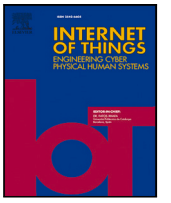
Internet of Things 32 (2025) 101648

Contents lists available at ScienceDirect

Internet of Things

journal homepage: www.elsevier.com/locate/iot



Research article

Enhancing medical digital twins within metaverse using blockchain, NFTs and LLMs

Ruba Islayem [a], Ahmad Musamih [b,*], Khaled Salah [a], Raja Jayaraman [c], Ibrar Yaqoob [d]

[a] Department of Computer and Information Engineering, Khalifa University, Abu Dhabi, United Arab Emirates
[b] Department of Management Science and Engineering, Khalifa University, Abu Dhabi, United Arab Emirates
[c] Department of Industrial Engineering, New Mexico State University, Las Cruces, NM, USA
[d] Artificial Intelligence & Cyber Futures Institute, Charles Sturt University, Bathurst, NSW 2795, Australia

ARTICLE INFO

Keywords:
Blockchain
Digital twins
Metaverse
NFTs
LLMs

ABSTRACT

Medical digital twins (MDTs) are rapidly emerging as transformative tools in healthcare. They provide virtual representations of medical devices and systems that facilitate real-time analysis and enhance decision-making. However, challenges such as secure data management, access control, and the lack of immersive and intelligent patient interactions limit their effectiveness. In this paper, we propose a solution integrating blockchain technology, Non-Fungible Tokens (NFTs), and Large Language Models (LLMs) within a metaverse environment to enhance MDT functionality. Blockchain and NFTs ensure secure ownership and access control, while the metaverse offers an engaging platform for user interaction. An LLM-powered non-player character (NPC) enables intelligent real-time user interactions and personalized insights. We develop two blockchain smart contracts for user registration, NFT ownership, and access control, and utilize decentralized InterPlanetary File System (IPFS) storage for the metaverse, MDT metadata, and interaction logs. We present the system architecture, sequence diagrams, and algorithms, along with the implementation and testing details. We conduct cost, security, and response time analyses to evaluate the smart contracts and LLM performance and compare our solution with existing approaches. We discuss practical implications, as well as challenges and limitations of the proposed solution. Finally, we explore the generalization of our system for various applications. The smart contract code and metaverse files are publicly available on GitHub.

1. Introduction

In recent years, Digital Twins (DTs) have gained significant attention across industries, including healthcare, manufacturing, and smart cities. A DT is a digital replica of a physical object, system, or process that integrates real-time data from sensors and wearable devices to bridge the physical and digital worlds [1]. In healthcare, Medical Digital Twins (MDTs) offer real-time replicas of medical devices and biological systems, allowing for advanced monitoring, analysis, and personalized treatment plans [2]. One of the key advantages of MDTs is their ability to simulate scenarios and predict outcomes in a virtual environment, enabling the evaluation and validation of treatments in a controlled setting [3]. MDTs are also revolutionizing medical device maintenance by enabling

* Corresponding author.
E-mail address: ahmad.musamih@ku.ac.ae (A. Musamih).

https://doi.org/10.1016/j.iot.2025.101648
Received 17 October 2024; Received in revised form 20 March 2025; Accepted 12 May 2025
Available online 27 May 2025

predictive maintenance. Through data integration from embedded sensors, MDTs provide valuable insights into device health and predict failures before they occur, thus enhancing patient safety and extending the service life of critical medical devices [4,5].

Despite the potential of MDT technologies in the healthcare sector, they face challenges in data integrity and security, access control, and intelligent interactive engagement. Current MDTs struggle to ensure the security, integrity, and immutability of data collected from medical devices [6]. Maintaining consistent, tamper-proof records is crucial for building trustworthy digital representations of medical devices. This ensures that clinicians and healthcare systems can rely on them for real-time monitoring, predictive maintenance, and accurate diagnostics. Additionally, it helps prevent incorrect medical decisions, privacy violations, and compromised patient outcomes. Moreover, data privacy and access control remain major concerns [2]. Medical devices usually handle sensitive patient data, and any breach could have serious medical consequences and implications for patient privacy and safety. However, current MDT technologies lack robust mechanisms for data security and access control, making them vulnerable to unauthorized access and potential data manipulation. Effective authentication, authorization, and access control are essential to ensure that only authorized individuals can access, retrieve, or interact with MDT data. Additionally, effectively visualizing and interacting with the data collected and monitored by the MDTs is another challenge [5]. The lack of immersive and intelligent interaction capabilities limits the ability of MDTs to empower patients and healthcare providers. For instance, patients in remote areas face challenges in accessing healthcare services, which makes it difficult for them to stay informed about their medical device status without frequent hospital visits. Moreover, the absence of efficient interactive systems hinders the ability to receive real-time updates and respond promptly to potential issues.

1.1. Motivation

The challenges associated with MDTs underscore the need for a more robust and comprehensive solution to ensure their reliability and effectiveness in real-world applications. This work is driven by the increasing risks of healthcare data breaches, limited access control, and the difficulties patients face in accessing and understanding their MDT records. Healthcare data security remains a major concern, with 725 data breaches reported in 2023 alone, exposing over 133 million records and compromising patient confidentiality [7]. Additionally, studies show that 21% of patients identify errors in their medical records, with over 40% perceiving them as serious [8]. Unauthorized access to sensitive patient data further exacerbates privacy concerns, with nearly 75% of patients expressing concerns regarding who can access their data [9]. Moreover, healthcare literacy remains a significant challenge, as nearly half of hospitalized patients struggle to interpret medical records [10], leading to misinformed decisions and reduced engagement in care. Lastly, the lack of accessible and affordable healthcare services leads to delayed diagnoses, untreated conditions, and increased mortality rates. Globally, over 4.5 billion people lack essential healthcare services [11], while 6.4% of U.S. adults cannot afford necessary medical care due to cost [12].

These statistics highlight the urgent need for a secure, decentralized, and intelligent MDT framework. This could be achieved by leveraging emerging technologies such as the metaverse, blockchain, and Generative Artificial Intelligence (GenAI). The metaverse create immersive environments for patients to interact with their MDTs virtually, while blockchain and Non-Fungible Tokens (NFTs) enhance data security, transparency, and ownership control [13,14]. Moreover, GenAI and Large Language Models (LLMs)-powered virtual assistants facilitate more personalized and intelligent interactions between patients and their MDTs.

Implementing such a solution in real-world has significant practical implications. By providing secure and verifiable MDT data through blockchain, healthcare institutions can enhance compliance with data protection regulations while minimizing the risk of data breaches and unauthorized access. The integration of LLM-powered assistants further empowers patients with a better understanding of their medical device records and reduces misinterpretation. Additionally, metaverse-driven MDT interactions enable remote monitoring and virtual consultations, which reduces the need for frequent hospital visits and reduces healthcare costs. This also alleviates the burden on healthcare facilities, which allows medical professionals to allocate resources more efficiently and focus on urgent cases.

1.2. Contributions

In this paper, we propose a solution that integrates blockchain, NFTs, and LLMs to overcome the limitations of current MDTs. Specifically, we create a blockchain and NFT-based interactive metaverse environment where patients can access their MDTs, monitor their status, and interact with them in a user-friendly and engaging manner. Our solution ensures secure and transparent ownership and access control through blockchain smart contracts (SCs) and NFTs, with each MDT being minted as an NFT. This NFT serves as a certificate of ownership, which restricts access to the metaverse environment exclusively to the MDT owner. The blockchain is also integrated with the decentralized storage of the InterPlanetary File System (IPFS) [15] to address the challenges associated with large data sizes. Additionally, LLM-powered non-Player Characters (NPCs) [16] within the metaverse are utilized to offer intelligent real-time responses to patient inquiries. The LLM is enhanced with a Retrieval-Augmented Generation (RAG) system to enable access to external MDT records and allow patients to ask NPCs questions about their devices and receive clear, accurate, and contextually relevant information instantly. By overcoming the limitations of current MDT technologies, our solution seeks to improve medical device management, enhance patient outcomes, and contribute to better healthcare delivery.

The main contributions of this paper are as follows:

- We propose a blockchain and NFT-based metaverse environment to enhance the functionality and security of MDTs. Users can seamlessly access the metaverse and interact with their MDTs to monitor and review real-time data.

- We integrate an LLM-powered NPC within the metaverse, enhanced with a RAG system, to enable patients to ask text-based questions and receive accurate, relevant responses in real-time.
- We develop blockchain-based SCs for secure user registration, MDT NFT minting, and recording user interactions. These NFTs ensure transparent ownership and restrict MDT access to authorized users.
- We provide a detailed explanation of the system architecture, sequence diagrams, and algorithms developed to implement our solution. In addition, we evaluate how the proposed system achieves its objectives and conduct cost, security, and response time analyses.

The paper is organized as follows: Section 2 provides background information on key concepts relevant to our system. Section 3 presents the related work and highlights the research gap. Sections 4 and 5 explain the system architecture and implementation details. Section 6 outlines the testing processes conducted on the developed system. Sections 7 and 8 analyze the system's cost, security, and scalability and discuss its effectiveness in meeting key objectives, its broader applicability, and its key challenges and limitations. Lastly, Section 9 concludes the paper.

2. Background

This section provides a brief overview of key concepts relevant to our proposed system, including the metaverse, blockchain and NFTs, and LLMs with RAG systems, emphasizing their role in enhancing MDTs.

2.1. Metaverse

The metaverse is a virtual world that is fully immersive and interactive. It is driven by advancements in technology such as the Internet, AI, and Extended Reality (XR) [17]. Initially conceptualized in science fiction, the metaverse has evolved into a rapidly growing domain, integrating virtual and augmented reality to create persistent, shared digital spaces. These environments are increasingly used in various sectors, including gaming, education, and healthcare, and enable novel ways for users to interact with digital content and each other in a seamless and realistic manner [18]. Therefore, immersive simulations can be created using the metaverse to allow patients to access 3D visualization of their MDTs in a virtual environment. This environment enables the simulation of complex medical scenarios and patient interactions, thus allowing individuals to virtually monitor their devices' status and review historical medical records. Moreover, NPCs are autonomous digital entities that exist within virtual environments to interact with users and enhance virtual engagement. Unlike human-controlled avatars, NPCs operate based on pre-programmed behaviors and scripted interactions, which makes them essential for creating dynamic and responsive virtual environments [16]. In MDTs, LLM-powered NPCs in the metaverse can act as virtual doctors that enhance patient interactions by providing real-time guidance, answering questions about MDT data, and improving access to medical information in virtual healthcare environments.

2.2. Blockchain and NFTs

Blockchain is a decentralized and distributed ledger that ensures security, transparency, and immutability through cryptographic and hashing mechanisms [13]. Its decentralized nature eliminates the need for intermediaries to ensure that data remains tamper-proof and verifiable. Moreover, blockchain SCs, which are self-executing agreements with code-based logic, enable automation by autonomously executing predefined rules to ensure transparency and generate immutable records of all transactions [19]. In healthcare, blockchain and SCs have the potential to enable tamper-proof logging of MDT data and ensure a verifiable history of device records and performance. Moreover, they maintain data integrity and prevent unauthorized modifications, which are essential to ensure accurate monitoring and predictive maintenance.

