THE HONG KONG
POLYTECHNIC UNIVERSITY
香港理工大學
Pao Yue-kong Library
包玉剛圖書館

Copyright Undertaking

This thesis is protected by copyright, with all rights reserved.

By reading and using the thesis, the reader understands and agrees to the following terms:

1. The reader will abide by the rules and legal ordinances governing copyright regarding the use of the thesis.
2. The reader will use the thesis for the purpose of research or private study only and not for distribution or further reproduction or any other purpose.
3. The reader agrees to indemnify and hold the University harmless from and against any loss, damage, cost, liability or expenses arising from copyright infringement or unauthorized usage.

IMPORTANT

If you have reasons to believe that any materials in this thesis are deemed not suitable to be distributed in this form, or a copyright owner having difficulty with the material being included in our database, please contact lbsys@polyu.edu.hk providing details. The Library will look into your claim and consider taking remedial action upon receipt of the written requests.

Pao Yue-kong Library, The Hong Kong Polytechnic University, Hung Hom, Kowloon, Hong Kong

http://www.lib.polyu.edu.hk

CONSISTENT PHYSICAL-TO-VIRTUAL MAPPING IN DIGITAL TWINS

YINFENG CAO

PhD

The Hong Kong Polytechnic University

2025

The Hong Kong Polytechnic University

Department of Computing

Consistent Physical-to-Virtual Mapping in Digital Twins

Yinfeng Cao

A thesis submitted in partial fulfillment of the requirements for
the degree of Doctor of Philosophy
April 2025

CERTIFICATE OF ORIGINALITY

I hereby declare that this thesis is my own work and that, to the best of my knowledge and belief, it reproduces no material previously published or written, nor material that has been accepted for the award of any other degree or diploma, except where due acknowledgment has been made in the text.

Signature: ____________________

Name of Student: Yinfeng Cao

Abstract

Digital Twin (DT) technology enables creating virtual representations of physical-world entities, thus facilitating advanced monitoring, simulation, and interaction by mapping physical objects and assets into virtual platforms like the metaverse, which is considered as the key technique in Industry 5.0. Based on DT, many enhanced applications such as smart cities, smart manufacturing, and immersive learning have emerged in recent years, providing users with immersive experience and interactions with the physical world.

In DT systems, the physical-to-virtual mapping is the key process ensures that digital representations (twins) consistently reflect their physical counterparts' attributes and states in real time, thus ensuring the fidelity. This process is typically implemented by physically deployed sensors and servers collecting and processing data from massive and distributed physical objects and assets in untrusted and dynamic environments, which pose several challenges. First, maintaining state consistency between rapidly changing physical objects and their digital twins is difficult due to distributed data sources, environmental uncertainties like potential attacks and device failures, which will emit incorrect states and eventually compromises the accuracy of virtual interactions. Second, the uniqueness of the mapping is hard to verify. Specifically, ensuring each digital twin uniquely corresponds to its physical entity (e.g., one physical human only have one identify in metaverse) is essential to prevent misrepresentation and Sybil attacks, particularly when valuable digital assets are involved. However,

due to the limited observation ability of virtual world, such uniqueness is difficult to verify and guarantee. Third, enabling the interoperability to securely relaying of physical and digital assets across different systems (including traditional financial databases, diverse blockchains, and metaverse platforms) is vital for asset utilization and exchange. However, due to the heterogeneity the lack of trust among different systems, designing a trustless and efficient interoperability solution is challenging.

This thesis proposes a comprehensive framework to address these consistency issues in physical-to-virtual mapping by integrating edge computing and blockchain technology. To achieve real-time projection and ensure state consistency, we develop edge blockchain-based systems (inspired by PolyVerse and PolyTwin). These utilize edge devices and sensors coordinated by a blockchain network to extract physical attributes and states, generating DTs. Crucially, a Proof-of-Consistency (PoC) protocol, run via localized consensus among edge devices, cross-verifies the generated DT data against physical inferences before propagation, ensuring mapping accuracy and resilience to edge-level inconsistencies.

To guarantee the uniqueness of the mapping between physical entities and their digital representations, thereby preventing Sybil attacks, the framework incorporates Eden, an edge-empowered Proof-of-Personhood (PoP) protocol. Eden combines physical-world human verification on privacy-preserving edge devices with on-chain transactional analysis. Its decentralized Proof-of-Trustworthiness (PoT) consensus assigns a verifiable score, securely binding a unique user to a single digital identity and ensuring the authenticity of the mapped identity.

