IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 15 February 2024, accepted 8 March 2024, date of publication 18 March 2024, date of current version 22 March 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3376509

RESEARCH ARTICLE

Trust-Free Blockchain Framework for AI-Generated Content Trading and Management in Metaverse

VU TUAN TRUONG, HUNG DUY LE, AND LONG BAO LE, (Fellow, IEEE)
Institut National de la Recherche Scientifique (INRS), University of Quebec, Montreal, QC H5A 1K6, Canada

Corresponding author: Long Bao Le (long.le@inrs.ca)

This work was supported in part by the Innovation for Defence Excellence and Security (IDEaS) Program from the Department of National Defence (DND) under Grant MN3-005.

ABSTRACT The rapid development of the metaverse and generative Artificial Intelligence (GAI) has led to the emergence of AI-Generated Content (AIGC). Unlike real-world products, AIGCs are represented as digital files, thus vulnerable to plagiarism and leakage on the Internet. In addition, the trading of AIGCs in the virtual world is prone to various trust issues between the involved participants. For example, some customers may try to avoid the payment after receiving the desired AIGC products, or the content sellers refuse to grant the products after obtaining the license fee. Existing digital asset management (DAM) systems often rely on a trusted third-party authority to mitigate these issues. However, this might lead to centralization problems such as the single-point-of-failure (SPoF) when the third parties are under attacks or being malicious. In this paper, we propose MetaTrade, a blockchain-empowered DAM framework that is designed to tackle these urgent trust issues, offering secured AIGC trading and management in the trustless metaverse environment. MetaTrade eliminates the role of the trusted third party, without requiring trust assumptions among participants. Numerical results show that MetaTrade offers higher performance and lower trading cost compared to existing platforms, while security analysis reveals that the framework is resilient against plagiarism, SPoF, and trust-related attacks. To showcase the feasibility of the design, a decentralized application (DApp) has been built on top of MetaTrade as a marketplace for metaverse AIGCs.

INDEX TERMS Metaverse, blockchain, AI-generated content (AIGC), digital asset management.

I. INTRODUCTION

In recent years, the metaverse [1] has gained significant attention as a virtual world enabled by the convergence of advanced technologies such as digital twins (DT) [2] and virtual/augmented reality (VR/AR) [3]. This new digital realm has sparked a tremendous demand for different forms of digital assets. To meet this demand, AI-generated content (AIGC) has emerged as a promising solution, enabling the automatic creation of massive virtual assets, surpassing the scale of traditional methods such as professional-generated content (PGC) and user-generated content (UGC) [4]. However, managing AIGCs would be more challenging than real-world assets since their associated information may be leaked on the Internet by malicious actors. As a type of digital asset, AIGC-based assets may lose their value quickly if they are accessible uncontrollably over the digital space, leading to a significant financial loss for the creators.

Various solutions have been employed to guarantee the copyright and ownership of content creators. One widely used technique is non-fungible token (NFT) [5], which secures the ownership of digital assets based on blockchain. If a digital asset is tokenized as an NFT, its unique identification is stored permanently on-chain in a smart contract, acting as a reliable proof of ownership. Therefore, the content owners can still prove that they are the unique owners of

The associate editor coordinating the review of this manuscript and approving it for publication was Kostas Kolomvatsos.

their NFT assets, even if other users also get access to the assets' source data. As a result, the value of NFT assets primarily stems from their uniqueness and historical record. This type of digital asset is referred as *uniqueness-based assets*. On the other hand, there is another type of digital asset called *access-based assets* or *license-based assets*, which are distributed to multiple customers through the sale of license. The value of access-based assets does not rely on uniqueness but rather on their functionality and usefulness in a specific environment. Therefore, NFT is not a feasible solution for access-based digital assets as they can be owned by multiple users.

To manage access-based AIGCs, traditional digital asset management (DAM) systems rely on a trusted intermediary to distribute the assets and its license to the customers. In particular, the asset's source data is often stored in a third-party *distribution channel* (e.g., cloud storage), while the license is distributed via a *license broker* [11]. Consequently, a significant proportion of profit must be shared with the intermediaries, leading to a higher fee for customers. Furthermore, the assets' source data stored on distribution channels can be leaked or modified by hackers or such the authorities who manage the storage channels. Similarly, the third-party license brokers can also act maliciously (e.g., leaking the licenses) for their own illegal gains.

Blockchain has been employed in several frameworks to mitigate the mentioned trust issues in asset trading and management [6], [7], [8], [9]. In general, blockchain is used in these designs to replace the role of the intermediary, thereby solving the SPoF and certain trust issues. However, existing frameworks still require additional trust assumptions to ensure secure trading. For example, the DAM framework in [9] requires an assumption that the data must not be modified on the cloud storage or during the communication among participants. On the other hand, the design in [6] relies on a third-party *arbitrator*, who is assumed to be honest, to resolve any dispute between the customers and sellers. In a trustless metaverse in which every user interacts with each other via virtual identities instead of real-world interaction [12], the mentioned trust assumptions become inappropriate and should be eliminated to ensure the benefits of the involved stakeholders.

In this paper, we propose MetaTrade, a novel blockchain-based design for secure AIGC trading and management in the metaverse. MetaTrade is specifically designed to resolve trust-related issues that remained unsolved in existing studies. Our framework also takes advantage of perceptual hashing (pHash) [13] and multi-layer encryption techniques to offer advanced security features. As a result, MetaTrade enables secure AIGC trading even if all data are leaked or modified on the transmission, while both the customers and sellers are not required to trust each other or trust any third-party arbitrator. MetaTrade also incorporates a plagiarism prevention system that resolves the unauthorized reproduction issue (e.g., some malicious customers resell the products they have purchased to compete with such the original creators). The framework also offers lower processing cost and higher performance in comparison with existing works.

A. RELATED WORKS AND MOTIVATIONS

Prior to MetaTrade, several works have investigated the use of blockchain for distributed trading systems for digital assets [6], [7], [8], [9]. For instance, the authors in [6] proposed a blockchain-based proof of delivery (PoD) scheme for digital asset trading. In this framework, a smart contract replaces the role of the intermediary in regulating the trading operations. The PoD generated by smart contracts can be used to prove the delivery of assets. Based on the proof, customers cannot avoid the payment as every information is transparent and verifiable on the blockchain. If the customers are not satisfied or cannot download the purchased digital asset after the payment, they can request a dispute for a refund, whose final decision is made by a third-party arbitrator. Consequently, this design still relies on a trust assumption that the arbitrator is always honest, available, and operational, which is impractical in a trustless virtual environment. The blockchain-based DAM framework proposed in [7] also faces difficulties in resolving disputation reports claimed by customers. Although customers are allowed to report mismatched data, there is no mechanism verifying whether the reports are honest, or the customers just try to avoid the payment after receiving the valid products. On the other hand, our design guarantees that the transactions will be canceled automatically if at least one party acting maliciously.

Some other existing works investigate privacy-preserving mechanisms for data/asset trading. SPChain presented in [10] proposed to encrypt the traded data based on the user's public key to prevent data leakage. However, applying asymmetric encryption directly on the source data is not a scalable solution due to its resource-intensive computational demands, particularly when dealing with large data. On the other hand, the authors in [8] supplemented a security manager layer to protect the proposed marketplace design for data trading. This layer consists of multiple storage operators, who offer decentralized data storage to replace centralized storage. In addition, encryption techniques are also deployed to protect the traded assets from data leakage. Although data leakage can be mitigated by encryption techniques in [8] and [10], the attackers can further interfere in the storage environment or the transmission of data to modify the AIGC's source data. Another approach for blockchain-based DAM is presented in [9], which leverages the Attribute-Based Access Control (ABAC) model to manage the access of clients. Specifically, the ABAC's policies are embedded into smart contracts, which automatically grant access of digital assets when some predefined conditions are satisfied by the clients. However, the framework requires an assumption that the data must not be modified on the cloud or during the communication among participants. In contrast, our framework can recognize data alteration attack, thereby canceling the purchase automatically without any financial

loss for both parties. Furthermore, none of the existing works can resist against plagiarism and unauthorized reproduction, which have been resolved in MetaTrade.