NFTs are unique digital tokens stored on the blockchain that represent ownership or proof of authenticity of a specific item or asset [14]. Unlike cryptocurrencies like Bitcoin or Ethereum, which are interchangeable, NFTs are non-fungible, meaning each one is distinct and cannot be replaced with an identical token. They are typically created using standards like ERC-721 [20] or ERC-1155 [21] on Ethereum, enabling secure and verifiable ownership transfer. NFTs enhance MDTs by serving as tokenized representations that facilitate ownership management and access control. Each MDT can be minted as a unique NFT and linked to verifiable metadata that defines its identity, status, and associated permissions. Through SCs, NFTs enable access control by verifying ownership and granting authenticated NFT owners permission to access, share, modify, or interact with their MDT records [14]. Additionally, dynamic NFTs play a crucial role in MDTs by enabling real-time metadata updates to ensure that the latest changes in the MDT's state are accurately reflected [22]. This is essential for tracking modifications and allowing LLM-powered interactions to access the most up-to-date information. By integrating NFTs, MDTs benefit from secure ownership tracking, enhanced access control, and real-time data synchronization.

2.3. LLMs and RAG systems

LLMs are advanced GenAI models trained on vast amounts of text data to understand and generate human-like responses based on context [23]. These models, such as GPT and BERT, use deep learning techniques to process natural language, which enables them to serve as advanced virtual assistants that understand user queries, answer questions, summarize information, and generate contextually relevant responses. LLMs excel at general knowledge retrieval and conversational AI but have limitations in accessing real-time or domain-specific information beyond their training data [24]. To overcome these limitations, Retrieval-Augmented Generation (RAG) enhances LLMs by integrating external information retrieval mechanisms into the text generation process [25]. Unlike traditional LLMs, which rely solely on pre-trained knowledge, RAG dynamically fetches relevant data from external sources, such as databases, documents, or real-time records, before generating responses. It works by embedding textual data into high-dimensional vector representations and storing them in a vector database [25]. When a query is made, the system retrieves the most relevant documents based on a similarity search and provides the LLM with additional context before generating an answer. This improves accuracy, ensures up-to-date information, and allows models to provide more reliable answers. In MDTs, RAG is essential for ensuring that LLM-powered interactions are based on the most current MDT data. Since MDT records are continuously updated with new sensor readings, operational logs, and diagnostic information, a traditional LLM may provide outdated or inaccurate responses. By incorporating RAG, the LLM can automatically retrieve the latest MDT data stored on decentralized storage or blockchain. This ensures that patient interactions, diagnostic insights, and decision support are always informed by real-time and verifiable information.

3. Related work

MDTs have been widely applied in cardiology, diabetes management, and implantable medical devices. For instance, MDTs assist in optimizing insulin dosing [26] and customizing implantable device therapies [4]. Moreover, they are used for early disease detection and predictive modeling of treatment outcomes [27,28]. While these applications demonstrate the potential of MDTs, recent research has focused on enhancing their capabilities and addressing existing limitations, particularly in data integrity, access control, and intelligent analysis. In this section, we provide an overview of the current literature and developments related to DTs, including the integration of blockchain, NFTs, metaverse, and AI to enhance DT functionalities and capabilities.

3.1. Blockchain and NFTs in DTs

The authors in [29,30] highlight blockchain's role in enhancing MDTs security, integrity, and automation in healthcare. Azzaoui et al. [29] focus on data security, transparency, and real-time updates for disease monitoring, while Amofa et al. [30] utilize Ethereum SCs to automate and secure personal health data management in patient DTs.

Similarly, the authors in [31,32] propose blockchain-based frameworks for managing DTs in manufacturing, with a focus on secure data sharing, traceability, and automation. Huang et al. [31] introduce a peer-to-peer network for DT data management across the entire product lifecycle by leveraging SCs to automate updates and prevent tampering. Mandolla et al. [32] extend this concept to additive manufacturing (AM) in the aircraft industry to ensure supply chain transparency, regulatory compliance, and secure digital threads for AM processes.

Moreover, the integration of blockchain and NFTs has been explored to enhance DTs interoperability, asset management, and traceability. Gebreabet al. [33] proposes an NFT-based framework for DTs to address cross-metaverse interoperability, trusted monetization, and ownership verification. This framework enables seamless teleportation and NFT transfers across decentralized metaverse platforms. Similarly, Hasan et al. [34] presents industrial use cases where dynamic and composable DT NFTs facilitate real-time updates, asset integration, and tradeability, which improve simulation accuracy and operational efficiency. In healthcare, the work of [35] leverages NFTs for DTs in the medical supply chain to capture the attributes and metadata of DTs throughout their lifecycle. This approach ensures secure ownership, lifecycle tracking, and data integrity through SCs and decentralized storage.

3.2. Metaverse and AI in DTs

Jagatheesaperumal et al. [36] and Mourtzis [37] explore the integration of DTs with the metaverse to emphasize their role in enhancing cyber–physical systems and advancing Industry 5.0. Both studies introduce a service-oriented DT architecture that leverages extended reality (XR), IoT, and software analytics to enable real-time synchronization, predictive analytics, and adaptive industrial workflows.

The authors in [38–40] explore the integration of the metaverse and AI algorithms to enhance MDTs for improved diagnostics and treatment planning. In [38,39], the authors develop MDTs of cancer patients within the metaverse and apply machine learning (ML) algorithms, such as support vector machines and random forest, to enable dynamic patient interactions and assist doctors in selecting optimal treatment strategies. Moztarzadeh et al. [40] extend this concept to dental applications by integrating the MobileNetV2 deep learning (DL) algorithm into a blockchain-based metaverse. They create an MDT of cervical vertebral maturation to enable automated and precise diagnostic analysis.

Furthermore, a study by Prakash et al. [41] introduces an industrial metaverse system that integrates blockchain and NFTs for data security and decentralized ownership, along with AI for DT security and fake DT detection. This integration enables dynamic process adjustments and enhances trust and transparency in industrial manufacturing. Sai et al. [42] also proposes an ML, NFTs, and

Table 1
Comparison of the proposed solution with existing ones.

Papers	Blockchain-based	Access control	Metaverse integration	Intelligent interactions	NFTs utilization	AI integration
[29–32]	Yes	NA	NA	NA	NA	NA
[33–35]	Yes	Yes	NA	NA	NFTs for access control and ownership verification	NA
[36–39]	No	NA	Yes	NA	NA	[38,39] utilizes ML algorithms to collect and analyze healthcare data for cancer patients' DTs
[40,41]	Yes	NA	Yes	NA	[41] leverages NFTs for DTs security and ownership management	ML and DL algorithms for automated diagnosis and personalized treatment
[42]	Yes	Yes	NA	NA	Incentive NFTs to encourage patient participation and manage access control	ML algorithms to remotely monitor patients DTs and create personalized treatment plans
Our solution	**Yes**	**Yes**	**Yes**	**Yes**	**NFTs to manage access control and secure ownership**	**GenAI and LLM models for intelligent real-time user interactions**

blockchain-based MDT platform for continuous remote monitoring and personalized treatment. Their approach utilizes ML models for drug recommendations and disease stage detection, while NFTs incentivize patient participation and securely store historical data.

For GenAI integration, the authors in [43,44] discussed the benefits of leveraging GenAI technology in enhancing MDTs. They highlighted the potential of GenAI models in improving treatment efficacy and diagnostic accuracy, providing real-time insights, and supporting personalized patient care. However, despite their promising applications, no implementations or frameworks integrating GenAI with MDTs have been proposed.

3.3. Research gap

Based on the aforementioned works, we conclude that although recent advancements in MDTs offer promising enhancements for medical applications, significant gaps remain in fully integrating advanced technologies to address critical issues. While blockchain and NFTs have been explored to improve the security and traceability of MDTs, there is a notable gap in access control and ownership management. Existing solutions primarily focus on MDT traceability and data security but fail to address access control mechanisms that are crucial for safeguarding privacy and regulating who can interact with MDTs. Moreover, most of these approaches do not leverage metaverse environments, which can introduce more immersive and interactive user experiences. For instance, solutions [29–32,36–39] utilize blockchain for data security and privacy but lack access control mechanisms, immersive user experience, and intelligent user interactions. Conversely, access control mechanisms have been partially addressed in some solutions [33–35] using NFTs, but these implementations do not incorporate immersive experiences or AI capabilities.

Moreover, current AI and ML applications in MDTs such as [38–42] are primarily centered around predictive analytics, such as early disease detection and personalized treatment recommendations. However, they do not utilize GenAI capabilities that can automatically retrieve real-time data and enable intelligent, contextually relevant interactions. LLMs and RAG systems are capable of creating personalized, engaging experiences with MDTs, but their absence limits the ability to provide dynamic, tailored responses. This lack of integration hinders user engagement and reduces the potential for more effective healthcare solutions. Moreover, [36–39] lack blockchain integration and depend on centralized storage, which raises concerns regarding data reliability, availability, and immutability.

Table 1 summarizes the key differences between our proposed solution and existing approaches in the literature. This comparison highlights the novelty and key advantages of our solution compared to other MDT frameworks. We integrate blockchain and NFTs to ensure secure ownership and access control while also delivering an immersive experience through the metaverse. Additionally, we enhance user engagement and DT data interpretations by leveraging LLM and RAG-powered NPCs that provide real-time, personalized intelligent interactions. Therefore, by combining these technologies, our approach stands out by offering a more secure, interactive, and intelligent experience compared to other existing solutions.

4. System design

In this section, we present an overview of the architecture and interaction flow of our proposed solution. We begin by detailing the system architecture, which outlines the relationships and roles of the key components. Additionally, we include sequence diagrams to illustrate the interactions among these components and demonstrate how they integrate to effectively manage and interact with the MDT.