Finally, to facilitate secure asset relaying and sharing across different technological domains, we introduce MAP, a scalable and trustless blockchain interoperability protocol. MAP employs a unified relay chain architecture to connect heterogeneous systems efficiently, eliminating quadratic scaling issues. Its optimized zk-SNARK-based hybrid light client significantly reduces the computational and on-chain costs associated with cross-domain transaction verification, enabling secure and efficient

mapping and transfer of assets between diverse platforms.

In summary, this thesis systematically investigates the requirements for and provides solutions to achieve consistent physical-to-virtual mapping for Digital Twins in environments like the metaverse, leveraging edge computing and blockchain. The proposed solutions (PolyVerse/PolyTwin with PoC, Eden PoP, and MAP interoperability) collectively address the critical challenges of state consistency, identity uniqueness, and cross-domain asset relaying in a decentralized manner. Through extensive evaluation and practical prototype implementations, we demonstrate the effectiveness and efficiency of this framework, contributing to the development of more reliable and integrated physical-virtual systems. Future directions focus on enhancing this mapping framework further.

Publications Arising from the Thesis

1. Yinfeng Cao, Jiannong Cao, Baoxia Du, and Ruidong Li, “Decentralized Digital Twin Network”, *IEEE Communications Magazine*, 2025.

2. Yinfeng Cao, Jiannong Cao, Dongbin Bai, Long Wen, Yang LIU, and Ruidong Li, “MAP the Blockchain World: A Trustless and Scalable Blockchain Interoperability Protocol for Cross-chain Applications”, in *The ACM Web Conference 2025 (WWW 2025)*, Sydney, Australia, 2025.

3. Yinfeng Cao, Jiannong Cao, Hongbo Liu, Zeyang Cui and Long Wen, “Eden: An Edge Computing Empowered Proof-of-Personhood Protocol for Anti-Sybil in Blockchain-based Metaverse”, in *The 2nd International Conference on Intelligent Metaverse Technologies & Applications (iMETA2024)*, Dubai, UAE, 2024. **Best Paper Award Nomination**.

4. Yinfeng Cao, Jiannong Cao, Zeyang Cui, Dongbin Bai, Mingjin Zhang, Long Wen, “PolyTwin: Edge Blockchain-empowered Trustworthy Digital Twin Network for Metaverse”, in *The 2nd Annual IEEE International Conference on Metaverse Computing, Networking, and Applications (IEEE MetaCom 2024)*, Hong Kong, 2024.

5. Yinfeng Cao, Jiannong Cao, Dongbin Bai, Zhiyuan Hu, Kaile Wang, Mingjin

Zhang, “PolyVerse: An Edge Computing-Empowered Metaverse with Physical-to-Virtual Projection”, in *IEEE International Conference on Intelligent Metaverse Technologies and Applications (iMETA2023)*, Tartu, Estonia, 2023. **Best Paper Award Nomination**.

6. Yinfeng Cao, Jiannong Cao, Yuqin Wang, Kaile Wang, and Xun Liu, “Security in Edge Blockchains: Attacks and Countermeasures”, in *ZTE COMMUNICATIONS*, 20(4), 2022.

7. Shan Jiang, Jiannong Cao, Juncen Zhu, and Yinfeng Cao, “Polychain: a generic blockchain as a service platform”, in *Blockchain and Trustworthy Systems: Third International Conference, BlockSys 2021*, Guangzhou, China, 2021. **Best Paper Award**.

Acknowledgments

With profound reflection, I contemplate the transformative journey that began nearly six years ago when I first set foot in Hong Kong in July 2019. The Intelligent Mobile Computing Laboratory (IMCL) at The Hong Kong Polytechnic University became not merely an academic setting, but rather the crucible in which my scholarly identity was forged. Within these hallowed walls, I experienced the trepidation and subsequent elation of my inaugural academic discussion, the nervous anticipation preceding my first presentation, the indescribable satisfaction of publishing my maiden paper, the humbling honor of receiving my first citation, and ultimately, the culmination of these efforts in my first top-tier conference publication. These milestones have bestowed upon me not only comprehensive professional research acumen and collaborative competencies, but also an ineffable sense of scholarly purpose. Throughout this odyssey, I have been blessed with profound friendships among my IMCL colleagues—bonds tempered in the fires of global pandemic, strengthened during recovery, and deepened through countless moments of shared triumph and tribulation. These connections have transcended mere professional association to become the most cherished treasures of my academic sojourn.

