Bachelor's Thesis

Information & Communications Technology

2025

Jani Sorvoja

Integrating Smart Contracts & Non-Fungible Tokens into a Video Game

Bachelor’s Thesis | Abstract

Turku University of Applied Sciences

Information & Communications Technology

2025 | 54 pages

Jani Sorvoja

Integrating Smart Contracts & Non-Fungible Tokens into a Video Game

The integration of blockchain technology into video games introduces new models of digital ownership which enable players to truly own and manage their in-game assets. This thesis implements non-fungible tokens (NFTs) and smart contracts within a game environment by developing a prototype of a collectible card game. Each card in the game is represented as a ERC-721 token, minted and traded with other players via Ethereum smart contracts on Sepolia testnet.

The prototype, built in Unity and integrated with blockchain using Nethereum and Infura, allows players to mint unique cards, view their collection, and trade with other players. The trading mechanisms include on-chain logic to verify ownership and ensure that swaps between players are conducted securely.

The project demonstrates feasibility and challenges of using blockchain in games, which includes smart contract design, token approval mechanisms, gas limitations, and decentralized user authentication. While this prototype leaves out full gameplay mechanics, it successfully creates a foundation for blockchain-based asset management. The results suggest that blockchain can enhance transparency and player agency in video games, although scalability and UX remain for future development.

Keywords:

Blockchain, blockchain technology, NFT, non-fungible token, smart contract

Opinnäytetyö (AMK) | Tiivistelmä

Turun ammattikorkeakoulu

Tieto- ja viestintätekniikka

2025 | 54 sivua

Jani Sorvoja

Älysopimusten ja NFT:iden sisällyttäminen peleihin

Lohkoketjuteknologian integrointi videopeleihin tuo mukanaan uudenlaisia digitaalisen omistajuuden malleja, joiden avulla pelaajat voivat aidosti omistaa ja hallita pelinsisäisiä omaisuuksiaan. Tämä opinnäytetyö implementoi NFT:eitä ja älysopimuksia peliympäristöön kehittämällä prototyypin keräilykorttipelistä. Työssä käytetyt pelikortit olivat ERC-721-standardin mukainen NFT-rahake, jonka pystyi luomaan ja vaihtamaan muiden pelaajien kanssa Ethereumin älysopimusten avulla Sepolia-testiverkossa.

Prototyyppi rakennettiin Unityllä ja integroitu lohkoketjuun käyttäen Nethereum-kirjastoa sekä Infura etäkäsittelykutsua. Se salli pelaajien luoda ainutlaatuisia pelikortteja, katsoa heidän kokoelmaansa sekä vaihtaa korttejaan muiden pelaajien kanssa. Vaihtomekanismi perustui älysopimuksiin, jotka varmistavat omistajuuden ja suorittavat vaihdon turvallisesti lohkoketjussa.

Opinnäytetyö osoittaa lohkoketjuteknologian olevan sovellettavissa peleihin, mutta se toi mukanaan haasteita, kuten älysopimuksien suunnittelun, NFT:iden hyväksyntäprosessit, palkkiorajoitukset ja hajautetun tunnistautumisen. Vaikka tässä työssä ei keskitytty pelimekaniikkojen rakentamiseen, se luo toimivan pohjan lohkoketjupohjaiselle omaisuudenhallinnalle. Tulokset osoittavat, että lohkoketjujen käyttö peleissä voi lisätä omistajuuden läpinäkyvyyttä ja pelaajan vaikutusmahdollisuuksia pelissä. Skaalautuvuus ja käyttökokemuksien parantaminen vaativat jatkokehitystä.

Asiasanat:

Lohkoketju, lohkoketjuteknologia, NFT, älysopimus

Contents

Figures

List of abbreviations

BFT	Byzantine Fault Tolerance
DAG	Direct Acrylic Graph
DPoS	Delegated Proof of Stake
ERC	Ethereum Request for Comment
ID	Identifier
IDE	Integrated Development Environment
IPFS	InterPlanetary File System
JSON	JavaScript Object Notation
NFT	Non-Fungible Token
PBFT	Practical Byzantine Fault Tolerance
P2P	Peer-to-Peer
PoS	Proof of Stake
PoW	Proof of Work
SDK	Software Development Kit
UI	User Interface
URI	Uniform Resource Identifier
UX	User Experience

1 Introduction

Blockchain came to be with the realization of Bitcoin on October 30th, 2008, when a person with the pseudonym Satoshi Nakamoto published a paper titled “Bitcoin: A Peer-to-Peer Electronic Cash System” [1] on cryptography mailing list. This paper explained the basic principles of Bitcoin and blockchain technology. Bitcoin is a peer-to-peer (P2P) [2] electronic cash system meaning anyone can join the Bitcoin network at any given time and transact within the network.

Nakamoto’s paper paved way for creation of payment networks which were not reliant on the traditional financial institutions’ trust-based model. Bitcoin replaces the trust model with a system based on cryptographic proof, and thus the need for a trusted third party is no longer necessary. Cryptographic measures ensure that the transactions within the network are computationally impractical to reverse. [1]

For the double-spending problem, a problem where a user tries to spend the same cryptocurrency units twice making duplicate transactions [3], a proposed solution was to use a P2P distributed timestamp server to generate computational proof of the chronological order of transactions [1]. This P2P distributed timestamp server is what we these days usually refer to as blockchain, meaning that blockchains are essentially records of every transaction from their birth to present day.

Transactions are organized into blocks which are then, in turn, linked back to the previous block of transactions forming a chain of blocks [2]. Blockchains can be visualized as a vertical stack where the first created block, a genesis block, is the bottom block of the chain. Newer blocks are stacked on top of older blocks and the highest up in the vertical stack is the most recent block.

Blockchain consists of four major supporting technologies: distributed ledger technology, consensus algorithm, secure cryptographic protocol and smart contracts [2]. Distributed ledger technology is used to store and verify the data, consensus algorithms are used to generate and update the data inside the distributed ledger, cryptography technologies are used to ensure the integrity of the data transmission and access, while smart contracts are used to programming and operating the data [2]. Use of these technologies, by nature, gives blockchain systems qualities of decentralization, and a tamper-resistant protocol in which its participants can execute, verify and audit digital transactions securely with each other [4].

Regardless of these common qualities, blockchains can vary widely depending on their use case. The system qualities blockchains often are linked to, are mobile concepts and not absolute. As indicated in many research papers regarding blockchain technology [5], [6], [2], improvements in these three goals of blockchain systems (decentralization, scalability and privacy), usually referred to as the “blockchain trilemma” [7], are often met with compromises in one or more sections of this trilemma. It has been scientifically proven by distributed database researchers to be impossible to build systems that meet all these three requirements [8]. Hence, until this trilemma is solved, blockchain-based applications must resort to neglecting some areas to make blockchain technology more suitable to their applications’ needs.

As a conclusion from this trilemma, financial applications call for prioritizing their security and decentralization over scalability or efficiency. However, for some other use-cases, such as gaming, performance of the blockchain could be more beneficial for faster transaction registering on-chain.

Apart from cryptocurrencies, the main domains of studies regarding blockchain technology, are in the following sectors: supply chain, education, agriculture and finance [9]. It is suggested for businesses and organizations to start small when it comes to the assimilation of the technology to existing or new processes and offerings [10].

In gaming, interest in blockchain-based games has been on the rise since the introduction of non-fungible tokens (NFTs) [11]. This growing interest has been seen on Google search trends as well [12] [13].

NFTs are technically speaking enabled by smart contracts. Smart contracts are protocols of code that execute on a blockchain automatically if conditions are met, to perform trackable and irreversible transactions. [14]

By utilizing blockchain technology, game assets can exist independently of games as NFTs, reducing the risk of losing digital items due to shutdowns of game servers or policy changes by game developers. Players can trade, sell, and transfer these unique identification codes and metadata, commonly known as NFTs, which correspond to off-chain virtual assets like digital art. As a result of unique metadata NFTs contain, they can not be traded or exchanged at equivalent like fungible tokens can be. The metadata also links the NFTs to a specific crypto-wallet address solidifying the ownership into it. [15]

In modern-day video games, players often rely on centralized game servers and developer-controlled economies. In other words, this means that players often do not have the true ownership of their in-game assets such as items, skins, characters or other collectibles which exist within these closed ecosystems. If a game shuts down, players usually lose all their in-game assets. With some game developers, their games' in-game purchases are considered as services and non-refundable purchases, unless it is required by applicable law. Lack of true ownership of digital assets raises questions how blockchain technology, specifically NFTs and smart contracts, can address these challenges for the consumers. [16]

This thesis investigates how NFTs and smart contracts can be efficiently used to represent in-game assets in a blockchain, and what challenges and benefits they might bring to gaming. To explore this, a prototype is developed demonstrating the basic functionalities of NFT minting, wallet-based authentication, asset verification, and trading.

The focus of this thesis is on blockchain integration and will not explore gameplay mechanics. As a result, the prototype will not include a full game loop.

2 Blockchain technology in game industry

This chapter investigates the fundamental principles of blockchain technology and explores its applications, benefits, and challenges in the game industry. Blockchain technology has gained some attention in the gaming landscape due to its capabilities of bringing decentralized ownership to digital assets and transparent transactions in the form of permissionless and auditable blockchains. Traditional gaming ecosystems depend on centralized game servers and developer- or publisher-controlled economies limiting players' control over their in-game assets. Blockchain integration to games could include verifiable ownership of digital assets and player-run marketplaces, allowing new economic models by not having centralized entity control over in-game assets' pricing.

2.1 Introduction to blockchain technology

This section covers the fundamentals of blockchain, including distributed ledger technology, consensus mechanisms, cryptographic protocols, and smart contracts. Together, these components make up the collection of technologies commonly referred to as blockchain. That collection of technologies usually consists of well-known computer science mechanisms and cryptography together with record keeping concepts.

2.1.1 Distributed ledger technology

In blockchain technology, used solution for data storage is called distributed ledger. A Distributed ledger is a database of assets that are jointly governed and shared in a network composed of multiple participants, with usually widely different geographic locations and backgrounds. From computer science perspective, a ledger is a data structure which contains information and transactions. These transactions can happen between multiple parties exchanging funds or digital items. In blockchain, transactions are organized into blocks, which are then linked to each other as logical chains, forming a chain of blocks. This data structure makes blockchains ever-growing databases that build on top of older information that is accumulated over time. [2]

2.1.2 Consensus mechanisms

As a result of blockchains' distributed nature, their ledgers rely on a P2P network where each node (network participant) stores a copy of valid ledger status and the unvalidated data set. This data set will be added as a new block to the ledger. For the blockchain network to function properly, nodes need to have a decided method on agreeing to a ledger's content and the process of grouping transactions into blocks. This is achieved by consensus mechanisms which validate the correct order of data for nodes to follow. [17]

In blockchain technology, there are varying methods for implementing blockchain's consensus. A consensus mechanism is the key technology of a blockchain and directly affects its efficiency, scalability, and resource consumption [17]. In the prototype implementation of this thesis, we will be utilizing Ethereum and its Proof-of-Stake (PoS) [18] consensus algorithm. Ethereum moved from Proof-of-Work (PoW) [18] algorithm to PoS on September 15th 2022 in order to reduce network's energy consumption by approximately 99.95% [19].