For uniqueness-based assets, several works investigated the use of NFT-based methods for AIGC. The authors in [14] devised a technique called Crypto-dropout that aims to ensure the uniqueness of UGC and AIGC. In particular, this method uses the crypto information associated with the user's blockchain account to generate a unique hash, which is used to generate an AIGC/UGC with uniqueness guaranteed by the difficulty of the hash collision. However, the method's purpose is only to generate a unique asset, while the trading process is not discussed in the paper. To enhance the trading of digital assets in the metaverse, the authors in [15] proposed NFTPrivate, a privacy-preserving version of NFT. Although this protocol requires higher computation and storage overhead than the conventional NFT, it can hide users' address thanks to cryptographic commitments. Nevertheless, these techniques are only for uniqueness-based AIGCs, while they are not suitable for access-based digital assets.

B. EXISTING OPEN ISSUES IN AIGC MANAGEMENT

Although different issues in AIGC management have been tackled in existing works as presented above, there remain various open challenges hindering the wide adoption of AIGC in the metaverse. The major problems are presented in this section, and our design aims to address them.

1) ENABLING TRUSTED TRADING IN A TRULY TRUSTLESS ENVIRONMENT

In an open platform like the metaverse, it is often assumed that any involved participants can potentially be malicious (i.e., they want to harm the platform and other participants) or selfish (i.e., they may act dishonestly to maximize their benefit). For example, once receiving the product fee, dishonest sellers can intentionally send incorrect products to the buyers. This is often referred as a *free-riding attack*. In this case, the buyers have no chance to receive their tokens back although they only obtained invalid products. Some existing works such as [6] and [7] allow buyers to claim for a refund by submitting a dispute request, reporting that the final product is invalid or it cannot be downloaded. However, selfish buyers can abuse this function, trying to avoid the payment although they have received the correct products. This is considered a *false-reporting attack*. In this circumstance, it is almost impossible to verify whether the seller or the buyer was dishonest. Although the designs in [6] and [7] integrated a third-party arbitrator to verify the dispute reports, such the arbitrator can be malicious. As a result, none of existing frameworks can operate in a truly trustless environment where all entities could be malicious or selfish.

These issues suggest an open challenge associated with the decentralized asset trading: how to ensure that the customer will obtain the correct asset, and the seller can receive the full product's fee, without requiring the buyers and sellers to trust each other or trust a third-party arbitrator? We aim to tackle this challenge in this paper.

2) PROTECTING THE AIGC DATA

Even if internal actors like sellers and buyers are both honest, external threats such as hackers may pose severe threats to the AIGC data. Although *data leakage* can be prevented by using certain traditional encryption techniques and credential exchange protocols, the attackers can further interfere in the storage or transmission of data to modify the AIGC's source data, which is referred as *data alteration* or *data injection attack*. This is especially challenging since it is not easy to distinguish whether the data is modified by external attackers or internal participants (e.g., the sellers and customers). Consequently, data alteration remains unsolved by most existing frameworks [9] where they often require an assumption that the data must not be modified on the cloud or during the communication among participants. In contrast, our framework can recognize data alteration, thereby cancelling the purchase automatically without any financial loss for both parties.

3) PLAGIARISM AND UNAUTHORIZED REPRODUCTION

Unlike uniqueness-based assets, licence-based or access-based AIGC products are particularly susceptible to plagiarism and reproduction. For instance, a customer, after purchasing an AIGC, might re-sell such the product for illegitimate financial gain. Consequently, this can lead to a significant financial loss for honest AIGC creators and sellers. This issue has not been addressed by existing AIGC frameworks since verifying plagiarism in an automatic and decentralized manner is especially challenging. In MetaTrade, we design a proof of plagiarism (PoP) technique based on both technical and incentive mechanisms to efficiently tackle this problem.

C. CONTRIBUTIONS AND STRUCTURE

Table 1 summarizes the contributions of our work in comparison with other related works in terms of security and privacy. It can be seen from this table that our proposed MetaTrade framework offers more extensive security/privacy protection and desired aspects compared to existing designs. Other numerical comparisons will be presented later in Section III. The main contributions of this paper can be summarized as follows:

- We propose MetaTrade, a novel blockchain-based design for AIGC trading that can operate in a trustless environment, while still ensuring the benefits of involved stakeholders. The proposed trading scheme can resist to data leakage, data alteration, free-riding, and false-reporting attacks without requiring the sellers and customers to trust each other.
- An AIGC management scheme is also integrated in MetaTrade, including a proof of plagiarism technique,

TABLE 1. Comparison between MetaTrade and existing related frameworks.

Framework		PoD	DRM	FastData	DAMChain	SPChain	MetaTrade
Attack/Issue	Cause/Method	[6]	[7]	[8]	[9]	[10]	Ours
Free-Riding	Malicious content sellers avoid granting the data/ products after receiving the full payment.	✓	✓		✓	✓	✓
False-Reporting	Dishonest customers falsely report that a product is invalid to avoid paying the license fee.						✓
Data Leakage	Hackers eavesdrop the communication between users or attack the storage to steal the products.			✓	✓	✓	✓
Data Alteration	Attackers interfere in the transmission or storage environment to modify the products' source data.						✓
Plagiarism	Selfish customers resell the products they have purchased and compete with the original sellers.						✓
SPoF	The centralized distribution channels are under attack, thus disrupting the entire trading system.	✓		✓	✓	✓	✓
Third-Party	The third-party intermediaries are malicious and want to obtain illegal benefits from the products.		✓	✓	✓		✓

an inheritable architecture for AIGC development, and a rating-based reputation system. They act as an efficient incentive mechanism encouraging honest contribution and preventing malicious actors.

- We implemented MetaTrade as a decentralized application (DApp) with two versions published on Github. One version is deployed on a public blockchain,[1] while the other is implemented on a consortium blockchain.[2] Experimental results on public blockchain show that MetaTrade offers a higher cost efficiency than existing related frameworks, while the consortium-based version reveals that our framework, with an optimal blocksize of 400-500 transactions, offers a throughput of nearly 1500 transactions per second with negligible delay.

To the best of our knowledge, MetaTrade is the first work that can securely protect the traded assets and the benefits of stakeholders even when the attackers successfully eavesdrop on all information transmitted by both the customers and the content sellers. This remains true even in trustless environments in which anyone can potentially be malicious.

Preliminary results of our paper were presented in [16]. This journal version, however, makes major extensions compared to the conference version as described in the following. First, more detailed discussions of the related works and open issues related to metaverse DAM are presented in this journal version. Second, the conference version mostly focuses on the design of the DAM smart contract while the current manuscript includes the designs of both the DAM smart contract and the marketplace smart contract. Third, we propose the inheritable architecture for AIGC development, which was not available in the conference version. Fourth, this journal version presents the implementation of MetaTrade on both public and consortium blockchains for different possible deployment scenarios. Finally, much more extensive numerical results are presented in this manuscript compared to those in the conference paper.

The rest of this paper is presented as follows. Section II proposes MetaTrade, a blockchain-based framework for secure AIGC trading and management with detailed security analysis. The implementation and performance of MetaTrade on both public and consortium blockchains are analyzed in Section III. Finally, Section V concludes the paper and discusses some open challenges to be solved in our future work.

II. METATRADE: BLOCKCHAIN-BASED FRAMEWORK FOR SECURE AIGC TRADING AND MANAGEMENT

In this section, we present MetaTrade, a framework for trading and managing AIGC based on blockchain and smart contracts.

