CR_w$ as $CR$ and $CL$
8          $artwork\_CC.upload(token_w, mhash_w, CR_w, CR_w, t)$;
9          $log\_CC.update(pk_{CR}, pk_{AG}, pk_{DP}, t, op)$;
10         $ret = success$;
11 Return $ret$;
```

The process of transferring artwork including related parameters is shown in more detail in Algorithm 2. At first, the public keys and the signatures of the ***CL***, ***NCL***, and the ***DP*** are verified (line 1). If the authentication is successful, the transfer process is approved. Two off-chain repositories are involved in this scenario denoted by db_{CL} and db_{NCL}. ***NCL*** refers to the new collector who wants to collect the transferred artwork. The db.get() can be regarded as an API from orbitdb, which is used to read the metadata of the artwork to be transferred in the database (line 5). Thereafter, the price (P_w), the predefined royalty rate (R_y), the ***NCL***, ***CL***, and the ***CR*** are passed to wallet_CC and the balance of the participants is cleared (line 9). The process of updating balances has been listed in Algorithm 3. It is worth noting that if it is not the firsthand transaction, the ***CR***'s balance will increase by P_w*R_y, while the existing collector ***CL*** will get P_w*(1-R_y) (Algorithm 3, line 7-8). After the balance is cleared, the artwork itself, along with metadata are appended to ***NCL***'s orbitdb and the ***CL*** in the ledger is changed to the ***NCL***'s updated information (line 10-11). In the same way, log_CC records the entire operation in the end (line 12).

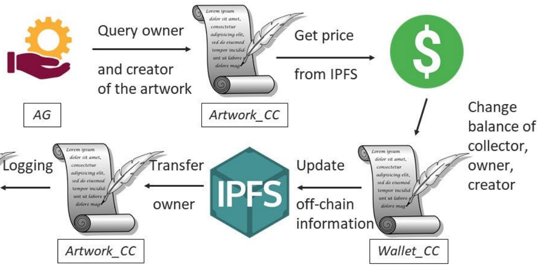

FIGURE 5. Flow of transferring artworks.

E. DESIGN OF COMBINING MODELS

AI-based pre-trained models have been adopted or employed as an assistant to create artworks as described in Section II. Immutability, availability and integrity are the unique blockchain properties that enable and adopt the secure distributed environment. The transparency, data usage and blockchain-based secure log are considered beneficial Therefore, combining the process of the model deployment and the adoption with respect to the aforementioned artwork's management will significantly save the indirect cost of time and ensure that ***MD*** receives the remuneration based on the transactions related to the models in the reliable ledger (either during the auction or when the model is invoked). The design of combining the models can be separated into two parts, namely uploading the models and invoking of the models requested from the ***CR***. In order to be felicitous toward the more micro-service oriented architectures prevalent nowadays, a virtualized container technology is regarded as one of the obvious choices since these containers are independent of each other and make the entire deployment process lighter and easier. The details of two parts have been provided as follows:

– Uploading Models

In general, the MDs and the DevOps engineers work separately. We let the ***MD*** or the ***AG*** replace the roles of the DevOps engineers for simplification in SPChain. The workflow for uploading the model is shown in Figure 6. The developed model M (which could include the API service and the corresponding DockerFile (DF_m)) is uploaded by the ***MD***. The DF is used to automatically install the dependency of the packages

Algorithm 2: Transferring Artworks Process

```
Input:
$\mathbb{D}_{CL}$: The orbitdb of $\boldsymbol{CL}$;
$\mathbb{D}_{NCL}$: The orbitdb of $\boldsymbol{NCL}$;
$token_w$: A random unique UUID;
$R_y$: The royalty rate (%);
$pk_{CL}$: The public key of $\boldsymbol{CL}$;
$pk_{NCL}$: The public key of $\boldsymbol{NCL}$;
$pk_{AG}$: The public key of $\boldsymbol{AG}$;
$pk_{DP}$: The public key of $\boldsymbol{DP}$;
$Sig_{CL}$: The siganture of $\boldsymbol{CL}$;
$Sig_{NCL}$: The siganture of $\boldsymbol{NCL}$;
$Sig_{DP}$: The siganture of $\boldsymbol{DP}$;
$op$: Operation toward DS's Data;
$\mathscr{V}$: The verifying signature algorithm;
Output:
$ret$: Returned result;
Init:
$ret$ = error;
Variable:
$W$: The transferring artwork which includes $\{PH_w, N_w, Desc_w, P_w\}$;
$CR_w$: $\boldsymbol{CR}$ of the transferring artwork;
$CL_w$: $\boldsymbol{CL}$ of the transferring artwork;
$NCL_w$: $\boldsymbol{CL}$ of the transferring artwork;
$P_w$: Current price of the transferring artwork;
$db_{CL}$: A promise resolves to $\mathbb{D}_{CL}$;
$db_{NCL}$: A promise resolves to $\mathbb{D}_{NCL}$;
$mhash_w$: The multi-hash after uploading artwork to IPFS;
$t$: Current time;
$check$: Signature verify result;
1  $check = \mathscr{V}(pk_{CL}, Sig_{CL})$ && $\mathscr{V}(pk_{DP}, Sig_{DP})$ && $\mathscr{V}(pk_{NCL}, Sig_{NCL})$;
2  if $check$ then
3      $CL_w, CR_w = arwork\_CC.query(token_w)$;
4      $db_{CL} = orbitdb.open(\mathbb{D}_{CL})$;
5      $W = db_{CL}.get(token_w)$;
6      $P_w = W.P_w$;
7      if $P_w \neq NULL$ then
8          $t = Time.now(), op = “R''$;
9          $wallet\_CC.update(R_y, NCL_w, CL_w, CR_w, P_w)$;
10         $mhash_w = db_{NCL}.put(token_w, W)$;
11         $artwork\_CC.update(token_w, mhash_w, NCL_w)$;
12         $log\_CC.update(pk_{NCL}, pk_{AG}, pk_{DP}, t, op)$;
13         $ret = success$;
14 Return $ret$;
```

Algorithm 3: Updating Wallet Process

```
Input:
$CR_w$: $\boldsymbol{CR}$ of the transferring artwork;
$CL_w$: $\boldsymbol{CL}$ of the transferring artwork;
$NCL_w$: $\boldsymbol{CL}$ of the transferring artwork;
$P_w$: Current price of the transferring artwork;
$R_y$: The royalty rate (%);
Variable:
$W_{CR}$: $\boldsymbol{CR}$'s digital wallet;
$W_{CL}$: $\boldsymbol{CL}$'s digital wallet;
$W_{NCL}$: $\boldsymbol{NCL}$'s digital wallet;
// Get wallets from the ledger
1  $W_{CR} = GetState(CR_w)$;
2  $W_{CL} = GetState(CL_w)$;
3  $W_{NCL} = GetState(NCL_w)$;
// Calculate and update the balance
// Firsthand transaction
4  if $CR_w == CL_w$ then
5      $W_{CR} += P_w$;
6  else
7      $W_{CR} += P_w * (R_y)$;
8      $W_{CL} += P_w * (1 - R_y)$;
9  $W_{NCL} -= P_w$;
// Update wallets to the ledger
10 $PutState(W_{CR}, W_{CL}, W_{NCL})$;
```

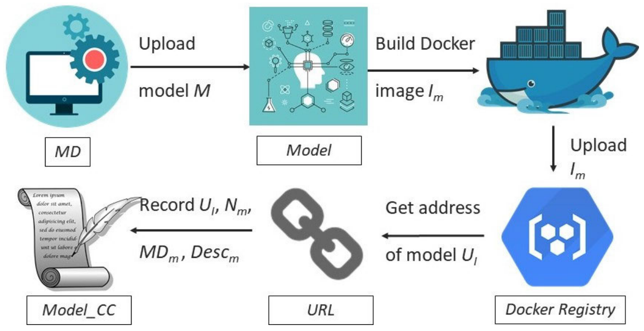

FIGURE 6. Flow of uploading models.

and execute the commands of the applications. The $\boldsymbol{MD}$ or the $\boldsymbol{AG}$ package M into a Docker image I_m according to the DF_m, and the I_m is then uploaded to the Docker Registry to be saved and await being picked by the $\boldsymbol{CR}$. The source address of the model uploaded to the Docker Registry (U_I), as well as, the name (N_m), the developer (MD_m), and the associated description ($Desc_m$) of M is recorded into the ledger via model_CC. Even though the operation is recorded on the Docker Registry at the same time, the evidence in the blockchain enables the reference for the $\boldsymbol{MD}$ to successfully obtain the premium.