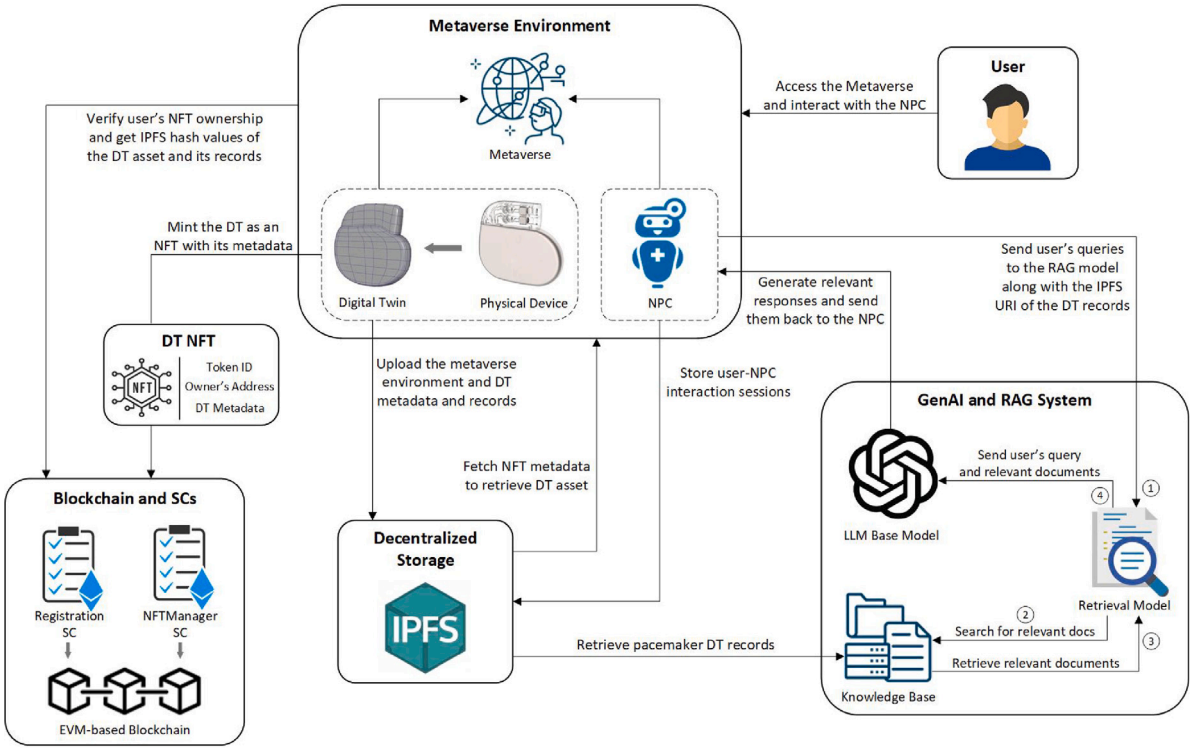


Fig. 1. High-level system architecture of the proposed solution.

4.1. System architecture and components

Fig. 1 provides a high-level view of the architecture of our proposed solution. The diagram visually represents the key components and their interactions within the system. At the core of the architecture is the blockchain, which ensures security, transparency, and immutability. Surrounding this core, the metaverse environment serves as the user interface, where interactions with the DT are enabled through the integration of GenAI and the RAG system. The figure also highlights the flow of data and the role of SCs and NFTs in managing access and interactions.

Below, we provide a detailed explanation of each component illustrated in the system architecture:

- **User:** The user in our system represents a patient who owns an MDT and wants to access it within the metaverse for interaction. To gain access to the metaverse, all users must be registered in the system and own the NFT of their DT.
- **Metaverse Environment:** The metaverse environment is the immersive component of our system. It provides users with a virtual experience where they can access and interact with their DT and obtain information through an NPC. The NPC knowledge is powered by an LLM and RAG that retrieves data from the DT and responds to users' queries accordingly.
- **Blockchain and SCs:** The blockchain serves as the backbone of our proposed system which ensures security, transparency, and immutability of transactions and interactions within the metaverse environment. Moreover, SCs are crucial to the system's operations. Once they are deployed, they become immutable, which guarantees that all functions are executed securely and transparently. Our system comprises two SCs: the Registration SC and the NFTManager SC. The Registration SC is used to register users in the system and assign them specific roles. These roles are linked to the users' unique Ethereum addresses to control access to certain functions. Moreover, the Registration SC is used to record interaction sessions between the user and the NPC on blockchain. It also enables users to register specific healthcare providers to grant them access to these records. The NFTManager SC, which adheres to the ERC-721 standard, is used to mint the MDT NFT to grant users access to the metaverse. Additionally, it is used within the metaverse to verify users' NFT ownership and retrieve the IPFS hash values of the DT metadata and records.
- **NFT:** In our system, the DT should be minted as an NFT with its metadata to create a unique digital asset for its owner. This minting process generates a unique token linked to the user's address to ensure secure ownership and access control. When the user accesses the metaverse, the NFT token is used to retrieve the user's DT metadata from the IPFS and render it into the virtual environment. This ensures that only NFTs owners can access and manage their DTs, which provides an additional layer of security and personalization.

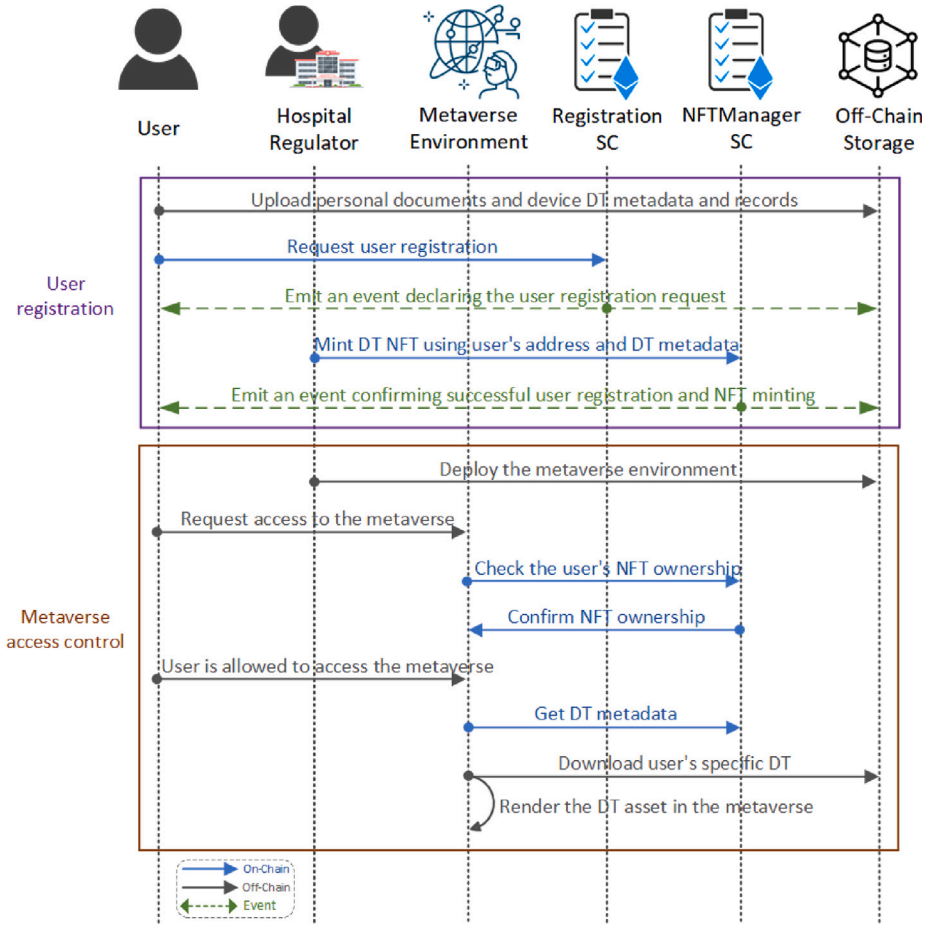


Fig. 2. Sequence diagram illustrating the user registration process and metaverse access control.

- **Decentralized Off-Chain Storage:** The decentralized storage is a critical component of our system. It offers cost-effective off-chain storage that ensures the reliability, accessibility, and integrity of data [15]. In our system, the DT metadata and records are stored in a decentralized storage system such as the IPFS. The hash values of these stored files are then recorded on the blockchain and used to mint the NFT. Additionally, the metaverse environment resides on the decentralized storage, where users can access it through the associated hash value.
- **GenAI and RAG System:** GenAI plays a crucial role in our system by enabling intelligent user interactions. We employ a RAG system to enhance the output of the LLM by giving it access to external knowledge beyond its training data. The RAG in our system enables a retrieval model to access the DT records stored on the IPFS storage. Then, it attaches these records to the LLM base model to ensure that it delivers accurate and relevant information to the users.

4.2. Interactions and process flow

This subsection presents sequence diagrams to illustrate the relationships and interactions among the different system components. It describes the system functionality by detailing the process flow and providing a clear explanation of how each component operates and interacts within the overall architecture.

4.2.1. User registration and metaverse access

Fig. 2 outlines the process for registering users and granting them access to the metaverse environment. Initially, the Registration and NFTManager SCs are deployed by a regulator from the hospital, who manages and oversees the entire process. Users who own DTs need to be registered on the system. They must upload their documents to the off-chain storage, including personal identification and health records for verification, DT metadata, and DT-generated records. Then, to begin registration, users call the registration function of the Registration SC and provide a unique content identifier (CID) of the uploaded documents and their DT metadata. This function emits an event notifying the regulator of the registration request. The regulator reviews the submitted documents to

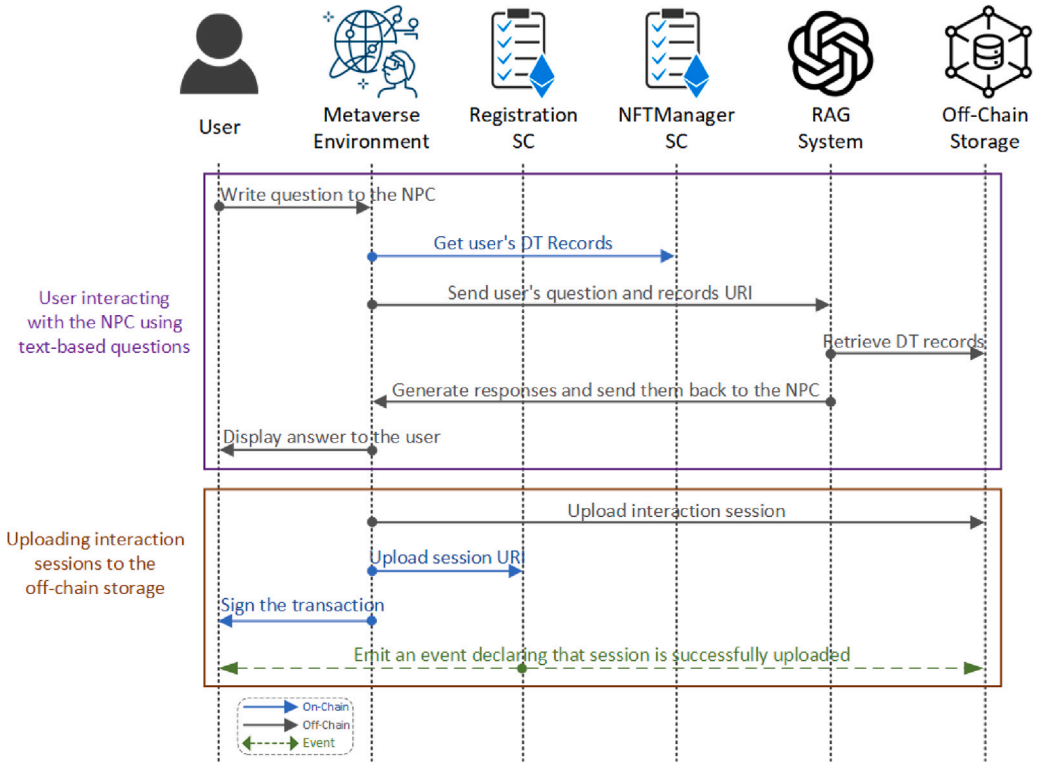


Fig. 3. Sequence diagram illustrating the process of interacting with the DT through the NPC.

verify the user's ownership of the DT and then calls the NFT minting function of the NFTManager SC to register the user and mint an NFT for his DT. The NFT is minted with the DT metadata provided by the user, and an event is emitted for confirmation. When a user wants to access the metaverse, which is hosted on the off-chain storage, the metaverse environment uses the NFTManager SC to verify the user's NFT ownership. If the user owns a valid NFT, access to the metaverse is granted. The metaverse then calls a function to get the DT metadata Uniform Resource Identifier (URI) to download and render the user's specific DT.