A. PRELIMINARIES

1) BLOCKCHAIN

Blockchain technology has evolved from a potential solution for digital currencies to a revolutionary tool enabling various applications across different sectors. At its core, blockchain functions as a decentralized ledger, recording transactions across multiple computers in such a way that the registered transactions are immutable and transparent. These decentralization and immutability properties not only enhance security but also eliminate the need for a central authority, thereby fostering trust among users. However, there remains certain limitations in terms of blockchain storage. In general, a blockchain is often replicated and stored by numerous nodes, while its size will grow continuously as new transactions/data are appended onto the ledger. This increases

[1] https://github.com/duyhung2201/MetaTrade-Ethereum
[2] https://github.com/duyhung2201/MetaTrade-Consortium

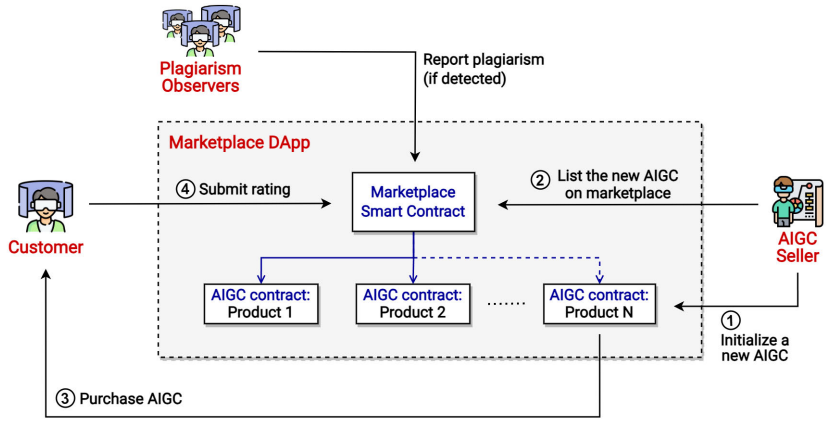


FIGURE 1. A high-level overview of MetaTrade's operation according to the different entities involved in the system.

the storage cost and hinders on-chain-storage applications. Consequently, data of large sizes should not be added to the blockchain. In our design framework, we propose to store the AIGC data off-chain to reduce this storage burden.

There are two main types of blockchains: public and consortium blockchains. Public blockchains, like Bitcoin and Ethereum, are completely open, allowing anyone to join and participate in the network, contributing to the operations of the underlying tamper-resistant ledger. On the other hand, consortium blockchains represent a semi-decentralized approach where the consensus process is controlled by a pre-selected subset of network nodes. This type of blockchain is particularly appealing in business environments where privacy and speed are prioritized.

2) SMART CONTRACT

Smart contracts have redefined the capabilities of blockchain technology far beyond simple transactional functionalities. In general, a smart contract is similar to a computer program with built-in rules and logic. However, the contract's code will be executed by multiple nodes on the blockchain network via the consensus mechanism, and the final results are decided by the majority of the network instead of a single computer. Therefore, as long as the majority of nodes are honest, no individual can manipulate the contract's results. Furthermore, all smart contracts' codes and results are stored on-chain so anyone can verify their validity and correctness. On the other hand, this transparency property might also become a limitation of smart contract. In fact, sensitive information such as private keys and credentials should not be processed by smart contracts, since they are also transparent on the blockchain and any node can read these data. Currently, there are several blockchain platforms that support smart contracts, the two probably most widely known being Ethereum [17] and Hyperledger [18]. These platforms are designed to run smart contracts without fraud, downtime, or any third-party interference.

3) IPFS STORAGE

The InterPlanetary File System (IPFS) is a distributed peer-to-peer storage system [19]. By leveraging a decentralized approach, IPFS allows files to be stored across multiple nodes, ensuring redundancy and resilience against data loss. Each file and all the blocks within it are uniquely identified by cryptographic hashes, ensuring data integrity and enabling a more efficient mode of retrieval. Instead of relying on centralized servers, IPFS operates through a network of peers who host and distribute content, dramatically improving the speed of access and reducing the bandwidth load. IPFS is particularly important in applications that demand high data integrity and availability such as blockchain-based DApps. In these applications, the data are often stored on IPFS whereas only the IPFS's URLs (or the data hashes) are store on blockchain, thereby reducing the storage cost enormously while enhancing data availability.

B. SYSTEM OVERVIEW

MetaTrade is a decentralized marketplace that facilitates the trading and management of license-based AIGC. Fig. 1 illustrates a high-level architecture of MetaTrade, including the following entities:

- **AIGC Seller:** They own the AIGC products and would like to sell their AIGCs to customers.
- **Customer:** They are interested in the AIGC products and would like to pay metaverse tokens to purchase the wanted AIGCs.
- **AIGC Smart Contract:** This smart contract regulates all trading operations between the interested customers and the AIGC seller of a specific product. Each AIGC product is associated with an AIGC smart contract, which is initialized by the corresponding AIGC seller.
- **Marketplace Smart Contract:** This smart contract is unique and acts as the back-end of MetaTrade's DApp. All AIGC smart contracts must be declared in the marketplace smart contract to be visible to the customers. The marketplace smart contract manages the

status of every AIGC contract, allowing customers to submit rating for their purchased AIGCs and commit plagiarism reports.

- **Plagiarism Observer:** They can be any party such as customers and AIGC sellers. If the observers recognized an abnormal similarity between two AIGCs listed on the marketplace, they can commit a plagiarism report to the marketplace smart contract, then earn token rewards if the report is verified to be correct (Section II-D1).

MetaTrade is implemented on both public and consortium blockchains (i.e., Ethereum network and Hyperledger Fabric) to offer different design choices. While the public-chain-based Metatrade offers a higher extent of decentralization and transparency, the version implemented on a consortium blockchain achieves better controllability, scalability, and privacy. Depending on the specific metaverse environment, a suitable version can be selected to satisfy the desirable requirements. It is worth noting that the scope of MetaTrade is not limited to metaverse AIGC, but can also be utilized as a trading platform for other types of digital assets in general.

C. AIGC TRADING

In MetaTrade, the trading process can be depicted in three main phases, including (i) AIGC initialization and request, (ii) granting AIGC access (by sellers), and (iii) finalizing the purchase (by customers). There are several stages involved in each phase. In general, the workflow of the AIGC trading process is presented in Fig 2.

1) PHASE 1 - AIGC INITIALIZATION AND REQUEST

When an AIGC seller wants to list a new product on the marketplace, the following blockchain transaction must be committed:

$$Tx^{\text{init}} = \{Addr_{mp}, Addr_{pr}, P_{\text{Price}}, P_{\text{Desc}}, \text{pHash}, Sig\}, \quad (1)$$

where $Addr_{mp}$ is the marketplace contract's address, $Addr_{pr}$ is the address of the ''parent'' smart contract (Section II-D3), P_{Price} is the desired price for the AIGC product, P_{Desc} is the product's description, pHash is the perceptual hash [13] of the product for plagiarism prevention (Section II-D1), and Sig is the digital signature of the transaction.

To buy an AIGC product, interested customers must submit a purchasing request and deposit the corresponding license fee to the AIGC smart contract (①). Consequently, a purchasing event is emitted by the smart contract to inform the AIGC seller about the new customers.

2) PHASE 2 - GRANTING AIGC ACCESS

The AIGC seller follows the algorithm 1 to grant the AIGC product to customers in a trust-free manner. Specifically, if the AIGC seller decides to sell the product to a customer, they first extract the customer's public key K_{cus}^{pub} from the purchasing transaction (stage ②). Next, the AIGC seller randomizes a 256-bit symmetric key K^{sym}, then use the encryption unit provided in the DApp to conduct the following tasks: (i) use the randomized symmetric key to encrypt the AIGC's source data based on AES-256 encryption algorithm; (ii) hash the encrypted data with SHA-256 algorithm to obtain a hash value $\mathcal{H}$; (iii) use the customer's public key K_{cus}^{pub} to encrypt the symmetric key K^{sym}, thus obtaining a signed key K_{signed}^{sym}. In summary, the encryption unit inputs the source data, symmetric key, and customer's public key, then outputs the signed key, encrypted data, and hash value (see the encryption unit in Fig 2).

Algorithm 1 Seller - Granting AIGC Access

Input: Purchasing transaction Tx^{buy}; AIGC's source data D_{src}; AES encryption function $\text{AES}_{enc}(\cdot)$; hashing function $\text{SHA}(\cdot)$; elliptic curve encryption function $\text{ECDSA}_{enc}(\cdot)$;
Output: Signed key K_{signed}^{sym}; AIGC's hash $\mathcal{H}$; AIGC's IPFS link $\mathcal{U}$;

1: Extract the customer's public key K_{cus}^{pub} from Tx^{buy};
2: Randomize a symmetric key K^{sym};
3: # Activate the encryption unit
4: Encrypt AIGC data: $D_{enc} \leftarrow \text{AES}_{enc}(D_{src}, K^{sym})$;
5: Hash the encrypted data: $\mathcal{H} \leftarrow \text{SHA}(D_{enc})$;
6: Encrypt the symmetric key K^{sym} using the elliptic curve algorithm: $K_{signed}^{sym} \leftarrow \text{ECDSA}_{enc}(K^{sym}, K_{cus}^{pub})$;
7: Upload the encrypted data D_{enc} to IPFS and obtain the corresponding URL: $\mathcal{U}_{\text{IPFS}} \leftarrow \text{IPFS\_up}(D_{enc})$;
8: Confirm the sale of AIGC by submitting this transaction to the AIGC contract: $Tx^{\text{grant}} = \{\mathcal{U}_{\text{IPFS}}, \mathcal{H}, K_{signed}^{sym}\}$;
9: **return** $D_{enc}, \mathcal{H}, K_{signed}^{sym}$.