I wish to express my most heartfelt and profound gratitude to Prof. Jiannong Cao, whose exceptional mentorship illuminated my path in research with unwavering brilliance. Prof. Cao did not merely guide my academic endeavors; he fundamentally transformed my understanding of scholarly pursuit and equipped me with the intellectual foundations essential for my professional trajectory. Through his exemplary mentorship and inspirational leadership, Prof. Cao personified the quintessential attributes of academic excellence. In moments of profound uncertainty and personal challenge, Prof. Cao extended beyond his role as academic supervisor to provide paternal wisdom and moral guidance, imparting lessons on the essence of principled manhood: embracing responsibility with dignity, exercising leadership with equity, and confronting adversity with unwavering resolve.

I am immeasurably indebted to Prof. Ruidong Li of Kanazawa University, Japan, whose exceptional tutelage during my six-month research residency proved transformative. Prof. Li's profound insights illuminated the true essence of rigorous scholarship and revealed the indispensable qualities inherent to the researcher's vocation. Beyond his academic counsel, Prof. Li instilled in me the virtues of contemplative self-examination, steadfast perseverance, and unwavering dedication to purpose—principles that have become the cornerstone of my scholarly ethos.

My deepest appreciation extends to my distinguished co-authors: Dr. Long Wen, Dr. Mingjin Zhang, Mr. Dongbin Bai, Mr. Baoxia Du (Kanazawa University), Mr. Hongbo Liu, Mr. Zhiyuan Hu, Mr. Yuqing Wang, Ms. Kaile Wang, and others whose contributions have been instrumental to my academic progression. Their intellectual generosity, unwavering support, and scholarly camaraderie enabled me to surmount seemingly insurmountable research challenges with renewed determination and clarity of purpose.

I hold in equal esteem my esteemed colleagues from both the IMCL at The Hong Kong Polytechnic University and the INCL at Kanazawa University. These extraordinary individuals shared in the intellectual rigors and emotional vicissitudes of research life—offering encouragement during moments of doubt, enlightenment during periods of confusion, and steadfast companionship throughout this arduous yet rewarding journey. Their presence sustained my resolve during moments of hesitation and reinforced my commitment to scholarly excellence.

Finally, I reserve my most profound and heartfelt gratitude for my beloved parents and family, whose boundless love, unwavering faith, and silent sacrifices have constituted the emotional bedrock upon which my academic aspirations have been built. Their unconditional support transcends mere acknowledgment; it represents the very foundation of my achievements and the wellspring of my perseverance. I shall eternally cherish the profound privilege of their presence in my life and strive to honor their sacrifices through continued dedication to scholarly excellence.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

This research studies how to map the physical-world objects and assets into a virtual-world platform (such as metaverse) while ensuring the their consistency in terms of attributes and states by projecting, relaying, and securing them with edge computing and blockchain techniques and components. In this chapter, we first introduce the background and motivations of mapping the physical-world objects assets into virtual world in digital twins. In Section 1.1, we discuss the background and motivations of this research. We then present the research objectives and framework in Section 1.2. Finally, we provide an overview of the thesis organization in Section 1.3.

1.1 Background and Motivations

Digital Twin (DT) technology has emerged as a transformative paradigm that creates virtual representations of physical objects and environments. Since its initial conception for industrial applications, DT implementations have rapidly evolved into complex ecosystems that leverage sensing technologies, edge computing, and extended reality (XR) to enable real-time synchronization between physical assets and their digital counterparts (twins) [61, 12]. Recent advancements in blockchain technology have further enhanced DTs by enabling decentralized ownership, secure data exchange, and tamper-proof historical records of physical object states [70]. These capabilities have expanded DT applications beyond traditional manufacturing into diverse domains including smart cities, healthcare, and immersive metaverse platforms.

The core value of Digital Twins lies in their ability to create high-fidelity virtual representations that accurately model both the physical characteristics and dynamic behaviors of real-world objects. By continuously collecting and processing data from sensors, DTs bridge the physical-virtual divide, enabling remote monitoring, simulation, and interaction with real-world systems [61]. This capability has profound implications for various applications, particularly in creating immersive virtual environments like metaverse platforms. For example, DT-enabled educational environments can provide realistic campus experiences for remote or disabled students by capturing and rendering real-time on-campus activities, enhancing their sense of presence and participation [28].