In this section, we give an example of uploading a neural network model by the $\boldsymbol{MD}$. Suppose there is a developer Johnson (MD_m) who would like to upload a real-time and super-resolution style transfer model

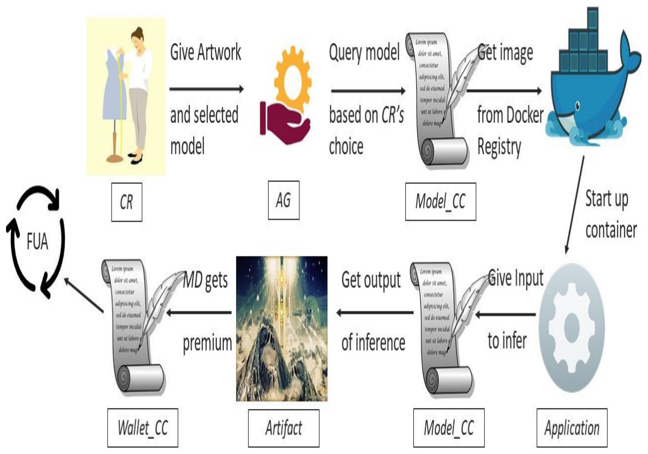

FIGURE 7. Flow of invoking models.

(N_m) of his research [35]. The model gives a fast and super-resolution image style transfer, which can be used by the ***CR*** to quickly transform their works (content-images) into various styles based on different style-images. In addition to uploading the binary file of the model, the uploaded materials also contain some additional description $Desc_m$ (e.g, advantages, functionalities, input, and output of the model). At the beginning of uploading the models, the related packages and execution commands used by the model such as pytorch, cuda, etc. are written into DF_m and compiled into I_m. I_m is then shared to the public or private Docker Registry, and the location of model U_I = https://hub.docker.com/repository/docker/johnson/fast_neural_style is returned by the Docker Registry. Finally, a 4-tuple parameter $\{MD_m, N_m, Desc_m, U_I\}$ is passed on to model_CC and this entry is recorded into the ledger.

– Invoking Models

After the ***MD*** and the ***AG*** has installed model_CC and the ***MD*** has recorded the relevant information of the model into the ledger, the ***CR*** can start to select the desired model to add flavor to the artwork. In this regard, relevant information about the models has been listed for the ***CR*** to choose from. We further refer to the entire process from selecting a model to obtaining inference results by the ***CR*** as Invoking Models where the general workflow has been shown in Figure 7.

As shown in Figure 7, the ***AG*** will make a request for the corresponding address of the model image which is accordingly selected by the ***CR***, and the container of applications created. The ***AG*** is accountable for providing the authentication token to the Docker Registry (in the case of private repositories) as the authentication for access when necessary, that is to say, 3A_ledger or log_ledger is not required. It is essential to reiterate once again that the processes or services of containers are autonomous among each other and multiple applications can be in progress simultaneously. The service waits for input to make inferences after the container is successfully run up. The interface between input, output, and model itself is an important design. The invokeModel function in model_CC is to interact with the model whose detailed process is included in the bottom of Algorithm 4 and is explained in the following section. As soon as the model creates the outputs, wallet_CC is automatically executed to give the bonus to the ***MD***. Finally, the artifact is returned to the ***CR*** and can be fed into the Flow of Uploading Artworks (FUA).

The detailed algorithm for invoking the models is depicted in Algorithm 4. Initially, the ***AG*** is provided with the artwork created by the ***CR*** and the model to be put into effect. The artwork may be various multimedia data such as images or audio, which is indicated by W; additionally, the name (N_m) and the developer (MD_m) of the model are also provided. For the reason that the asset stored in the ledger in the HLF is comprised of a key-value pair, the composition key combined with N_m and MD_m is set to represent the unique key, which is called Com_{key} (line 1).The ***AG*** queries the reference address of the model image U_I from the ledger according to Com_{key} and accesses the image (line 2-3). The aforementioned container-based model service awaits the input to create or decorate artifacts once the image is activated. In line 6, invokeModel function acts on main communication with the service. The service interface along with the input retrieved from the ***CR*** are passed as arguments to invokeModel, and invokeModel then sends a request to the service via the service interface to get the response (line 6, 12-15). As an illustration, the transfer style model mentioned in the previous section could be integrated with some web application tools (e.g., Flask). After the virtualized service is available, it acts as the backend server providing the external API and expects to be invoked. N_m and MD_m are then passed on to wallet_CC when the service successfully returns the result (line 8). As a result, ***AG*** obtains the predetermined premium R_{md}.

In this section, an example of invoking the model has been described. It is assumed that the model N_m is what has been mentioned in the previous section (i.e., the real-time and super-resolution style transfer model). The ***Creator*** hopes to use this model to convert the artwork "Tulips" which is shown in Figure 8 with a size of 663KB into a mosaic style artwork. The ***Art Gallery*** invokes model_CC after receiving the request for invoking the model from the ***Creator*** in order to get the address of the model U_I = https://hub.docker.com/repository/docker/johnson/fast_neural_style. Thereafter, the ***AG*** starts up the container application and passes W as a parameter to call the API of the model. As soon as the model converts the original "Tulips" artwork into the mosaic style, the artifact is outputted, as shown in Figure 8(b).

Algorithm 4: Invoking Models Process

```
Input:
$MD_m$: $\boldsymbol{MD}$ of the invoking model;
$N_m$: The name of the invoking model;
$W$: The artwork which includes metadata such as price, image, name, etc;
$R_{md}$: The premium for $MD_m$;
$access_t$: Access token for validating with Docker Registry;
Output:
$A_m$: The artifact outputed from the model;
$ret$: Returned result;
Init:
$ret$ = error;
Variable:
$U_I$: Source address of the model on Docker Registry;
$I_m$: Docker image of the model;
$C_I$: Container run upon $I_m$;
$Com_{key}$: The composite key of the model;
1  $Com_{key} = MD_m + N_m$;
2  $U_I = model\_CC.query(Com_{key})$;
3  $I_m = GET(U_I, access_t)$;
4  if $I_m$ then
5      $C_I = RUN(I_m)$;
6      $A_m = model\_CC.invokeModel(C_I, W)$;
7      if $A_m$ then
8          $wallet\_CC.update(MD_m, R_{md})$;
9          $ret = success$;
10         Reutrn $ret, A_m$;
11 Return $ret$, [];
12 Function invokeModel(api, metadata):
13     $m = (map[string]string)$;
14     $m["mt"] = metadata$;
15     $resp, err = POST(api, Json(m))$;
16     if $err$ then
17         Return $NULL$;
18     Return $resp$;
```

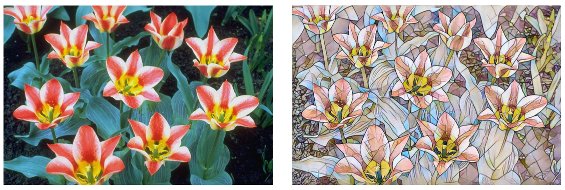
(a) Orginal Tulips Artwork (b) Mosaic Style Tulips Artwork

FIGURE 8. Illustration of invoking style transfer model.

F. PROCESS OF DATA REQUESTS

Taking into account the GDPR, the DS is given the Right of Data Portability. Consequently, the ***TP*** can request structural data from the ***CR*** or the ***CL*** by providing services or benefits (e.g., data monetization). The process of how the TP obtains data stored in orbitdb is described in this section. The Right of Erasure which is also clearly stipulated in the GDPR should be considered at the same time. That is, the ***CR*** and the ***CL*** are entitled to make a request to expunge the data at any time in the case the consent has been revoked or personal data has been collected unlawfully by the ***AG***. As per the guidelines of GDPR [46], the combination of identifiers is considered personal data or personal identifiers. Personal identifier data can include first and last names, ID numbers, information about a person's location, an online identifier factor, and so on. Three processes involved in the data request are set forth in the following, namely Request Consent, Request Data, and Revoke Consent.

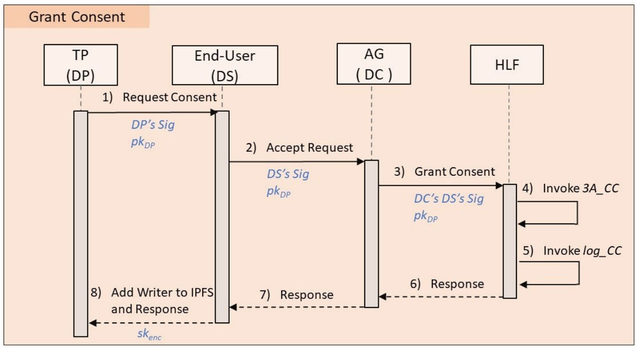

FIGURE 9. Process of granting consent from DS.

- Request Consent

  The interaction diagram for the ***TP*** to obtain the data stored in an off-chain storage is shown in Figure 9. The general process is similar to that of the ***AG*** getting the consent from the DS. The difference lies in that the snapshot of assets in 3A_ledger are updated instead of being created. In other words, the policy $\mathbb{P}$ to be granted is updated in 3A_ledger (step 4), and a logging is added to log_ledger afterward (step 5). Furthermore, when all the ledgers have been recorded, the DS not only writes the permission of the ***TP*** to the IPFS but also returns the private key used to decrypt the address of orbitdb to the ***TP*** (sk_{enc}, step 8). sk_{enc} which comes in handy during the process of requesting data.