4.2.2. User interaction with the DT through LLM

Fig. 3 depicts the process of user interaction with the DT in the metaverse, facilitated by an LLM-powered NPC that provides contextualized real-time responses. After gaining access to the metaverse, the user can engage in text-based conversations with the LLM to ask questions related to their DT status, conditions, and records. When a user sends a question, a function is called to get the latest DT records from off-chain storage and send it along with the user's question to the RAG system. The RAG system retrieves the relevant records and generates a response based on the provided data. This response is then sent back to the NPC, which displays it to the user. Once the interaction session between the user and the NPC terminates, the session logs are uploaded to the off-chain storage. Subsequently, a function of the Registration SC is called to record the unique CID of the uploaded session on the blockchain and map it to the user. This function is restricted to the owner of the DT, so the user needs to sign the transaction in order to execute this function. Healthcare providers can later be registered by users to grant access to these session logs as needed.

5. Implementation details

In this section, we provide the implementation details of our proposed system, including a detailed description of the algorithms used in the developed SCs, the metaverse environment implementation details, and the LLM model integration. The implementation presented in this section focuses on the use case of a pacemaker DT. The metaverse files and SCs are made publicly available on GitHub.[1]

[1] https://github.com/rubak3/PacemakerMetaverse.

5.1. Smart contracts implementation

The SCs in our solution are tailored for deployment on the Ethereum blockchain. Ethereum provides a decentralized platform that supports SCs and facilitates secure and transparent transactions [19]. By leveraging Ethereum's robust infrastructure, our SCs benefit from the blockchain immutability and distributed ledger capabilities. Moreover, Ethereum support for standards like the ERC-721 for NFTs enhances the functionality and integration of our solution within the blockchain network. The SCs are written in Solidity language and compiled and tested using the Remix IDE [45]. Remix is an online web-based development environment that enables users to write and execute SCs on the Ethereum blockchain, as well as provide debugging and testing capabilities for Solidity code. We present below detailed algorithms that represent various functions and working principles of our SCs.

Algorithm 1 Registering Users and Minting DT NFTs

Input: *user, tokenID, docs, metadata, records, regUsers, DTRecordsURI, DTMetadata, NFTOwners*
regUsers: A mapping of the registered users
DTRecordsURI: A mapping of the users' DT records URI
DTMetadata: A mapping of the users' DT metadata URI
NFTOwners: A mapping of NFT owners and their token ID
Upload user DT metadata and records to the IPFS
Function `mintDTNFT(`*user, metadata, records*`)`**:**
  if *(caller == regulator)* $\wedge$ *(user* $\notin$ *regUsers)* **then**
    Increment tokenID counter
    Call *_safeMint(tokenID)*
    `/* _safeMint() function is called from the OpenZeppelin library */`
    Update *regUsers*, *NFTOwners*, *DTRecordsURI*, and *DTMetadata* mappings
    Emit an event confirming successful user registration and DT NFT minting
  else
    Show an error and revert the contract state
`/* User registration and DT NFT minting are complete */`
Function `updateDTRecords(`*DT Records*`)`**:**
  if *(caller* $\in$ *regUsers)* **then**
    Update *DTRecordsURI* mapping
    Emit an event declaring the IPFS hash of the newly updated DT records
  else
    Show an error and revert the contract state
`/* DT records are updated */`
Function `tokenURI(`*tokenID*`)`**:**
  `/* Override the _setTokenURI() function to enable dynamic NFTs */`
  return *dtRecordsUri[_tokenId]*;

Algorithm 1 outlines the process for minting NFTs for users' DT. Users first request registration and upload the necessary data through the *requestUserRegistration()* function. Then, the *mintDTNFT()* function proceeds with user registration by adding the user to a mapping of registered users. Within this function, the *_safeMint()* function is called from the OpenZeppelin ERC721 library to mint the NFT for the user. The URI of the DT records and metadata are then added to their respective mappings, and an event is emitted. To implement dynamic NFTs and support data updates, the *updateDTRecords()* function takes the IPFS hash of new records and updates the *DTRecordsURI* mapping accordingly. Additionally, the *tokenURI()* function is overridden to dynamically return metadata based on on-chain updates instead of storing a fixed URI, where it always returns the most recent metadata associated with the NFT based on the *DTRecordsURI* mapping.

Algorithm 2 represents the *getDTMetadata()* and *getDT-Records()* functions of the NFTManager SC. These functions can be called by registered users only and are used within the metaverse. These functions works by first accessing the *NFTOwners* mapping to obtain the user's token ID, and use it to return the associated DT records and metadata URIs. The DT metadata is used within the metaverse to render the DT asset during runtime, while the DT records are used by the RAG to provide relevant and accurate responses.

Algorithm 3 outlines the process of uploading session logs to the IPFS and granting doctors access to them. When a user-NPC interaction session in the metaverse concludes, the session logs are uploaded to the IPFS. Then, *uploadSession()* function is called to map the IPFS URI to the user address on-chain. If users want to grant a doctor access to their session logs, they call the *registerDoctors()* function to register them. This function adds the doctor's address to the *regDrs* mapping and links it to the user's address. Registered doctors can later retrieve a session URI by calling the *getSessionURI()* function and providing the session ID. The user's address mapped to the doctor must match the address mapped to the provided session ID in order to return the session URI.

Algorithm 2 Accessing DT Metadata and Records

Input: *user, regUsers, DTMetadata, NFTOwners*
Function `getDTMetadata()`:
  if *(caller ∈ regUsers)* **then**
    return *DTMetadata[NFTOwners[caller])]*
  else
    Show an error and revert the contract state
`/* The URI of the DT metadata is returned */`
Function `getDTRecords()`:
  if *(caller ∈ regUsers)* **then**
    return *tokenURI(NFTOwners[caller])*
  else
    Show an error and revert the contract state
`/* The URI of the DT records is returned */`

Algorithm 3 Registering Doctors and Uploading Sessions

Input: *sessionID, regDrs, doctorEA, sessions, sessionURI, regUsers*
doctorEA: Doctor Ethereum address
regDrs: A mapping of the registered doctors
sessions: A mapping of the sessions uploaded to the IPFS
Upload session logs to the IPFS
Function `UploadSession(`$sessionURI$`)`:
  if *(caller ∈ regUsers)* **then**
    Update *sessions* mapping with *sessionID* and user address
    Increment *sessionID*
    Emit an event confirming the successful upload of the session
  else
    Show an error and revert the contract state
`/* Session URI is uploaded and mapped to the user */`
Function `registerDoctors(`$doctorEA$`)`:
  if *(caller ∈ regUsers) ∧ (doctorEA ∉ regDrs)* **then**
    Update *regDrs* mapping with *doctorEA* and user address
    Emit an event confirming the successful registration of the doctor
  else
    Show an error and revert the contract state
`/* Doctor registration is complete */`
Function `getSessionURI(`*sessionID*`)`:
  if *(caller ∈ regDrs) ∧ (regDrs[caller].patientAddr == sessions[SessionID].patientAddr)* **then**
    return sessions[sessionID].sessionURI
  else
    Show an error and revert the contract state
`/* Session URI is retrieved by a registered doctor */`

5.2. Metaverse environment implementation

The metaverse environment is developed using Unity 3D and programmed with C# scripts. It includes two scenes. The initial scene, shown in Fig. 4, offers an introductory interface for users. To integrate blockchain and Web3 functionalities, the Thirdweb Unity SDK [46] is utilized. This SDK enables seamless connections to users' Ethereum wallets, interactions with SCs, message signing, and utilization of common standards like tokens and NFTs. It also provides built-in RPC URLs and IPFS gateways. In our implementation, the SDK initially connects the user's wallet and then connects to the NFTManager SC using the Ethereum address. Then utilizing the ERC-721 NFT standard, the SDK retrieves the NFT balance of the connected wallet to ensure that only users holding a valid DT NFT are granted access to the metaverse.

The second scene, shown in Fig. 5, serves as the main interactive scene where users locate their pacemaker DT and interact with the LLM-powered NPC. The pacemaker DT asset is not loaded initially in the scene but is rendered during runtime. When the user accesses the metaverse, the Thirdweb Unity SDK is used to call the *getDTMeadata()* function, described in Algorithm 2, to retrieve the DT metadata URI. This URI is used to download the DT asset from the IPFS and render it into the scene. Moreover, Fig. 6

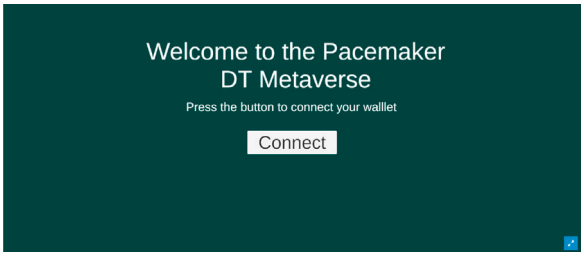


Fig. 4. Start scene.

Fig. 5. Main scene including the DT and the NPC.

Fig. 6. Chat interface for interacting with the NPC.

shows the chat interface where users can interact with the NPC using text-based questions. This chat is designed to provide real-time responses and detailed information about the user's pacemaker status. The dynamic loading of the DT asset and the real-time interaction capabilities ensure personalized and engaging user experiences within the metaverse.

5.3. GenAI model implementation

The GenAI model is implemented using OpenAI's RAG framework (AI Assistant), which enhances the base LLM by enabling access to external knowledge sources. OpenAI Assistants can be provided with specific instructions to adjust their personality and ensure they provide relevant and accurate responses [47]. They can access various tools, such as the file search tool, to retrieve and process external data.

In our implementation, when a user initiates a conversation with the NPC, the NPC first utilizes the Thirdweb Unity SDK to call the *getDTRecords()* function and retrieve the user's DT records, as detailed in Algorithm 2. This URI is used to download the records file from the IPFS. Next, an OpenAI Assistant is initialized with the GPT-4o-mini base model to handle user interactions. A persistent thread is created to maintain the conversation context and manage message history. Then, the DT records file is attached and referenced within this thread to provide relevant information to the LLM. When a user sends a message, the LLM utilizes attached files to retrieve related data and generate a contextually aware response. At the end of the session, the Thirdweb Unity SDK is used to upload the session logs to the IPFS, and the *uploadSession()* function, described in Algorithm 3, is called to store the session URI on-chain.

6. Testing and validation

This section details the testing and validation process carried out to verify the system's implementation and ensure its correct functionality. We conducted a series of tests on the SCs, the metaverse environment, the LLM integration, and the dynamic data

Table 2
Ethereum addresses of deployed SCs and involved participants.