However, existing DT solutions face three key challenges when deployed in open, decentralized environments. First, generating DTs for complex physical systems requires simultaneous processing from large-scale, geo-distributed objects and assets, demanding substantial computing resources. Traditional centralized DT architectures fail to scale efficiently, resulting in high latency and hardware costs when processing data from multiple distributed sensors. Additionally, DTs deployed in unmanaged environments are vulnerable to malicious attacks or device malfunctions, leading to *inconsistent* representations that compromise accuracy in subsequent processing stages. Second, DT identity security remains inadequately addressed, particularly in open environments. Current identification systems are vulnerable to *Sybil attacks*, where malicious actors create multiple fake identities to manipulate the system, potentially compromising DT integrity and trustworthiness. Third, as DTs increasingly function as tokenized digital assets in applications like metaverse platforms, their secure sharing across different technological domains remains unresolved. Existing systems typically operate within isolated ecosystems, while cross-domain solutions are underdeveloped, leaving digital assets isolated and underutilized.

1.2 Research Objectives and Framework

In this thesis, we study the consistent physical-to-virtual mapping problem in digital twins. Specifically, given a set of physical objects (e.g, vehicles, buildings, and pedestrians for general application purposes) and assets (e.g, balance, properties, and equipment for financial purposes), we aim to design methods of consistently and uniquely project physical objects into virtual objects in real time and relay assets into virtual world as digital assets by designing and optimizing the blockchain consensus with edge computing techniques. Within the framework, I develop an edge blockchain-based digital twin projection system. Above the system, I designed and integrated a proof-of-personhood protocol for unique identity management and and a cross-chain protocol for secure digital twin asset relaying and sharing, respectively.

The system architecture of our consistent physical-to-virtual mapping system for digital twins is illustrated in Figure 1.1. The architecture comprises two parts: the physical worlds consisting of objects and assets, and the virtual world consisting the mapped objects and assets for applications. Two worlds are connected by our three mapping layers. In the

The bottom layer consists of decentralized edge devices equipped with sensor clusters from multiple stakeholders. These edge devices jointly generate digital twins by continuously collecting attributes and states from physical objects and processing them into digital twin asset semantics, thereby achieving low latency and high scalability through localized computation.

The middle layer forms our primary research focus, where we study blockchain-based methods for digital twin asset validation, management, and sharing while ensuring trustworthiness. In this layer, we aim to leverage a blockchain network overlaid on the edge device infrastructure, with each edge device functioning as a blockchain node. This blockchain network performs three critical functions: validating consistency for each newly generated digital twin asset, verifying and assigning unique identifiers to these assets, and facilitating their secure relay across different metaverse platforms or blockchains. These capabilities enable the framework to achieve decentralized security that withstands various malicious adversaries and failures. The top layer, the application layer, retrieves and aggregates digital twins and renders them into virtual environments applications. Based on this architecture, we have implemented and deployed several prototype systems, such as immersive metaverse campus environment and digital twins for smart city traffic monitoring.