At stage ④, the AIGC seller uploads the encrypted data to IPFS [19] and correspondingly obtains the URL of the data. Then, to grant the AIGC access to the customer, the seller submits the signed key, hash value, and IPFS URL to the AIGC smart contract, specifying the customer's address (Stage ⑤). At this point, only the IPFS URL is accessible by the associated customer, while the hash value and signed key cannot be accessed by anyone, even the customer, as they are hidden under a private state.

3) PHASE 3 - FINALIZING THE PURCHASE

To confirm and finalize the purchase, the customer must follow the algorithm 2. In particular, at stages ⑥-⑦, the customer obtains the URL from the AIGC smart contract, then uses it to download the encrypted data D_{enc} from IPFS. So far, the customer can still cancel the request to withdraw the deposited license fee because the symmetric key has not been revealed. Otherwise, to confirm the purchase, the customer computes the hash of the encrypted data (denoted by $\mathcal{H}_c$ or hash_c), then activates the ''comparing hashes'' function of the AIGC smart contract, passing the hash_c as an argument (stage ⑧).

By activating this function, the AIGC smart contract automatically compares hash_c with the hash provided by the seller from stage ⑤. If the two hashes are mismatched, it indicates one of the following reasons: (i) the seller was dishonest by uploading an incorrect hash or encrypted

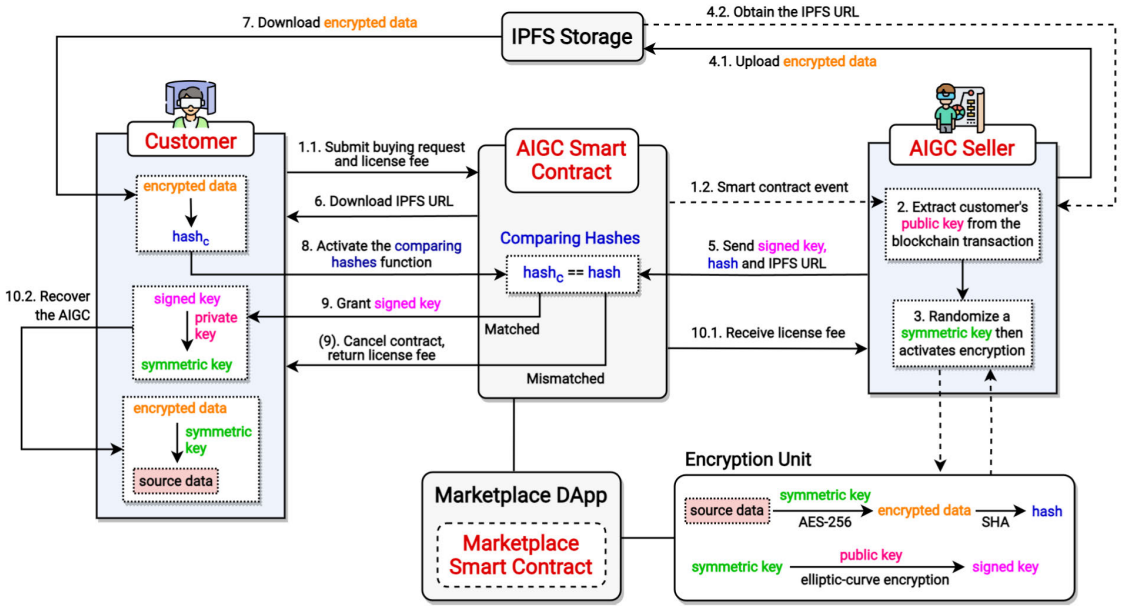


FIGURE 2. The AIGC trading process of the proposed MetaTrade's marketplace.

Algorithm 2 Customer - Finalizing the Purchase

Input: Customer's private key K_{cus}^{pri}; AIGC's IPFS link $\mathcal{U}$; AES decryption function $\text{AES}_{dec}(\cdot)$; hashing function $\text{SHA}(\cdot)$; elliptic curve decryption function $\text{ECDSA}_{dec}(\cdot)$;
Output: The recovered AIGC's source data D_{src};

1: Download encrypted data: $D_{enc} \leftarrow \text{IPFS\_down}(\mathcal{U})$;
2: Compute the data's hash: $\mathcal{H}_c \leftarrow \text{SHA}(D_{enc})$;
3: Activate the *comparing hashes* function of the AIGC smart contract;
4: **if** $\mathcal{H}_c \neq \mathcal{H}$ **then**
5: AIGC contract automatically cancels the purchase;
6: Customer receives the fund back;
7: Go to END;
8: **else**
9: Download the signed key K_{signed}^{sym};
10: Decrypt the signed key to obtain the symmetric key: $K^{sym} \leftarrow \text{ECDSA}_{dec}(K_{signed}^{sym}, K_{cus}^{pri})$;
11: Recover the AIGC: $D_{src} \leftarrow \text{AES}_{dec}(D_{enc}, K^{sym})$;
12: **end if**
13: **return** D_{src}.

data; (ii) the customer intentionally reported an incorrect hash_c; (iii) both the seller and buyer are honest, but the encrypted data has been altered by hackers somewhere on the transmission or storage environment between stage ④ to ⑦. No matter the reason, the purchase will be canceled automatically, and the license fee is sent back to the customer. On the other hand, if the hashes are matched with each other, the AIGC contract automatically publishes the signed key so the customer can download it (stage ⑨). Although other users can also obtain the signed key, only the associated customer can decrypt it to obtain the symmetric key since this process requires the customer's private key. Finally, the customer can use the decrypted symmetric key to recover the product's source data from the encrypted data, while the license fee is transferred to the seller at the last stage.

To prevent denial of service (DoS) attacks, each customer address can only fail in comparing hashes at most 2 consecutive times. If a customer reaches this threshold, the marketplace smart contract will ban the suspicious address permanently. This design guarantees that only one of the following two possible circumstances can occur: (i) the customer obtains the correct source file and the AIGC seller receives the license fee; (ii) the purchase is automatically canceled without financial loss for both parties. Although a significant number of stages involved in the design, its practical operation is simple and easy-to-use for both customers and AIGC sellers.

D. AIGC MANAGEMENT

1) PROOF OF PLAGIARISM

After a customer successfully purchases an AIGC product, the customer might intentionally re-sell such the product on the marketplace to compete with the original seller. To address this concern, MetaTrade proposes the implementation of pHash [13] as a means to prevent unauthorized reproduction, which can assess the similarity between two different contents. For instance, if two AIGCs closely resemble each other, their corresponding pHash values will be abnormally close to each other.

As mentioned in (1), every AIGC seller is required to include a pHash of their product when initializing a new AIGC smart contract. To prevent AIGC sellers from

committing a wrong pHash, customers who successfully bought the AIGC can compute its pHash and attach that pHash to their rating (II-D2). Consequently, the pHash linked to an AIGC product will follow the result of the majority of customers who purchased the AIGC.

If any MU recognizes that there are two AIGC products on the marketplace that resemble each other, they can commit a transaction to the marketplace smart contract to report the plagiarism:

$$Tx^{\text{report}} = \{Addr_{mp}, Addr_1, Addr_2, Sig\}, \tag{2}$$

where $Addr_1$ and $Addr_2$ are the addresses of the two AIGC smart contracts being reported.

Consequently, the marketplace smart contract compares the pHashes of the two reported AIGC products. If the similarity value exceeds a predefined threshold, the one created later will be considered malicious and eliminated from the marketplace. It should be noted that the time that an AIGC is added to the marketplace is determined by the timestamp associated with its initialization transaction. For each successful plagiarism report, the reporter is rewarded σ metaverse tokens derived from the malicious AIGC smart contract. These tokens were deposited by the AIGC sellers when they listed the new product. If a blockchain address suffers from more than 2 successful plagiarism reports, the address will be banned permanently from the marketplace. Similarly, any plagiarism observer will also be banned if they commit more than 2 consecutive false reports.

