International Journal on Advanced Science Engineering Information Technology

Vol.16 (2026) No. 2
ISSN: 2088-5334

Exploring NFT Marketplace Development through Extended AoM

Cheah WaiShiang [a,*], Mahir Pradana [b], Eaqerzilla Phang [a], Nurfauza bt Jali [a], Marlene Lu [c]

[a] *Faculty of Computer Science and Information Technology, Universiti Malaysia Sarawak, Sarawak, Malaysia*
[b] *Telkom University, Bandung, Jawa Barat, Indonesia*
[c] *Swinburne University of Technology Sarawak, Kuching, Sarawak, Malaysia*

*Corresponding author: *wscheah@unimas.my*

***Abstract*—Blockchain application development is a complex process. To reduce development complexity, various software methodologies have been introduced to support the systematic development of blockchain applications. However, despite these efforts, many existing methodologies still rely on software models that are not well-suited to the unique characteristics of blockchain systems. Most models are generic and do not treat blockchain concepts as first-class entities. Hence, there is a gap in current modeling practices, particularly in supporting seamless transformation from design to implementation. To bridge the gap, we introduce a new insight into blockchain-based application development through extended Agent-Oriented Modeling (eAOM). AOM is a methodology for complex socio-technical system development, and we believe that it can become an alternative methodology for blockchain application development. This paper demonstrates the adoption of eAOM for NFT application development. We present a walkthrough example of modeling and developing an NFT marketplace using eAoM, from conception to deployment. The systematic development of the NFT marketplace validates the suitability of eAOM for blockchain application development. A comparative analysis with existing methodologies, along with the limitations of eAOM, is also presented to guide future research.**

***Keywords*—Non-fungible tokens (NFTs); NFT marketplace; extended Agent-oriented modeling (eAOM); Blockchain; application development.**

Manuscript received 27 Mar. 2025; revised 24 Aug. 2025; accepted 22 Nov. 2025. Date of publication 30 Apr. 2026.

I. INTRODUCTION

Non-fungible tokens (NFTs) have gained significant attention in recent years as a solution for protecting the ownership, authenticity, originality, and copyright of digital assets using blockchain [1]. These include artwork, collectibles, virtual real estate, and simulation models [2], digital twins [3] etc, each possessing its unique properties and characteristics [4]. Cryptocurrencies are commonly used to trade, purchase, and exchange NFTs on dedicated NFT marketplaces. Apart from that, these digital platforms provide features for creating and storing NFTs, commonly referred to as minting. Prominent NFT marketplaces such as OpenSea, Rarible, and Nifty Gateway have emerged since 2017. OpenSea, launched in beta in December 2017 by Devin Finzer and Alex Attallah, is often regarded as the first open NFT marketplace built on the Ethereum blockchain, enabling users to mint, buy, and sell NFTs with ease [5]. Rarible, officially launched in 2020 by Alexei Falin and Alex Salnikov, offers a unique feature in the form of its proprietary cryptocurrency, $RARI, which promotes and encourages community participation [6].

Similarly, Nifty Gateway, founded in 2018 by Duncan and Griffin Cock Foster, was later acquired by Gemini LLC in 2020 [7], provides users with the flexibility to register with or without a cryptocurrency wallet, requiring only an email address. These marketplaces employ different approaches to enhance accessibility and usability in the NFT ecosystem. In recent years, the NFT marketplace has seen significant growth in transaction volume, with numerous platforms entering the market. While the potential benefits of NFT marketplaces are widely acknowledged, their development is far from trivial. From the review, several challenges emerged in blockchain application development. These challenges encompass a wide range of issues, including security vulnerabilities, scalability limitations, resource inefficiencies, low throughput, bandwidth constraints, blockchain forks, and usability concerns. The development and deployment of blockchain applications are further complicated by the need to maintain the integrity of smart contract code and ensure seamless system interoperability.

Additional challenges include preserving data privacy and immutability, managing data storage, navigating governance complexities, and integrating with legacy systems. Apart from scalability and integration issues, blockchain applications face challenges in gaining general acceptance due to inadequate regulatory policies, high maintenance, high development costs, and the steep learning curve required to understand and implement the technology [8]. High energy consumption, latency, and slow transaction speed hinder the widespread adoption and optimization of blockchain systems [9].

To date, several development processes or software methodologies have been introduced to address the challenges in blockchain application development. These methodologies aim to overcome the limitations of traditional software development approaches, which are not well-suited to the blockchain domain and require an understanding of decentralized systems, cryptographic principles, and consensus mechanisms [10]. Additionally, they provide essential activities, guidelines, checklists, design principles, and heuristics for blockchain-based software development [11]. A systematic engineering approach can help reduce the likelihood of user dissatisfaction, unforeseen security issues, and other costly defects that arise after deployment [11].

Górski [10] introduced the use of a smart contract design pattern to restructure blockchain applications effectively. These patterns provide reusable solutions that enhance the efficiency and security of smart contract implementations. By defining verification rules for multiple transaction types, they help ensure consistency and reduce redundancy in smart contract development

The Blockchain-oriented Software Engineering Approach for Higher Adoption Possibility (BOSE-HAP) has been proposed to support the development of blockchain applications [11]. BOSE-HAP is structured around three core layers: the design-based thinking flow, the engineering workflow, and the iteration stages.

To evaluate blockchain's potential and design persuasive use cases, the Action Design Research (ADR) and Situational Method Engineering (SME) approaches to the Blockchain Use Case Development (BUD) method [14] were developed. The ADR approach begins with problem formulation, followed by an iterative cycle of evaluating methods or constructs, reflecting on findings, and generalizing the learning outcomes to develop practical and scalable blockchain solutions.

In addition to structured approaches like ADR and SME, agile methodologies, such as Scrum and Lean Kanban, are well-suited for the early stages of blockchain engineering, particularly in identifying potential requirements and defining smart contracts [12]. Agile fosters effective communication between product owners and developers to ensure alignment on product acceptance testing and certification. Additionally, Agile methodologies have been introduced for testing blockchain applications [13], leveraging user stories to validate smart contract functionality. By defining distinct user stories, developers can create diverse use cases for blockchain smart contracts, enabling tailored functionalities for different end users. Furthermore, Agile security testing supports multiple iterations throughout the blockchain development lifecycle, enhancing system resilience and reliability.