- Request Data

  Upon successfully obtaining the consent of the DS, the ***TP*** can make a request for obtaining data. The process of data being requested from the DS is shown in Figure 10. This request is authenticated and validated by 3A_CC and log_CC. As soon as the authentication is successful, the encrypted address of orbitdb, as well as the private key to decrypt the data are returned to the ***TP***. Since the ***TP***s are not involved in the blockchain network, the authentication is initiated by the ***AG***. As stated in Figure 10, if pk_{TP} is stored in 3A_ledger and has a corresponding policy record, *enhash* is provided and subsequently, log_ledger is

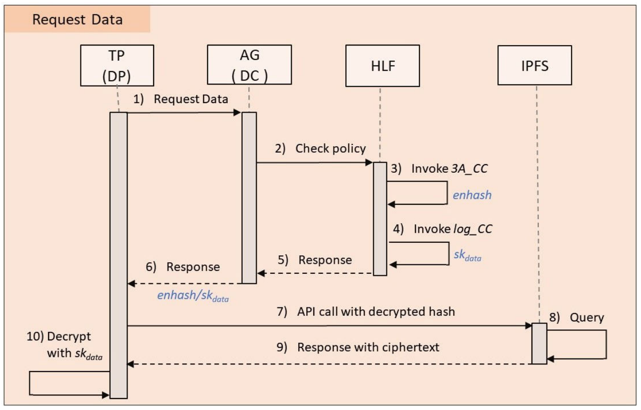

FIGURE 10. Process of requesting data from DS.

invoked. (step 3-4). log_ledger is notified to validate the operation and provide sk_{data} to the ***TP***. At this point, sk_{enc} which has already been obtained during the process of granting consent comes in handy for decrypting *enhash*. After *enhash* is decrypted, the ciphertext can be accessed through an API call (step 7-8). Originally in [18], a double-check with validating *acess_token* was done at this point. But thanks to the off-chain authorization, the token validation is no longer needed. *acess_token* is substituted by the authorization of orbitdb and data encryption, which is easier and intuitive. After the ciphertext has responded, the plaintext is obtained by decrypting them with sk_{data} (step 9-10).

- Revoke Consent

  The process of revoking the consent given by the DS is shown in Algorithm 5. When the DS wants to withdraw the consent, there are roughly three steps: (1) 3A_CC is called and information returned about the access permissions of the DP, and then these permissions are updated according to the operation issued by the DS (line 3-4). (2) log_CC is then invoked to record the operation (line 5). (3) In the end, the permission of the DP is correspondingly modified in orbitdb so that the DP can no longer access the data (line 6).

IV. EXPERIMENTS

In this section the experimental environment settings and reference indicators such as read throughput, write throughput, read latency, and write latency are described first, and then the experimental results of different functions in the SCs based on the reference indicators are elaborated. In addition to the results of the reference indicators, the resource consumptions of the containers in the blockchain network are also measured. Subsequently, some optimization steps such as changing the speed of the block generation and the maximum number of transactions allowed in a block are adopted so that the overall performance of SPChain can be raised. Finally, the differences with the previous works are sorted out and the compliance with the GDPR is reviewed, along with the review of the benefits from the AI model combination.

Algorithm 5: Revoke Consent Process

```
Input:
$pk_{DS}$: The public key of DS;
$pk_{DP}$: The public key of DP;
$pk_{AG}$: The public key of AG;
$Sig_{DS}$: The siganture of DS;
$Sig_{AG}$: The siganture of AG;
$op$: Operation toward DS's Data;
$\mathscr{V}$: The verifying signature algorithm;
Output:
$ret$: Returned result;
Init:
$ret$ = error;
Variable:
$t$: Current time;
$policy$: $\mathbb{P}$ defined in 3A_ledger;
$check$: Signature verify result;
1 $check = \mathscr{V}(pk_{DS}, Sig_{DS})$ && $\mathscr{V}(pk_{AG}, Sig_{AG})$;
2 if $check$ then
3     $policy = 3A\_CC.query(pk_{DS}, pk_{AG})$;
4     $3A\_CC.update(pk_{DS}, pk_{AG}, policy - op, t)$;
5     $log\_CC.update(pk_{DS}, pk_{AG}, pk_{DP}, t, op)$;
6     $orbitdb.removeWriter(pk_{DP})$;
7     $ret = success$;
8 Return $ret$;
```

TABLE 3. Environment of the blockchain network.

CPU	Intel(R) Core(TM) i7-3770 CPU @ 3.40GH
Memory	16 GB RAM
Operating System	Ubuntu 16.04 LTS (64-bit)
SC Language	Golang v1.12

A. EXPERIMENTAL ENVIRONMENT

The experimental environment of the blockchain network is shown in Table 3. The functions in artwork_CC, wallet_CC, model_CC, 3A_CC, log_CC are implemented in the Golang programming language version 1.12. The blockchain network is made up of two peers per organization, five orderers, and Fabric CA is built on the Ubuntu 16.04 LTS running on a CPU Intel(R) Core(TM) i7-3770 and 16 GB RAM.

B. EXPERIMENTAL INDICATORS AND FRAMEWORK

At present, although there are no unified indicators or metrics for evaluating the performance of the blockchain, most of the evaluation lies on the following indicators: read latency, read throughput, write latency, and write throughput. The four indicators depicted in the metrics worked by the Hyperledger [36] Performance and Scale Working Group (PSWG) are briefly introduced as follows:

- Read Latency

  Read latency is the time between when the read request is submitted and when the reply is received, as defined in Equation(5.1).

$$\begin{aligned} &\textit{Read Latency} \\ &\quad = \textit{Time when response received} - \textit{submit time} \end{aligned} \tag{1}$$

- Read Throughput
Read throughput is defined as the number of read requests completed in a certain time period and is expressed as reads per second (RPS), as defined in Equation(5.2).

$$\begin{aligned} &\textit{Read Throughput} \\ &\quad = \textit{Total read operations / total time in seconds} \end{aligned} \tag{2}$$

- Write Latency
Write latency is a network-wide view of the time taken for a submitted transaction across the network which includes the time of propagation and consensus, as defined in Equation(5.3).

$$\begin{aligned} &\textit{Write Latency} \\ &\quad = (\textit{Confirmation time @ network threshold}\ ) \\ &\quad - \textit{submit time} \end{aligned} \tag{3}$$

- Write Throughput
Write throughput is defined as the number of valid write transactions committed in a certain time period and is expressed as transactions per second (TPS). Note that it is not the rate at a single node but across the entire System Under Test (which in SPChain is Fabric). Write throughput is defined in Equation(5.4).

$$\begin{aligned} &\textit{Write Throughput} \\ &\quad = \textit{Total committed transactions} \\ &\quad /\textit{total time in seconds @ committed nodes} \end{aligned} \tag{4}$$

The evaluation of blockchain applications can be cumbersome, mainly because of the complex and resource-heavy distributed network. As a result, the Hyperledger Caliper [37] has been used as our testing framework for evaluating the aforementioned indicators. The Hyperledger Caliper is a benchmark tool aimed at serving different platforms of blockchain and is aimed at assisting users to measure performance of a System Under Test (SUT) while executing the smart contracts in terms of throughput, latency, and resource consumption. The high-level architecture of Caliper is shown in Figure 11. There are three input files for Caliper, including the workload module that defines user behavior or processes logic, the benchmark configuration that defines the internal interaction with the SC, and the network configuration that represents the network topology. The monitoring information is compiled into a complete report containing throughput, latency, success rates, resource consumption, and so on.

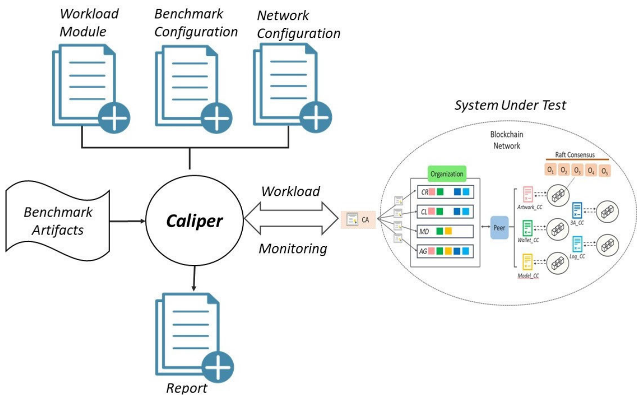

FIGURE 11. Bird's-eye view of hyperledger caliper [37].

TABLE 4. Summarisation of functions in SCs.

SC	R/W	Function	Description
artwork_CC	R	*readArtwork()*	Read the artwork provenance.
		getHistoryForArtwork()	Read history of the artwork provenance.
	W	*uploadArtwork()*	Add the artwork into ledger.
		transferArtwork()	Transfer the owner of the artwork.
model_CC	R	*readModel()*	Read the model information.
		invokeModel()	Invoke model API.
	W	*addModel()*	Add the model into ledger.
wallet_CC	R	*readBalance()*	Read the balance of a wallet.
	W	*addWallet()*	Add the wallet into ledger.
		transferBalance()	Transfer balance among wallets.
3A_CC	R	*readConsent()*	Read the consent.
	W	*initialConsent()*	Initial the consent in the initial phase.
		grantRevokeConsent()	Grant or revoke the consent.
log_CC	R	*readLog()*	Read the log ledger.
	W	*addLog()*	Add the log into ledger.