2) RATING AND REPUTATION MECHANISM

After a customer successfully purchases an AIGC product, the customer's address is added into a *customer list* declared in the corresponding AIGC smart contract. This is a proof of purchase which can be verified by the marketplace smart contract. Based on the proof, customers can submit a rating (from 0 to 5) after purchasing an AIGC product via the following transaction:

$$Tx^{\text{rate}} = \{Addr_{\text{AIGC}}, \mathcal{S}, \text{pHash}_c, Sig\}, \tag{3}$$

where $Addr_{\text{AIGC}}$ is the address of the rated AIGC smart contract, $\mathcal{S}$ is a score from 0 to 5, and pHash_c is the perceptual hash computed by the customer's DApp.

This transaction triggers the marketplace smart contract to verify the proof of purchase. If the proof is available, the marketplace contract accepts the rating and accumulates it into the product's overall rating. The rating information is verifiable and public on the marketplace, acting as a reliable reference indicating the quality of each AIGC product. In addition, the accumulated rating of all products from every seller is also visible to assess the seller's reputation.

3) INHERITABLE AIGC DEVELOPMENT

As mentioned earlier in (1), AIGC sellers must declare the parent contract's address (i.e., $Addr_{pr}$) when initializing a new AIGC smart contract. In case the AIGC product is not inherited from any other AIGC, it is not necessary to set the parent contract address. By linking an AIGC contract to a parent contract, the sellers acknowledge that their product is an inherited version of another AIGC that is associated with the declared parent smart contract. In this case, a minor proportion of the product's profit will be shared with the parent contract. This proportion is decided by the owner of the parent AIGC contract. Despite this financial loss, the AIGC seller is incentivized to do so since his product can potentially reach more customers if its parent AIGC product is popular and of high rating. In addition, if the AIGC seller does not declare the parent contract, the listed AIGC is prone to plagiarism reports and the corresponding blockchain address can be banned permanently from the marketplace.

4) INCENTIVE-BASED ECONOMIC MODEL

Besides using the reputation system to implement the incentive mechanism, MetaTrade also utilizes economic solutions to discourage malicious participants. From the customer's side, if a malicious customer tries to obtain any product P^i without paying the product's fee P^i_{Price}, they will end up paying an additional transaction fee for the comparing-hash operation (at stage ⑧) without actually possessing the product. As a result, the attackers' balance would be drained quickly while they gain no benefit from the system.

On the other hand, if a seller intentionally lists an invalid AIGC product on the marketplace, the AIGC smart contract will eventually cancel the transactions and return the corresponding deposited tokens to customers. In this case, the seller loses a significant number of tokens for sending invalid information (i.e., $\mathcal{U}_{\text{IPFS}}$, $\mathcal{H}$, and K^{sym}_{signed}) to the AIGC smart contract at stage ⑤. Moreover, the seller already lost certain tokens for the initialized transaction Tx^{init}. Consequently, they disincentivize AIGC sellers from committing malicious products.

Finally, all the transaction fees from Tx^{init}, Tx^{buy}, and Tx^{grant} are distributed to consensus nodes who maintain the blockchain's consensus mechanism. In the public-blockchain implementation, the revenue of consensus nodes (i.e., miners) are predefined by the blockchain platform itself. On the other hand, in the consortium-chain version, the leader in our Raft-based consensus algorithm receives 50% of the transaction fee in every operation round, while the rest of fee is distributed equally to other consensus nodes. This reward incentivizes metaverse users to contribute their computational resources to validate MetaTrade's transactions. As a result, the framework achieves high sustainability and viability as it is maintained seamlessly based on these incentivized nodes.

E. SECURITY ANALYSIS

In this section, we show how MetaTrade can resist all presented security threats.

1) FREE-RIDING ATTACK

MetaTrade ensures that AIGC sellers cannot earn benefits without granting the product. This is because the license fee is only paid after the AIGC contract confirms that the two hashes are matched with each other in stage ⑧. If there are issues leading to an incorrect data downloaded by the customer, the hash_c will be completely different, making the contract canceled without financial loss from the customer side.

2) FALSE-REPORTING ATTACK

If a customer intentionally submits a wrong hash_c in stage ⑧, the purchase is simply canceled. As a result, the customer gains no information about the product's source data D_{src} as it is encrypted by the symmetric key K_{sym}. If the customer confirms the correct hash, the process becomes irreversible and the payment is settled. Therefore, it is guaranteed that customers cannot obtain the AIGC products without paying the associated fee.

3) DATA LEAKAGE

While the product's source data is encrypted by the symmetric key K^{sym}, the symmetric key is also protected by asymmetric encryption using the customer's public key K_{cus}^{pub}. Therefore, even if an attacker can steal all data during the transmission (stages ④-⑨), the attacker still cannot obtain the valid source file D_{src}. The only requirement must be met is that the attacker cannot steal the customer's private key K_{cus}^{pri}, which is reasonable to any blockchain frameworks.

4) DATA ALTERATION

Since IPFS storage is immutable, the product's data D_{src} can only be altered during the transmission in stage ④ or ⑦. This also leads to the mismatch between the hashes, making the contract canceled safely.

5) PLAGIARISM AND REPRODUCTION

These threats are mitigated by the plagiarism report scheme presented in Section II-D1. On the other hand, authorized reproduction is encouraged in MetaTrade through the inheritable development architecture, as it is beneficial to both the customers and sellers.

6) SPOF AND THIRD-PARTY ISSUES

MetaTrade does not rely on third-party authorities to maintain its operation. Instead, the distribution channels, license brokers, and centralized databases are replaced by blockchain, smart contracts, and IPFS storage. Therefore, third-party and SPoF-related issues are totally eliminated.

Furthermore, MetaTrade also offers transparency and accountability since every trading information is public on the blockchain and cannot be manipulated by any party. Besides, it requires no direct communication between the sellers and customers. Instead, the participants mostly interact with the platform via the AIGC and marketplace smart contracts.

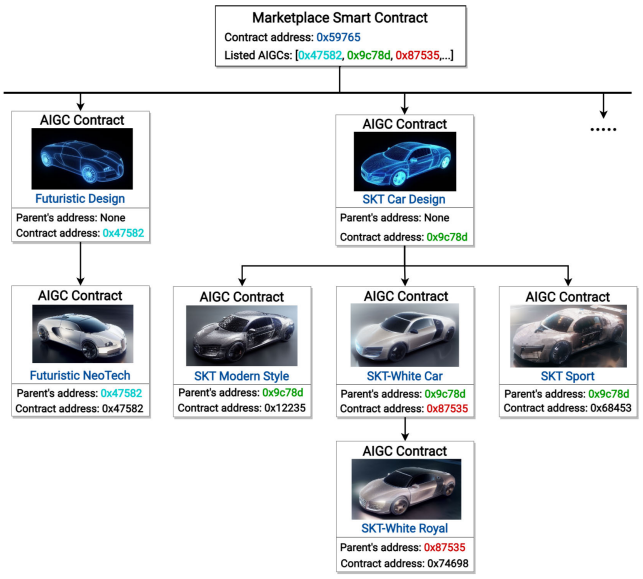


FIGURE 3. The inheritable content development architecture. The car models in this example are generated by stable diffusion version 1.5 [20], styled by LoRA [21], and constrained by ControlNet using edge maps [22].

III. EXPERIMENTAL RESULTS

In this section, we first describe our implementation setup for MetaTrade. Then, we present the cost/performance analysis of our framework on two different types of blockchain, which are public and consortium chains. In specific, the public-chain version is mostly evaluated by cost efficiency. Other factors such as performance and processing speed totally depend on the specific public blockchain that MetaTrade is built on, thus cannot be evaluated directly. In contrast, the consortium-chain version is analyzed using performance metrics such as throughput, latency, and the proportion of transactions that the framework can afford when facing different transaction workloads.

The collaborative AIGC creation scheme achieved by the proposed inheritable development architecture for MetaTrade is illustrated in Fig 3. In this demonstration, there are raw AIGC designs of 3D virtual cars listed on the marketplace supported by the proposed design and development.