Based on the review, conventional agile approaches may not fully address the unique challenges of blockchain application development, particularly in managing decentralized architectures and smart contract configurations. To bridge this gap, the Agile Blockchain Development Engineering (ABCDE) methodology extends Agile principles by incorporating structured modeling techniques specifically designed for blockchain systems [14]. ABCDE follows the core values of the Agile Manifesto while introducing a systematic approach to defining actors, user stories, and system interactions early in the development process. This methodology enhances clarity in smart contract design and reduces failure risks by utilizing modeling notations to define actors, requirements, and relationships [15]. The framework integrates Unified Modeling Language (UML) diagrams, such as use case, class, sequence, and state chart diagrams, to visualize interactions. Following a structured process, ABCDE begins with identifying system goals, defining actors and user stories, and progresses through smart contract development, blockchain component design, testing, and final deployment, ensuring a robust and scalable blockchain application."

Model-driven engineering (MDE) is another software development method that introduces higher levels of abstraction and automates code generation. MDE provides a systematic way to model blockchain applications at different levels of granularity, ensuring that smart contracts and system architectures are well-defined and thoroughly tested [16], [17], [18]. MDE emphasizes platform-independent design, preventing vendor lock-in and facilitating interoperability of smart contracts. With MDE, automated code generation can be derived through model transformation. For instance, smart contracts can be automatically generated from models, significantly reducing the likelihood of logical errors. The models are not platform-specific, enabling ease of migration from one blockchain development platform to another. Graphical models, like the smart contract model, improve the communication between stakeholders [12].

To provide a deeper understanding of business requirements, value exchanges, and actor interactions within a decentralized ecosystem, an extended Agent-Oriented Modeling (AOM) approach has been proposed [19]. This methodology integrates the e^3value model to capture business requirements and value flows, along with agent-oriented models such as goal, role, and interaction models, to enhance the design process. By following a structured five-step process, extended AOM provides a systematic approach for requirement elicitation, domain modeling, and platform-independent blockchain design, making it particularly beneficial for developers new to blockchain application development.

This paper presents the adoption of our extended AOM to demonstrate the development of NFT marketplace application. By applying the framework to a real-world use case, we validate its ability to systematically capture business requirements, value exchanges, and actor interactions while guiding platform-independent blockchain design. The findings contribute to the growing body of research on blockchain-oriented software engineering by offering practical insights for developers.

II. Materials and Methods

To address the gaps in blockchain development, we adopted our extended AOM for blockchain applications as stated in [22]. The extended AOM consists of five phases prior to implementation. The first phase, requirement elicitation, employs the Human-Oriented Method for Eliciting Requirements (HOMER) to gather stakeholders' needs. This is followed by the second phase, computation-independent modeling, where the elicited requirements are formalized through a goal model, a role model, an organization model, and a domain model. The first two phases constitute the problem understanding stage.

Once the problem space is defined, the next phase involves early identification of blockchain use cases through e^3 value modeling and agent role mapping. Then, platform-independent modeling is used to design a blockchain-enabling application. Finally, a platform-specific modeling layer is designed to transform the design model into a blockchain construct. In summary, it can be briefly described as follows:

1) Phase 1: Requirement elicitation layer: Requirement elicitation collects the needs of the domain experts and stakeholders to understand and analyze the problem at hand. At AOM, HOMER is adopted for requirement elicitation [20] which is based on the organizational metaphor of hiring staff.

2) Phase 2: Computation Independent Modeling: In this phase, the modeler transforms the elicited answers into a goal model, role model, organization model, and domain model. It has been reported that the models are designed for non-technical users, making them easy to understand while remaining comprehensive [20], [21]. In addition, the models capture what is expected of the system by describing the context in which it operates [22]. These models may also help analysts to examine the system and identify the problem. Furthermore, they bridge the gap between design and development teams by providing a shared visual and conceptual framework [23].

The goal model illustrates the project's objective and motivation. The goal model comprises a hierarchical structure with a main objective and sub-objectives, and depicts the system's quality goal. It aims to maintain the project's vision and mission throughout development and to achieve consensus on the knowledge, problem, and solution among technical and non-technical clients. Figure 1 shows the goal model for NFT marketplace, which consists of four actors (e.g., buyer, creator, seller, and blockchain provider). The main goal is to manage NFT. This involves two sub-goals, like 'buy NFT' and 'Sell NFT'.

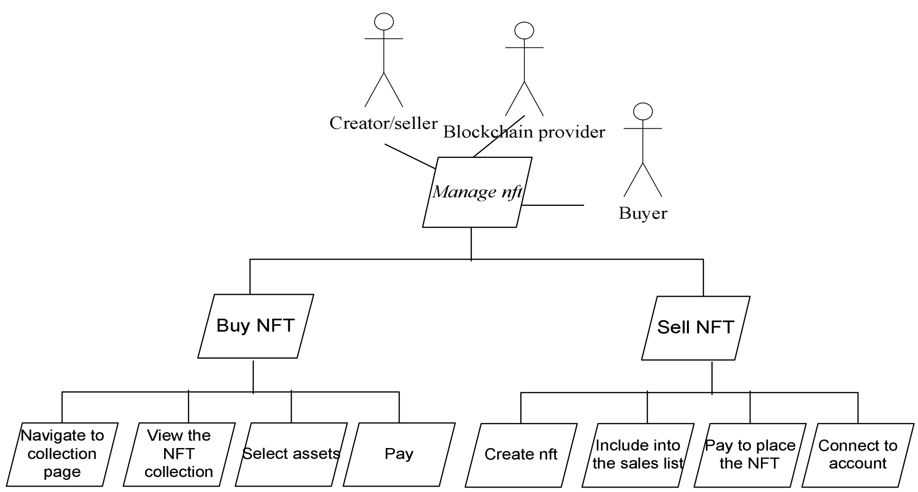

Fig. 1 Goal model for NFT marketplace system

A role model models the responsibilities of a staff member in solving the problem at hand [24]. An example of a role model, a creator, is presented in Table 1. A creator is responsible for creating an NFT to sell, listing the NFT collection, transferring the ownership to the buyer, and handling the virtual coin.