The first successful implementation of PoW in a blockchain was Bitcoin [1]. While Bitcoin's PoW was not the first ever PoW implementation, as it was proposed by Adam Back to throttle systematic abuse of un-metered internet resources in 1997 [20], it was a clever implementation for using computational power to secure the transactions. Since then, PoW has been widely used in different cryptocurrency applications as their consensus algorithm. In PoW, the consensus mechanism ensures that transactions are added to blockchain securely by requiring network participants, miners, to solve a computational puzzle. The computational puzzle revolves around figuring out a random number, nonce, that is added to a block header. With a correct nonce, the hash value of the block is either lower than or equal to the difficulty target. For instance, if the difficulty target has 32 leading zeros, miners need an average of 4.3 billion attempts to find a valid hash. Once a miner finds a valid hash for a block, it is broadcasted to the network, miners earn a block reward, and it is added to nodes copies of the ledger [2]. [5]

As referred in Figure 1, there are no miners in PoS. In the miners place are validators which, instead of computing power, compete with the amount of cryptocurrency they have staked. This means that the more coins a validator has, the easier it will be to find a correct answer for new block creation. Consequently, PoS has eliminated the need for high energy consumption by the network and can shorten the time needed for reaching consensus. However, in extreme cases PoS can bring centralized results which are unwanted in most blockchain applications. [2]

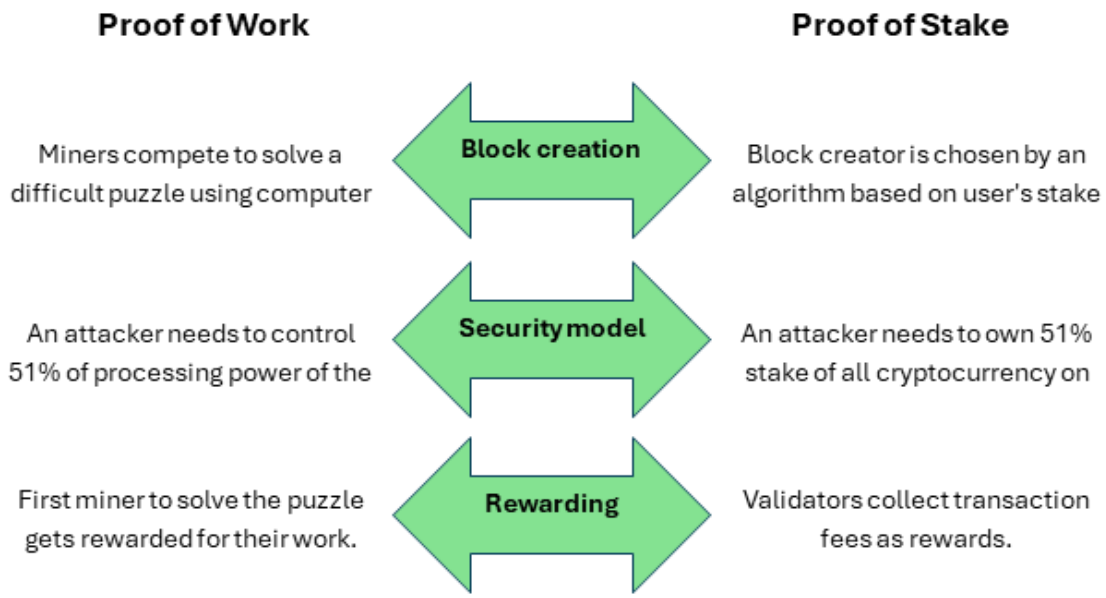

Figure 1. Proof of Work vs. Proof of Stake.

This thesis examines mainly blockchain applications which use either PoS or PoW. Other notable consensus mechanisms include Delegated Proof of Stake (DPoS) [5], Byzantine Fault Tolerance (BFT) [21], Practical Byzantine Fault Tolerance (PBFT) [21] and Direct Acrylic Graph (DAG) [22].

2.1.3 Cryptographic protocols

As visualized in Figure 2, asymmetric encryption is a method that uses a pair of mathematically related keys for encryption and decryption in order to secure communication. Hence, this method is also known as public-key cryptography. A public key is available to anyone and is used to encrypt a message while only its intended recipient can decrypt the message using their own private key. Private keys are intended to be kept confidential by their owner and should not be shared. One big advantage asymmetric encryption has over symmetric encryption is the elimination of the need for key distribution. In asymmetric encryption public keys are exchanged via public-key servers. The disclosure of public keys is not detrimental to the security of encrypted messages at the time, because public keys cannot be used to derive private keys [23]. In blockchain applications, asymmetric encryption is used to verify transactions [2] and in wallet address derivation [24].

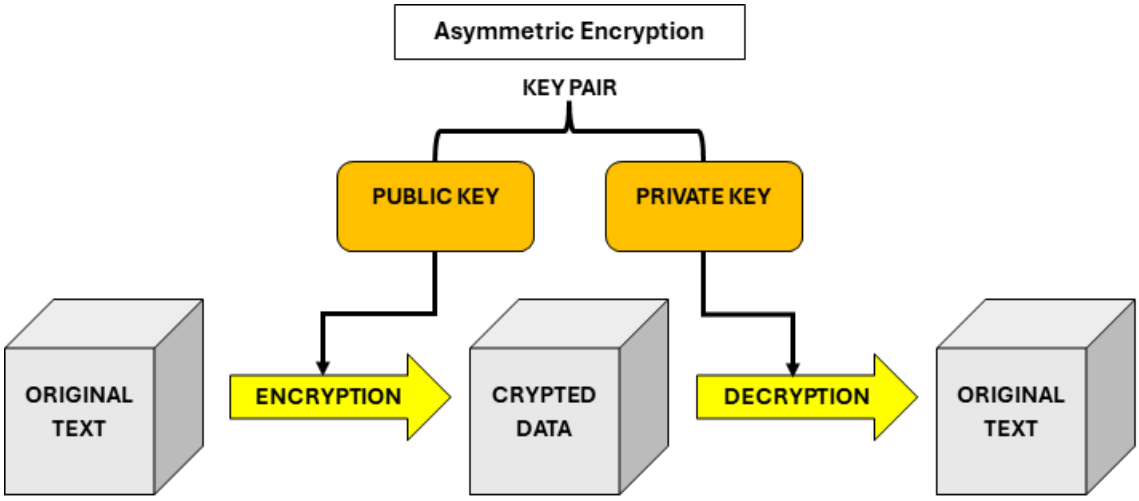

Figure 2. Asymmetric encryption.

In blockchain, transactions are stored in a block in the form of a Merkle Tree: a hash tree which can quickly summarize and verify the integrity of the data it contains [25]. In Merkle Tree method, transactions are transformed into hashes by using different hash functions (for instance SHA-256 or Keccak-256). The hashes of the nodes higher up in Merkle Tree hierarchy level are hashed with the hashes of the lower-level nodes. If one of these hashes are tampered with, it changes the whole hash tree's outcome, usually referred to as root hash. This way it is easy to confirm whether blocks have been tampered with by verifying the root hash with no need to verify all individual transaction hashes [26].

On top of verification benefits, with the use of Merkle Tree method blockchain participants can save a lot of disk space due to qualities of Merkle Tree. They allow older blocks' transaction lists to be pruned, effectively you are cutting off branches of the tree by having them contain only block headers with relevant information to previous blocks and the root hash which is unique to the list of transactions it contains [1]. This can be useful for network participants which do not have as many resources to justify downloading the whole blockchain.

2.1.4 Smart contracts

Nick Szabo was first to define smart contract as "a computerized transaction protocol that executes the terms of a contract." in his 1994 publication [27]. Szabo explained smart contracts as computerized transaction protocols which executes the terms of a contract. The objectives for smart contracts were to satisfy common contractual conditions, to minimize malicious and accidental exceptions, and to minimize the need for trusted third parties. In its essence,

smart contracts are program code which is stored in a blockchain after being signed by all parties. The program code determines several predefined states, conversion rules, trigger conditions, and response operations, among other elements [28]. [27]

In conclusion, both Bitcoin and Ethereum hosts several smart contracts. However, Ethereum's smart contracts are more complex by nature allowing more complicated smart contracts on it [29]. Of course, this complexity also means more possible attack vectors for malicious attackers to exploit.

2.2 Blockchain use in gaming

In blockchain games, data for ownership of an asset is offloaded to on-chain [30]. On the contrary, in traditional games players' in-game possessions are stored on centralized servers in control of game developers [31]. Item drop rates, whether its in-game boss loot or loot box rewards, are often not disclosed transparently to players [31]. Players' in-game assets are tied to one game offering no NFT-enabled interoperability [32] of their possessions. By nature, blockchain can offer more transparency and fraud prevention to games since it is easier to detect counterfeit items in digital marketplaces using NFTs [31] [33]. The main issues in traditional games that blockchain-based games aim to fix include lack of true ownership, limited transparency and item scarcity, poor interoperability, and insufficient protection against fraud and security threats. [32]

Pixudi is a traditional race board game built on blockchain technology. There players compete each other by overcoming obstacles, completing missions, and collecting in-game assets. In Pixudi, blockchain integration comes in to enhance transparency, fairness, and digital ownership of in-game assets. For instance, they have used smart contracts to ensure that each roll on the dice is fully randomized, all in-game purchases, wins and losses can be viewed on-chain, and all in-game items are minted as NFTs on their partner blockchains granting players verifiable ownership of their assets. [34]

Pixels on the other hand is still a game that is being developed [35]. It launched its alpha version in November 2022 on Ethereum blockchain. Since then, they have announced migration to Ronin blockchain [36]. Pixels is an open-ended game about farming and exploration. Players gather resources, advance their skills, and build relationships while exploring the world through quests and story. Pixels focus on interoperability and decentralization aspects of blockchain having integrated over 90 different NFT collections to the game allowing their players to walk around the game world using their NFTs. If the collection is

supported by Pixels, players can use those supported NFTs in-game [35]. Pixels is also planning on bringing NFTs from other projects and blockchain games with currently being worked on feature called “realms” [36]. The aim with realms is to create technology which allows 2d persistent, multiplayer, virtual spaces that users can build and play on. All integrated to blockchain. [35]

Interestingly, Pixels developers state in their Litepaper from 2024 [35] that initially they’ve decided to have all in-game item ownership integrated on-chain, while leaving most game mechanics server-side to allow for faster development and responsiveness. This would allude to blockchain technology or their choices of blockchain technology not being performant enough for their needs.

Axie Infinity is a collection of games [37] developed by Vietnamese studio called Sky Mavis [38]. Its first entry was a creature collector game where these creatures, Axies, can battle, be traded and bread. Axie Infinity’s blockchain integration lies in player-owned economy and true ownership of in-game assets. Axie Infinity uses smart contracts to enable various game mechanics, like NFT minting, claiming in-game rewards, selling Axies and breeding them [39]. Sky Mavis has also created Ronin blockchain which is a Ethereum Virtual Machine blockchain tailored for gaming [40].

These examples highlight how blockchain is being practically applied to existing games. As a conclusion from these real-life examples, we can determine that the most common use cases for blockchain integration are smart contracts which enable plethora of other features like digital ownership of in-game assets through NFTs, player-run economies, and verifiable game mechanics auditable through blockchain. All of the examples above were chosen randomly from the top 20, at the time of writing, most popular games on DappRadar based on their unique active wallets [41].