C. EXPERIMENTAL RESULTS

In this work, experiments were conducted with the read latency, write latency, read throughput, and write throughput of the functions in each smart contract. At the same time, the resource consumption of peers and orderers in the containers was monitored. The functions and relative descriptions of each round of tests have been summarized in Table 4. Among these functions, two major categories can be identified, namely querying the ledger (R) and updating the ledger (W). Caliper was used to continuously send transaction requests to these functions using a fixed-backlog schedule which aims to maintain a defined backlog of transactions within the system by modifying the driven TPS. Notably, invokeModel was tested with a fast style transfer [38] model from whose architecture was proposed by [35]. A 664 KB input image was fed into the model and then transferred to a mosaic style image. Subsequent to obtaining the various indicator results with preset settings, techniques such as adjusting different block generation speeds and increasing the maximum transaction volume in a block were introduced to improve performance. Lastly, the latency of different numbers of orderers under the Raft consensus algorithm was experimented with during the normal operation of the blockchain network.

D. PERFORMANCE ON USING DEFAULT BATCH TIMEOUT AND MAX MESSAGE COUNT

With the default settings, BatchTimeout (BT) which represents the time to wait for another transaction after the first

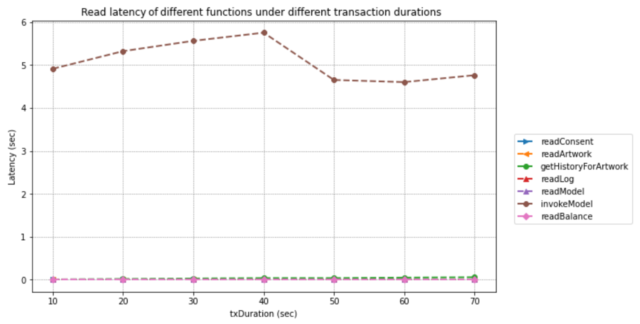

FIGURE 12. Read latency of different functions under default settings.

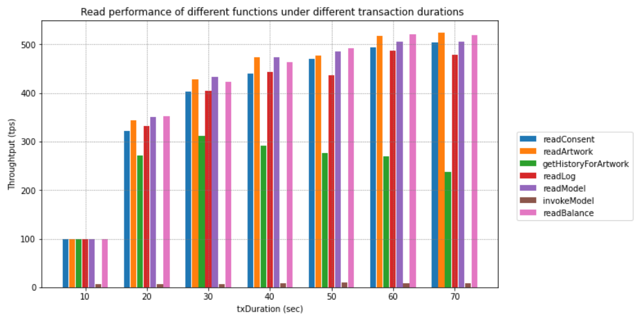

FIGURE 14. Read throughput of different functions under default settings.

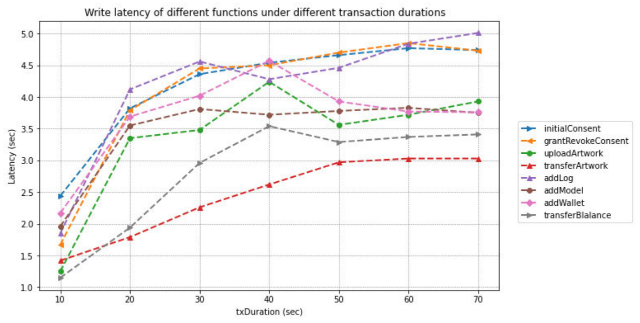

FIGURE 13. Write latency of different functions under default settings.

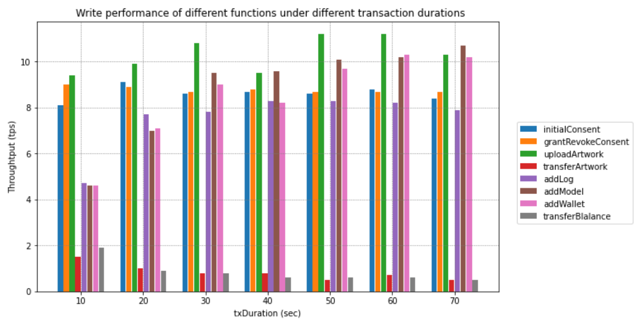

FIGURE 15. Write throughput of different functions under default settings.

transaction has arrived before cutting into blocks in the network is set to the default configuration in Hyperledger Fabric (i.e., 2 seconds). Meanwhile, MaxMessageCount (MMC) which defines the number of transactions that can be accommodated in a single block is set to the default configuration in the Hyperledger Fabric (i.e., 10 transactions per block). The time of sending the transaction (i.e., txDuration) for each function summarized above was increased from 10s to 70s and the latency and the TPS observed. The duration of sending transactions was then determined based on the results of the experiment. It was observed that there was no greater increase in throughput when txDuration was higher than 70 seconds, which means that the number of sent transactions had reached saturation. The latency of read and write are shown in Figures 12 and Figure 13, respectively. It can be seen that the read latency of each function is almost equal to 0s, no matter how long the transaction requests lasted except invokeModel for those whose latency ranged from 4s to 6s. Most of this time was spent on the inference of the model. As for the write latency, the time rose slowly according to txDuration. Finally, the latency of each function ranged from 3s and 5s when txDuration lasted for 70s.

The read and write throughputs measured under the default settings are shown in Figure 14 and Figure 15, respectively. For the read request, the throughput was seen to improve from 100 TPS to up to 500 TPS when txDuration was gradually increased. It is worth noting that there were two functions with lower throughput, namely invokeModel and getHistoryForArtwork. The reason why invokeModel reached only about 10 TPS, as has been described in the above section, is that most of the time was spent on the inference. For getHistoryForArtwork, the reason for reaching fewer TPS was due to the larger amount of data stored in the ledger (which is the whole transaction history of the artwork). Before these read requests were executed, different assets (e.g., consent, artwork, wallet, log, model) were written into the ledger in the beginning in order to avoid any failure in reading the corresponding assets and erroneously affecting the experimental latency and throughput. Consequently, getHistoryForArtwork read all the historical changes to the artwork, which reduced the throughput.

In terms of write, the throughput of various functions was also seen to improve with the increase of txDuration as shown in Figure 15. Nevertheless, once txDuration exceeded 50s, TPS did not increase much, which was due to the bottleneck of the hardware resources under the default settings. Also, it was observed that there were two functions with very poor throughput, namely transferArtwork and transferBalance. The main reason for this was the problem of synchronization. The ledger was activated to use lock-free optimistic concurrency with rollback in case of dirty read/writes so that multiple workers needed to wait for each other in the Caliper.

E. PERFORMANCE ON ADJUSTING MAX MESSAGE COUNT

In general, throughput can be enhanced by increasing MMC at the expense of latency. The reason for higher latency is that when MMC increases, the time it takes for the orderer to package transactions into blocks is expected to become longer. In this section, the performance for different values of MaxMessageCount is compared with the default setting of MMC = 10 transactions per block. As shown in Figure 16 and Figure 17 respectively, the read and write

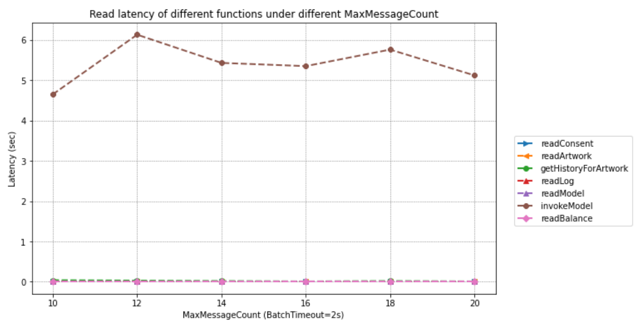

FIGURE 16. Read latency of different functions under different MaxMessageCount.

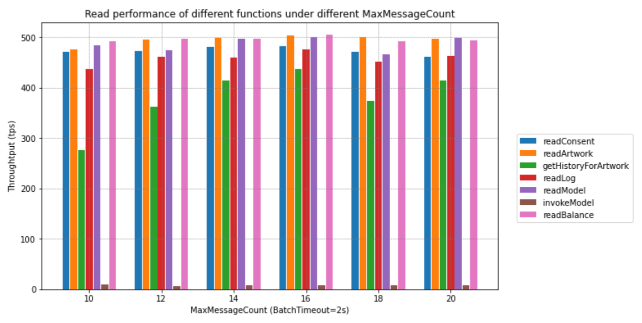

FIGURE 18. Read throughput of different functions under different max message count.

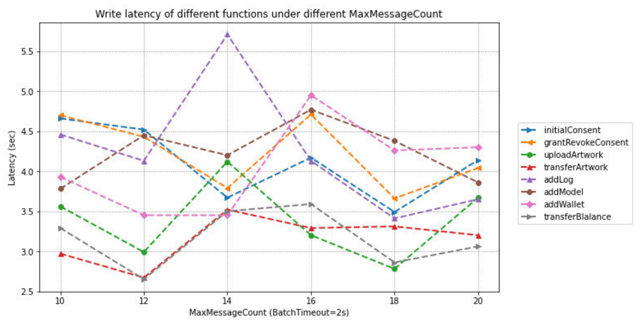

FIGURE 17. Write latency of different functions under different MaxMessageCount.