TABLE I
ROLE MODEL FOR NFT MARKETPLACE

Role name	**Creator**
Responsibilities	Create an NFT to sell
	Receive the order
	Transfer the ownership to the buyer
	Receive the virtual coin
	List the NFT collection
Constraints	Must have internet to access the marketplace
	Must have a token to handle the transaction
	Must have a wallet account

The organization model captures the structural relationships among a group of actors and their roles in achieving shared goals. Research identifies five key interaction types between humans and organizations: control, peer, benevolence, dependency, and ownership [25] . Control defines the role's authority within a system; peer states the equal authority of the roles; benevolence describes the shared interests among the roles; dependency specifies how the roles depend on one another to achieve the goal; and ownership determines the corporate boundaries. Figure 2 illustrates the organization model for the NFT marketplace. It models the relationship among the actors, in which all actors communicate in a bi-directional manner and depend on one another. For instance, buyers engage with creators to browse listed collections, while transactions are facilitated and validated by the blockchain provider.

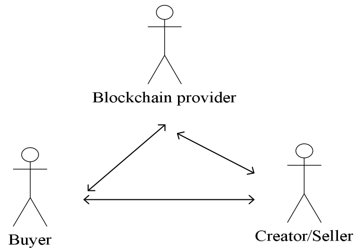

Fig. 2 Organization model for NFT marketplace

The domain model represents shared knowledge as domain entities and shows the relationships between agents and these entities under various conditions. Figure 3 shows the domain model for the NFT marketplace. The knowledge items consist of user, cart, sale, NFT collection, NFT artwork, admin, and announcement. In this model, users can add NFT artworks to their carts, where each NFT artwork belongs to a specific NFT collection.

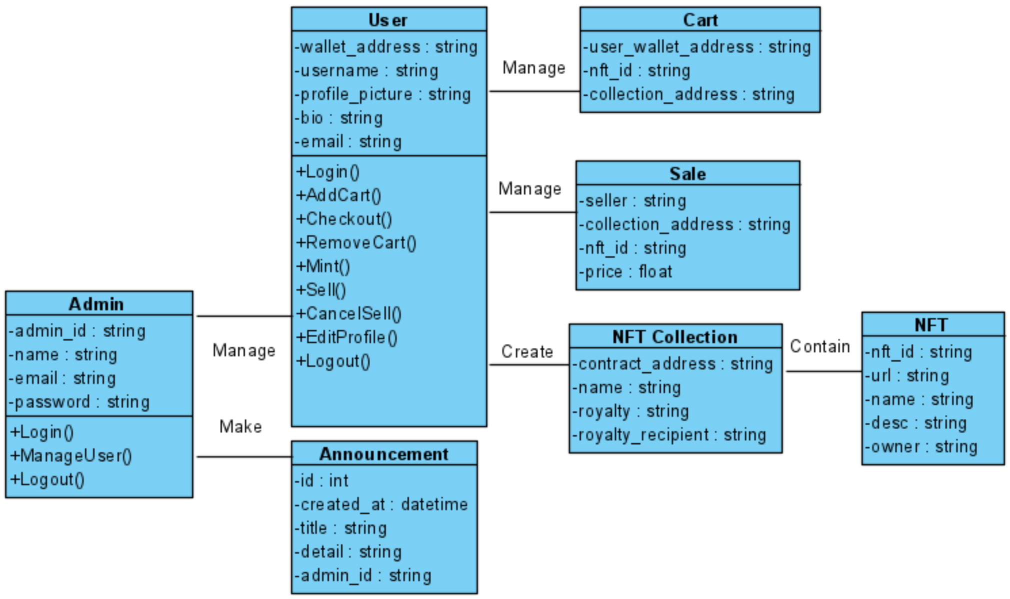

Fig. 3 Domain model for NFT marketplace

3) Phase 3: Early Identification of Blockchain Use Case: We adopted the work from E^3 value to elicit the blockchain requirements by using e^3value model. The e^3value model represents the governance operations and behavior of the existing system and aids in identifying opportunities for blockchain-enabling entities, such as cryptographic transactions and smart contracts. In addition, e^3value models can fulfill the requirements for replacing intermediaries, a market structure that supports peer-to-peer and immutable transactions for a blockchain business case.

Once the blockchain use case is identified, the modeler can begin modeling the design aspects of the blockchain application. During the design phase, the modeler must transform actors from the conceptualization domain modeling and early identification of the blockchain use case into a dedicated agent using an agent-role mapping diagram. The agent-role mapping diagram models the human actor as a software/human agent.

4) Phase 4: Platform Independent Modeling (PIM)- In this phase, the blockchain properties are designed for the use case by using the knowledge model, interaction model, scenario model, and behavior model. The knowledge model defines the knowledge or information that agents require to execute their behavior. Knowledge models are ontologies that provide a knowledge framework, storing essential concepts and relationships within the problem domain.

The interaction model defines the protocol among the agents, including the purpose, initiator, responder, inputs, outputs, and processing. It shows the potential information exchange or request in interaction patterns between agents in a sequential order. The interaction model enables stakeholders to clearly visualize and comprehend agent interactions and message flows within the system.

The scenario model contains the steps and strategy of each agent to accomplish a goal in sequential order. The scenario model is based on a schema that depicts the goal, the initiator, the trigger event, and a list of steps. Each step will be written based on the agent's reaction to the environment and the environment's response to the agent.

From the scenario model, we can construct a behavior model that explains an agent's behavior in an execution cycle [28]. An agent's behavior is defined by its perceptions and knowledge, expressed as rules. A start event will trigger a rule when the condition is true, update the agent's mental state, and send a message or perform an action. The agent may also perceive messages sent by other agents or observe physical actions they carry out.