These trends indicate a broader shift in the gaming industry. Over the past decade blockchain technology has gained popularity beyond just cryptocurrencies and is now being developed in various applications, including gaming. By utilizing decentralized networks, smart contracts, and NFTs, blockchain based games give players full ownership over their in-game assets and enable secure P2P transactions. As video games have become increasingly digital and monetized through in-game purchases, questions like whether players truly own their digital assets has become more relevant. Blockchain gaming directly challenges the traditional model and instead offers a more transparent, decentralized, and player-empowered alternative.

2.3 Benefits of blockchain for gaming

2.3.1 True digital ownership

Use of microtransactions as an alternative monetization mechanic has been on a rise in gaming: according to research [42] online microtransaction market size is anticipated to grow at a compounding annual growth rate of 7.11 %. This means that it is projected to double its market size nominated in U.S. dollars by 2034.

In some traditional online games, players can often purchase in-game items or cosmetics via microtransactions. An issue with digital purchases is that you as a consumer are at the mercy of a central entity, usually the game developers [16]. Blockchain gaming changes this by allowing players to truly own their in-game assets.

With NFT-based assets collectibles are stored on a blockchain, releasing them from centralized entities and confirming players' ownership over their assets. In the case of game shutdowns, blockchain assets would remain accessible even if the game ceases to exist, given that the blockchain is permissionless and not centralized. Events like this would probably have a big economic impact on their in-game NFTs but nevertheless, they would still be available to be freely traded regardless of the outcome.

2.3.2 Interoperability

Interoperability refers to the capability of two or more systems to connect, share, and use exchanged information effectively. Blockchain assets are not restricted based on applications. Instead, they are limited to applications which support the blockchain that the assets are native to. This allows for cross-game and cross-platform asset usage fostering gaming experience and use cases for in-game assets. [43]

Some studies have tried to resolve interoperability between multiple blockchains unsuccessfully whilst adhering to the classical definition of blockchain [43]. This means that blockchain games must really consider the underlying blockchain solution before they should start the development process. And besides, migrating to a different blockchain later might prove itself to be a difficult task and require deep technical understanding. Hence, if current blockchain games want to enhance their game's interoperability, they could offer their players multi-chain support for minting NFTs. Alternatively, they could use bridges (i.e.

LayerZero [44]) or multi-chain marketplaces (such as Rarible [45] or Magic Eden [46]) to allow the players to trade their NFTs across different chains.

2.3.3 Decentralization & transparency

In traditional games, game developers have full control over rules, assets, and economy which can lead to unfair practices [47] or lack of trust for certain game mechanics, like loot boxes. There have been improvements made to these around different jurisdictions as distrust over game mechanics has increased [48]. Whilst some countries have been more lenient towards loot boxes, other countries have outright banned or heavily limited the accessibility for consumers to access certain features in games [49]. Blockchain technology's standard for transparency could be used in the game industry to enhance the transparency of various game mechanics so that every player would have the information and means to look it up. With standardized transparency, governments could more easily evaluate the mechanics of each game. Thus, legislation regarding age ratings would be allowed to be made more effectively on a per-country basis thanks to clearly auditable information.

Blockchain enables decentralization of certain aspects of gaming. Its decentralization eliminates single point of failure risk if fully deployed to gaming. Asset utilization could already be decentralized with NFTs and marketplaces allowing players to freely trade without restrictions. A game's economy which incorporates blockchain technology is not controlled by single entity and can be participated from anywhere, without having to log into the game itself. If game states and assets were to be on-chain, they will remain available and accessible even if individual game instances were to go offline.

A study [50] suggests that InterPlanetary File System (IPFS) could be used as a layer on top of blockchain. By doing so, it would offer an alternative to centralized game servers while connecting nodes (players) directly to a P2P hypermedia network. IPFS network could be utilized to synchronize game states across all gamers in multiplayer games, while in single-player games it could be used to communicate the game state of their game while receiving rendering instructions from proxies. In this suggestion, blockchains would be utilized to record all transaction on-chain after being validated by game's smart contracts. [50]

2.3.4 Play-to-Earn models

While play-to-earn model was already introduced in late 1980s and became more popular in early 2000s with the founding of eBay and emerging popularity of several massively multiplayer online games, such as Runescape and World of Warcraft [51], utilizing blockchain allows play-to-earn game model to level up from relying on 3rd party trading platforms for trading to move it directly on-chain. This would make the whole process of transacting with a seller on 3rd party websites more transparent, secure, and decentralized. On-chain transactions would be visible and secured by smart contracts. [32]

2.3.5 Enhanced security

Blockchain enhances game security by adding several layers of protection through its decentralized and immutable nature. Traditional games often depend on centralized servers for data storage. This approach can be vulnerable for hacking, data tampering, and insider manipulation. Blockchain technology offers a solution to this with a distributed ledger where all transactions are recorded and cryptographically secured making it difficult to tamper. [5]

Smart contracts play a crucial role in fair and secure gameplay mechanics. Smart contracts operate autonomously executing predefined rules without the possibility of alteration as they exist on the underlying blockchain [5]. This reduces the risk for exploits and cheating as the logic cannot be manipulated after deployment.

Furthermore, players are in control of their assets via private keys which can be used by players to initiate transactions or game mechanics. This decentralized ownership reduces the risk for mass data breaches and account compromises. Because sensitive data is stored on-chain, even a compromised game server cannot alter the state of the blockchain. [23]

Lastly, blockchain networks benefit from their consensus mechanisms, which are designed to ensure that all participating nodes agree on the current state of the network [2]. This eliminates single point of failure and adds redundancy against distributed denial-of-service attacks to the nodes.

2.4 Challenges of blockchain in gaming

Despite the potential benefits of blockchain technology in gaming, there are several challenges preventing its widespread adoption.

From a technical standpoint, it is currently not possible to achieve all three critical aspects of blockchain technology: security, scalability, and decentralization. This “blockchain trilemma” was first introduced by Vitalik Buterin, a co-founder of Ethereum, in a blogpost in 2017 [52]. In it, he stated that blockchain could only achieve at most two out of three beforementioned properties. From the early blockchain inventions like Bitcoin and Ethereum, this challenge has diverged different blockchain projects to varying directions. Some prefer to focus on decentralization and security sacrificing scalability while others have focused more on scalability and security sacrificing decentralization. [52] [2]

Fighting the trilemma, one solution to fix usually the unscalability of the main network has been invention of layer-2 solutions in blockchain technology. Layer-2 solutions have been worked on since 2016 with the upcoming of Bitcoin Lightning Network [53]. Layer-2 solutions are secondary protocols built on top of a layer-1 blockchains to improve scalability, reduce fees, and increase transaction speed while still relying on main chain for security [54]. Layer-2 solutions can approach improving main network’s scalability issue in different ways. Rollup layer-2 solutions ‘roll up’ bunch of transactions into a single transaction on the layer-1 network easing transaction fees of the main network by distributing it to everyone in the rollup (i.e. Arbitrum, zkSync) [54]. Other solutions include sidechains which run parallel to layer-1 blockchain but operate their own consensus mechanisms (i.e. Polygon PoS, Ronin) [55], and state channels which open a temporary off-chain channels between participants while only the final state is recorded on the layer-1 blockchain (i.e. Lightning Network) [53].

Why this matters from gaming perspective is that scalability of a blockchain network offers: lower gas fees which is needed for transacting within a blockchain environment, faster transactions which eliminate any delays in confirming transactions, and better user experience not having to wait for block confirmations. [6] [7] [54]

According to a survey by YouGov [56], two thirds of the surveyed played games to ‘pass the time or relax’ and ‘to have fun’. This suggests that most players are looking for enjoyment in video games rather than financial gain. As a result, focusing too much on play-to-earn models or trading aspects can result in alienating a large portion of the gaming audience. This is why it is suggested to treat blockchain features as complementary additions to gameplay and not the core attraction. However, even when blockchain is used carefully, it introduces another major challenge: poor user experience [57]. Having to setup your crypto wallet can already put off some players without mentioning proper private key management practises or the fact that some blockchain games might require you to purchase cryptocurrency or NFTs before playing. Because of the nature of cryptocurrencies, many games prioritize wrong aspects of gaming rather than the core reasons why most people play games.

Both cryptocurrencies and game industry being relatively young industries, they lack regulatory and legal clarity. Governments worldwide are still defining how blockchain assets, cryptocurrencies, and NFTs should be regulated. This regulatory uncertainty creates more economical risk factors for developers to weigh in on. Also, as it is currently in many jurisdictions, cryptocurrencies and NFTs are seen as investments and taxed as such. Crypto earnings might be subject to taxation complicating game economies further. [58] [59]

Security concerns over blockchain technology can be seen as a barrier for entry. While blockchain technology is expanding and the underlying blockchains are developing new ways to take the industry forward, new exploits are being discovered and abused. These exploits and hacks usually end up in early adaptors losing their money which receives lots of news coverage further distancing average consumer away from cryptocurrencies and blockchain technology. Recently, Dubai based cryptocurrency exchange called ByBit got hacked and lost $1.5 billion USD worth of customer funds [60].

3 Core components of blockchain game

Blockchain gaming is powered by decentralized technologies that transform how players interact with digital assets, make transactions, and engage in virtual economies. At the heart of these technologies are smart contracts and NFTs, which enable trustless automation and verifiable digital asset ownership. Smart contracts function as self-executing agreements that govern various game mechanics, from rewards distribution to trading. Meanwhile, NFTs provide players with true digital ownership of their in-game assets, that are available to trade, sell, or utilize across different platforms. This chapter explores these core components in more detail.

3.1 Smart contracts in gaming

As it has been established by now, smart contracts are self-executing programs stored on a blockchain. Their rules and logic are encoded within them. Once they’re deployed, they run autonomously and cannot be altered. This ensures transparency and trust in their execution.

When a smart contract is executed on a blockchain, it creates a transaction which requires computational resources from the network. This requirement incurs fees to be paid. That is because these fees compensate network validators for processing transactions and executing smart contract logic [61]. If it were to be implemented to a PoW blockchain instead of PoS, the fees would go to miners instead of validators [1]. The fee amount depends on multiple factors such as contract complexity, how much data is being stored, and network congestion. Additionally, blockchain platforms set computational limits to prevent excessive resource consumption [1], ensuring the network keeps operating efficiently. This, however, can be bypassed to a certain extent with use of layer-2 solutions as discussed earlier.

3.1.1 On-chain vs. off-chain

An important consideration when implementing smart contracts to games is deciding which logic should be executed on-chain and what should remain off-chain as incurring transactions on-chain will have its limitations. On-chain logic benefits from blockchain technology’s typical traits, immutability and transparency, making it ideal for ownership tracking and making transactions. However, computationally more intensive tasks such as physics calculations, AI

behaviour, and rendering should remain off-chain. Even in a compensated solution, off-chain game logic can still reference on-chain data if needed. This could be done through different web3 libraries such as web3.js [62] or ethers.js [63] which allow applications to retrieve data from blockchain.

3.1.2 Smart contract security issues

Inclusion of smart contracts in blockchain technology introduces new security risks that must be addressed to prevent exploits and attacks from malicious actors. Ethereum smart contract vulnerabilities can be divided into the following categories: unchecked send/other function call, re-entrancy vulnerabilities, permission control vulnerabilities, function visibility vulnerabilities, random number vulnerabilities, and integer overflow/underflow. [64]

Unchecked send/other function calls refer to calling important functions like send or transfer in a smart contract. It is important to check whether the function executed successfully or not before proceeding. Failing to do so can lead to unexpected errors or vulnerabilities. E.g. an attacker could potentially trigger a smart contract even without meeting the requirements of the contract and thus failing the transaction [64].