A. SYSTEM SETUP AND IMPLEMENTATION

1) BLOCKCHAIN IMPLEMENTATION

In terms of public blockchain, we implement MetaTrade on Ethereum, a well-known proof-of-stake blockchain, because of its reliability and popularity. Both AIGC smart contract and marketplace smart contract are written in Solidity and published on Github.[3]

Regarding the consortium-chain implementation, we built MetaTrade based on Hyperledger Fabric v2.3 [18], an open-source development platform for private blockchains. We choose this platform for implementation because it provides various built-in consensus mechanisms with data

[3]https://github.com/duyhung2201/MetaTrade-Ethereum

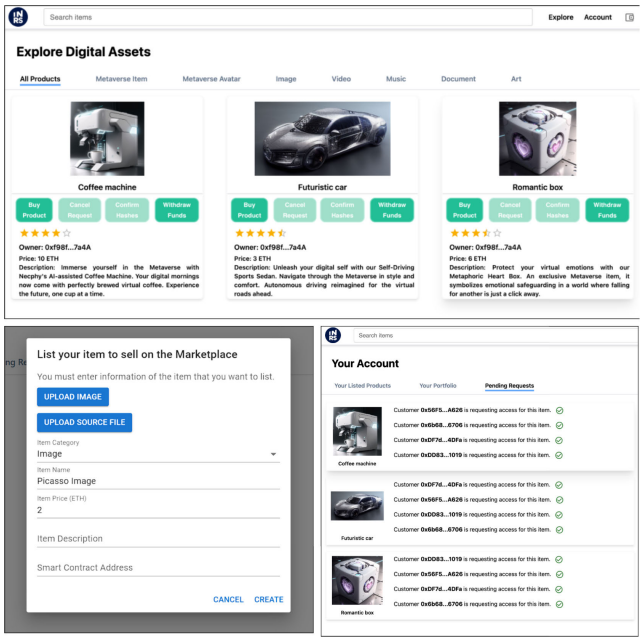


FIGURE 4. An overview of the decentralized application built on top of MetaTrade, connecting to a blockchain and IPFS storage.

privacy preservation. As a general blockchain design, MetaTrade can also be implemented on top of other consortium blockchain platforms. This provides flexibility as we can choose a suitable one with the desired scalability and decentralized extents. The source code is also provided on Github,[4] in which the blockchain is developed using the Go programming language and smart contracts are based on JavaScript and Node.js. Raft consensus algorithm [23] is used to offer up to 50% crashing fault tolerance. The simulated network consists of 100 peer-to-peer nodes, with 50 of them being orderers (i.e., consensus nodes) that maintain the network's operation. Each node is equipped with an independent Docker container to join the system. For evaluation, a blockchain benchmarking tool named Hyperledger Caliper is used to simulate transaction workloads and evaluate the framework's performance. The experiment is deployed on a computer possessing a CPU Intel Core i9-13900K (3.2 GHz) with 64 GB of RAM.

2) DAPP DESIGN PRINCIPLES

The framework is provided in a complete DApp connecting to the blockchain and the IPFS network with a user-friendly interface, illustrated in Fig. 4. The design principles guiding the user interface (UI) and user experience (UX) of MetaTrade focus on making the application intuitive and easily navigable for users. These principles are:

- **Adopt Familiar Mental Models:** MetaTrade's design incorporates familiar mental models, simplifying the transition for users new to blockchain platforms. The interface and operations are akin to conventional marketplace platforms, reducing the learning curve.
- **Simplify Complex Blockchain Concepts:** MetaTrade demystifies complex blockchain terminologies and operations, presenting them in a user-friendly manner. The aim is to ensure users can easily interact with and understand the platform.
- **Deliver the Value of DApps Effectively:** The platform clearly communicates the benefits of using a DApp, such as enhanced security and immediate digital asset transfers. These benefits are made tangible through the platform's design and user interactions.
- **Ensure Strong Usability and Interaction Experience:** Emphasizing a seamless user experience, MetaTrade is designed for high usability. Features and functionalities are intuitive, promoting efficient and hassle-free user interactions.
- **Responsive and Inclusive Design:** The responsive nature of MetaTrade ensures functionality across various devices. The design also accounts for inclusivity, catering to a diverse range of users with different abilities and preferences.
- **Highlight Security Features in User Interface:** The UI of MetaTrade underscores its robust security features, building trust regarding the safety of transactions. Features such as clear indicators for the status of transactions (pending, confirmed, failed) and detailed information on the block including the transaction are prominently integrated, enhancing user confidence and understanding of the transaction process.

B. PUBLIC BLOCKCHAIN EVALUATION

The cost efficiency in Ethereum-based MetaTrade is evaluated based on the amount of gas required for each contract function. It should be noted that the gas fee is only applied for functions that modify the blockchain's state, while read-only functions do not result in gas and latency.

Although the amount of gas usage is fixed for every function, the gas price often fluctuates according to the market's demand. During our analysis on July 2023, each unit of gas is equal to 10^{-8} ETH, while 1 ETH is about $1,895 US. Fig. 5 shows the average cost of the AIGC smart contract's functions. Comparing hashes and granting AIGC are two main functions that must involve additional information to decide the purchase's result, thus they are of highest cost with $1.7 US and $1.93 US, respectively. Since the AIGC's data is stored off-chain on IPFS instead of on-chain storage, the trading cost is fixed no matter the data size.

Regarding the marketplace smart contract, the gas fee of different functions are presented in Fig. 6. The listing function results in the highest cost of more than 150,000 gas (i.e., about 0.0016 ETH or $3.02 US), which is acceptable because this function is only called once when initializing a new product. Although reporting plagiarism and submitting

[4]https://github.com/duyhung2201/MetaTrade-Consortium

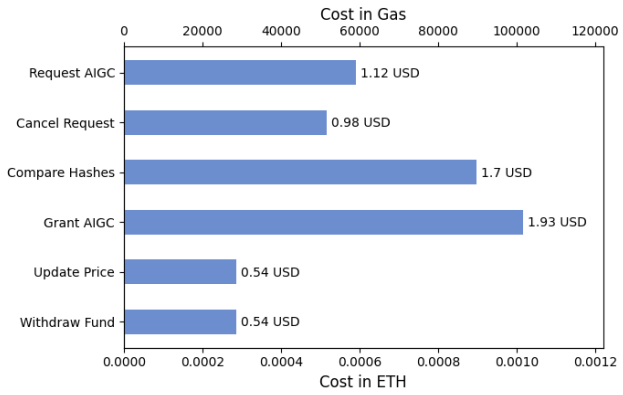


FIGURE 5. Cost analysis for AIGC smart contract's functions.

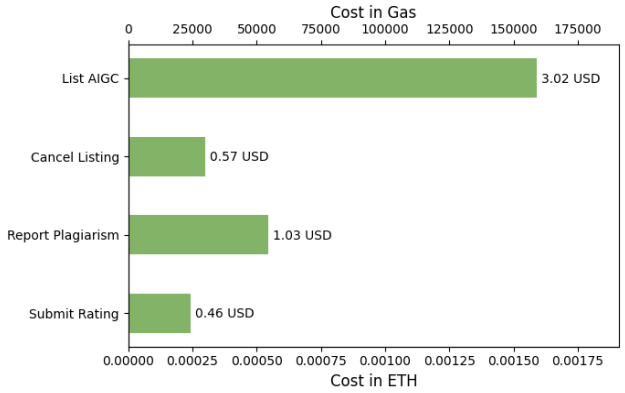


FIGURE 6. Cost analysis for marketplace smart contract's functions.

TABLE 2. Cost comparison for each successful operation cycle between MetaTrade and some data/asset access control frameworks.

Framework	Cost per Successful Cycle	
	Gas Usage	Cost in USD
[24]	304,243	5.76
[25]	5,006,610	94.85
[26]	310,136	5.87
MetaTrade	250,633	4.74

rating also cost certain gas, customers are encouraged to submit these transactions thanks to the incentive mechanism. This fee can also help preventing DoS and collusion attacks (e.g., attackers submitting massive dishonest plagiarism reports) since the balance of the attackers would be drained quickly when launching the attacks.