Participant/SC	Ethereum address
Regulator	0xCf91699bde28437A700c81Eb548D5753ef84E23c
User	0xcCB35879a8ff534AA149018c9046B37352029adA
Doctor	0xFF8880B24939aA9DCCABb35040cDFc02a664aB08
Registration SC	0xDA4BAc5620c9CBDD5d4E135774a0CF56F689abc8
NFTManager SC	0xd9A75c00B0360642a65354B21Bb42A7c7E7478d1

Table 3
Transaction hashes of executed functions on Sepolia testnet.

Function	Transaction hash
requestUserRegistration()	0x346d7aadcfd89f369751e86c67bfad7101d36f618b9c72ea7d5cf592f083f721
mintDTNFT()	0x542482549bf1b9b8c98450b959e79e3c6b6a3ef029124d886e5565634fd27275
UploadSession()	0x2acbdcdfac2d9f1f384acf170397e368c2f7150b5ad4e3d97b7f09e0d69c49b1
registerDoctors()	0xd86797662aff46c056da96dbb0ace63a6d054cc02eead03b8ad9cd827b076f45
updateDTRecords()	0xca6734f7683bb8cde8f9d5295a6b49f6d7ea8b8e1ef0ae019633b3193a2ac05f

Table 4
IPFS hashes of the uploaded files.

File	IPFS hash
Metaverse	QmUhVS4zmhecT4grfN2ma7tg9MfqXYSCZ5ZsbH4gHwRUHx
DT metadata	QmTzSTqN3fPe3N5sHzbEbCp4kUefDwdXqqJjYxMaKmKqPb
DT records	QmaoMrCgT9RfyJRqy2yJHhUtux7pDssmoAHgTWuBLb6gqC
Session logs	QmW1CX5yccuS4PXDCEUDkctzKYXnKYJiHSofPMUXTRRpDQ

Fig. 7. Output logs showing successful execution of *mintDTNFT()* function.

updates to assess their functionality and adherence to the system requirements. The main goal is to ensure the accurate deployment and execution of the functions and algorithms outlined in Section 5, and verify that the outcomes align with expected behavior.

For testing purposes, SCs are deployed on the Sepolia testnet. Table 2 lists the Ethereum addresses of the deployed SCs and the participants involved, which serves as a reference during the testing phase. Additionally, Table 3 provides transaction hashes for all executed functions to enable readers to review transaction details on the Sepolia testnet explorer.[2]

6.1. Registering users and minting NFTs

The registration process begins when a user calls the *requestUserRegistration()* function. To call this function, the user must own a DT, and both the DT metadata and its records must be uploaded to the IPFS and passed to the function. Table 4 provides the IPFS hash values for all uploaded documents. Once the user submits the request, the SC owner, typically a regulator from the hospital, initiates the minting process by invoking the *mintDTNFT()* function of the NFTManager SC. This function is restricted to the SC owner, and any unauthorized attempt to call it will result in an error. Fig. 7 displays the event confirming successful registration and NFT minting which includes the user's address and the NFT token ID.

6.2. Accessing the metaverse

Once an NFT is minted for the user, they are granted access to the metaverse. The metaverse is stored on the IPFS for enhanced security and immutability, and the user accesses it through the IPFS hash provided in Table 4. The users connect their wallet, and they are granted access to the metaverse only if they own a valid NFT, as illustrated in Figs. 8 and 9. Once authenticated, the DT asset is downloaded from the IPFS and rendered into the main scene.

[2] https://sepolia.etherscan.io/.

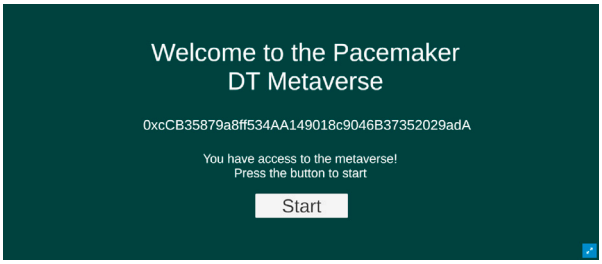


Fig. 8. User with NFT granted access.

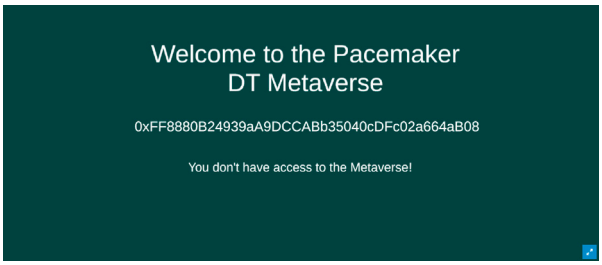


Fig. 9. User without NFT denied access.


```
"operational_data": {
  "heart_rate": {
    "current": 72,
    "historical": [
      {"timestamp": "2024-07-01T08:00:00Z", "value": 70},
      {"timestamp": "2024-07-02T08:00:00Z", "value": 75},
      {"timestamp": "2024-07-03T08:00:00Z", "value": 68}
    ]
  },
  "pacing_events": [
    {"timestamp": "2024-07-01T09:00:00Z", "event": "atrial_pacing"},
    {"timestamp": "2024-07-03T10:30:00Z", "event": "ventricular_pacing"}
  ]
},
"diagnostic_data": {
  "ecg_data": [
    {"timestamp": "2024-07-01T08:00:00Z", "ecg": "ecg_data_1"},
    {"timestamp": "2024-07-02T08:00:00Z", "ecg": "ecg_data_2"}
  ],
  "impedance": [
    {"timestamp": "2024-07-01T08:00:00Z", "value": 600},
    {"timestamp": "2024-07-03T08:00:00Z", "value": 605}
  ],
  "thresholds": {
    "atrial": 1.5,
    "ventricular": 1.0
  }
}
```


Fig. 10. A sample of the pacemaker DT records uploaded to the IPFS for testing.

6.3. Interacting with the LLM

To test the LLM functionality and performance, synthetic pacemaker DT records were uploaded to the IPFS, with a sample displayed in Fig. 10. When initiating an interaction session with the LLM-powered NPC, an AI Assistant is initialized, and the records file is downloaded from IPFS and attached to the AI Assistant, as shown in Fig. 11. The user then sends text-based questions, to which the LLM generates responses based on the attached records. Fig. 12 illustrates how the LLM effectively provides accurate and relevant answers to user's inquiries.

6.4. Handling dynamic updates of DT records

Our system implements dynamic NFTs to enable real-time updates of DT records. The *updateDTRecords()* function is called to associate new DT records with the NFT. The *getDTRecords()* function is continuously called within the system to retrieve the current

```
DT records data downloaded from IPFS and saved to:
/idbfs/d8fa93b4521ec5b7b92fcceb390b5771/PacemakerRecords.json
Records file attached to the AI assistant with ID: file-
xOXMQx9ASK8inedWI2opDuK
```

Fig. 11. DT records successfully downloaded and attached to the AI assistant.

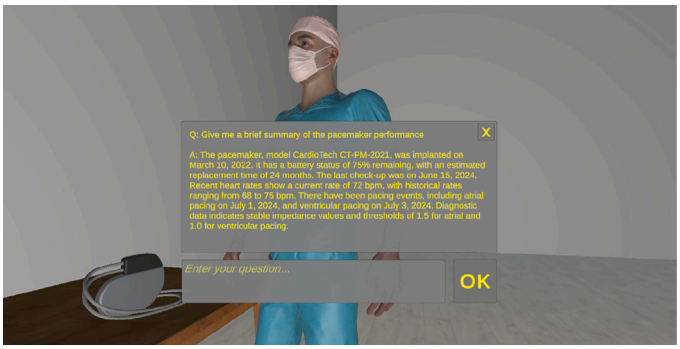


Fig. 12. LLM generating accurate responses to user's questions.

```
Records updated!
Old records: QmaoMrCgT9RfyJRqy2yJHhUtux7pDssmoAHgTWuBLb6gqC
New records: QmbMvkveFBL9916bg66ThWBZ7pbDY9B8wgP5qYqscmV9fM
Records retrieved from IPFS and attached to the LLM successfully
```

Fig. 13. Successful retrieval and attachment of updated DT records.

```
Session logs uploaded to IPFS: QmW1CX5yccuS4PXDCEUDkctzKYXnKYJiHSofPMUXTRRpDQ
Session logs uploaded to the blockchain
TransactionResult:
{
  "to": "0xDA4BAc5620c9CBDD5d4E135774a0CF56F689abc8",
  "from": "0xcCB35879a8ff534AA149018c90468373520 29adA",
  "transactionHash": "0x2acbdcdfac2d9f1f384acf170397e368c2f7150b5ad4e3d97b7f09e0d69c49b1",
  "event": "SessionUploaded",
  "eventSignature": "SessionUploaded(address,uint256)"
}
```

Fig. 14. Output logs showing successful execution of *uploadSession()* function.

records. If the retrieved records differ from those originally attached to the model, the updated records are fetched from IPFS and integrated into the LLM, as shown in Fig. 13. This allows the LLM to provide up-to-date responses based on the latest DT records to ensure accurate and relevant information.

6.5. Uploading session logs to the IPFS

At the end of each user-NPC interaction session, the session logs are uploaded to the IPFS and can be accessed by registered doctors through the blockchain. A session successfully uploaded with the IPFS hash is presented in Table 4. This hash is securely stored on the blockchain through the *uploadSession()* function, as shown in Fig. 14. To allow doctors to access these sessions, the user calls the *registerDoctors()* function and passes the doctor's address. Once registered, the doctor can use the *getSessionURI()* function to retrieve the session IPFS URI. The error shown in Fig. 15 is displayed when a doctor tries to access a session that is associated with a different user.

7. System evaluation and analysis

In this section, we present a comprehensive analysis of the system's cost, security, and response time. We conduct a detailed cost assessment of SCs and LLM integration, evaluate security measures to ensure data protection and privacy, and analyze response time and scalability to assess the system's efficiency under increasing load.

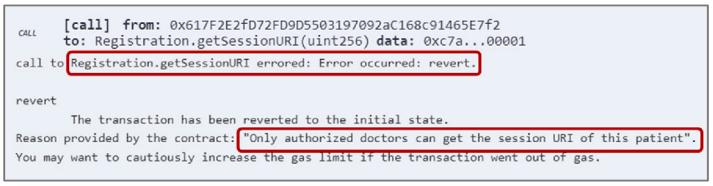


Fig. 15. Error message indicating unauthorized access to the session logs.

Table 5
Execution costs of the functions of the SCs.

Function name	Transaction gas	Transaction fee (Ether)	Cost using Ethereum (USD)	Cost using Polygon (USD)	Cost using zkSync (USD)
requestUserRegistration()	32 453	0.00016	0.392	0.00006	0.00003
mintDTNFT()	265 067	0.00133	3.255	0.00052	0.00021
updateDTRecords()	41 813	0.00021	0.512	0.00008	0.00003
registerDoctors()	55 255	0.00028	0.685	0.00011	0.00004
uploadSession()	127 802	0.00064	1.566	0.00025	0.00010

7.1. Cost analysis

We analyze the financial aspects of integrating the SCs and the LLM into the system. The aim is to find out their impact on overall operational costs.