Re-entrancy vulnerabilities happen when a smart contract does not properly manage function execution order. This creates an opportunity for attackers to continuously call a function before the previous execution finishes. In an example, attacker deposits Ethereum into a bank contract and then calls the withdraw function. When the contract transfers Ethereum to the attacker it triggers their fallback function, which re-enters the withdraw function before the balance updates. This creates a recursive loop where attacker can drain the funds which are stored in the smart contract. Proper coding practises such as updating balance before transfers can prevent this exploit happening [65].

Permission control vulnerabilities are fairly self-explanatory. Without proper access control measurements, attackers can exploit high-privilege functions in smart contracts leading to unwanted consequences. For instance, if balance-modifying functions lack restraints, an attacker could arbitrarily manipulate account balances, and in worst cases, even destroy the contract itself transferring its remaining funds to themselves [64].

Function visibility vulnerabilities refer to functions being public in Solidity by default. If a developer forgets to set the visibility of a private function, it remains visible to anyone allowing unwanted consequences [64].

Random number vulnerabilities mean exploits where an attacker can precisely predict the outcome of a random number generator. This can be done if a smart contract relies on a publicly known variable used as a seed for generating random numbers [64].

And lastly, integer overflow/underflow vulnerabilities mean the logic behind uint256 type integers are handled in 256-bit Ethereum virtual machines. The maximum representable value is $2^{256} - 1$, which is a very large number. If the maximum value is added by 1, it changes to 0. And alternatively, if you subtract 1 from 0, you will have the maximum value. This can be abused if not handled properly in code [64].

3.1.3 Smart contracts for NFTs

In gaming, smart contracts facilitate mechanics such as item minting, ownership transfers, and marketplace interactions. This is enabled through code that smart contracts are made of. To improve blockchain technologies interoperability, security and adoption, different standards have been proposed through its development to tackle these issues. This all started on Ethereum in 2015 with Ethereum improvement proposals and ERC-20 token standard which laid the groundwork for allowing any tokens on Ethereum to be re-used by other blockchain applications [66]. This led to introduction of fungible tokens on blockchains. When it comes to Non-Fungible Tokens, NFTs, there are two standards which are more widely used: ERC-721 and ERC-1155. ERCs are predefined protocols implemented through smart contracts to establish token standards on Ethereum blockchain [50]. [30]

ERC-721 is the Non-Fungible Token standard. A sample of it can be seen in Figure 3 below. Suggested in January 2018 [67], it enabled an implementation of a standard application program interface (API) for NFTs within smart contracts and provides basic functionality to track and transfer NFTs. With that a framework for NFTs was laid out. This token standard is the key feature enabling unique tokens existing on a blockchain and ownership of distinct digital assets.

```
// contracts/GameItem.sol
// SPDX-License-Identifier: MIT
pragma solidity ^0.6.0;

import "@openzeppelin/contracts/token/ERC721/ERC721.sol";
import "@openzeppelin/contracts/utils/Counters.sol";

contract GameItem is ERC721 {
    using Counters for Counters.Counter;
    Counters.Counter private _tokenIds;

    constructor() public ERC721("GameItem", "ITM") {}

    function awardItem(address player, string memory tokenURI)
        public
        returns (uint256)
    {
        _tokenIds.increment();

        uint256 newItemId = _tokenIds.current();
        _mint(player, newItemId);
        _setTokenURI(newItemId, tokenURI);

        return newItemId;
    }
}
```

Figure 3. ERC-721 token standard example used in the prototype development. [68]

ERC-1155, however, can be seen as a combination of two previously proposed standards, ERC-20 and ERC-721. Suggested in June 2018 [69], this standard allows the management of multiple token types by the same smart contract. Previously it was required to deploy separate contracts for each token type (fungible or non-fungible) placing a lot of redundant bytecode on the Ethereum blockchain. But this standard made it possible to handle both token types under the same smart contract.

By adhering to these standards developers can create interoperable smart contracts that work across different platforms improving user experience, security and adoption.

3.1.4 Smart contract integration with game engines

For a game to interact with blockchain-based smart contracts, games need to integrate libraries, which assist communication between the game client and blockchain networks. Popular libraries for Ethereum-based smart contracts include web3.js and ethers.js which are both JavaScript-based libraries. But considering web3.js libraries were archived in early March 2025 [70], it would be the best to opt in for using some other library, like ethers.js [63] or viem [71].

Alternatively, game developers could utilize existing solutions for different game engines, software development kits (SKDs) . SDKs are collections of related software packages designed to work together making it easier for developers to build applications with them. These package collections usually include tools, libraries, frameworks, and documentation which help to streamline development. [72]

SDKs are usually either company-provided or open-source. Web3.unity SDK offered by ChainSafe [73] is an example of SDK that acts as a bridge between Unity game engine and the Ethereum blockchain. This allows developers to easily incorporate blockchain features like token management or wallet integration within their Unity projects.

Currently there are similar solutions available for other game engines. Sequence has SDKs for both Unreal Engine [74] and Unity [75] while there are some open-source SDKs available for Godot-based blockchain games [76] [77]. Depending on what kind of game and which game engine the game is being developed, it might limit your options in case SDK utilization in development is being considered.

3.2 NFTs as in-game assets

To learn what NFTs really are, we need to look at them at a more technical level. In this chapter, what makes an NFT is being explored with examples of functions that are related to NFTs and their creation.

As was established previously and in Figure 3, NFTs boil down to ERC-721 standard. By looking at OpenZeppelin’s example of an ERC-721 token contract [68] a better idea of what NFTs are can be formed.

From the minting function it can be seen that when an NFT is awarded, it links together three different properties: the wallet address, uniform resource identifier (URI) [78], and a token ID.

Starting with the wallet address, it is simply a string of letters and numerals which link the NFT to a certain wallet address marking its current owner. In blockchain games, this would be the player's wallet address they use in conjunction with the game.

Token URIs are compact strings of characters for identifying an abstract or physical resource [79]. It often points to a JSON (JavaScript Object Notation) type file containing all the metadata regarding the NFT. In game use, NFTs' metadata files could contain information such as the item name, description, image location, and its in-game attributes.

Last part of ERC-721 NFTs is token ID. It is a unique uint256 number which each NFT has. It is created and assigned for the token in smart contract's NFT minting code.

So, essentially how NFTs are implemented to a game is that first smart contracts need to be utilized to mint NFTs with unique token ID, player's wallet address, and metadata URIs which is the representation part of the NFT (i.e. what the NFT actually is). Then, the game needs to be able to parse the NFT metadata which is located somewhere off-chain. And finally, be able to represent it in-game for players to enjoy.

4 Prototype design

So far, blockchain and its core technologies has been discussed. Next, blockchain game prototype that was developed during this thesis is to be examined more thoroughly.

4.1 Game concept overview

The basis for this prototype was to be a minimalistic collectible card game environment in which each card players have are represented as unique NFTs. Players log into the game using their Ethereum-based wallet address. Wallet address acts as identification of the player replacing more commonly seen user accounts. As can be seen in Figure 4 below, in-game players can view their existing NFT collection for the game, mint new NFTs, send trade offers to other players, and either accept or deny incoming trade offers from other players.

Figure 4. Main menu of the prototype.

As will be discussed in limitation of this thesis, core gameplay loop was chosen to be omitted from this prototype. It was deemed unnecessary to demonstrate asset ownership with NFTs and smart contracts.

4.2 Technical stack

The technical implementation of the prototype depended on a combination of technologies, game development tools, and blockchain integration libraries. Each of the components played a specific role in enabling the functionality required for minting new NFTs, displaying, and trading them on blockchain.

Unity was chosen to serve as the game development platform for this prototype due to it being one of the top choices for game development tools [80] for building blockchain games as well as previous familiarity with Unity development. Other notable similar solutions include Unreal Engine and Godot. In the prototype, Unity was used to build user interface as well as to manage player interactions like wallet input, card selection, and blockchain transaction initialization. Unity's scripting language is C# which made it possible to integrate external libraries such as Nethereum and handle asynchronous blockchain communication.

Nethereum [81] is an open-source .NET library that allows Unity to interact with Ethereum blockchain. In essence, it works as a translation layer between Unity application and Ethereum clients which pass calls to Ethereum blockchain [81]. In the prototype, Nethereum was used to send transactions, call different smart contract functions, and to monitor transaction status. Nethereum also handled all Application Binary Interface (ABI) [63] encoding and decoding, contract interaction, and signing of transactions with player's private key.

Infura [82] is a third-party blockchain node provider. It exposes Remote Procedure Call (RPC) [83] endpoint to connect applications to their offered blockchain networks. Instead of having to run a full Ethereum node locally, the prototype was chosen to use Infura's RPC to send transactions and queries via its Sepolia testnet endpoint. This choice significantly reduced setup complexity and investment, as well as ensured reliable access to Ethereum network during development.

Sepolia is one of Ethereum's official testnets. At the time of this thesis, it was the recommended default testnet for application development by Ethereum foundation [84]. It allows developers to deploy and test smart contracts in a blockchain environment without any real funds. All tokens and transactions in the prototype exist solely on Sepolia and have no real-world monetary value. This testnet was essential for testing blockchain logic and debugging smart contracts with realistic constraints like gas limits and transaction confirmation times without needing monetary investment. But since Sepolia is only a testnet, it cannot properly simulate the stress Ethereum network experiences especially during peak usage times [85].

Solidity is Ethereum's native smart contract language. It was used to write two smart contracts: NFT contract to mint and manage the cards, and a trade contract which allows players to propose, accept and decline card-for-card trades. Development of these smart contracts was done using Remix integrated development environment [86] which is a browser-based environment for testing smart contracts.

IPFS was used to host the NFT imagery used by the prototype. This was done so that even third-party marketplaces could fetch the image data from this P2P file storage system. ChatGPT was used to generate all the images used in this prototype.

4.3 Blockchain integration

The prototype allows players to mint, view, and trade collectible card NFTs, all of which either happen directly on-chain or reference on-chain data. Blockchain-related operations are handled through Ethereum smart contracts which are deployed on the Sepolia testnet. Unity, in turn, interacts with these contracts using C# library called Nethereum.

In the prototype, blockchain serves as the source of truth for asset ownership and trades. Next, blockchain integration is observed from feature set perspective. Some of the actions happen on-chain while others remain off-chain. On-chain, all game assets are managed via two smart contracts: minting contract and trading contract. Generally speaking, off-chain actions mean actions that happen on Unity and Nethereum's side while on-chain actions are mostly smart contract function executions that leave a record on the blockchain. Refer to Figure 5 below for further differentiation.