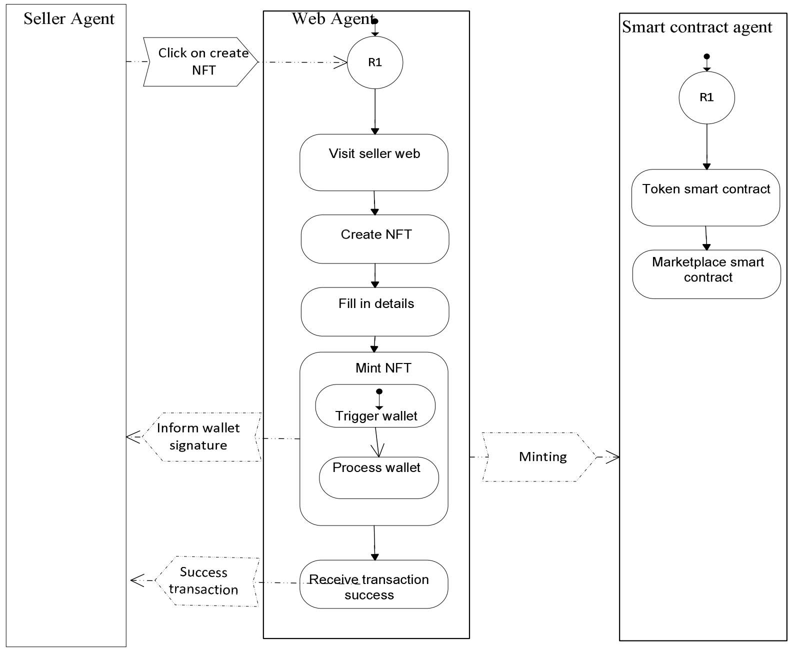

Fig. 4 Behavior model for the seller in the NFT marketplace

A seller interacts with the Token Factory Smart Contract to create an NFT collection. The Token Factory deploys a Token Smart Contract for the collection. Then, the seller will mint NFTs, and the digital assets will be uploaded to the Inter Planetary File System (IPFS) and stored in the Token Smart Contract. The seller sets a listing price and grants approval for the marketplace contract to manage NFT transfers. Upon transaction, the marketplace smart contract validates the price and charges a platform fee.

The steps from Phase 1 to Phase 4 aid in planning and designing a blockchain application. In phase 1, the modeler understands and analyzes the problem through the HOMER technique. Then, the modeler maps the answer from HOMER to the computation-independent modeling. To identify and design a blockchain use case, the modeler constructs the e^3 value model to extract the blockchain requirement. Once the blockchain use case is confirmed, the modeler will design the blockchain component through platform-independent design and modeling. Building upon the design model, developers can proceed to implement the blockchain-enabled system through platform-specific modeling.

The platform-specific model contains core functionalities of NFT marketplace, enabling end users to mint, buy, and sell NFTs. It is a Web 3.0 application that operates in a partially decentralized manner, where all NFTs are minted on the Ethereum blockchain, and transactions are executed through a smart contract deployed on Ethereum. Users can access the platform via any web browser with the MetaMask wallet extension installed. NFT browsing is open to all, but buying, minting, or selling NFTs requires users to log into their MetaMask wallet.

The proposed NFT marketplace was built using the MEVN stack, Solidity, and Hardhat on the Ethereum blockchain. The MEVN stack, a full-stack development technology for building modern web applications, comprises MongoDB (a NoSQL database), Express.js (a backend framework), Vue.js (a frontend framework), and Node.js (a runtime environment for server-side development). This stack provides a seamless, scalable development process, while Solidity and Hardhat are used for smart contract development, testing, and deployment on the Ethereum blockchain.

MongoDB is used to store users' data, including wallet addresses, email addresses, and usernames. The stored data is managed and updated via the system's API server. For NFTs, metadata such as name, description, and image are securely stored on the IPFS using Pinata for reliable pinning and accessibility. Meanwhile, key blockchain-related data, including ownership details, listing status, and NFT pricing, is permanently recorded on the Ethereum blockchain. This helps to ensure transparency, security, and decentralization.

Beyond understanding the technology stack, the critical part of NFT development is implementing smart contracts. A smart contract is an agreement in code that automatically enforces and processes a transaction when the pre-defined conditions are met. It was designed to eliminate the need for intermediaries to ensure efficiency and transparency in digital transactions [26]. Using smart contracts can reduce documentation, time, and costs while enhancing trust and security in transactions. For instance, in a typical asset or liability purchase, a smart contract can replace a lawyer by automatically verifying and executing agreements, ensuring that transactions occur only when conditions stated in the contract are fulfilled. This automation not only enhances trust in online trading but also minimizes human error and potential disputes.

The smart contract that is developed is as follows:

1) Token Smart Contract: The ERC-721 standard is used for NFT creation, transfer, and management. ERC-721 ensures that each token is unique and transferable while maintaining secure ownership structures.

2) Token Factory Smart Contract: A factory method pattern is implemented for deploying token smart contracts.

This design pattern provides a structured approach to dynamically creating and managing NFT collections.

3) Marketplace Smart Contract: This contract facilitates NFT trading, including listing, purchasing, and transferring ownership, while ensuring security and compliance.

III. Results and Discussion

A web-based NFT marketplace called "Elysium" was developed for the Faculty of Computer Science and Information Technology (FCSIT), Universiti Malaysia Sarawak (UNIMAS). Figure 5 shows the UI for seller behavior, as presented in Figure 4, the behavior model. Figures 5 through 8 present the user interface for minting and selling NFTs within the marketplace. To sell an NFT, users must first mint the NFT, which involves creating a collection and adding digital assets to it. The collection can be created by filling out a form that includes setting a royalty percentage, allowing the user to earn a commission whenever their NFT is resold. Once the collection is created, the system prompts the user to approve the transaction via MetaMask.