7.1.1. Smart contracts cost analysis

The SCs in our solution are written in solidity language and are compatible with any EVM-based blockchain. For this analysis, we show the cost implications in case the SCs are deployed on a public Ethereum blockchain, where executing functions incurs gas fees that are paid in Ether. These fees consist of execution gas, which is related to function code execution and state changes, and transaction gas, which covers contract deployment and data transmission [48]. Gas prices are measured in Gwei, and fluctuate with network congestion. Table 5 shows the gas costs associated with various functions utilized in our SCs, along with their conversion into fiat currency (USD) using different blockchain platforms. For this analysis, we considered an average gas price of 5 Gwei and Ether price of $2447 based on the pricing data accessed on August 08, 2024 [49].

The analysis shows variations in gas usage for different functions. The *mintDTNFT()* function has the highest gas usage due to its role in minting an NFT, which incurs high gas fees due to the complex computations, extensive state changes, and additional storage required to create and record new token data. Similarly, uploading session logs using the *uploadSession()* function also incurs high gas usage compared to others because it updates multiple variables and mappings in the SC state. In contrast, the remaining functions have lower gas usage, as they involve simpler operations and fewer resource-intensive changes.

When considering USD costs, they are notably high for some functions due to the current high cost of Ether. For instance, the *mintDTNFT()* function costs $3.255. Although this is relatively high, it should be executed only once during user registration. Conversely, the *uploadSession()* function is called for every session, which may be costly for users frequently interacting with the NPC. These rising Ether costs present a substantial challenge to the public Ethereum network. To address this, since our SCs can be deployed on any EVM-based blockchain, we can utilize Layer 2 (L2) solutions like Polygon [50] and zkSync [51]. L2 solutions are built on top of the main Ethereum network to provide scalable and cost-effective solutions by processing transactions off-chain and then settling them on the main chain [52]. As shown in Table 5, transaction costs on Polygon and zkSync are significantly lower, at a maximum of $0.0006 and $0.0003, respectively. Alternatively, we can deploy the SCs on a private blockchain network and set gas prices to zero. This approach is particularly advantageous for our system as it offers enhanced privacy for personal health records.

7.1.2. LLM cost analysis

The cost of operating an LLM-powered system can significantly affect its scalability, which makes it essential to evaluate its financial aspects. The choice of GPT model plays a pivotal role in determining the overall cost of operation. OpenAI offers several GPT models, each with different capabilities and costs. For this analysis, we consider three models: GPT-3.5 Turbo, GPT-4o Mini, and GPT-4o. These models differ in token processing abilities, response generation, and pricing. The OpenAI Assistant API charges are based on token usage for the selected GPT models, with additional fees for specific tools like the File Search tool used in our system [53]. Token usage depends on the number of tokens processed and includes both input tokens (text sent to the model) and output tokens (text generated by the model), where one token is approximately equivalent to 0.75 words.

Table 6 provides an overview of the pricing for different GPT models, including costs associated with input tokens, output tokens, and file search tool utilization. For our analysis, we assume the model uses 500 input tokens and generates 1000 output tokens per interaction, which is equivalent to approximately 350 and 750 words, respectively, along with a file size of 1 GB. These estimates reflect the expected usage levels that are generally not exceeded during a typical interaction session with the pacemaker DT in our case. The analysis reveals that the costs are minimal and negligible compared to traditional healthcare expenses. Even with the

Table 6
Cost analysis of different GPT models and associated tools.

	GPT-3.5 Turbo	GPT-4o Mini	GPT-4o
Input token cost	$0.00050/1K tokens	$0.00015/1K tokens	$0.00500/1K tokens
Output token cost	$0.00150/1K tokens	$0.00060/1K tokens	$0.01500/1K tokens
File search tool cost	$0.10/1 GB	$0.10/1 GB	$0.10/1 GB
Total cost	$0.1018	$0.1007	$0.1175

```
Compiled with solc
Total number of contracts in source files: 20
Source lines of code (SLOC) in source files: 807
Number of  assembly lines: 0
Number of optimization issues: 0
Number of informational issues: 1
Number of low issues: 0
Number of medium issues: 0
Number of high issues: 0

ERCs: ERC165, ERC721

+-----------------+-------------+---------------+------------+--------------+--------------------+
| Name            | # functions |          ERCS | ERC20 info | Complex code |           Features |
+-----------------+-------------+---------------+------------+--------------+--------------------+
| IERC20Errors    |           0 |               |            |           No |                    |
| IERC1155Errors  |           0 |               |            |           No |                    |
| IERC721Receiver |           1 |               |            |           No |                    |
| ERC721Burnable  |          51 | ERC165,ERC721 |            |           No |           Assembly |
| Strings         |           7 |               |            |           No |           Assembly |
| Math            |          20 |               |            |          Yes |           Assembly |
| SignedMath      |           4 |               |            |           No |                    |
| NFTManager      |          68 | ERC165,ERC721 |            |           No |           Assembly |
| Registration    |          14 |               |            |           No | Tokens interaction |
+-----------------+-------------+---------------+------------+--------------+--------------------+
INFO:Slither:Registration.sol analyzed (20 contracts)
```

Fig. 16. Output of Slither tool showing vulnerability-free smart contracts.

most advanced GPT-4o model, the total cost per session is just $0.1175. This cost will remain very low, even with increased token usage or larger files. Therefore, leveraging LLM-powered interactions in a virtual environment enables patients to access crucial information about their medical devices in real-time without the need for frequent and expensive in-person visits.

7.2. Security analysis

We present the security analysis of the SCs and the LLM model within the proposed system. By evaluating the integration of these cutting-edge technologies, we reveal how they safeguard data protection, uphold privacy, and enhance resilience against a wide range of threats.

7.2.1. Smart contracts security analysis

Our proposed system fundamentally relies on blockchain and SCs. Blockchain technology offers inherent security features that ensure data integrity, consistency, and immutability through cryptographic hashing, consensus mechanisms, and a decentralized ledger that prevents tampering and ensures synchronization across all nodes. However, SCs can be vulnerable to exploitation, which makes it crucial to rigorously test them for bugs and vulnerabilities to maintain the integrity and security of the entire system. To evaluate the security of our SCs, we utilized the Slither analysis tool [54]. Slither is a Python-based static analysis framework designed to detect vulnerabilities in Solidity contracts and assist developers in swiftly identifying and remedying potential issues. During the initial analysis, Slither identified some low-risk issues related to variable naming standards. These were resolved by adhering to best practices for naming conventions. Additionally, a medium-risk issue was detected in the *mintDTNFT()* function, where a reentrancy vulnerability was present. To address this, we utilized the ReentrancyGuard SC from the OpenZeppelin library, which mitigates reentrant call risks by locking the contract during sensitive operations. Following these optimizations, a subsequent analysis confirmed that our SCs were free from any remaining issues, as illustrated in Fig. 16. This analysis ensures that our SCs are robust, secure, and well-suited to support the system's blockchain-based operations without exposing users to potential risks.

7.2.2. LLM security analysis

The security of the LLM model integrated into our system is a critical aspect that ensures the confidentiality, integrity, and availability of patient data. OpenAI models are designed with a comprehensive approach to security, privacy, and operational integrity. They are built with security measures that ensure the protection of user data and adherence to ethical guidelines. For instance, strong encryption methods are used to secure all communications between the user and the LLM models [55]. These methods are used during data transmission and when data is stored to protect user information from unauthorized access. In terms of model vulnerabilities, OpenAI continuously monitors and tests the models to identify and address potential weaknesses [55]. They are also regularly updated to defend against new attacks and vulnerabilities. In addition, OpenAI adheres to strict standards and regulations such as the General Data Protection Regulation (GDPR) and the California Consumer Privacy Act (CCPA). These

Table 7
Response time for key system functionalities, including SCs, IPFS, and LLM interactions.

Functionality	Process description	Response time
Minting user's NFT	Calling SC function	14.13 s
Verifying access control	Calling SC function	0.85 s
Retrieving MDT asset	Calling SC function + retrieving data from IPFS	1.91 s
Retrieving MDT records	Calling SC function + retrieving data from IPFS + attaching retrieved data to the LLM	2.99 s
Interacting with the LLM	Sending API requests to OpenAI Assistant	2.58 s
Uploading session logs	Uploading data to IPFS + calling SC function	12.35 s
Updating DT records	Calling SC function	13.26 s
Retrieving updated records	Detecting data updates + retrieving updated data from IPFS + attaching data to the LLM model	8.00 s

standards and regulations ensure that the models maintain legal and ethical use of the technology at all times [55]. Moreover, the LLM relies on blockchain-based secure and unalterable records to provide accurate and reliable responses. By leveraging NFTs for access control and maintaining data integrity through IPFS-stored DT records, the system guarantees that only authenticated users can interact with the LLM and access these immutable records.

7.3. Response time and scalability analysis

In this subsection, we evaluate the system's performance by assessing the response speed of key system functionalities to ensure seamless user experiences. Table 7 presents the measured response times for various operations, including SC execution, data retrieval from IPFS, and interactions with the LLM.

Among all functionalities, SC operations that involve writing to the blockchain, such as minting NFTs and updating DT records, have the highest response times at 14.13 s and 13.26 s, respectively. This is expected since on-chain transactions require validation through Ethereum's consensus mechanism, which leads to longer execution times. However, these times remain within Ethereum's typical latency range of 12 s to 15 s [49]. In contrast, retrieving and uploading data to IPFS shows significantly faster response times than blockchain transactions since IPFS allows decentralized storage without requiring consensus-based validation. Additionally, the retrieval of MDT assets and records, as well as access verification, involves calling getter functions of the smart contracts. These functions do not result in on-chain transactions or require consensus validation, instead, they execute locally on the Ethereum Virtual Machine (EVM), leading to minimal response times. This distinction explains the significantly lower latency of operations such as verifying access control (0.85 s), retrieving MDT assets (1.91 s), and retrieving MDT records (2.99 s).

Moreover, retrieving and processing newly updated MDT records takes 8 s, as it involves detecting updates, fetching the latest data from IPFS, and integrating it with the LLM. However, this time remains within a few seconds, which ensures efficient synchronization. Additionally, the interaction with the LLM, which involves sending API requests and receiving responses, remains relatively efficient at 2.58 s, ensuring seamless user experiences. Overall, these measured response times are adequately low to ensure a fast, smooth, and responsive user experience. These response times may vary slightly based on network conditions, blockchain congestion, and system load, but they remain within acceptable limits for real-time interactions.

In addition, to evaluate the system's scalability under higher loads, we conducted stress testing to assess its ability to handle an increasing number of concurrent requests. This was achieved through a JavaScript script that simulates multiple users executing various operations simultaneously. The script executes different requests in parallel to measure system response time, and the code is publicly available on GitHub[1]. The chart in Fig. 17 illustrates the average response time of key functionalities, including SC execution, LLM interaction, data retrieval, and IPFS upload, under different load sizes ranging from 1 to 20 concurrent requests. The chart shows that response times remain relatively stable across different load sizes, with only minor variations. This indicates that the system can efficiently scale to accommodate multiple users without significant delays or performance degradation. While SC execution remains the most time-consuming operation due to the blockchain validation process, its response time remains within the expected range and does not increase as the number of requests grows. This suggests that the blockchain network can handle the high loads without congestion or performance degradation. Similarly, data retrieval, LLM interactions, and IPFS uploads exhibit minimal fluctuations in response time, confirming the system's ability to process real-time MDT interactions seamlessly.