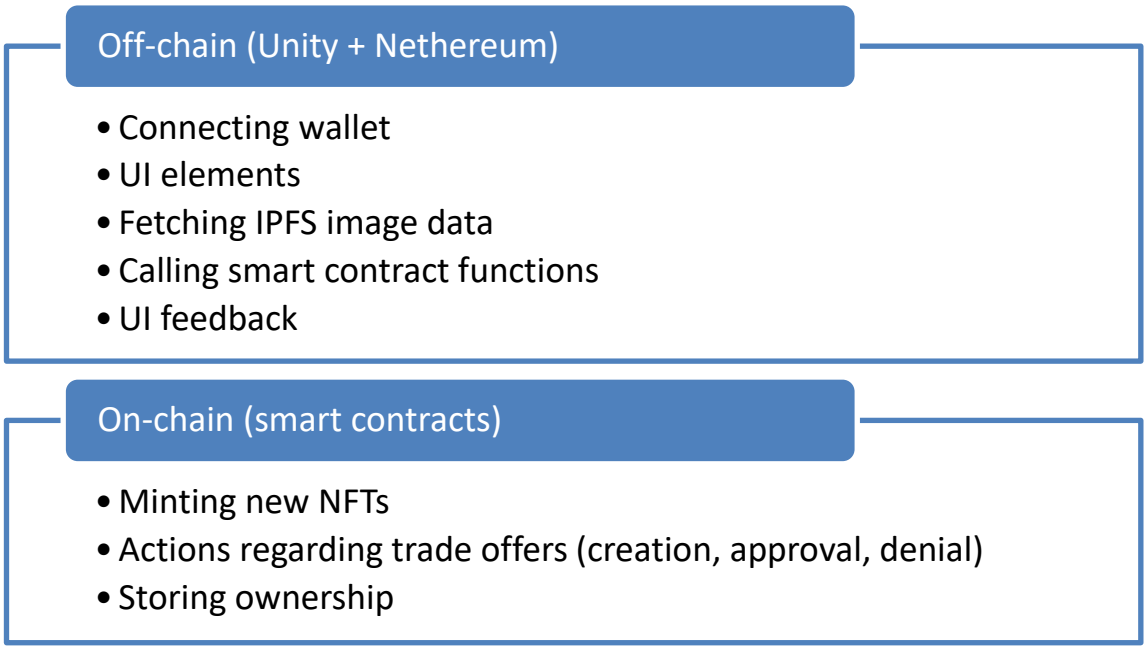

Figure 5. On-chain versus off-chain.

4.3.1 Minting cards

NFT minting mainly happens with a smart contract called NFTMint which is an ERC-721 compliant contract. The contract manages token ownership, minting, and metadata storage. When a card is minted, it is permanently recorded on the blockchain and is assigned with an ID to a player's wallet address. The logic behind card minting works as described in Figure 6.

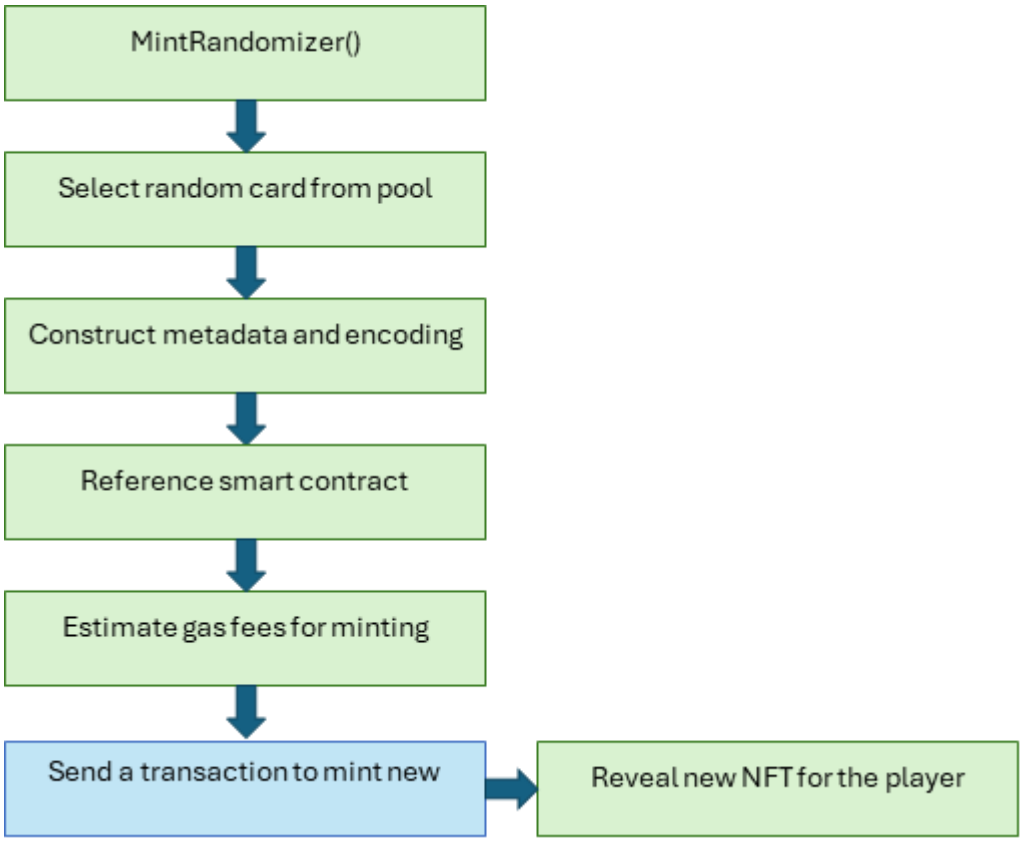

Figure 6. Flowchart for card minting in prototype. Green boxed represent off-chain logic while blue boxes represent on-chain logic.

Here, MintRandomizer() function is being called by the player from Unity's UI elements. Then, after a randomized selection from the card pool data, a card is being constructed off-chain. After relevant information such as player's private key and contract address is referenced with NFTMint smart contract, the gas fee estimation occurs on-chain for the minting function on-chain. Lastly, after gas estimation has gone through without errors, a transaction is sent on-chain to instil the newly minted NFT.

4.3.2 Fetching player-owned NFTs

This functionality offers a way for players to check their already owned cards in-game. To achieve this, the prototype must identify which token IDs belong to the logged-in player wallet address and display all relevant metadata inside Unity, as referred in Figure 7 below.

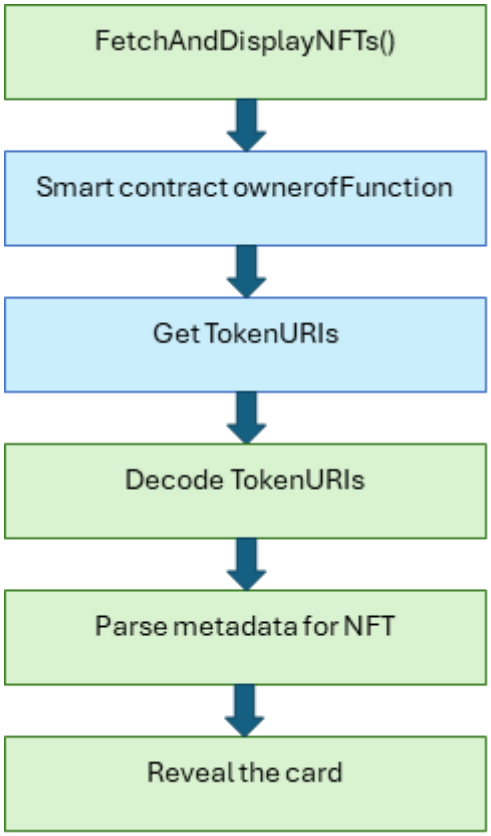

Figure 7. Flowchart of the fetching logic. Green boxed represent off-chain logic while blue boxes represent on-chain logic.

This logic is initiated from player's click of a UI button. It creates references to the NFTMint smart contract and then loops through predefined range of possible token IDs. Each tokenID is being queried to the blockchain using ownerOf() function to determine if the token is owned by the current player's wallet address. If ownership is confirmed, the function then retrieves and decodes the metadata with tokenURI() function. This metadata contains all relevant info for the card.

After this, each matching player-owned NFTs are instantiated in the game as card prefabs within an UI container and finally rendered on to the screen as in Figure 8 below.

Figure 8. Image from the collection view of the prototype.

4.3.3. Trade offer creation

Trading is a core mechanic of the prototype allowing a player to send a trade offer to another player. The trade offer contains information regarding the card offered for a trade, receiver's wallet address, as well as the token ID of which the sender wants in return for their card. Before this trade offer is sent to the smart contract, the player must grant the contract permission to transfer the token by calling standard ERC-721 method called approve(). Then, the trade offer is only recorded if the proposer of the trade deal owns the token they are offering, and the contract has the approval to manage said token. This ensures secure and authorized creation of P2P trade offers without needing a third party to act as an intermediary. After the trade offer has been accepted by the smart contract, Unity UI will give confirmation for the offer sender as an Etherscan link for the verification. This logic flow is further visualized in Figure 9.

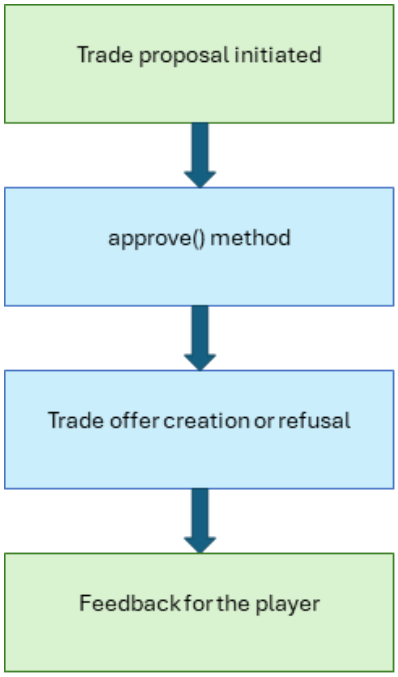

Figure 9. Trade proposal logic. Green boxed represent off-chain logic while blue boxes represent on-chain logic.

4.3.4 Accepting or declining trade offers

Once a trade offer has been created, the recipient of that offer has the ability to either accept or decline the offer via an UI panel in-game. This happens in two steps: first, all incoming trade offers for the player who is currently logged in are being fetched from the blockchain. And secondly, regarding of what player chooses to do with their incoming trade offers, they can either decline or accept it. This functionality provides core interactions of the prototype and demonstrates bidirectional NFT exchange through smart contracts.

First step of this functionality, as referred in Figure 10, is self-explanatory. When player moves into the trading panel in Unity, all incoming trade offers are being fetched from the blockchain and displayed on-screen as UI elements. Then, player clicks a trade offer they want to interact with and chooses to either accept or decline it. In the case of approval, the following happens:

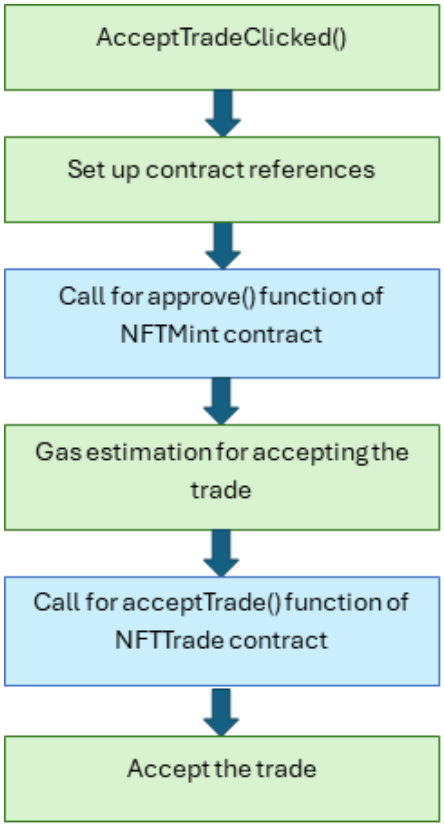

Figure 10. Logic behind accepting a trade offer. Green boxed represent off-chain logic while blue boxes represent on-chain logic.