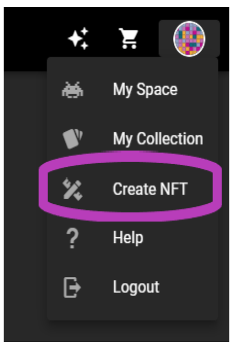

Fig. 5 "Create NFT" Navigation Button

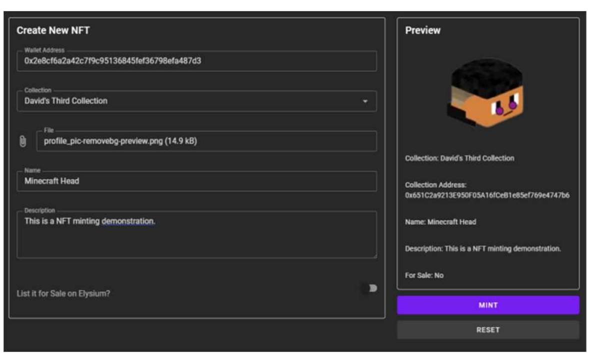

Fig. 6 Minting NFT Page

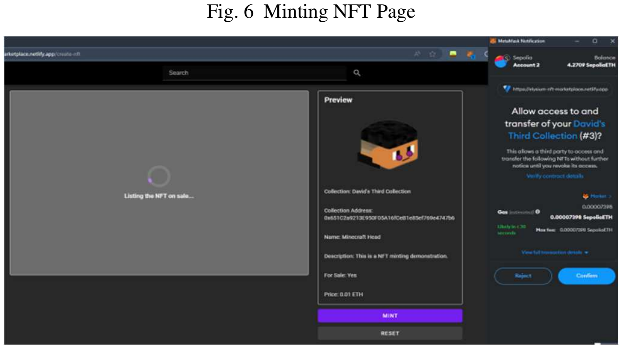

Fig. 7 MetaMask Requesting for Approval, Transferring Ownership to Elysium

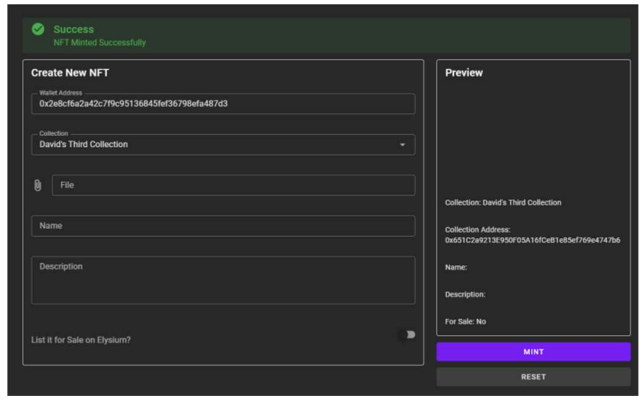

Fig. 8 NFT Successfully Listed for Sale

With a collection in place, the next step is to mint an NFT by uploading the associated digital file. Upon submission, a MetaMask confirmation pop-up appears, and once approved, the NFT is successfully minted and stored on the Ethereum blockchain. To list the NFT for sale, the user needs to locate their newly minted NFT and set the selling price. This will trigger two MetaMask approvals to ensure security and authenticity. Once confirmed, the NFT is officially listed under "On Sale", making it available for potential buyers on the marketplace.

To buy an NFT, a user is presented with two options: to buy immediately or to add it to the cart for later checkout. For both options, users are required to navigate to a collection page by clicking "View" button in the home page, as shown in Figure 9. When the chosen page loads, the user should click "On Sale" to view the NFTs listed for sale. To purchase, the user must click the "View" button again for the specific NFT (see Figure 10), which leads to Figure 11.

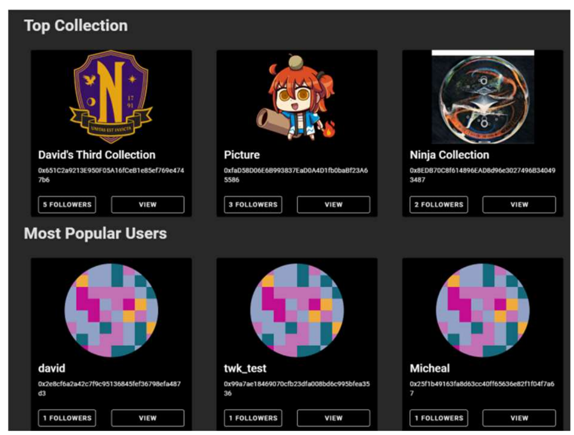

Fig. 9 Elysium Home Page

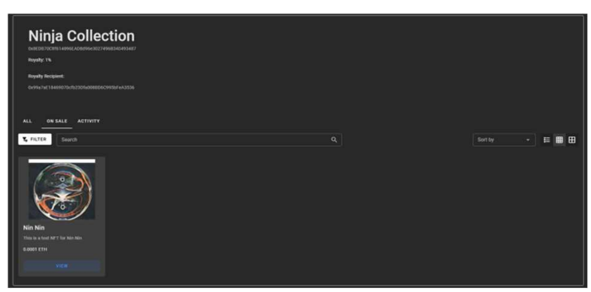

Fig. 10 An Example of NFT Collection's Page

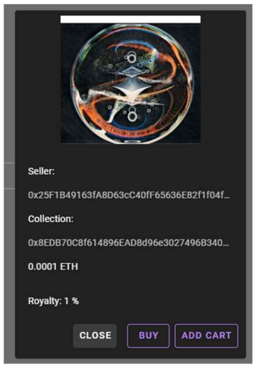

Fig. 11 Viewing on Sale NFT

When a user clicks the 'Buy' button, MetaMask prompts them to confirm payment for both the NFT price and associated gas fees (Figure 12). Upon successful completion of the transaction, the system displays a confirmation message, as shown in Figure 13.

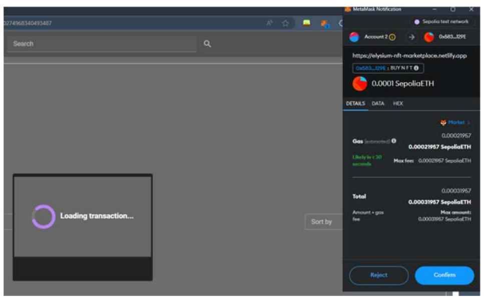

Fig.12 MetaMask Requesting for Buying Confirmation

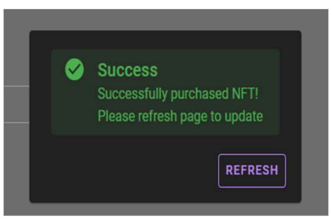

Fig. 13 NFT Successfully Purchased

This section presents the modeling and implementation of the NFT marketplace, demonstrating the practical application of eAOM in decentralized application development. While eAOM effectively guides both the design and implementation phases, our evaluation reveals certain limitations in its modeling capabilities, as detailed in Tables 2 and 3.

TABLE II
COMPARING EAOM WITH STATE-OF-THE-ART SOFTWARE METHODOLOGIES FOR BLOCKCHAIN APPLICATION DEVELOPMENT

Existing Methods	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**
Design pattern Gorski et al. 2024	-	-	-	Yes	-	-	-	-
Agile Marchesi L. et al 2020	Yes	-	Yes	Yes	Yes	Yes	-	-
MDE 2022	Yes	-	Yes	Yes	Yes	Yes	-	-
eAOM	Yes	Yes	Yes	Yes	Yes	-	-	-

1. Modeling and notation 2. Early requirement 3. Requirement 4. Design 5. Implementation 6. Testing 7. Maintenance 8. Deployment.