This stress testing evaluated scalability under moderate concurrent requests to ensure the system's stability within practical usage scenarios. Large-scale scalability testing, involving hundreds or thousands of users, requires additional resources and is beyond the scope of this paper. However, if scalability issues arise, solutions such as L2 can optimize blockchain transactions, while caching mechanisms and parallel processing can enhance efficiency and reduce delays under higher loads.

8. Discussion

In this section, we evaluate and discuss how the system meets its main objectives. Moreover, we discuss how it can be generalized for wider applications, and we address the key challenges and limitations that need to be considered.

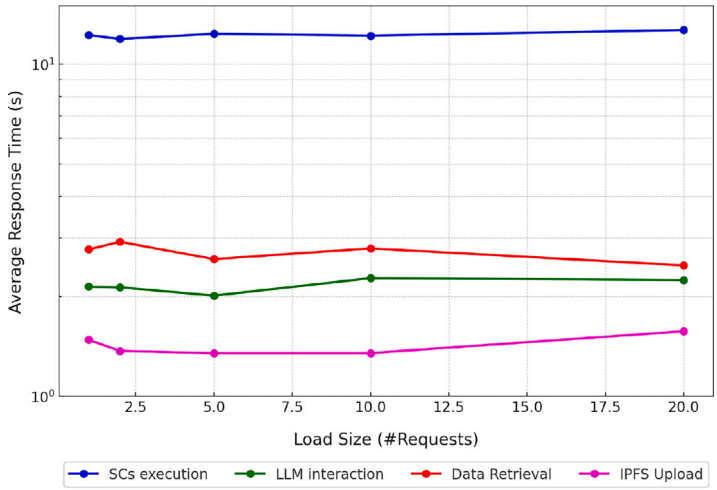


Fig. 17. Average response time of key system functionalities under increasing concurrent requests.

8.1. Evaluation of system objectives

Herein, we discuss how our proposed system meets the following main objectives defined in the paper.

- **Immersive Simulation Experience:** The immersive simulation experience is achieved through the metaverse environment, which is developed using Unity 3D. It allows users to locate their personalized MDTs and interact with an NPC that has access to their records. This approach enhances the users' engagement and interaction with their medical devices.
- **Secure Ownership and Access Control:** We ensure secure ownership and access control by integrating blockchain and NFTs within the metaverse environment. To access the metaverse, users must possess an NFT that represents their MDT. The system checks the SCs to verify NFT ownership to restrict access to authorized users. Additionally, the user's MDT is uploaded to the metaverse during runtime. This is achieved using the NFT metadata retrieved from the IPFS to offer a secure and personalized experience for each user.
- **Intelligent User Interactions:** The LLM-powered NPC is used to achieve intelligent user interaction within the metaverse. When users interact with the LLM, their questions are sent to the OpenAI RAG model. This model retrieves relevant data from the DT records, which are securely downloaded from the IPFS. Based on this data, the LLM generates accurate and contextually relevant responses for the user. This approach enhances user engagement and ensures accurate and real-time interactions within the metaverse. Additionally, dynamic NFTs enable real-time changes to DT records, further improving the user experience by ensuring interactions always reflect the most up-to-date information.
- **Decentralization, Privacy, and Data Integrity:** Our solution ensures security, decentralization, immutability, and data integrity by integrating blockchain, SCs, and IPFS storage. The blockchain provides a decentralized and tamper-proof ledger that records all transactions and interactions immutably at all times. SCs further enhance security and privacy by managing access control. They are used to map DT records to their owners and restrict data access to authorized users. Additionally, the DT data and the metaverse environment reside on the IPFS storage, which distributes information securely across multiple nodes. These technologies together guarantee that data within our system is always protected and trustworthy.

8.2. Generalization

Our proposed system is specifically designed to enhance MDTs in healthcare settings. However, its underlying principles and technologies offer a versatile framework applicable to a broad range of applications. Key features of the system, including LLM-powered interactions, secure ownership and access control, and decentralized data storage, address challenges across various fields. The system architecture, illustrated in Fig. 1, is modular and can be easily adapted for different use cases. For instance, the system can be extended to manage DTs of body organs, where LLM-powered NPCs can facilitate intelligent user interactions, track organ health, and offer personalized recommendations. Additionally, by fine-tuning the LLM with a dataset specific to medical assistance and training it for specific tasks, the LLM-powered NPCs can handle appointment scheduling, medication tracking, and reminders, all while ensuring data privacy and security. SCs can also be modified to include additional functions to be used by the LLM to access and update records and perform specific tasks requested by users.

The principles and technologies of our proposed system can further be adapted for non-healthcare applications. For instance, the system can be used in the manufacturing sector to monitor and interact with machinery and equipment DTs. By training the LLM on "what-if" questions, users can explore new possible scenarios and predict the outcomes of different maintenance operations.

Moreover, SCs can be modified to enable multiple users to access the metaverse and utilize the system's features. This can be achieved by minting access NFTs for several users instead of restricting access to the single owner of the DT. Overall, the ability of our solution to be adapted for different applications demonstrates its potential to enhance DT functionalities.

8.3. Challenges and limitations

The proposed system offers significant advantages for MDTs. However, it also presents several challenges and limitations that need to be addressed. One of the limitations is the need to fine-tune the LLM for MDT-specific applications. Pre-trained LLMs are typically trained on generalized datasets and may not provide precise or accurate answers for medical use cases. Therefore, fine-tuning them on relevant medical question–answer datasets, such as MedQA [56], is essential for specialization. This process involves selecting and preprocessing the dataset to clean, normalize, and structure it into QA pairs, followed by training the desired model using supervised fine-tuning, where it learns to generate accurate medical responses through gradient-based optimization. Finally, continuous evaluation on validation datasets is necessary to ensure the model generalizes well. This enhances the overall system's effectiveness and ensures that the LLM delivers reliable insights and analyses.

Moreover, metaverse-based MDT systems require high network bandwidth and compatible devices for real-time interactions and AI-driven responses, which may pose challenges for users in low-resource settings. This could be addressed by optimizing the metaverse for lower bandwidth usage and providing alternative access methods to accommodate different levels of connectivity. Additionally, adaptive streaming and cloud rendering can reduce computational demands, while a lightweight web-based interface ensures accessibility across various devices, including smartphones.

Lastly, given the sensitive nature of medical data, the system must adhere to strict regulatory standards, such as the Health Insurance Portability and Accountability Act (HIPAA), to ensure patient privacy and data security. To achieve this, strong encryption mechanisms and zero knowledge proofs (ZKPs) could be implemented to further enhance data protection and allow secure access control without exposing sensitive information. Overall, these challenges and limitations must be carefully addressed to ensure the system's feasibility, scalability, and effectiveness.

9. Conclusion

In this paper, we proposed a novel solution to enhance MDTs in the metaverse by leveraging blockchain, NFTs, and LLMs. This approach ensures secure ownership and access control while offering an immersive environment, with LLM-powered NPCs enabling intelligent user interactions. We connected the decentralized storage of IPFS to the blockchain to host the metaverse environment, along with MDT metadata and records, to address the large-scale data problem. We developed SCs and presented algorithms, along with their implementation and testing details. We provided a cost analysis demonstrating the minimal costs associated with LLM integration. We presented the security analysis to confirm that our SCs are free of vulnerabilities and secure against well-known attacks and threats. Moreover, our analysis demonstrated minimal system response times, with blockchain transactions requiring the highest execution time, while stress testing verified the system's scalability under higher concurrent requests. We showed the novelty of our solution by comparing it with other existing solutions and discussed its potential for generalization in other healthcare and non-healthcare applications. While our proposed solution provides a strong foundation, large-scale validation and clinical deployment remain beyond the scope of this paper and will be explored in future work. Moreover, further research will focus on fine-tuning the LLM with a specialized QA dataset specific to medical devices, and deploying the solution on a private blockchain to enhance security, privacy, and cost efficiency.

CRediT authorship contribution statement

Ruba Islayem: Writing – original draft, Software, Formal analysis, Data curation. **Ahmad Musamih:** Writing – review & editing, Validation, Supervision, Methodology, Conceptualization. **Khaled Salah:** Writing – review & editing, Validation, Supervision, Project administration, Methodology, Funding acquisition, Conceptualization. **Raja Jayaraman:** Writing – review & editing, Validation, Conceptualization. **Ibrar Yaqoob:** Writing – review & editing, Validation.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgment

This publication is based upon work supported by the Khalifa University of Science and Technology under Award No. RIG-2023-049. During the preparation and review of this work, the authors used ChatGPT to improve readability and language. After using this tool, the authors reviewed and edited the content as needed and took full responsibility for the content of the publication.

Data availability

The data and code supporting the findings of this study are openly available on GitHub.

References

[1] T. Sun, X. He, X. Song, L. Shu, Z. Li, The digital twin in medicine: A key to the future of healthcare? Front. Med (Lausanne) 9 (2022) http://dx.doi.org/10.3389/fmed.2022.907066.

[2] T. Sun, X. He, Z. Li, Digital twin in healthcare: Recent updates and challenges, Digit. Heal. 9 (2023) 205520762211496, http://dx.doi.org/10.1177/20552076221149651.

[3] T. Erol, A.F. Mendi, D. Dogan, The digital twin revolution in healthcare, in: 2020 4th International Symposium on Multidisciplinary Studies and Innovative Technologies, ISMSIT, IEEE, 2020, pp. 1–7, http://dx.doi.org/10.1109/ISMSIT50672.2020.9255249.

[4] H. Yang, Z. Jiang, Decision support for personalized therapy in implantable medical devices: A digital twin approach, Expert Syst. Appl. 243 (2024) 122883, http://dx.doi.org/10.1016/j.eswa.2023.122883.

[5] B.R. Barricelli, E. Casiraghi, D. Fogli, A survey on digital twin: Definitions, characteristics, applications, and design implications, IEEE Access 7 (2019) 167653–167671, http://dx.doi.org/10.1109/ACCESS.2019.2953499.

[6] M. Attaran, B.G. Celik, Digital twin: Benefits, use cases, challenges, and opportunities, Decis. Anal. J. 6 (2023) 100165, http://dx.doi.org/10.1016/j.dajour.2023.100165.

[7] U. Okwudilli Mathew, R. Lopez Rosa, O. Oluyemisi Adenike, D. Zegarra Rodriguez, Advancing cybersecurity use of sensitive data in electronic healthcare system: A review of privacy and regulations, 2024.

[8] S.K. Bell, et al., Frequency and types of patient-reported errors in electronic health record ambulatory care notes, JAMA Netw Open 3 (6) (2020) e205867, http://dx.doi.org/10.1001/jamanetworkopen.2020.5867.

[9] Patient Survey Shows Unresolved Tension over Health Data Privacy, American Medical Association, 2022, (Accessed, 17 March 2025) [Online]. Available: https://www.ama-assn.org/press-center/press-releases/patient-survey-shows-unresolved-tension-over-health-data-privacy.