Once player clicks to accept a trade, approve() method of the minting smart contract and acceptTrade() method of the trade smart contract are being called. But this time the approve() function is being called for the receiver of the initial trade offer. As a result, the trade is now accepted, and the trade smart contract handles the transfer of ownership. After everything is done, player receives a visual cue from this as an Etherscan link for the completed transaction.

Alternatively, if player chooses to decline an incoming trade offer, declineTrade() method of the trade smart contract is being called instead of acceptTrade(). Declining a trade marks the offer as void on-chain.

This structure ensures fairness and prevents unauthorized trades. Smart contract checks act as a final layer of defence, even if incorrect or malicious data is submitted off-chain.

4.4 NFTs in the prototype

In this prototype, each collectible card is represented as a unique NFT following the ERC-721 standard [67]. These tokens serve as digital representations of cards that players can own, view, and trade in-game. NFTs are central to the ownership model of the game. They ensure that every card a player has is cryptographically verifiable and linked to their Ethereum wallet address.

In the prototype's NFT design, it was chosen to store all data regarding NFTs directly on-chain apart from the images themselves. Metadata follows a widely known JSON schema and stores information such as card name, description, image address, and attributes. This JSON data is encoded in base64 format and returned via the smart contract's tokenURI() function. When queried, the contract provides full decoded data URI containing the metadata. This is what game clients or NFT marketplaces can use to display the card.

In this prototype, card metadata is embedded directly in base64. The image field in the metadata points to a card illustration address hosted on IPFS. This ensures that the card visuals are not reliant on centralized servers and are generally accessible.

4.5 Smart contracts in the prototype

This prototype depends on two smart contracts written in Solidity: one for managing and minting NFTs called NFTMint, and another for handling trade offers between players called NFTTrade. These contracts were deployed to Ethereum Sepolia testnet and are the foundation of blockchain-based asset ownership and trading mechanisms implemented in the prototype.

The smart contracts used in the game were separated from each other depending on their main use-case. This was done mostly to simplify the contract logic and to make it easier to upgrade or redesign the logic, if necessary, as these two features work independently of each other. But in this way of separating the contracts, you will need to assign a constructor parameter which references to your NFTMint contract. Otherwise NFTTrade contract does not know which NFTMint contract to interact with.

Both contracts extend OpenZeppelin's ERC-721 implementation [68] providing standard interface for minting, transferring, and querying token ownership. By using trusted libraries like OpenZeppelin, as referenced in Ethereum's development pages [87], the implementation benefits from industry-standard practises in security and interoperability.

NFTTrade contract introduces a custom TradeOffer struct. As shown in Figure 11, that is the information which touch the blockchain and change depending on what actions player takes in-game regarding trade offers. This structure allows that each token can have only one active offer at a time making the lookups for querying trade offer states faster.

```
struct TradeOffer {
    address proposer;
    address recipient;
    uint256 proposerTokenId;
    uint256 requestedTokenId;
    uint256 timestamp;
    bool isActive;
}
```

Figure 11. Custom struct used by NFTTrade contract.

To enable off-chain tracking and user feedback, several key events of the smart contract are emitted. These events include: TradeOfferCreated, TradeOfferAccepted, TradeOfferCancelled, and TradeDeclined. These events can be useful for integrating with block explorers, which are used for inspecting transactions happening on blockchains, or for real-time updates in the Unity game client.

All smart contracts used in this prototype were written and tested using Remix IDE [86], a browser-based Solidity development environment. The contracts were not formally audited. But due to the limited scope and educational purpose of the prototype, the design of the smart contracts prioritized clarity and simplicity.

4.6 Implementation challenges

Integrating blockchain functionality to a video game prototype brought a wide range of technical and practical challenges. These challenges stemmed not only from blockchain’s complexity but also from the friction between decentralized technologies and real-time, user-centric game development workflows.

4.6.1 Working with blockchain transactions in Unity

Unlike web apps which can interact with JavaScript-based libraries such as Web3.js, Unity required use of .NET-based library for Ethereum interaction. A library called Nethereum was chosen for this prototype. While it turned out to be a powerful library, it brough in additional complexity such as asynchronous programming, gas estimation, and limited debugging tools if something went wrong.

Lack of debugging tools turned out to be problematic during development. Some of the issues regarding smart contract interactions had to be mostly resolved by trial and error or by inspecting blockchain data in Etherscan. The most frequent problem that occurred during the development was transaction signings failing due to tokens not being properly approved for change of ownership.

4.6.2 Approvals and on-chain constraints

One of the core blockchain design requirements was to make sure that the trade contract could only transfer NFTs if the owner had approved them for trading. This led to a couple of hindrances.

First of which, unapproved tokens caused failed transactions. This needed to be handled in the logic pre-emptively. Secondly, approval confirmations on Sepolia testnet sometimes forced an artificial delay to be added into the code to ensure that the approval was confirmed on-chain before continuing with trade offer creation.

4.6.3 Smart contract limitations

There were a few necessary trade-offs that were made during the development to keep smart contracts minimal.

Firstly, TradeOffer mapping currently works in a way where only one active trade is allowed per proposer token ID. Meaning that a player can only send one trade offer for each of their own cards. This simplifies indexing but reduces flexibility for the players.

Secondly, there was no timeout or expiry cleanup beyond timestamp check, meaning that trades remain in the contract’s state unless the recipient of those trade offers manually cancels or declines them.

And lastly, more advanced trading logic like multi-token trades or auctioning were deliberately omitted to keep the minimal viable product aspect of the prototype.

4.6.4 Metadata management

The metadata storage method of this prototype made it possible to have completely immutable on-chain data since the metadata was embedded into the token URIs. This method comes with its own drawbacks.

As all metadata is embedded in the token URIs, it is impossible to modify the data later. In some game development scenarios, it might be desirable to have the possibility of modifying some data later as the game progresses. In collectible card games, some older cards might become less relevant over time and require some tweaking to make them viable to use. In this case, game developers might want to look into dynamic metadata solutions that allow data modification after the token has been minted.

This metadata design also caused some complexity in Unity's metadata parser as it has to decode and parse these metadata containing strings to extract values for rendering cards.

4.6.5 User experience considerations

Despite not being the focus of this thesis, one of the differences between blockchain applications and traditional games is the lack of instant feedback. Blockchain actions can often take several seconds to confirm, fail due to unpredictable network conditions or gas issues, or require external wallet approval. All these seemingly small issues that might occur during blockchain usage will need to be informed to the user if they experience such events. In the prototype these were implemented in UI as status messages for approval processes, Etherscan links to view transactions, and visual confirmation cue of successful or failed trades.

Despite these efforts the user experience remained less fluid than in traditional, centralized games.

4.6.6 Performance considerations

Even though the developed prototype was not built for production-scale deployment, performance still played a role in design decisions. Since blockchain operations are inherently slower than in-memory or server-based systems, care had to be taken to manage latency and maintain responsiveness in the user experience.

One of the most notable performance-related challenges was the delay between sending a transaction and seeing its confirmation on the blockchain. Even on Sepolia testnet, transaction confirmations could take sometimes anywhere from seconds to tens of seconds. To mitigate these issues, artificial delay was added after approvals and before subsequent actions. While functional, this approach was more of a workaround than a proper solution. In production, this would ideally be replaced by an event listening system to detect transaction receipts before moving forward with code.

Another type of delay that was added was to mitigate possible Infura rate limits and to maintain stable UI behaviour in Unity. Lengthy blockchain operations such as loading multiple token URIs could cause UI to lag if not properly managed. This sometimes caused not all player-owned cards to be rendered on-screen.

Additionally, the blockchain itself is also a performance limiter. While Ethereum's Sepolia testnet was chosen for this prototype, it might not be the best option depending on its use-case as talked about earlier in chapter 2.4 of this thesis. For more performance-dependent use-cases, a blockchain or layer-2 solution with higher transactions per second output could be more capable in processing transactions.

5 Conclusions

In this last chapter of the thesis key outcomes are summarized and the broader significance of blockchain integration to games reflected upon. It also outlines the limitations encountered during development and proposes directions for future improvements and research.

5.1 Summary of findings

This thesis explored the integration of blockchain technology into a video game through the development of the prototype of a collectible card game. The primary goal was to investigate how NFTs can be utilized to represent in-game assets and what challenges and benefits smart contract implementations bring to gaming.

This prototype successfully demonstrated several key blockchain features including NFT minting and ownership verification. In the prototype, players could successfully mint unique cards stored as ERC-721 token on Ethereum's Sepolia testnet. These tokens were verifiably tied to wallet address which was used to initially mint them. As shown below in Figure 12, all this can be verified using on-chain ownership records such as Etherscan.

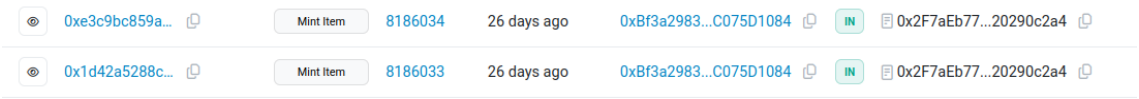

Figure 12. Screen capture from Etherscan NFT mint operation.

Other smart contract features included bidirectional trading between two players. Here, smart contract enforced secure and trustless exchanges, including logic for trade offer creation, acceptance, and cancellation.

All this was developed to a system where a Unity-based UI was combined with Nethereum and Infura to interact with Ethereum-based smart contracts whenever it was necessary. All Unity-side development happened within a context of a desktop application while smart contracts themselves were developed separately.

These findings validated the feasibility of using blockchain as decentralized monetary management platform in games, even if the prototype was limited in

gameplay functionality. NFTs were successfully used as in-game assets as well as smart contract benefits and challenges were discussed. All the while, the prototype maintained a clear separation between on-chain and off-chain operations, offering an architectural foundation for future expansion.

Developing such a prototype required a vast skill set which unifies traditional game development with blockchain engineering. From the game development perspective, working with Unity and C# was essential to design the user interface, gameplay logic, and integrate with backend services. On the blockchain side, understanding smart contracts, Solidity programming, and ERC-721 standard was crucial to define the rules for minting, trading, and ownership. In addition, Nethereum library and Infura required familiarity with asynchronous programming, RPC calls, gas management, and Ethereum testnet infrastructure. Overall, this prototype demanded both technical knowledge and understanding of blockchain principles, to apply blockchain technology to interactive systems such as games.

5.2 Implications of blockchain in gaming

The results suggest that blockchain introduces new possibilities to how digital assets are managed in games. Traditional in-game assets are controlled by game developers and hosted on centralized platforms. By contrast, blockchain allows these assets to exist independently of the game or the developer. This is achieved by enabling players to truly own their in-game assets by retaining the access for them, having the possibility for the assets to be used elsewhere, and enhancing the transparency since on-chain transactions are public and can be viewed by anyone.

However, blockchain adoption does raise questions around its usability, scalability, and regulation, which are not yet fully solved as it is still a developing technology.