TABLE III
FINDINGS AND RECOMMENDATIONS IN NFT APPLICATION DEVELOPMENT

Software phases	**Findings**	**Recommendation**
Testing	Financial considerations, resource costs, and system performance, low acceptance of users on NFT application, issues like the direct relationship between artist and fan [31], lack of NFT knowledge among the community [28]	Training to increase the user awareness of the NFT application training and seek a consultant.
Deployment	Law and regulation [27], [29], [30]. Performance issues, CPU cycle [1]. Market abuse. Cryptographic crash[27], [30]	Raising awareness of the needs of law and regulation to the government
Maintenance	The immutability of the records makes the maintenance a trivial task	Continuous integration and development through automated tools? Possible

A. Testing

Testing is one of the phases after the implementation of the NFT application. Testing blockchain applications is challenging due to the high volume of transactions required for verification and validation. Large-scale testing in blockchain environments can be impractical, as the immutability of blockchain makes it impossible to modify or roll back recorded transactions [31].

User Acceptance Testing (UAT) is a standard approach in software validation, but conducting UAT for an NFT marketplace introduces significant financial barriers. One major issue is transaction costs, as blockchain interactions incur gas fees. A minimum transaction of 0.01 ETH (approximately RM20) is needed, making large-scale testing unrealistic without substantial funding. Additionally, NFT prices vary significantly, with some digital assets costing up to 5 ETH, further complicating test scenarios. Additionally, there is concern about the direct relationship between creators and buyers (e.g., artists and fans) established through NFTs by eliminating intermediaries [27]. In the case of music NFTs, for example, the absence of traditional intermediaries raises questions about whether NFTs can effectively enhance artists' visibility, since individual creators are solely responsible for promoting themselves within the NFT marketplace.

Another challenge arises from the instability of testnets, causing frequent transaction failures and delays. This aligns with the usability challenges highlighted by [17], in which blockchain networks often exhibit inconsistent performance, leading to frustration during testing [17]. Financial constraints are also a significant issue. Ethereum transactions are subject to fluctuating gas fees, which can increase or decrease unpredictably based on network demand [18]. This introduces additional costs and requires a substantial initial investment to conduct thorough testing. Furthermore, NFT-related expenses, such as minting fees and platform charges, are key factors in determining user trust and adoption [32].

Beyond financial considerations, resource costs and system performance are two key blockchain testing parameters [26], [33]. In our experience, failed debugging or pending transactions consumed significant time and effort. One major setback was the sudden unavailability of free Ether testnets, forcing us to migrate to the Polygon testnet. This transition required additional testing, debugging, and reconfiguration, increasing the cost and complexity of the testing process.

Finally, NFT literacy is important during the UAT testing of the NFT marketplace. Without strong knowledge, NFT acceptance among participants will be lower. During our UAT test, we faced low NFT literacy among our participants, as shown in Table IV.

TABLE IV
LEVEL OF AWARENESS ON NFT

Questions	1 Strongly Disagree	2 Disagree	3 Agree	4 Strongly Agree
I have heard of Non-Fungible tokens (NFT).	6(13.3%)	8(17.8%)	5(11.1%)	26(57.8%)
	31.1%		68.9%	
I have used NFT before.	29(64.4%)	5(11.1%)	3(6.7%)	8(17.8%)
	75.5%		24.5%	
I am familiar with NFT.	16(35.6%)	15(33.3%)	10(22.2%)	4(8.9%)
	68.9%		31.1%	
I have a good understanding of NFT technology.	15(33.3%)	15(33.3%)	13(28.9%)	2(4.4%)
	66.6%		33.3%	

Table IV presents our early survey to understand respondents' awareness, familiarity, and engagement with NFTs. The survey included 45 respondents; the majority (68.9%) were from Sarawak, 24.4% from West Malaysia, and 6.7% from Sabah. The results indicate that the majority (68.9%) of respondents are aware of NFTs, while 31.1% remain unaware. Despite this awareness, actual NFT usage remains low: 75.5% of respondents report they have never used NFTs, while only 24.5% (11 respondents) have had prior experience with them. Understanding of NFT technology is notably weak, as reflected in the survey data: 66.6% of respondents rated their understanding at the lowest levels (1–2), compared with 33.3% who rated it higher (3–4).

This suggests that while many respondents have heard of NFTs, their practical knowledge and comprehension of the technology remain inadequate. Figure 14 presents the bar chart of the transaction status version during the testing periods. It showcases the complexity while testing the NFT marketplace.

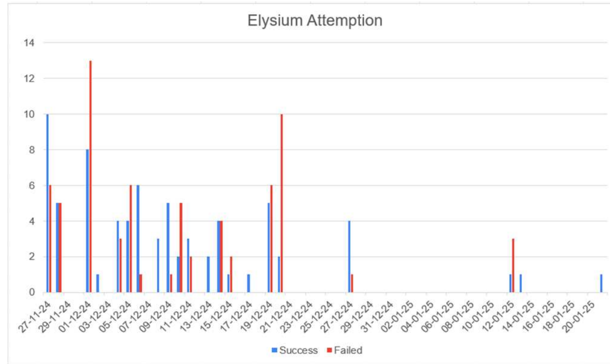

Fig. 14 Transaction status during testing periods

The graph represents the number of successful and failed attempts over time for the "Elysium Attempt". The x-axis marks the dates from November 27, 2024, until January 21, 2025, while the y-axis indicates the number of attempts. The blue bar denotes successful attempts, while the red bars represent failed attempts. In late November and early December, the highest number of attempts occurred on December 1, 2024, with 8 successes and 13 failures, making it the most challenging day with the highest failure. Other significant dates with serious attempts include November 27, 2024, with 10 successes and 6 failures, and December 5, 2024, with 4 successes and 6 failures.

As the month progressed, the number of attempts fluctuated, indicating inconsistent performance. Some days, like December 6, 2024 (6 successes, 1 failure) and December 8, 2024 (3 successes, 0 failures), show higher success rates with minimal or no failures. This is because the transaction during Elysium works well with just 1 or 2 attempts. However, on December 20, 2024, with 2 successes and 10 failures, it was another day with high failure rates, reflecting a difficult session.