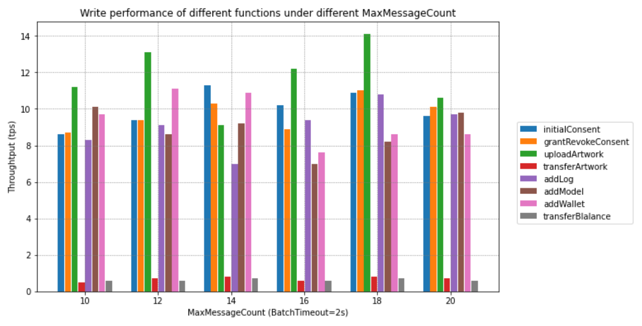

FIGURE 19. Write throughput of different functions under different max message count.

latencies were configured for 10 to 20 MMC with an increase of two transactions per block each step. The read latencies of the different functions were negligibly close to 0, while the latency of the function invokeModel depended largely on the model itself, which was around 4 to 6 seconds. As for the experimental results of write latency, the write delays did not increase with MaxMessageCount, as otherwise expected theoretically. It was observed that when MMC = 18, the write latencies of most functions were almost lower than the preset settings. It can be speculated that the reason behind this behavior is what Thakkar et al. [39] had observed, that is, if the transactions arrival rate reaches a saturation point, the latency decreases as the MaxMessageCount increases. In this case, the time of verification and committing of a block with size n is less than the time for verifying and committing of m blocks with size n/m of each block. As a result, the write latency of each function was reduced roughly by 0.5 second to 1 second when compared to the default setting of MMC=10.

The result of the read and write throughputs under different values of MMC are shown in Figure 18 and Figure 19, respectively. It can be observed that there was not much fluctuation in the read throughput compared to the default setting (i.e., MMC = 10) when MMC was increased to 20, except for the increase from approximately 280 TPS to 400 TPS in the case of the function getHistoryForArtwork. The reason for the increase was that the read operation did not require endorsement policy and block generation. With regard to the write throughput, there was an average increase of 2 to 3 TPS when MMC = 18. Since a lower latency and higher throughput were achieved when MMC was set to 18, this MMC value will be used. BatchTimeout will be further adjusted in the next section to achieve better performance.

F. PERFORMANCE ON ADJUSTING BATCH TIMEOUT

In general, raising the BatchTimeout can increase the number of transactions placed in a block, with the sacrifice of increasing the interval generation time among blocks, that is latency. Under the conditions of a fixed transaction duration of 50 seconds and MaxMessageCount of 18 transactions per block, experiments were conducted by increasing BatchTimeout every 0.5s, ranging from 2s to 5s. The experimental results of read and write are as shown in Figure 20 and Figure 21, respectively. In terms of read latency, latency was seen to increase slightly because of the increase in the number of transactions contained in the block, whereas it was still maintained at around 0s. Similar to adjusting MMC, the result of write latency was not in line with the general expectation. The latency of each function from 2s to 5s slightly fluctuated but the throughput was indeed seen to improve which is depicted later in the next subsection. The reason may be as guessed in the experiment of adjusting MMC. When BatchTimeout becomes higher, the possibility that a block stores up to 18 transactions becomes more and more likely. In other words, a fewer amount of blocks is conducive to decreasing latency.

The experimental results of read and write throughput adjusted according to different BatchTimeout values are respectively shown in Figure 22 and Figure 23. Similar to the adjustment of MMC values, read did not generate blocks so that the throughput was between 400 TPS and 500 TPS.

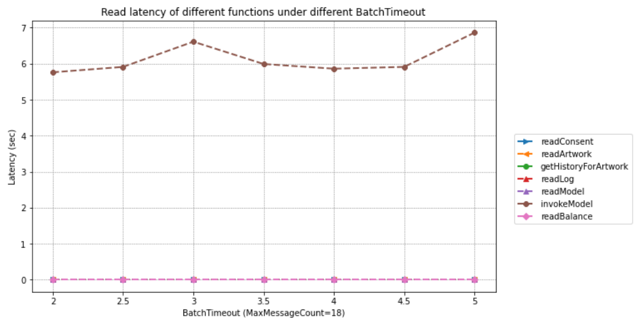

FIGURE 20. Read latency of different functions under different BatchTimeout.

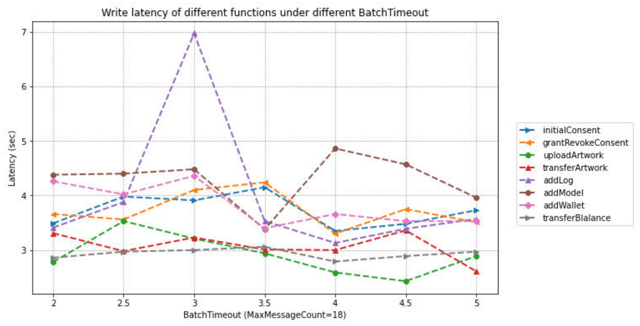

FIGURE 21. Write latency of different functions under different BatchTimeout.

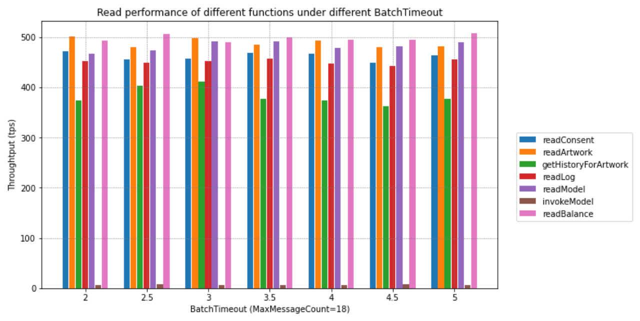

FIGURE 22. Read throughput of different functions under different BatchTimeout.

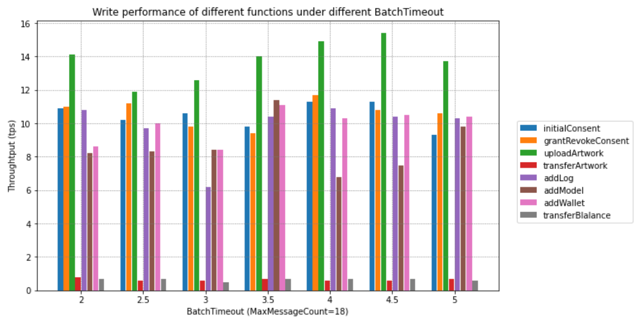

FIGURE 23. Write throughput of different functions under different BatchTimeout.

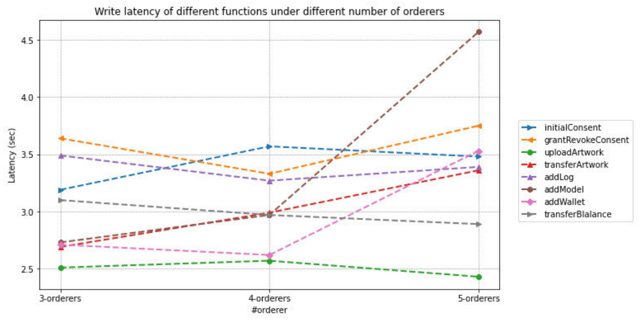

FIGURE 24. Write latency of different functions under different numbers of replicas.

On the other hand, write throughput was improved due to what has been mentioned in the previous subsection. Each block had a better chance of accommodating the maximum number of transactions. Therefore, it can be observed that when the value of BT was 4.5s, the overall increments were about 1 to 2 TPS in the end, where most of the functions achieved 10 to 12 TPS, except the highest 16 TPS of uploadArtwork. Functions transferArtwork and transferBalance still exhibited low TPS due to the lock-free optimistic concurrency with rollback in the ledger.

G. LATENCY ON VARYING NUMBER OF REPLICAS

In this section, the delays caused by different numbers of replicas in the Raft consensus will be compared. As described in Section III-C, there are preset five orderers in HLF for achieving the maximum fault tolerance rate of two malicious or faulty nodes. In Raft, a leader node is elected firstly, which is responsible for all communications with the outside environment. Through the replicated state machine, each orderer executes the received commands in the same sequence to obtain the consensus result. Accordingly, the more replicas there are, the higher the latency expected. The experiment results of write latency under three to five orderers are shown in Figure 24. In the initial round of five orderers, the delay of each function was around 2.5s to 4.5s. After that, the leader orderer of each round was stopped until three orderers were left and it was observed that the delay dropped to a range of 2.5s and 3.6s while maintaining crash fault tolerance. To summarize, different numbers of replicas needed to be adjusted to achieve a suitable network delay under different resources environment and consumption.

V. DISCUSSIONS

In this section, a discussion is presented about whether the architecture of SPChain complies with the six guidelines and three rights in the GDPR, followed by a discussion on the benefits of combining AI models with DAM. Eventually, a comprehensive comparison is conducted with other existing methods, including the GDPR compliance, smart contract design, decentralized off-chain storage, performance evaluation, and combining with AI models.