5.3 Limitations of the thesis

When discussed about benefits and challenges of adopting blockchain technology into video games, it is important to note that this was mostly done from end-user’s perspective. There are financial incentives from game developers and publishers’ perspective as to why it would not be beneficia for them to have their games to utilize blockchain technology instead of the current traditional model.

As for the prototype itself, even though it demonstrated blockchain functionality, there were some important limitations that need to be acknowledged. Prototype omitted a full game loop and lacked in user experience polish. The prototype was initially designed to be as minimal as possible and only demonstrate key blockchain functionality. That is why a proper game loop was left out as well as proper error handling and transaction monitoring.

Some performance-related issues occurred during the development which were mostly due to the blockchain not being able to process the transactions as quickly as the code in Unity was proceeding. This caused some artificial delays to be added to the code. In addition, some of the methods used in the scripts regarding off-chain execution were not designed with the most efficient outcome in mind. For instance, a brute-force scanning approach in a form of a loop was used to query the blockchain to determine whether each token is owned by the logged in player's wallet address.

It is also important to note that while Sepolia testnet was chosen to be utilized in the prototype, it can not properly reflect scalability issues during high usage periods of the blockchain as it is not the mainnet environment with actual users engaging with the blockchain.

Finally, security was not addressed in the development of the prototype. While smart contracts were kept simple, production-level implementations would require audits and additional development.

5.4 Future work & research directions

This thesis lays out groundwork for several possible areas of further research, including topics such as gameplay integration and alternative layer-1 or layer-2 solutions. Future projects could embed blockchain mechanics into actual game loops where, for instance, card battle mechanics could be published as smart contracts in the blockchain. Layer-1 or layer-2 solution studies could explore alternative blockchains to Ethereum or sidechain and rollup possibilities for the underlying blockchain and hopefully improve scalability and reduce gas costs.

Related to the metadata implementation of the prototype, another interesting topic for further research could be alternative choices for metadata storage. In the prototype, all metadata was embedded on-chain and only the image assets themselves were off-chain on IPFS. Not minting the actual metadata on-chain could allow modification of such data later as the application progresses and modifications to the earlier assets become necessary.

Marketplace integration was not investigated during this thesis. Instead, a simple implementation of it was carried out in-game. Further research could focus on integrating existing NFT marketplaces expanding the trading functionality beyond the prototype's scope and reducing the development time.

Additionally, smart contract security was not addressed during the prototype development. Deeper investigation into access control, re-entrancy, and gas optimizations would be necessary areas of research for achieving production readiness.

In summary, this thesis has demonstrated that blockchain can act as a functional backbone for ownership and trading in video games. However, realizing its full potential does require overcoming several technical and experiential hurdles. The results point towards a promising future where players may truly own their digital worlds.

References

[1] S. Nakamoto, “Bitcoin: A peer-to-peer electronic cash system,” 2008.

[2] W. Li, M. He and S. Haiquan, “An Overview of Blockchain Technology: Applications, Challenges and Future Trends,” in *2021 IEEE 11th International Conference on Electronics Information and Emergency Communication (ICEIEC)*, Beijing, China, 2021, pp. 31-39, doi: 10.1109/ICEIEC51955.2021.9463842.

[3] A. Kumar, B. K. Sah, T. Mehrotra and G. K. Rajput, “A Review on Double Spending Problem in Blockchain,” in *2023 International Conference on Computational Intelligence and Sustainable Engineering Solutions (CISES)*, Greater Noida, India, 2023, pp. 881-889, doi: 10.1109/CISES58720.2023.10183579.

[4] M. Atzori, “Blockchain Governance and The Role of Trust Service Providers: The TrustedChain® Network,” *The Journal of the British Blockchain Association,* vol. 1, pp. 1-17, 2018, doi: 10.31585/jbba-1-1-(3)2018.

[5] R. Yang, F. R. Yu, P. Si, Z. Yang and Y. Zhang, “Integrated Blockchain and Edge Computing Systems: A Survey, Some Research Issues and Challenges,” *IEEE Communications Surveys & Tutorials,* vol. 21, no. 2, p. 1508–1532, 2019, doi: 10.1109/COMST.2019.2894727.

[6] M. A. Islam and S. Madria, “A Permissioned Blockchain Based Access Control System for IOT,” in *2019 IEEE International Conference on Blockchain (Blockchain)*, Atlanta, GA, USA, 2019, pp. 469-476, doi: 10.1109/Blockchain.2019.00071.

[7] Coinbase, “What is the blockchain trilemma?,” Coinbase, [Online]. Available: https://www.coinbase.com/learn/crypto-glossary/what-is-the-blockchain-trilemma. [Accessed 14 March 2025].

[8] V. Kustov, N. Beksaev and R. Ravi, “The Blockchain SSD Trilemma or Chasing Three Birds with One Stone,” in *2023 16th International Conference on Advanced Technologies, Systems and Services in Telecommunications (TELSIKS)*, Niš, Serbia, 2023, pp. 219-222, doi: 10.1109/TELSIKS57806.2023.10316137.

[9] M. AlShamsi, M. Al-Emran and K. Shaalan, “A Systematic Review on Blockchain Adoption,” *Applied Sciences,* vol. 12, no. 9, p. 4245, 22 April 2022, doi: 10.3390/app12094245.

[10] J. Angelis and E. R. da Silva, “Blockchain adoption: A value driver perspective,” *Business Horizons,* vol. 62, no. 3, pp. 307-314, 26 December 2019, doi: 10.1016/j.bushor.2018.12.001.

[11] W. Chujitarom and C. Panichrutiwong, “Storytelling for Non-Fungible Token via Blockchain Technology A Case Study of Layer Randomly model for Digital Art Profile Picture (PFP),” in *2023 International Conference on Information Management (ICIM)*, Oxford, U.K., 2023, pp. 88-91, doi: 10.1109/ICIM58774.2023.00022.

[12] Google Trends, “Search interest for 'nft',” [Online]. Available: https://trends.google.com/trends/explore?date=all&q=nft&hl=en-GB. [Accessed 14 March 2025].

[13] Google Trends, “Search interest for 'blockchain games',” [Online]. Available: https://trends.google.com/trends/explore?date=all&q=blockchain%20games&hl=en-GB. [Accessed 14 March 2025].

[14] S. M. Beillahi, G. Ciocarlie, M. Emmi and C. Enea, “Behavioral Simulation for Smart Contracts,” in *41st ACM SIGPLAN International Conference on Programming Language Design and Implementation (PLDI)*, London, UK, 2020, p. 17, doi: 10.1145/3385412.3386022.

[15] B. Egliston and M. Carter, “Cryptogames: The promises of blockchain for the future of the videogame industry,” *New Media & Society,* vol. 26, p. 6756–6778, November 2024, doi: 10.1177/14614448231158614.

[16] Supercell, “Terms of Service,” 6 November 2024. [Online]. Available: https://supercell.com/en/terms-of-service/. [Accessed 14 May 2025].

[17] Y. Yuan and F. Wang, “Blockchain: The state of the art and future trends,” *Acta Automatica Sinica,* vol. 42, p. 481–494, April 2016, doi: 10.16383/j.aas.2016.c160158.

[18] H. R. Andrian, N. B. Kurniawan and Suhardi, “Blockchain Technology and Implementation : A Systematic Literature Review,” in *2018 International Conference on Information Technology Systems and Innovation (ICITSI)*, Bandung, Indonesia, 2018, pp. 370-374, doi: 10.1109/ICITSI.2018.8695939.

[19] Ethereum Foundation, “ethereum.org,” 12 February 2025. [Online]. Available: https://ethereum.org/en/roadmap/merge/. [Accessed 17 March 2025].

[20] A. Back, “Hashcash - A Denial of Service Counter-Measure,” 2002.

[21] S. Gao, T. Yu, J. Zhu and W. Cai, “T-PBFT: An EigenTrust-based practical Byzantine fault tolerance consensus algorithm,” *China Communications,* vol. 16, no. 12, p. 111–123, December 2019, doi: 10.23919/JCC.2019.12.008.

[22] H. Li, H. Liu and J. Li, “Workflow scheduling algorithm based on control structure reduction in cloud environment,” in *2014 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, San Diego, CA, USA, 2014, pp. 2587-2592, doi: 10.1109/SMC.2014.6974317.

[23] Z. Meng and Y. Wang, “Asymmetric Encryption Algorithms: Primitives and Applications,” in *2022 IEEE 2nd International Conference on Electronic Technology, Communication and Information (ICETCI)*, Changchun, China, 2022, pp. 876-881, doi: 10.1109/ICETCI55101.2022.9832032.

[24] S. Suratkar, M. Shirole and S. Bhirud, “Cryptocurrency Wallet: A Review,” in *2020 4th International Conference on Computer, Communication and Signal Processing (ICCCSP)*, Chennai, India, 2020, pp. 1-7, doi: 10.1109/ICCCSP49186.2020.9315193.

[25] H. Li, R. Lu, L. Zhou, B. Yang and X. Shen, “An Efficient Merkle-Tree-Based Authentication Scheme for Smart Grid,” *IEEE Systems Journal,* vol. 8, no. 2, pp. 655-663, June 2014, doi: 10.1109/JSYST.2013.2271537.

[26] H. Liu, X. Luo, H. Liu and X. Xia, “Merkle Tree: A Fundamental Component of Blockchains,” in *2021 International Conference on Electronic Information Engineering and Computer Science (EIECS)*, Changchun, China, 2021, pp. 556-561, doi: 10.1109/EIECS53707.2021.9588047.

[27] N. Szabo, “Smart Contracts,” 1994. [Online]. Available: https://www.fon.hum.uva.nl/rob/Courses/InformationInSpeech/CDROM/Literature/LOTwinterschool2006/szabo.best.vwh.net/smart.contracts.html. [Accessed 18 March 2025].

[28] X. Wang, J. He, Z. Xie, G. Zhao and S.-C. Cheung, “ContractGuard: Defend Ethereum Smart Contracts with Embedded Intrusion Detection,”

IEEE Transactions on Services Computing, vol. 13, no. 2, p. 314–328, March-April 2020, doi: 10.1109/TSC.2019.2949561.

[29] River, “Bitcoin vs. Ethereum,” River, [Online]. Available: https://river.com/learn/bitcoin-vs-ethereum/?utm_source=chatgpt.com#why-ethereum-was-invented. [Accessed 18 March 2025].

[30] T. Min and W. Cai, “A Security Case Study for Blockchain Games,” in *2019 IEEE Games, Entertainment, Media Conference (GEM)*, New Haven, CT, USA, 2019, pp. 1-8, doi: 10.1109/GEM.2019.8811555.

[31] A. Carvalho, “Bringing transparency and trustworthiness to loot boxes with blockchain and smart contracts,” *Decision Support Systems,* vol. 144, p. 113508, 2021, doi: 10.1016/j.dss.2021.113508.

[32] E. Mannai, H. K. B. Ayed, I. A. Abdelmoula and B. Zaghdoudi, “Toward Interoperability of NFT Marketplaces,” in *2023 International Wireless Communications and Mobile Computing (IWCMC)*, Marrakesh, Morocco, 2023, pp. 1679-1683, doi: 10.1109/IWCMC58020.2023.10182583.

[33] R. K. Kaushal, N. Kumar, S. N. Panda and V. Kukreja, “Immutable Smart Contracts on Blockchain Technology: Its Benefits and Barriers,” in *2021 9th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO)*, Noida, India, 2021, pp. 1-5, doi: 10.1109/ICRITO51393.2021.9596538.