As shown in Table 2, in comparison with several data access control frameworks in [24], [25], and [26], MetaTrade offers higher cost efficiency for each successful operation cycle (i.e., from the AIGC request until the access/license is granted). This includes requesting AGIC, granting AIGC, and comparing hashes functions.

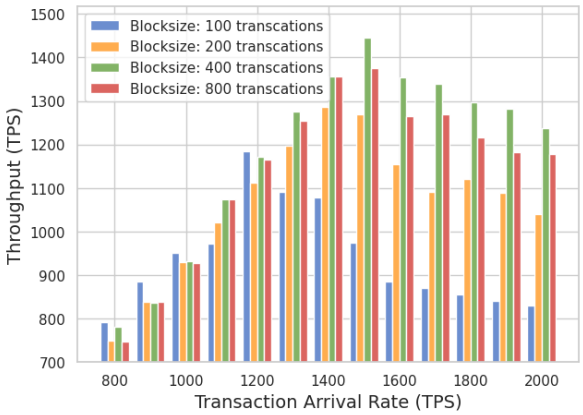


FIGURE 7. MetaTrade's throughput according to varying blocksizes.

C. CONSORTIUM BLOCKCHAIN EVALUATION

Unlike public blockchain, there is no gas fee involved in the consortium-chain implementation. Thus, we mainly evaluate the framework's performance instead of cost efficiency. Fig. 7 shows the throughput of MetaTrade according to different blocksizes and under varying transaction workloads. In this experiment, the blocksize increases exponentially from 100 transactions to 800 transactions per block. This is because the difference in the throughput becomes smaller for larger blocksizes. To observe the performance differences more clearly, we double the blocksize in each experiment. Besides, the range from 100 to less than 1000 transactions per block is also suitable for Hyperledger-based blockchains. It is observed that with a transaction workload of less than 1,000 transactions per second (TPS), the smallest-blocksize network with 100 transactions per block always achieves the highest throughput. However, this trend reverses quickly when there are more than 1,200 transactions submitted to the blockchain per consensus round. Under a high workload, the blocksize of 400 transactions offers the highest throughput. This is because there is a trade-off in terms of blocksize. A smaller block can only store fewer transactions, but a larger blocksize might require higher processing resource to validate such the huge number of transactions. Based on the above experiment, a suitable blocksize of 400-500 transactions should be chosen to offer the highest throughput.

We also investigate the impact of blocksize and transaction workload on the average latency of the blockchain system where the results are shown in Fig. 8. In this setting, blockchain latency is determined as the average latency of each transaction, from the time a transaction is submitted until it is appended to the blockchain. In general, the latency is negligible for all blocksizes when the transaction arrival rate is less than 1200. However, it increases significantly once reaching this limit, especially for the blockchain with the smallest blocksize of 100 transactions. It is also observed that

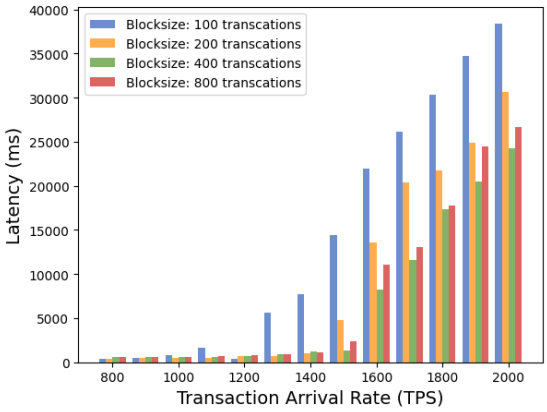
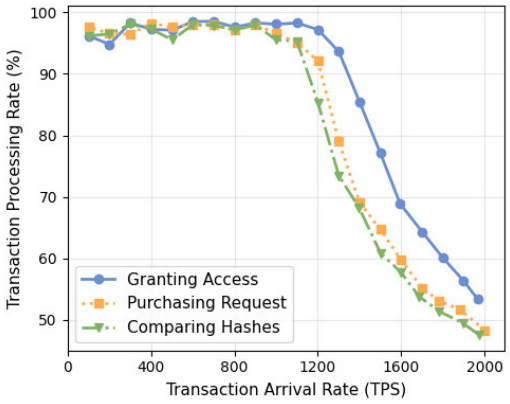
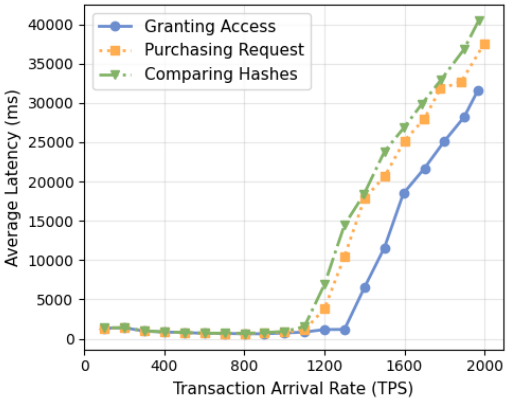


FIGURE 8. Transaction latency under different workloads and blocksizes.



(a) Processing Rate



(b) Average Latency

FIGURE 9. Monitoring processing rate and average latency under different workloads.

the blocksize of 400 transactions enables the lowest latency among the experimented settings.

We also analyze three main functions of the AIGC smart contract, which are purchasing request, granting AIGC access, and comparing hashes. Fig. 9(a) shows that almost 100% of transactions can be processed when the transaction arrival rate is less than 1300 TPS. When the workload increases beyond this saturation point, we observed a lower transaction processing rate for transactions triggering these three functions. On the other hand, Fig. 9(b) indicates that the network's average latency is negligible when the workload is low, but increases significantly once reaching the threshold of 1300 TPS. Compared to Visa with an average processing rate of around 1,700 TPS,[5] MetaTrade can be considered efficient because its purpose is only for AIGC trading/management without involving other financial activities. To maintain a low latency, additional mechanisms such as queueing can be applied to constrain the transaction workload to less than 1300 TPS.

IV. OPEN CHALLENGES

In terms of scalability, the design of MetaTrade reveals certain open challenges that will be investigated in our future research. Firstly, the trading cost in the public-chain version is unstable, relying greatly on the gas fee and token price of the blockchain that it is built on. This might lead to excessively high trading costs during certain periods in which the market's demand is huge. Secondly, the performance of the consortium-chain implementation is still limited at around 1,200 TPS, which should be further improved to support a very large-scale metaverse. Therefore, in our future work, we plan to research advanced consensus techniques for blockchain scalability to improve the framework's performance and cost efficiency. Potential solutions that can be considered are sharding, roll-up, cross-chain communication, and a variety of layer-2 blockchains. In particular, sharding techniques follow the divide-and-conquer strategy to divide a blockchain into multiple smaller sub-chains, thus reducing the computation and storage burdens in each chain. On the other hand, roll-up and layer-2 solutions aim to design certain supporting chains that offload the workload of the layer-1 blockchain.

In the public-chain implementation, MetaTrade can take advantage of such the blockchain platform it is built on (e.g., Ethereum) to enable interoperability. By directly using the blockchain's native currency like ETH, participants can spend MetaTrade's tokens on various applications on that blockchain or convert them to real-world currencies. On the other hand, the consortium-based MetaTrade operates on an independent consortium blockchain, posing interoperability challenges. While interoperability between different public blockchains can be considered a mature topic with various cross-chain communication techniques, the interaction between a consortium blockchain and other public blockchains is still in its infancy. Our future research will study the integration of the consortium-based MetaTrade with well-known public chains, thereby improving scalability and flexibility.

[5]https://phemex.com/blogs/what-is-transactions-per-second-tps

When adopting MetaTrade in real-world metaverse environments, there remain various challenges regarding both technical and social aspects. Firstly, the incentive mechanism of MetaTrade must be adjustable and integrated suitably to the metaverse's economic system, since each blockchain and metaverse platform may own various types of tokens and currency with different values and applications. Moreover, the value of each currency often fluctuates greatly over time. To mitigate these issues, the use of different stable coins, whose values remain more stable regardless of the market's demand, is a potential solution. On the other hand, the regulations in the virtual world may vary between different metaverse environments, and among different countries. As a result, MetaTrade must be designed and adapted to obey these regulations, thus improving social acceptance and ensuring the benefit of the involved participants. Finally, efficient integration of artificial intelligence based functions and tools into MetaTrade is desired to enable various other metaverse applications and services.