A. GDPR COMPLIANCE

The verification of six principles and three rights with respect to the DS have been summarized in Table 5. Among them, all relevant rules related to the deletion of the existing personal data stored in orbitdb (i.e., Accuracy, Storage limitation, Right to rectification, and Right to erasure) are discussed in

TABLE 5. Verification of the GDPR principle and right.

Article	Principle / Right	Verification
5.1 Lit.(a)	Lawfulness, fairness and transparency	Comply. Transactions in the consortium blockchain achieved a pseudonymous but transparent principle. On the other hand, digial assets stored in orbitdb are also transparent to the DS.
5.1 Lit.(b)	Purpose limitation	Comply. The process of the data usage is recorded on the tamper-resistant blockchain. SP acts as a delegate to provide services such as uploading artwork, transferring artwork through these data.
5.1 Lit.(c)	Data minimization	Comply. After granting consent by the DS, the SP only needs the minimal data to provide services.
5.1 Lit.(d)	Accuracy	Comply. Digital assets stored in the off-chain orbitdb can be updated or erased immediately where the blockchain only stored the hash of address and provenance.
5.1 Lit.(e)	Storage limitation	Comply. After the processing of personal data has elapsed, the orbitdb can be erased where the hash stored in the blockchain becomes invalid.
5.1 Lit.(f)	Integrity and confidentially	Comply. Integrity is achieved by leveraging the blockchain's tamper proof property and also, the decentralized orbitdb. On the other hand, because of the orginal crypto mechanism in blockchain and asymmetric encryption of plaintexts stored in orbitdb, as well as, hash address, data is preserved confidentially.
16	Right to rectification	Comply. The DS is entitled to request the DC for correct data, as well as the transaction process because the DS participates in both blockchain and orbitdb.
17	Right to erasure	Comply. Digital assets as well as the metadata stored in orbitdb can be erased in case the consent is withdrawn by the DS or illegally processed by the DC.
20	Right to data portability	Comply. As stated in Section III-F, a TP can request the data stored in orbitdb in a machine-readable format after requesting consent from the DS.

more detail. In essence, orbitdb has higher fault tolerance than traditional centralized databases, but ''all'' the data cannot be actively deleted under the IPFS protocol. Technically, the nodes that request data from a hash address of orbitdb replicates data which is actively cleared by the IPFS if there is no pin or usage. That is why the address of orbitdb and the personal data was encrypted. Consequently, only the nodes of specific data subjects, service provider, and third-party would know the actual hash address of orbitdb and the plaintexts (which is more like a private orbitdb). When the participating nodes are in compliance with GDPR and try to delete data stored in orbitdb, the new node would not be able to synchronize the data since it cannot find any online nodes (in the case of obtaining the hash address of orbitdb). Also, the key-pairs used for encryption are deleted by DS so that the erasure of data is achieved.

B. BENEFITS OF COMBINING AI MODELS

The benefits of combining digital assets and AI models are explored in this section. As mentioned in Section III-E, the ***CR*** can invoke the models developed by the ***MD*** to enhance the creativity of the created artwork. The design at this stage is often not a single process in most realistic scenarios, which is illustrated in Figure 25. For example, the ***CR*** may first obtain the artwork through a generating model and then perform other kinds of processing on the artwork (e.g., noise removal, style transfer). These processes can be executed in a separate container in an orderly manner without interference and moreover, the messages between each process can be transmitted to each other until a satisfactory artifact is generated. Technically, these processes are the workflows of containers where several engines can define these workflows such as Argo, KubeFlow, etc. Such a combination eliminates the need for additional model marketing, and is also oriented to more intelligent and automated applications. Even the ***AG*** can use AI models such as recommendation systems or

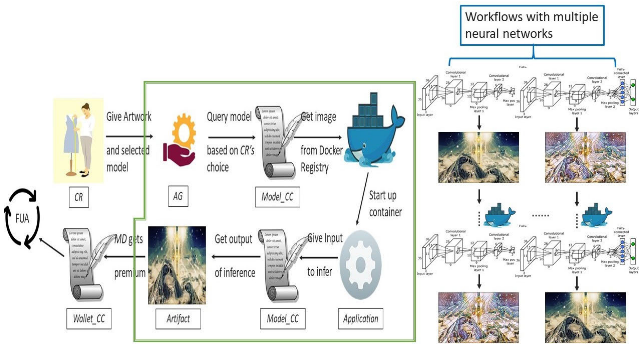
Workflows with multiple neural networks
Give Artwork and selected model
Query model based on CR's choice
Get image from Docker Registry
Start up container
Give Input to infer
Get output of inference
MD gets premium
FUA
CR
AG
Model_CC
Wallet_CC
Artifact
Model_CC
Application

FIGURE 25. Illustration of multiple model combinations.

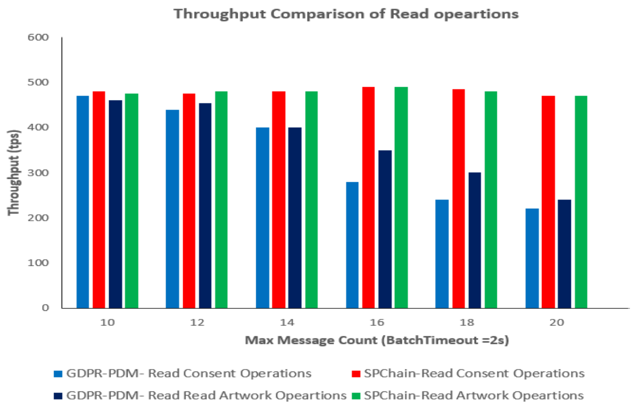

FIGURE 26. Read comparison with existing work- GDPR-PDM.

recommended pricing to increase the opportunity for trading. It is worth noting that although artwork images were used as an illustration, other digital assets like videos or audios can also be used to convey the same idea.

C. OVERALL COMPARISON WITH EXISTING WORKS

The comparison with existing works is shown in Table 6, which can be classified into five discussions, that is (1) Whether the GDPR is complied with (2) Whether there is a detailed smart contract design (3) Whether there is a decentralized Off-Chain Storage under the GDPR (4) Whether there is practical implementation and a complete experimental evaluation and (5) Whether it is combined with AI models. As Table 6 reveals, except for [26], other works are in compliance with GDPR in terms of design, but only [18] and the proposed SPChain has in-depth experimental performance evaluations.

The proposed Spchain is compared with the previous framework GDPR-PDM [16] in terms of throughput. Read Consent, write consent operations, and writer artwork operations are shown in Figures 26 and 27. When the MMC setting is increased to 20, the read and write throughput increases compared to the default setting of MMC = 10. The throughput of the digital transactions of the proposed Spchain compared to the previous work read and write transactions is increased in Max Message Count. Performance is one of the most important and cumbersome considerations when the design is applied in real scenarios for blockchain. Despite

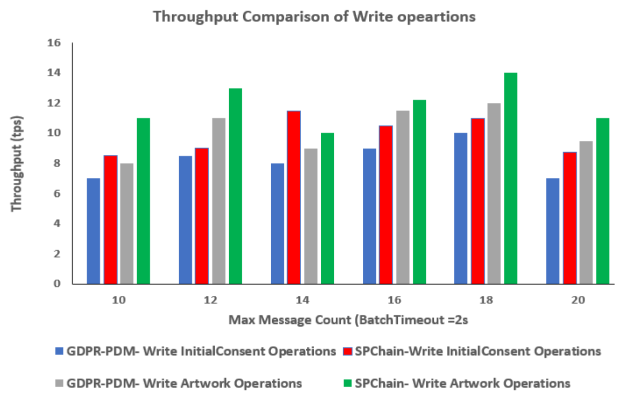

FIGURE 27. **Write comparison with existing work- GDPR-PDM.**

TABLE 6. **Comparison with existing works.**

	[18]	[16]	[26]	[27]	[28]	**SPChain**
GDPR Compliance	✓	✓	–	✓	✓	✓
SC Design	✓	✓	X	✓	X	✓
Decentralized Off-Chain Storage	–	–	X	X	–	✓
Performance Evaluation	✓	X	X	X	X	✓
Combining with AI models	X	X	X	X	X	✓

"✓" means addressed, "–" means no proof of concept, and "X" means did not address

some basic experimental results such as read throughput, read latency, write throughput, and write latency of smart contracts in [18], some performance adjustments were not taken into account. In addition, the reasons behind the results were not adequate and did not provide in-depth explanations, as has been provided in this work in the previous section. Additionally, the smart contract design is also a vital consideration when applying the blockchain to real scenarios. Reference [18] focuses on the smart contract design of 3A_ledger and log_ledger, where [16], [27] depicts smart contracts for storing and accessing encrypted personal data. In contrast, SPChain has designed five smart contracts for managing digital assets, GDPR-compliance, and AI model combinations. In terms of decentralized Off-Chain Storage, [16], [18], [28] had taken the concept into consideration, but it was not implemented. In comparison, a decentralized orbitdb was adopted with crypto artworks management in this proposed SPChain so that the issue of a single point of failure on off-chain could be resolved and brought closer to true decentralization. As far as combination with AI models is concerned, the concept introduced in this proposed SPChain is a novel one and combines DAM (or PDM) with AI models so as to not only eliminate the maintenance costs for individual platforms of blockchain and AI model marketplace, but also to orient towards more intelligent applications.