[34] Kapcrew, “Pixudi Litepaper,” 2024. [Online]. Available: https://docs.pixudi.com/pixudi-manual. [Accessed 20 March 2025].

[35] Pixels, “Pixels Litepaper,” [Online]. Available: https://whitepaper.pixels.xyz/pixels.xyz-lite-paper/core-pillars. [Accessed 20 March 2025].

[36] Pixels, “Pixels Roadmap,” 2023. [Online]. Available: https://whitepaper.pixels.xyz/roadmap. [Accessed 20 March 2025].

[37] Sky Mavis, [Online]. Available: https://hub.skymavis.com/games?partnershipType=sky-mavis,axie-builder-program. [Accessed 14 May 2025].

[38] Sky Mavis, “Sky Mavis team,” [Online]. Available: https://skymavis.com/team. [Accessed 14 May 2025].

[39] Sky Mavis, “Axie Infinity – Smart contracts and GitHub Repo,” 2022. [Online]. Available: https://whitepaper.axieinfinity.com/technology/key-smart-contracts. [Accessed 20 March 2025].

[40] Sky Mavis, “Introduction to Ronin,” 11 February 2025. [Online]. Available: https://docs.roninchain.com/basics/introduction. [Accessed 20 March 2025].

[41] DappRadar, “DappRadar,” [Online]. Available: https://dappradar.com/narratives/gaming/games. [Accessed 20 March 2025].

[42] S. Zoting, “Online Microtransactions Market Size, Shar and Trends 2024 to 2034,” Precedence Research, 2024.

[43] P. Lafourcade and M. Lombard-Platet, “About Blockchain Interoperability,” 2020.

[44] LayerZero Labs, “How it works,” [Online]. Available: https://layerzero.network/how-it-works. [Accessed 25 March 2025].

[45] Rarible, [Online]. Available: https://rarible.com/. [Accessed 14 May 2025].

[46] Magic Eden, [Online]. Available: https://magiceden.io/. [Accessed 14 May 2025].

[47] H. Cryer, “MMO dev fined record-breaking $9 million fee for misleading players over microtransactions,” gamesradar+, 4 Janruary 2024. [Online]. Available: https://www.gamesradar.com/mmo-dev-fined-record-breaking-dollar9-million-fee-for-misleading-players-over-microtransactions/. [Accessed 14 May 2025].

[48] D. Pearson, “China forces devs to reveal loot box drop rates in game,” Games Industry.biz, 9 December 2016. [Online]. Available: https://www.gamesindustry.biz/china-forces-devs-to-reveal-loot-box-drop-rates-in-game. [Accessed 26 March 2025].

[49] L. Y. Xiao, “Loot Box State of Play 2024: Another trip around the world of regulation,” Game Industry.biz, 17 December 2024. [Online]. Available: https://www.gamesindustry.biz/loot-box-state-of-play-2024-another-trip-around-the-world-of-regulation. [Accessed 26 March 2025].

[50] K. B. Muthe, K. Sharma and K. E. N. Sri, “A Blockchain Based Decentralized Computing And NFT Infrastructure For Game Networks,” in *2020 Second International Conference on Blockchain Computing and*

Applications (BCCA), Antalya, Turkey, 2020, pp. 73-77, doi: 10.1109/BCCA50787.2020.9274085.

[51] R. Heeks, “Current Analysis and Future Research Agenda on 'Gold Farming': Real-World Production in Developing Countries for the Virtual Economies of Online Games,” 2008, Development Informatics Working Paper no. 32, 2008, Available at SSRN: https://ssrn.com/abstract=3477387 or http://dx.doi.org/10.2139/ssrn.3477387.

[52] V. Buterin, “Sharding FAQ,” 31 December 2017. [Online]. Available: https://vitalik.eth.limo/general/2017/12/31/sharding_faq.html. [Accessed 26 March 2025].

[53] J. Poon and T. Dryja, “The Bitcoin Lightning Network: Scalable Off-Chain Instant Payments,” 14 January 2016. [Online]. Available: https://lightning.network/lightning-network-paper.pdf. [Accessed 27 March 2025].

[54] Ethereum Foundation, “What is layer 2?,” 12 March 2025. [Online]. Available: https://ethereum.org/en/layer-2/learn/. [Accessed 27 March 2025].

[55] Ethereum Foundation, “Sidechains,” 29 March 2024. [Online]. Available: https://ethereum.org/en/developers/docs/scaling/sidechains/. [Accessed 27 March 2025].

[56] YouGov, “2024 Essential Facts About the U.S. Video Game Industry,” Entertainment software association, 2024.

[57] N. Pirpattipanad and P. Ratanaworachan, “User Experiences on a Blockchain-Based Ticket Sales Platform,” in *2024 28th International Computer Science and Engineering Conference (ICSEC)*, Khon Kaen, Thailand, 2024, pp. 1-6, doi: 10.1109/ICSEC62781.2024.10770641.

[58] BBC News, “China declares all crypto-currency transactions illegal,” 24 September 2021. [Online]. Available: https://www.bbc.com/news/technology-58678907. [Accessed 27 March 2025].

[59] Regulation (EU) 2023/1114 of the European Parliament and of the Council of 31 May 2023 on Markets in Crypto-assets (consolidated version: 09 Jan. 2024), EUR-Lex. [Online]. Available: https://eur-lex.europa.eu/legal-

content/EN/TXT/?uri=CELEX%3A02023R1114-20240109. [Accessed 27 March 2025].

[60] T. Rajic and J. Brock, “The ByBit Heist and the Future of U.S. Crypto Regulation,” Center for Strategic and International Studies, 2025.

[61] V. Buterin, “Ethereum Whitepaper,” 2014.

[62] Web3.js contributors, “Web3.js v1.10.0 documentation,” [Online]. Available: https://web3js.readthedocs.io/en/v1.10.0/. [Accessed 28 March 2025].

[63] Ethers.js contributors, “Ethers.js documentation,” 2023.

[64] D. He, Z. Deng, Y. Zhang, S. Chan, Y. Cheng and N. Guizani, “Smart Contract Vulnerability Analysis and Security Audit,” *IEEE Network,* vol. 34, no. 5, p. 276–282, September/October 2020 , doi: 10.1109/MNET.001.1900656.

[65] C. Liu, H. Liu, Z. Cao, Z. Chen, B. Chen and B. Roscoe, “ReGuard: Finding Reentrancy Bugs in Smart Contracts,” in *2018 IEEE/ACM 40th International Conference on Software Engineering: Companion (ICSE-Companion)*, Gothenburg, Sweden, 2018, pp. 65-68.

[66] F. Vogelsteller and V. Buterin, "ERC-20: Token Standard," Ethereum Improvement Proposals, no. 20, November 2015. [Online serial]. Available: https://eips.ethereum.org/EIPS/eip-20.

[67] W. Entriken, D. Shirley, J. Evans and N. Sachs, "ERC-721: Non-Fungible Token Standard," Ethereum Improvement Proposals, no. 721, January 2018. [Online serial]. Available: https://eips.ethereum.org/EIPS/eip-721.

[68] OpenZeppelin, “ERC-721,” [Online]. Available: https://docs.openzeppelin.com/contracts/4.x/erc721. [Accessed 3 April 2025].

[69] W. Radomski, A. Cooke, P. Castonguay, J. Therien, E. Binet and R. Sandford, "ERC-1155: Multi Token Standard," Ethereum Improvement Proposals, no. 1155, June 2018. [Online serial]. Available: https://eips.ethereum.org/EIPS/eip-1155.

[70] E. Wells, “Web3.js to sunset: ChainSafe transitioning developers,” ChainSafe, 15 January 2025. [Online]. Available: https://blog.chainsafe.io/web3-js-sunset/. [Accessed 2 April 2025].

[71] viem, “viem,” [Online]. Available: https://viem.sh/. [Accessed 2 April 2025].

[72] J. M. Willenbring, S. S. Shende and T. Gamblin, “Providing a Flexible and Comprehensive Software Stack Via Spack, an Extreme-Scale Scientific Software Stack, and Software Development Kits,” *Computing in Science & Engineering,* vol. 26, no. 1, p. 20–30, January–March 2024, doi: 10.1109/MCSE.2024.3395016.

[73] ChainSafe, “Introducing ChainSafe Gaming,” [Online]. Available: https://docs.gaming.chainsafe.io/. [Accessed 3 April 2025].

[74] Sequence, “Game Engine SDKs - Unreal,” [Online]. Available: https://docs.sequence.xyz/sdk/unreal/introduction. [Accessed 3 April 2025].

[75] Sequence, “Game Engine SDKs - Unity,” [Online]. Available: https://docs.sequence.xyz/sdk/unity/overview. [Accessed 3 April 2025].

[76] Zen Republic, “Solana Godot SDK,” [Online]. Available: https://zenwiki.gitbook.io/solana-godot-sdk-docs. [Accessed 3 April 2025].

[77] Virus-Axel, “Godot Solana SDK,” [Online]. Available: https://github.com/Virus-Axel/godot-solana-sdk. [Accessed 3 April 2025].

[78] W. Van Lancker, D. Van Deursen, E. Mannens and R. Van de Walle, “HTTP adaptive streaming with Media Fragment URIs,” in *2011 IEEE International Conference on Multimedia and Expo (ICME)*, Barcelona, Spain, 2011, pp. 1-6, doi: 10.1109/ICME.2011.6012149.

[79] T. Berners-Lee, R. Fielding, U. Irvine and L. Masinter, “Uniform Resource Identifiers (URI): Generic Syntax,” August 1998. [Online]. Available: https://www.ietf.org/rfc/rfc2396.txt. [Accessed 3 April 2025].

[80] Metana Editorial, “Web3 Gaming Tools, Frameworks, and Engines for Building Decentralized Games,” Metana, 12 March 2025. [Online]. Available: https://metana.io/blog/web3-gaming-tools-frameworks-and-engines-for-building-decentralized-games/?utm_source=chatgpt.com#post-content-single-blog-content. [Accessed 2 April 2025].

[81] Nethereum, [Online]. Available: https://nethereum.com/. [Accessed 17 May 2025].

[82] Infura, [Online]. Available: https://www.infura.io/. [Accessed 17 May 2025].

[83] P. K. S. Shwetabh Srivastava, “Performance analysis of Sun RPC,” in *2013 National Conference on Parallel Computing Technologies (PARCOMPTECH)*, Bangalore, India, 2013, pp. 1-9, doi: 10.1109/ParCompTech.2013.6621405.

[84] Ethereum Foundation, “Networks,” 16 April 2025. [Online]. Available: https://ethereum.org/en/developers/docs/networks/. [Accessed 21 May 2025].

[85] M. Willson, “Mainnet vs Testnet,” Blockchain Council, 22 April 2025. [Online]. Available: https://www.blockchain-council.org/blockchain/mainnet-vs-testnet/. [Accessed 17 May 2025].

[86] Remix, [Online]. Available: https://remix-project.org/?lang=en. [Accessed 17 May 2025].

[87] Ethereum Foundation, “ERC-721 Non-Fungible Token Standard,” 19 November 2023. [Online]. Available: https://ethereum.org/en/developers/docs/standards/tokens/erc-721/. [Accessed 20 May 2025].