By mid- January, it was clear that things had settled down. The attempts became less frequent, but the results were more consistent, suggesting improved mastery or a reduced need for repeated trials. The overall trend suggests a learning curve that begins with intense attempts, faces many challenges, gradually improves, and finally reaches a point where success becomes more predictable. As the date shows no bars, it is because no transaction occurred for Elysium.

B. Deployment and Maintenance

The success of the deployment depends on several factors. First, not many people in our community own Ether or virtual coins. Our community is unfamiliar with virtual currency and does not trust the existing virtual currencies available in the marketplace. From the review, NFTs are subject to incidents such as market abuse, wash trading, round-trip trading, unprofitable trading, and hidden trading [34]. Round-trip trading is the unethical practice of repeatedly buying and selling assets from the original NFT holders. Unprofitable trading occurs when a buyer receives funds from a seller to buy an NFT. Finally, hidden trading implies continuous private trading. On the other hand, a collapse in the cryptocurrency market has demotivated NFT buyers or creators [30].

Furthermore, the lack of legal protection and regulatory frameworks for virtual currency poses a significant barrier to NFT adoption in Malaysia. The absence of clear laws and regulations hinders the efforts to safeguard the copyright of digital assets and protect creators' rights. Additionally, inadequate regulation increases the risk of misuse within NFT marketplaces, such as money laundering and market manipulation [30].

NFT transactions are resource-intensive and can significantly consume CPU power. The requirement to connect to MetaMask for transactions often slows page load times, especially on lower-spec devices. This performance issue can negatively impact the user experience, making it essential to optimize transaction handling to improve accessibility and smooth interactions. The immutability of blockchain presents challenges for maintaining NFT applications. Tasks such as data migration become problematic because records are permanently stored on blockchain nodes upon creation, making modifications or updates virtually impossible.

IV. Conclusion

The development of NFT marketplaces presents unique challenges that conventional software development methodologies are not fully equipped to address. This study evaluates the suitability of our proposed eAOM in developing an NFT marketplace, specifically through the case study of building "Ethereum" for FCSIT, UNIMAS. By examining the entire development process, from requirement analysis to deployment and maintenance, several limitations and challenges were identified for future enhancements to eAOM. The findings demonstrate that eAOM can support the design and development of an NFT marketplace. The agent models capture the key elements of the NFT application. Although the NFT application was successfully developed and deployed, several key issues were identified. One of the most significant is the limited understanding of NFT technology as reflected in the survey data. 66.6% of respondents rated their understanding at the lowest levels (1–2), double the percentage who rated it higher (3–4).

This suggests that while many respondents have heard of NFTs, their practical knowledge and comprehension of the technology remain inadequate. This led to low acceptance of the NFT application during UAT testing. Meanwhile, several key challenges remain unaddressed, including how to model the incompatibility of centralized models with decentralized architectures, the difficulty of modifying immutable blockchain records, the challenges of testing due to high transaction costs and unstable testnets, and the absence of clear legal and regulatory frameworks to protect digital assets and to address the misuse of NFTs in the marketplace. In the future, we would like to investigate how eAOM can be applied to testing NFT applications and to supporting the maintenance and deployment phases of blockchain-based application development.

Acknowledgment

We would like to express our gratitude to UNIMAS for funding this project. The project is funded under Unimas matching grant, with project id of INT/F08/TEL-U/86329/2024.

References

[1] H. R. Hasan, K. Salah, R. Jayaraman, I. Yaqoob, and M. Omar, "NFTs for combating deepfakes and fake metaverse digital contents," *Internet Things*, vol. 25, p. 101133, Apr. 2024. doi: 10.1016/j.iot.2024.101133.

[2] M. Ďuriška, H. Neradilová, G. Fedorko, V. Molnár, and N. Mikušová, "Use of non-fungible tokens for proof of ownership and originality of simulation model in logistics," *Simul. Model. Pract. Theory*, vol. 134, p. 102949, Jul. 2024. doi: 10.1016/j.simpat.2024.102949.

[3] S. A. Gebreab *et al.*, "NFTs for accessing, monetizing, and teleporting digital twins and digital artifacts in the metaverse," *Comput. Commun.*, vol. 228, p. 107965, Dec. 2024. doi: 10.1016/j.comcom.2024.107965.

[4] P. Lohar and K. Rathi, "Evaluation of non-fungible token (NFT)," *arXiv*, Aug. 2023. doi: 10.48550/arXiv.2308.14389.

[5] OpenSea, "About OpenSea," *OpenSea*. Accessed: 2025. [Online]. Available: https://opensea.io/about-opensea

[6] Bybit, "A beginner's guide: What is Rarible and the RARI token?," 2021. [Online]. Available: https://www.bybit.com/en/web3-m/raiders/47/.

[7] E. Banulescu, "Nifty Gateway review: Everything you need to know," *BeInCrypto*, Feb. 21, 2022. [Online]. Available: https://beincrypto.com/learn/nifty-gateway-review/.

[8] V. Sri Vigna Hema and A. Manickavasagan, "Blockchain implementation for food safety in supply chain: A review," *Compr. Rev. Food Sci. Food Saf.*, vol. 23, no. 5, Sep. 2024. doi: 10.1111/1541-4337.70002.

[9] U. N. Kayani, "Exploring prospects of blockchain and fintech: Using SLR approach," *J. Sci. Technol. Policy Manag.*, vol. 16, no. 1, pp. 5-41, Dec. 2023. doi: 10.1108/jstpm-01-2023-0005.

[10] T. Górski, "Smart contract design pattern for processing logically coherent transaction types," *Appl. Sci.*, vol. 14, no. 6, p. 2224, Mar. 2024. doi: 10.3390/app14062224.

[11] Z. Z. Li, H. Wang, D. Gasevic, J. Yu, and J. K. Liu, "Enhancing blockchain adoption through tailored software engineering: An industrial-grounded study in education credentialing," *Distrib. Ledger Technol. Res. Pract.*, vol. 2, no. 4, pp. 1-24, Dec. 2023. doi:10.1145/3632532.