VI. CONCLUSION

In this work, SPChain was proposed as a means of achieving a secure and GDPR-compliant digital asset management framework by leveraging the security and non-tampering features of blockchain. At the same time, AI models were combined to make digital assets more accessible to a larger number of applications and to enable better creativity. SPChain can be credited with three main contributions, including (1) A thorough design for securely and privately managing the digital assets with a digital artwork used for a complete illustration (2) The adoption of decentralized Off-Chain Storage (i.e., IPFS) to replace traditional centralized storage for avoiding the problem of single point of failure and (3) The combination of digital assets and AI models via virtualized container technology. The GDPR regulations that are in conflict with the principles of the blockchain such as Accuracy, Storage limitation, Right to erasure, etc., have been resolved by storing the private data in the IPFS, with the corresponding encrypted hash values stored in the secure blockchain ledger. Moreover, different SCs have been designed to solve different security and privacy issues. For example, artwork_CC was used to ensure the provenance of the digital art and the transfer process, wallet_CC was called to clear the balance when monetary flow was involved, model_CC was responsible for recording relevant information of the uploaded model and as a proof for the ***Model Developer*** to obtain the premium when uploaded model was invoked by a ***Creator***, and 3A_CC as well as log_CC proposed by [18] were modified to combine with the decentralized IPFS.

As for combining with AI models, different models can be invoked by a ***CR*** to create new artifacts of artwork which are more smart and diversified. Because of the use of virtualized independent environments, the problem of the outputs of the model being inconsistent in the heterogeneous environment was resolved. Also, multiple models could make inferences at the same time without interfering with each other. In the experimental results, the performance was initially measured, such as the read latency, write latency, read throughput, and write throughput of different functions in the smart contracts under different transaction durations ranging from 10s to 70s. The write throughput was about 8 to 11 TPS when the transaction duration was 50s, except for transferArtwork and transferBalance which required more time for waiting for the synchronization amongst multiple workers in the Hyperledger Caliper; and the read throughput reached about 500 TPS when the transaction duration was 70s, except for getHistroyForArtwork which had a larger amount of data. After the adjustments of MaxMessageCount and BatchTimeout, the write throughput increased to between 10 and 15 TPS when the transaction duration was 50s, MaxMessageCount was 18 and BatchTimeout was 4.5s. Under this configuration, the latency was almost the same as that for the default setting where the read throughput was still maintained at around 500 TPS. In conclusion, SPChain can be attributed as a solution for providing a secure, GDPR-compliant, and intelligent digital assets management method, enabling the data owner with more transparency and control towards their data.

For future works, three directions for further study can be considered. First, when more stakeholders or organizations are involved in the system, the design configuration of the blockchain network can be expanded and adjusted correspondingly. For instance, channel configurations based on the number of CPUs, effective optimization for the validation of the SCs, and the selection of an appropriate ledger database behind the snapshot. These detailed guidelines can be found in [39]. For the second direction, the proposed SPChain can be used to manage different digital assets, such as images, audio, personal data, etc. Taking audio files as an example, the sizes of audio files are often larger than images, resulting in increasing the latency. Therefore, optimizing the performance of orbitdb becomes one of the crucial issues; similarly, in terms of combining AI models, the inference time that outputs the artifacts (e.g, music creation, specific noise spectrum removal) should be reduced. The third direction is that when multiple models or steps are connected in a sequence as a pipeline (i.e., workflow), a larger number of workflows will consume huge resources, leading to difficulties in container deployment and management. Some automatic deployment tools such as Kubernetes [40] can be adapted to overcome the problem. Additionally, the security of AI models should be protected to avoid common attacks such as evasion, poisoning, and stealing, which means the attackers may influence the inference results or extract relevant information of the model by adversarial perturbation and continuously supplying input to the model. In this case, the model under attack may be used to generate the same digital asset (i.e., model reproduction). In summary, the foundation has been laid for the secure and private management of digital assets [48] and the combination with AI models. Many directions under different scenarios can be investigated further based upon this foundation.

REFERENCES

[1] (Sep. 2020). *Digital Media Revenue in Selected Countries Worldwide in 2020 (in Million U. S. Dollars)*. [Online]. Available: https://www.statista.com/statistics/459335/digital-media-revenue-countries-digital-market-outlook/

[2] (Jul. 2021). *Dapper*. [Online]. Available: https://www.dapperlabs.com/

[3] (Jul. 2021). *Pixura*. [Online]. Available: https://pixura.io/

[4] (Apr. 2020). *Cambridge Analytica and Facebook: The Scandal and the Fallout so Far*. [Online]. Available: https://www.nytimes.com/2018/04/04/us/politics/cambridge-analytica-scandal-fallout.html

[5] Y. Zhu, Y. Qin, Z. Zhou, X. Song, G. Liu, and W. C.-C. Chu, "Digital asset management with distributed permission over blockchain and attribute-based access control," in *Proc. IEEE Int. Conf. Services Comput. (SCC)*, Jul. 2018, pp. 193–200.

[6] H. R. Hasan and K. Salah, "Proof of delivery of digital assets using blockchain and smart contracts," *IEEE Access*, vol. 6, pp. 65439–65448, 2018.

[7] (Jan. 2021). *General Data Protection Regulation*. [Online]. Available: https://www.statista.com/statistics/459335/digital-mediarevenue-countries-digital-market-outlook/

[8] S. Xie, D. Hong-Ning, C. Xiangping, and W. Huaimin, "Lockchain challenges and opportunities: A survey," *Int. J. Web Grid Services*, vol. 14, pp. 352–375, Jan. 2018.

[9] M. A. Salek, M. Vecchio, M. Pincheira, K. Dolui, F. Antonelli, and M. M. Husain, "Applications of blockchains in the Internet of Things: A comprehensive survey," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 2, pp. 1676–1717, 2nd Quart., 2018.

[10] S. Singh and N. Singh, "BlockChain: Future of financial and cyber security," in *Proc. 2nd Int. Conf. Contemp. Comput. Informat. (IC3I)*, Dec. 2016, pp. 463–467.

[11] B. A. Tama, B. J. Kweka, Y. Park, and K.-H. Rhee, "A critical review of blockchain and its current applications," in *Proc. Int. Conf. Elect. Eng. Comput. Sci. (ICECOS)*, Aug. 2017, pp. 109–113.

[12] M. Franceschet, G. Colavizza, T. Smith, B. Finucane, M. Ostachowski, O. Lukas, S. Scalet, J. Perkins, J. Morgan, and S. Hernández, "Crypto art: A decentralized view," *Leonardos*, vol. 54, no. 4, pp. 109–113, 2020.

[13] L. van Haaften-Schick and A. Whitaker, "From the artist's contract to the blockchain ledger: New forms of artists' funding using equity and resale royalties," *SSRN Electron. J.*, vol. 2022, pp. 1–29, Jan. 2022.

[14] W. Gao, W. G. Hatcher, and W. Yu, "A survey of blockchain: Techniques, applications, and challenges," in *Proc. 27th Int. Conf. Comput. Commun. Netw. (ICCCN)*, Jul. 2018, pp. 1–11.

[15] L. Kiffer, D. Levin, and A. Mislove, "Analyzing Ethereum's contract topolog," in *Proc. 2nd Internet Meas. Conf.*, Oct. 2018, pp. 494–499.

[16] B. Faber, G. C. Michelet, N. Weidmann, R. R. Mukkamala, and R. Vatrapu, "BPDIMS: A blockchain-based personal data and identity management system," in *Proc. Annu. Hawaii Int. Conf. Syst. Sci.*, 2019, pp. 6855–6864.

[17] A. Bayle, M. Koscina, D. Manset, and O. Perez-Kempner, "When blockchain meets the right to be forgotten: Technology versus law in the healthcare industry," in *Proc. IEEE/WIC/ACM Int. Conf. Web Intell. (WI)*, Dec. 2018, pp. 788–792.

[18] N. B. Truong, K. Sun, G. M. Lee, and Y. Guo, "GDPR-compliant personal data management: A blockchain-based solution," *IEEE Trans. Inf. Forensics Security*, vol. 15, pp. 1746–1761, 2020.

[19] Z. Wang, J. Kuen, L. Ma, A. Shahroudy, B. Shuai, T. Liu, X. Wang, G. Wang, and J. Cai, "Recent advances in convolutional neural networks," *Pattern Recognit.*, vol. 77, pp. 354–377, Oct. 2017.

[20] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial networks," 2014, *arXiv:1406.2661*.

[21] A. Kumar, "Sketching an AI marketplace: Tech, economic, and regulatory aspects," *IEEE Access*, vol. 9, pp. 13761–13774, 2021.

[22] P. Dhamange, S. Soni, V. Sridhar, and S. Rao, "Market dynamics and regulation of a crowd-sourced AI marketplace," *IEEE Access*, vol. 10, pp. 54325–54335, 2022.

[23] G. Wood, "Ethereum: A secure decentralised generalised transaction ledger," *Ethereum Project Yellow Paper*, vol. 151, pp. 1–32, Apr. 2014.