[10] R. Shahid, M. Shoker, L.M. Chu, R. Frehlick, H. Ward, P. Pahwa, Impact of low health literacy on patients' health outcomes: a multicenter cohort study, BMC Health Serv. Res. 22 (1) (2022) 1148, http://dx.doi.org/10.1186/s12913-022-08527-9.

[11] Billions left behind on the path to universal health coverage, World Health Organization, 2025, (Accessed 17 March 2025) [Online]. Available: https://www.who.int/news/item/18-09-2023-billions-left-behind-on-the-path-to-universal-health-coverage.

[12] T. Norris, D. Adjaye-Gbewonyo, L. Bottoms-Mcclain, Early release of selected estimates based on data from the 2023 national health interview survey, 2024, [Online]. Available: https://www.cdc.gov/flu/fluvaxview.

[13] N. Kshetri, Blockchain and sustainable supply chain management in developing countries, Int. J. Inf. Manage. 60 (2021) 102376, http://dx.doi.org/10.1016/j.ijinfomgt.2021.102376.

[14] W. Rehman, H. e Zainab, J. Imran, N.Z. Bawany, NFTs: Applications and challenges, in: 2021 22nd International Arab Conference on Information Technology, ACIT, IEEE, 2021, pp. 1–7, http://dx.doi.org/10.1109/ACIT53391.2021.9677260.

[15] M. Alizadeh, K. Andersson, O. Schelen, Efficient decentralized data storage based on public blockchain and IPFS, in: 2020 IEEE Asia-Pacific Conference on Computer Science and Data Engineering, CSDE, IEEE, 2020, pp. 1–8, http://dx.doi.org/10.1109/CSDE50874.2020.9411599.

[16] H. Wan, et al., Building LLM-based AI agents in social virtual reality, in: Extended Abstracts of the CHI Conference on Human Factors in Computing Systems, ACM, New York, NY, USA, 2024, pp. 1–7, http://dx.doi.org/10.1145/3613905.3651026.

[17] S. Mystakidis, Metaverse, Encyclopedia 2 (1) (2022) 486–497, http://dx.doi.org/10.3390/encyclopedia2010031.

[18] Y. Wang, Z. Su, N. Zhang, R. Xing, A survey on metaverse: Fundamentals, security, and privacy, IEEE Commun. Surv. & Tutorials 25 (1) (2022) http://dx.doi.org/10.48550/arXiv.2203.02662.

[19] G.A. Oliva, A.E. Hassan, Z.M. Jiang, An exploratory study of smart contracts in the ethereum blockchain platform, Empir. Softw. Eng. 25 (3) (2020) 1864–1904, http://dx.doi.org/10.1007/s10664-019-09796-5.

[20] W. Entriken, D. Shirley, J. Evans, N. Sachs, ERC-721: Non-fungible token standard, ethereum improvement proposals, 2018, (Accessed: 11 March 2025) [Online]. Available: https://eips.ethereum.org/EIPS/eip-721.

[21] W. Radomski, A. Cooke, P. Castonguay, J. Therien, E. Binet, R. Sandford, ERC-1155: Multi token standard, ethereum improvement proposals, 2018, (Accessed: 11 March 2025) [Online]. Available: https://eips.ethereum.org/EIPS/eip-1155.

[22] M. Solouki, S.M.H. Bamakan, An in-depth insight at digital ownership through dynamic NFTs, Procedia Comput. Sci 214 (2022) 875–882, http://dx.doi.org/10.1016/j.procs.2022.11.254.

[23] X. Luo, D. Liu, F. Dang, H. Luo, Integration of LLMs and the physical world: Research and application, in: ACM Turing Award Celebration Conference 2024, ACM, New York, NY, USA, 2024, pp. 1–5, http://dx.doi.org/10.1145/3674399.3674402.

[24] W. Fan, et al., A survey on RAG meeting LLMs: Towards retrieval-augmented large language models, in: Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, ACM, New York, NY, USA, 2024, pp. 6491–6501, http://dx.doi.org/10.1145/3637528.3671470.

[25] Y. Gao, et al., Retrieval-augmented generation for large language models: A survey, 2023.

[26] T.A. SMITH, et al., 1058-P: The use of metabolic digital twins to personalize mealtime insulin dosing in type 1 diabetes clinical management, Diabetes 68 (Supplement_1) (2019) http://dx.doi.org/10.2337/db19-1058-P.

[27] J. Corral-Acero, et al., The 'digital twin' to enable the vision of precision cardiology, Eur. Heart J. 41 (48) (2020) 4556–4564, http://dx.doi.org/10.1093/eurheartj/ehaa159.

[28] Y. Liu, et al., A novel cloud-based framework for the elderly healthcare services using digital twin, IEEE Access 7 (2019) 49088–49101, http://dx.doi.org/10.1109/ACCESS.2019.2909828.

[29] A.E.L. Azzaoui, T.W. Kim, V. Loia, J.H. Park, Blockchain-based secure digital twin framework for smart healthy city, 2021, pp. 107–113, http://dx.doi.org/10.1007/978-981-15-9309-3_15.

[30] S. Amofa, et al., Blockchain-secure patient digital twin in healthcare using smart contracts, PLOS One 19 (2) (2024) e0286120, http://dx.doi.org/10.1371/journal.pone.0286120.

[31] S. Huang, G. Wang, Y. Yan, X. Fang, Blockchain-based data management for digital twin of product, J. Manuf. Syst. 54 (2020) 361–371, http://dx.doi.org/10.1016/j.jmsy.2020.01.009.

[32] C. Mandolla, A.M. Petruzzelli, G. Percoco, A. Urbinati, Building a digital twin for additive manufacturing through the exploitation of blockchain: A case analysis of the aircraft industry, Comput. Ind. 109 (2019) 134–152, http://dx.doi.org/10.1016/j.compind.2019.04.011.

[33] S.A. Gebreab, et al., NFTs for accessing, monetizing, and teleporting digital twins and digital artifacts in the metaverse, Comput. Commun. 228 (2024) 107965, http://dx.doi.org/10.1016/j.comcom.2024.107965.

[34] H.R. Hasan, et al., Non-fungible tokens (NFTs) for digital twins in the industrial metaverse: Overview, use cases, and open challenges, Comput. Ind. Eng. 193 (2024) 110315, http://dx.doi.org/10.1016/j.cie.2024.110315.

[35] S.A. Gebreab, H.R. Hasan, K. Salah, R. Jayaraman, NFT-based traceability and ownership management of medical devices, IEEE Access 10 (2022) 126394–126411, http://dx.doi.org/10.1109/ACCESS.2022.3226128.
[36] S.K. Jagatheesaperumal, M. Rahouti, Building digital twins of cyber physical systems with metaverse for industry 5.0 and beyond, IT Prof 24 (6) (2022) 34–40, http://dx.doi.org/10.1109/MITP.2022.3225064.
[37] D. Mourtzis, Digital twin inception in the era of industrial metaverse, Front. Manuf. Technol. 3 (2023) http://dx.doi.org/10.3389/fmtec.2023.1155735.
[38] M.B. Jamshidi, M. Ebadpour, M.M. Moghani, Cancer digital twins in metaverse, in: 2022 20th International Conference on Mechatronics - Mechatronika, ME, IEEE, 2022, pp. 1–6, http://dx.doi.org/10.1109/ME54704.2022.9983328.
[39] O. Moztarzadeh, et al., Metaverse and healthcare: Machine learning-enabled digital twins of cancer, Bioeng. 10 (4) (2023) 455, http://dx.doi.org/10.3390/bioengineering10040455.
[40] O. Moztarzadeh, et al., Metaverse and medical diagnosis: A blockchain-based digital twinning approach based on MobileNetV2 algorithm for cervical vertebral maturation, Diagn. 13 (8) (2023) 1485, http://dx.doi.org/10.3390/diagnostics13081485.
[41] R. Prakash, T. Thomas, Towards secure AI-driven industrial metaverse with NFT digital twins, in: 2025 17th International Conference on COMmunication Systems and NETworks, COMSNETS, IEEE, 2025, pp. 721–729, http://dx.doi.org/10.1109/COMSNETS63942.2025.10885606.
[42] S. Sai, A. Gaur, V. Hassija, V. Chamola, Artificial intelligence empowered digital twin and NFT-based patient monitoring and assisting framework for chronic disease patients, IEEE Internet Things Mag. 7 (2) (2024) 101–106, http://dx.doi.org/10.1109/IOTM.001.2300138.
[43] Y. Huang, J. Zhang, X. Chen, A.H.F. Lam, B.M. Chen, From simulation to prediction: Enhancing digital twins with advanced generative AI technologies, in: 2024 IEEE 18th International Conference on Control & Amp; Automation, ICCA, IEEE, 2024, pp. 490–495, http://dx.doi.org/10.1109/ICCA62789.2024.10591881.
[44] M. Swan, T. Kido, E. Roland, R.P. Dos Santos, AI health agents: Pathway2vec, ReflectE, Category Theory, Longev. Proc. the AAAI Symp. Ser. 3 (1) (2024) 426–433, http://dx.doi.org/10.1609/aaaiss.v3i1.31249.
[45] Remix - ethereum IDE & community, remix, 2019, (Accessed 07 August 2024) [Online]. Available: https://remix-project.org/.
[46] Thirdweb unity SDK, thirdweb docs, 2023, (Accessed 07 August 2024) [Online]. Available: https://portal.thirdweb.com/unity/v4.
[47] Assistants api overview, openai, 2024, (Accessed 07 August 2024) [Online]. Available: https://platform.openai.com/docs/assistants/overview.
[48] Z. Zheng, et al., An overview on smart contracts: Challenges, advances and platforms, Future Gener. Comput. Syst. 105 (2020) 475–491, http://dx.doi.org/10.1016/j.future.2019.12.019.
[49] Ethereum gas tracker, etherscan, 2018, (Accessed 07 August 2024) [Online]. Available: https://etherscan.io/gastracker.
[50] Web3, Aggregated, polygon, 2021, (Accessed 11 March 2025) [Online]. Available: https://polygon.technology/.
[51] The Elastic Network, Powered by zksync, zksync, 2024, (Accessed 11 March 2025) [Online]. Available: https://www.zksync.io/.
[52] W. Zhang, T. Anand, Ethereum architecture and overview, in: Blockchain and Ethereum Smart Contract Solution Development, A Press, Berkeley, CA, 2022, pp. 209–244, http://dx.doi.org/10.1007/978-1-4842-8164-2_6.
[53] Pricing | openai, openai, 2024, (Accessed 24 August 2024) [Online]. Available: https://openai.com/pricing.
[54] Slither: Static analyzer for solidity and vyper, GitHub, 2024, (Accessed 09 August 2024) [Online]. Available: https://github.com/crytic/slither.
[55] Security & privacy | openai, openai, 2024, (Accessed 25 August 2024) [Online]. Available: https://openai.com/security-and-privacy/.
[56] D. Jin, E. Pan, N. Oufattole, W.-H. Weng, H. Fang, P. Szolovits, What disease does this patient have? A large-scale open domain question answering dataset from medical exams, 2020.