[12] M. Fahmideh *et al.*, "Engineering blockchain-based software systems: Foundations, survey, and future directions," *ACM Comput. Surv.*, vol. 55, no. 6, pp. 1-44, Dec. 2022. doi: 10.1145/3530813.

[13] G. Gatla, K. Gatla, and B. V. Gatla, "Agile testing for blockchain development QA," *Am. Acad. Sci. Res. J. Eng. Technol. Sci.*, vol. 91, pp. 9-16, 2023.

[14] L. Marchesi, M. Marchesi, and R. Tonelli, "ABCDE—Agile block chain DApp engineering," *Blockchain Res. Appl.*, vol. 1, no. 1-2, p. 100002, Dec. 2020. doi: 10.1016/j.bcra.2020.100002.

[15] L. Cocco and R. Tonelli, "A self-sovereign identity–blockchain-based model proposal for deep digital transformation in the healthcare sector," *Future Internet*, vol. 16, no. 12, p. 473, Dec. 2024. doi:10.3390/fi16120473.

[16] M. Jurgelaitis, L. Ceponiene, and R. Butkiene, "Solidity code generation from UML state machines in model-driven smart contract development," *IEEE Access*, vol. 10, pp. 33465-33481, 2022. doi:10.1109/access.2022.3162227.

[17] A. Chandan, V. Potdar, and M. John, "Systematic literature review of blockchain technology's technical challenges: A tertiary study," *Information*, vol. 15, no. 8, p. 475, Aug. 2024. doi:10.3390/info15080475.

[18] P. Rani, R. K. Sachan, and S. Kukreja, "A systematic study on blockchain technology in education: Initiatives, products, applications, benefits, challenges and research direction," *Computing*, vol. 106, no. 2, pp. 405-447, Oct. 2023. doi: 10.1007/s00607-023-01228-z.

[19] M. T. LiBin, C. WaiShiang, M. A. B. Khairuddin, E. Mit, and A. Erianda, "Agent-oriented modelling for blockchain application development: Feasibility study," *JOIV Int. J. Inform. Vis.*, vol. 5, no. 3, p. 248, Sep. 2021. doi: 10.30630/joiv.5.3.670.

[20] D. Wilmann and L. Sterling, "Guiding agent-oriented requirements elicitation: HOMER," in *Fifth Int. Conf. Qual. Softw. (QSIC'05)*, pp. 419-424. doi: 10.1109/qsic.2005.34.

[21] S. Curty, F. Härer, and H. G. Fill, "Blockchain application development using model-driven engineering and low-code platforms: A survey," in *Enterprise, Business-Process and Information Systems*

Modeling, ser. Lecture Notes in Business Information Processing, vol. 450, Springer, Cham, 2022. doi: 10.1007/978-3-031-07475-2_14.

[22] B. Bousetta, O. El Beggar, and T. Gadi, "A methodology for CIM modelling and its transformation to PIM," *J. Inf. Eng. Appl.*, vol. 3, no. 2, pp. 1-21, 2013.

[23] M. Kardoš and M. Drozdová, "Analytical method of CIM to PIM transformation in model driven architecture (MDA)," *J. Inf. Organ. Sci.*, vol. 34, no. 1, pp. 89-99, 2010.

[24] L. S. Sterling and K. Taveter, *The Art of Agent-Oriented Modeling*. The MIT Press, Jul. 2009. doi: 10.7551/mitpress/7682.001.0001.

[25] L. Cernuzzi and F. Zambonelli, "Adaptive organizational changes in agent-oriented methodologies," *Knowl. Eng. Rev.*, vol. 26, no. 2, pp. 175-190, May 2011. doi: 10.1017/s0269888911000014.

[26] G. Poels, F. Kaya, M. Verdonck, and J. Gordijn, "Early identification of potential distributed ledger technology business cases using e3value models," in *Lecture Notes in Computer Science*, vol. 11822, Springer International Publishing, 2019, pp. 70-80. doi: 10.1007/978-3-030-34146-6_7.

[27] S. Dzihan-Zamagna and A. Pfeiffer, "Harmonizing rights and rewards: Music NFTs as a paradigm for equitable compensation in the digital era," *Eur. Conf. Cyber Warf. Secur.*, vol. 23, no. 1, pp. 769-773, Jun. 2024. doi: 10.34190/eccws.23.1.2292.

[28] S. I. Mustajap, A. Husain, S. B. Zaibon, M. A. H. Ismail, and M. N. Abdul Hamid, "The potential of non-fungible tokens (NFTs) in enhancing Malaysia's arts and cultural heritage preservation," *PaperASIA*, vol. 40, no. 6b, pp. 71-85, Nov. 2024. doi:10.59953/paperasia.v40i6b.282.

[29] D. Yoga, T. Susilowati, and M. Tohari, "Copyright violation in works of art using the form of non-fungible tokens (NFT) in the era of digitalization," *Return Stud. Manag. Econ. Bus.*, vol. 3, no. 8, pp. 561-569, Aug. 2024. doi: 10.57096/return.v3i8.266.

[30] A. A. Rabaai, S. A. A. Maati, N. B. Muhammad, and E. M. Eljamal, "Barriers to invest in NFTs: An innovation resistance theory perspective," *Uncertain Supply Chain Manag.*, vol. 12, no. 1, pp. 601-614, 2024. doi: 10.5267/j.uscm.2023.8.011.

[31] M. H. Miraz and M. Ali, "Blockchain enabled smart contract based applications: Deficiencies with the software development life cycle models," *Baltica J.*, vol. 33, no. 1, pp. 101-116, Jan. 2020. doi:10.48550/arXiv.2001.10589.

[32] R. Lahiri, S. Chakrabarti, and S. Saha, "Non-fungible tokens (NFTs): A systematic study of trust criteria," in *Proc. 26th Int. Conf. Distrib. Comput. Netw.*, Jan. 2025, pp. 370-376. doi:10.1145/3700838.3703666.

[33] S. H. Haeri *et al.*, "Mind your outcomes: Quality-centric systems development," Preprints, Dec. 2021. doi:10.20944/preprints202112.0132.v1.

[34] S. Chen, J. Chen, J. Yu, X. Luo, and Y. Wang, "The dark side of NFTs: A large-scale empirical study of wash trading," in *Proc. 15th Asia-Pacific Symp. Internetware*, Jul. 2024, pp. 447-456. doi:10.1145/3671016.3674808.