[24] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, vol. 25, Dec. 2012, pp. 1097–1105.

[25] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Feb. 2016, pp. 770–778.

[26] X. Zheng, R. R. Mukkamala, R. Vatrapu, and J. Ordieres-Mere, "Blockchain-based personal health Data sharing system using cloud storage," in *Proc. IEEE 20th Int. Conf. e-Health Netw., Appl. Services (Healthcom)*, Sep. 2018, pp. 1–6.

[27] C. Wirth and M. Kolain, "Privacy by blockchain design: A blockchain-enabled GDPR-compliant approach for handling personal data," in *Proc. 1st ERCIM Blockchain Workshop, Eur. Soc. Socially Embedded Technol. (EUSSET)*, May 2018, pp. 1–7.

[28] M. J. M. Chowdhury, A. Colman, M. A. Kabir, J. Han, and P. Sarda, "Blockchain as a notarization service for data sharing with personal data store," in *Proc. 17th IEEE Int. Conf. Trust, Secur. Privacy Comput. Commun. (TrustCom)*, Aug. 2018, pp. 1330–1335.

[29] F. Zemler and M. Westner, "Blockchain and GDPR: Application scenarios and compliance requirements," in *Proc. Portland Int. Conf. Manage. Eng. Technol. (PICMET)*, Aug. 2019, pp. 1–8.

[30] J. B. Bernabe, J. L. Canovas, J. L. Hernandez-Ramos, R. T. Moreno, and A. Skarmeta, "Privacypreservingsolutions for blockchain: Review and challenges," *IEEE Access*, vol. 7, pp. 164908–164940, 2019.

[31] P. Ryan and R. Brennan, "Support for enhanced GDPR accountability with the common semantic model for ROPA (CSM-ROPA)," *Social Netw. Comput. Sci.*, vol. 3, no. 3, pp. 1–16, May 2022.

[32] M. Koutli, N. Theologou, A. Tryferidis, D. Tzovaras, A. Kagkini, D. Zandes, K. Karkaletsis, K. Kaggelides, J. A. Miralles, and V. Oravec, "Blockchain and GDPR: Application scenarios and compliance requirements," in *Proc. Portland Int. Conf. Manag. Eng. Technol. (PICMET)*, Aug. 2019, pp. 263–270.

[33] P. Hacker and J.-H. Passoth, "Varieties of AI explanations under the law. From the GDPR to the AIA, and beyond," in *Proc. Int. Workshop Extending Explainable AI Beyond Deep Models Classifiers*. Cham, Switzerland: Springer, 2022, pp. 343–373.

[34] D. Ongaro and J. Ousterhout, ''In search of an understandable consensus algorithm,'' in *Proc. USENIX Annu. Tech. Conf.*, 2014, pp. 305–319.

[35] J. Johnson, A. Alahi, and L. Fei-Fei, ''Perceptual losses for real-time style transfer and super-resolution,'' in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, May 2016, pp. 694–711.

[36] L. Foschini, A. Gavagna, G. Martuscelli, and R. Montanari, ''Hyperledger fabric blockchain: Chaincode performance analysis,'' in *Proc. IEEE Int. Conf. Commun. (ICC)*, Jun. 2020, pp. 1–6, doi: 10.1109/ICC40277.2020.9149080.

[37] (Jan. 2021). *Hyperledger Caliper*. [Online]. Available: https:/hyperledger.org/

[38] (Jan. 2021). *Fast Neural Style in PyTorch*. [Online]. Available: https://github.com/pytorch/examples/tree/master/fastneuralstyle

[39] P. Thakkar, S. Nathan, and B. Viswanathan, ''Performance benchmarking and optimizing hyperledger fabric blockchain platform,'' in *Proc. IEEE 26th Int. Symp. Modeling, Anal., Simulation Comput. Telecommun. Syst. (MASCOTS)*, Sep. 2018, pp. 264–276.

[40] *Kubernetes*. Accessed: Jul. 30, 2022. [Online]. Available: http://wwwkubernetes.io

[41] L. Kamal and R. Raj, ''Blockchain: A compendium on contemporary privacy preservation approaches and its manifestation in varied realms,'' in *Proc. Int. Conf. Hybrid Intell. Syst.* Sehore, India: Springer, Dec. 2021, pp. 517–529.

[42] C. B. Fernandez, T. Braud, and P. Hui, ''Implementing GDPR for mobile and ubiquitous computing,'' in *Proc. 23rd Annu. Int. Workshop Mobile Comput. Syst. Appl.*, Mar. 2022, pp. 1–15.

[43] V. Wylde, N. Rawindaran, J. Lawrence, R. Balasubramanian, E. Prakash, A. Jayal, I. Khan, C. Hewage, and J. Platts, ''Cybersecurity, data privacy and blockchain: A review,'' *Social Netw. Comput. Sci.*, vol. 3, no. 2, pp. 1–12, Mar. 2022.

[44] D. Liu, C. Huang, J. Ni, X. Lin, and X. S. Shen, ''Blockchain-cloud transparent data marketing: Consortium management and fairness,'' *IEEE Trans. Comput.*, vol. 71, no. 12, pp. 3322–3335, Dec. 2022.

[45] O. P. Stan and L. Miclea, ''New era for technology in healthcare powered by GDPR and blockchain,'' in *Proc. 6th Int. Conf. Advancements Med. Health Care Technol.* Cham, Switzerland: Springer, 2019, pp. 17–20.

[46] V. Diamantopoulou, A. Tsohou, and M. Karyda, ''From ISO/IEC, 27002:2013 information security controls to personal data protection controls: Guidelines for GDPR compliance,'' in *Computer Security*. Cham, Switzerland: Springer, 2019, pp. 238–257.

[47] Y. Wang, Z. Su, N. Zhang, J. Chen, X. Sun, Z. Ye, and Z. Zhou, ''SPDS: A secure and auditable private data sharing scheme for smart grid based on blockchain,'' *IEEE Trans. Ind. Informat.*, vol. 17, no. 11, pp. 7688–7699, Nov. 2021.

[48] Y. Wang, Z. Su, N. Zhang, R. Xing, D. Liu, T. H. Luan, and X. Shen, ''A survey on metaverse: Fundamentals, security, and privacy,'' *IEEE Commun. Surveys Tuts.*, early access, Sep. 7, 2022, doi: 10.1109/COMST.2022.3202047.

WEI-SHAN LEE received the M.S. degree from the Department of Computer Science and Information Engineering, National Chung Cheng University, Taiwan, in 2021. His research interests include blockchain and machine learning.

JOHN A (Member, IEEE) received the B.Tech. and M.Tech. degrees in computer science and engineering from Pondicherry University, India, and the Ph.D. degree in computer science and engineering from Manonmaniam Sundaranar University, India, in 2019. He is currently doing his postdoctoral research with the AI and Sustainable Development Research Labaratory, Department of Computer Science and Information Engineering, National Chung Cheng University, Minxiong, Taiwan. His research interests include real time applications, machine learning, data analysis and prediction, and spatial and temporal database.

HSIU-CHUN HSU received the master's degree in business administration from the National Chung Cheng University (CCU), Taiwan, where she is currently pursuing the Ph.D. degree with the Department of Information Management. She is also a Manager with the Research Center on Artificial Intelligence and Sustainability, CCU. She has vast experiences in international collaborations with universities and companies from India, Taiwan, and other countries. She has played a key role in the international development of CCU while at the Office of International Affair. Her research interests include qualitative analysis of big data on societal and human aspects of information systems, including sustainability development and smart city applications.

PAO-ANN HSIUNG (Senior Member, IEEE) received the Ph.D. degree in electrical engineering from the National Taiwan University, Taipei, Taiwan, in 1996. Since August 2007, he has been a Full Professor at the National Chung Cheng University (CCU), Taiwan. Previously, he was the Department Chair, the Dean of International Affairs at CCU, and the Director-General of the Intelligence Technologies Department, Chiayi City Government. Since August 2015, he has been the Director of the Research Center on AI and Sustainability, CCU. He is currently a Chief Information Officer and the Director of the Taiwan-India Joint Research Center on Artificial Intelligence, CCU. He has published more than 300 papers in international journals and conferences. His research interests include smart system design, deep learning, the AIoT, reconfigurable computing and system design, and smart city technology, such as smart traffic and smart grid, cognitiveradio architecture, system-on-chip design, and embedded real-time system design. He is a fellow of the IET, a Senior Member of the ACM, and a Life Member of the IICM. He was a recipient of the 2010 Outstanding Research Award given by the National Chung Cheng University. He was also a recipient of the 2004 Young Scholar Research Award given by the National Chung Cheng University to five young faculty members per year. He received the 2001 ACM Taipei Chapter Kuo-Tingli Young Researcher for his significant contributions to design automation of electronic systems. He received several advisor awards for best master theses, embedded system competitions, and RFID design competitions. He helped the Chiayi City to achieve 2018 TOP7 Smart City in the Intelligence Community Forum (ICF) World Wide Competition. He has also helped obtain huge financial budget in the amount of NT$ 300million dollars.