V. CONCLUSION

In this paper, we identified various security risks and challenges in metaverse AIGC trading. To mitigate these open challenges, we proposed MetaTrade, a blockchain-empowered DAM framework for AIGC trading and management. The framework ensures the benefits of stakeholders in a trustless metaverse environment where anyone can be malicious. MetaTrade also resists to data leakage and data alteration thanks to multi-layer encryption techniques regulated by smart contracts. Furthermore, a concrete economic incentive mechanism is built to avoid malicious actors, while the rating and reputation scheme encourages honest collaboration. The framework is implemented and analyzed on both public and consortium blockchains as a complete DApp to show its feasibility and efficiency, while offering different privacy and scalability options.

REFERENCES

[1] H. Ning, H. Wang, Y. Lin, W. Wang, S. Dhelim, F. Farha, J. Ding, and M. Daneshmand, "A survey on the metaverse: The state-of-the-art, technologies, applications, and challenges," *IEEE Internet Things J.*, vol. 10, no. 16, pp. 1–34, May 2023.

[2] Y. Han, D. Niyato, C. Leung, D. I. Kim, K. Zhu, S. Feng, X. Shen, and C. Miao, "A dynamic hierarchical framework for IoT-assisted digital twin synchronization in the metaverse," *IEEE Internet Things J.*, vol. 10, no. 1, pp. 268–284, Jan. 2023.

[3] K. Li, B. P. L. Lau, X. Yuan, W. Ni, M. Guizani, and C. Yuen, "Toward ubiquitous semantic metaverse: Challenges, approaches, and opportunities," *IEEE Internet Things J.*, vol. 10, no. 24, pp. 21855–21872, Dec. 2023.

[4] M. Xu, W. C. Ng, W. Y. B. Lim, J. Kang, Z. Xiong, D. Niyato, Q. Yang, X. Shen, and C. Miao, "A full dive into realizing the edge-enabled metaverse: Visions, enabling technologies, and challenges," *IEEE Commun. Surveys Tuts.*, vol. 25, no. 1, pp. 656–700, 1st Quart., 2023.

[5] B. Hammi, S. Zeadally, and A. J. Perez, "Non-fungible tokens: A review," *IEEE Internet Things Mag.*, vol. 6, no. 1, pp. 46–50, Mar. 2023.

[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] A. Garba, A. D. Dwivedi, M. Kamal, G. Srivastava, M. Tariq, M. A. Hasan, and Z. Chen, "A digital rights management system based on a scalable blockchain," *Peer-to-Peer Netw. Appl.*, vol. 14, no. 5, pp. 2665–2680, Sep. 2021.

[8] A. Dixit, A. Singh, Y. Rahulamathavan, and M. Rajarajan, "FAST DATA: A fair, secure, and trusted decentralized IIoT data marketplace enabled by blockchain," *IEEE Internet Things J.*, vol. 10, no. 4, pp. 2934–2944, Feb. 2023.

[9] Y. Zhu, Y. Qin, Z. Zhou, X. Song, G. Liu, and W. 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.

[10] W. S. Lee, A. John, H. C. Hsu, and P. A. Hsiung, "SPChain: A smart and private blockchain-enabled framework for combining GDPR-compliant digital assets management with AI models," *IEEE Access*, vol. 10, pp. 130424–130443, 2022.

[11] V. T. Truong, L. Le, and D. Niyato, "Blockchain meets metaverse and digital asset management: A comprehensive survey," *IEEE Access*, vol. 11, pp. 26258–26288, 2023.

[12] 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.*, vol. 25, no. 1, pp. 319–352, 1st Quart., 2023.

[13] X. Wang, K. Pang, X. Zhou, Y. Zhou, L. Li, and J. Xue, "A visual model-based perceptual image hash for content authentication," *IEEE Trans. Inf. Forensics Security*, vol. 10, no. 7, pp. 1336–1349, Jul. 2015.

[14] H. Duan, Z. Lin, X. Wu, and W. Cai, "MetaCube: A crypto-based unique user-generated content editor for web3 metaverse," *IEEE Commun. Mag.*, vol. 61, no. 8, pp. 1–7, Jun. 2023.

[15] Y. Xiao, L. Xu, C. Zhang, L. Zhu, and Y. Zhang, "Blockchain empowered privacy-preserving digital objects trading in metaverse," *IEEE MultimediaMag.*, vol. 30, no. 2, pp. 1–11, Feb. 2023.

[16] V. Tuan Truong and L. Bao Le, "A blockchain-based framework for secure digital asset management," in *Proc. IEEE Int. Conf. Commun.*, Jun. 2023, pp. 1911–1916.

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

[18] E. Androulaki et al., "Hyperledger fabric: A distributed operating system for permissioned blockchains," in *Proc. 13th EuroSys Conf.*, Apr. 2018, pp. 1–15, doi: 10.1145/3190508.3190538.

[19] E. Daniel and F. Tschorsch, "IPFS and friends: A qualitative comparison of next generation peer-to-peer data networks," *IEEE Commun. Surveys Tuts.*, vol. 24, no. 1, pp. 31–52, 1st Quart., 2022.

[20] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, "High-resolution image synthesis with latent diffusion models," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 10674–10685.

[21] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen, "LoRA: Low-rank adaptation of large language models," 2021, *arXiv:2106.09685*.

[22] L. Zhang, A. Rao, and M. Agrawala, "Adding conditional control to text-to-image diffusion models," 2023, *arXiv:2302.05543*.

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

[24] J. P. Cruz, Y. Kaji, and N. Yanai, "RBAC-SC: Role-based access control using smart contract," *IEEE Access*, vol. 6, pp. 12240–12251, 2018.

[25] H. Guo, E. Meamari, and C.-C. Shen, "Multi-authority attribute-based access control with smart contract," in *Proc. Int. Conf. Blockchain Technol.*, Mar. 2019, pp. 6–11.

[26] Y. Zhang, M. Yutaka, M. Sasabe, and S. Kasahara, "Attribute-based access control for smart cities: A smart-contract-driven framework," *IEEE Internet Things J.*, vol. 8, no. 8, pp. 6372–6384, Apr. 2021.

VU TUAN TRUONG received the B.Eng. degree in electrical and computer engineering from Hanoi University of Science and Technology (HUST), Vietnam, in 2021. He is currently pursuing the Ph.D. degree with the Institut National de la Recherche Scientifique (INRS), University of Quebec, Montreal, QC, Canada. His research interests include blockchain, machine learning, and enabling technologies for metaverse, wireless networks, and future internet.

HUNG DUY LE received the B.Eng. degree in information systems from Hanoi University of Science and Technology (HUST), Vietnam, in 2022. He is currently pursuing the M.Sc. degree with the Institut National de la Recherche Scientifique (INRS), University of Quebec, Montreal, QC, Canada. His research interests include blockchain and enabling technologies for the metaverse and future internet.

LONG BAO LE (Fellow, IEEE) received the B.Eng. degree in electrical engineering from the Ho Chi Minh City University of Technology, Vietnam, in 1999, the M.Eng. degree in telecommunications from Asian Institute of Technology, Thailand, in 2002, and the Ph.D. degree in electrical engineering from the University of Manitoba, Canada, in 2007. He was a Postdoctoral Researcher with the University of Waterloo, from 2007 to 2008, and Massachusetts Institute of Technology, from 2008 to 2010. Since 2010, he has been with the Institut National de la Recherche Scientifique (INRS), University of Quebec, Montreal, QC, Canada, where he is currently a Full Professor. He is the coauthor of the books *Radio Resource Management in Multi-Tier Cellular Wireless Networks* (Wiley, 2013) and *Radio Resource Management in Wireless Networks: An Engineering Approach* (Cambridge University Press, 2017). His current research interests include smart grids, radio resource management, network control and optimization, and emerging enabling technologies for 5G-and-beyond wireless systems and the metaverse. He was a member of the Editorial Board of IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS and IEEE COMMUNICATIONS SURVEYS AND TUTORIALS. He is an Editor of IEEE TRANSACTIONS ON COMMUNICATIONS and IEEE TRANSACTIONS ON COGNITIVE COMMUNICATIONS AND NETWORKING.