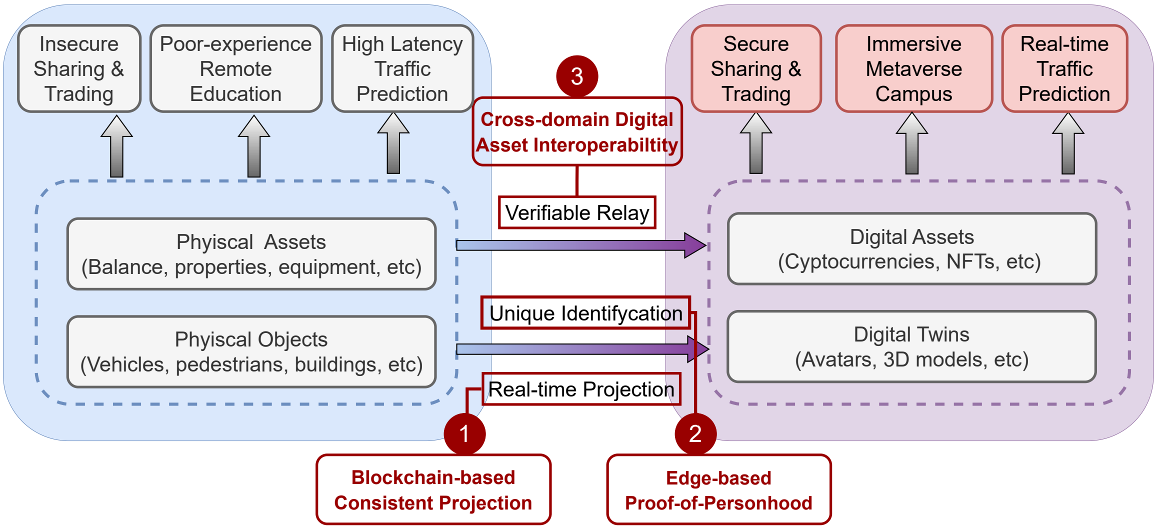

Insecure Sharing & Trading
Poor-experience Remote Education
High Latency Traffic Prediction
Phyiscal Assets (Balance, properties, equipment, etc)
Phyiscal Objects (Vehicles, pedestrians, buildings, etc)
3
Cross-domain Digital Asset Interoperabiltity
Verifiable Relay
Unique Identifycation
Real-time Projection
1
Blockchain-based Consistent Projection
2
Edge-based Proof-of-Personhood
Secure Sharing & Trading
Immersive Metaverse Campus
Real-time Traffic Prediction
Digital Assets (Cyptocurrencies, NFTs, etc)
Digital Twins (Avatars, 3D models, etc)

Figure 1.1: System architecture of blockchain-based edge computing for trustworthy digital twins in metaverse.

The rest of the thesis is organized as follows:

1.3 Thesis Organization

The remainder of this thesis is organized as follows:

- In Chapter 2, we review the literature relevant to this thesis, including digital twin and metaverse systems, metaverse identity management, and cross-domain asset sharing.

- In Chapter 3, we design and develop edge blockchain-based consistent projection systems (PolyVerse and PolyTwin). We introduce an edge computing architecture that efficiently collects physical object attributes through deployed sensors and AI models. To ensure digital twin consistency, we develop a Proof of Consistency (PoC) mechanism that cross-verifies digital twins against physical object inference results on the blockchain, and an optimized efficient data structure for storing digital twin states. We implement two prototypes—PolyCampus and PolyExchange—that enable immersive physical-virtual interaction in metaverse applications. Experimental results demonstrate the practical efficiency, and we formally prove its security under standard blockchain security boundaries.

- In Chapter 4, we propose Eden, an edge computing-empowered Proof-of-Personhood protocol for addressing Sybil attacks in the metaverse. Eden employs a hybrid approach that combines video-based human recognition with on-chain transactional activity analysis to establish unique bindings between users and metaverse identities (like wallet addresses). We develop specialized edge devices with on-device recognition models to verify human users in the physical world. To counter advanced AI-based Sybil attacks, we also analyze transactional patterns using a scorecard-based regression model. Results demonstrate that Eden can effectively identify human users with high accuracy in metaverse.

- In Chapter 5, we introduce MAP, a scalable and trustless blockchain interoperability protocol that can securely relay digital assets among heterogeneous chains. MAP is designed to minimize computational costs when scaling to new chains, which employs a relay chain architecture that eliminates the need for pairwise chain-to-chain light clients. To optimize transaction verification efficiency, we propose a zk-based hybrid light client scheme that adaptively decouples signature verification workloads based on their performance characteristics in on-chain smart contracts versus off-chain circuits.

- In Chapter 6, we conclude the thesis by summarizing the main contributions and outlining our vision for future research directions.

Chapter 2

Literature Review

2.1 Digital Twin Systems and Platforms

Industrial Platform. In recent years, several representative industrial digiltal twin metaverse platforms have emerged. Sandbox Games like Sandbox [3] and Decentraland [1] offer immersive experiences through digital asset trading based on blockchain and avatar interaction in virtual 3D environments. Simulation Platforms like NVIDIA's Omniverse [42] provide real-time 3D simulations and visualizations for industrial applications. Collaboration Tools, such as Meta Horizon Workrooms [5], facilitate productive and collaborative VR experiences for enterprise teams.

Research Efforts. Recent research efforts have also explored various aspects of the metaverse. Duan et al. built a blockchain-driven virtual campus and discussed its benefits for social goods [28]. Lam et al. proposed a human-avatar framework with full-body motion capture for metaverse [49]. Dhelim et al. proposed hybrid Fog-Edge computing architectures for metaverse tasks like 3D simulation [26]. Wang et al. designed a framework for Metaverse classrooms to achieve real-time synchronization of a large number of participants through VR equipment and sensors [92]. Shen et al. introduced the cyber-physical-social system (CPSS) paradigm to enhance lecturers' space immersion through sparse consumer-grade RGBD cameras [78]. Cai and Karunarathna et al. discussed networking optimization for efficient metaverse systems [20] [45]. Several works focusing on the synchronization perspective, proposing strong algorithms for digital twins [105] [106].

2.2 Identity Management in Digital Twins

General Biometric Authentication. The Humanity Protocol and Proof-of-Humanity Protocol leverage video-based recognition for PoP [8] [9]. Specifically, they use cameras to recognize faces and palm prints and link them to wallets. Only holders who pass such authentication can access the wallets, thus ensuring that a wallet holder is actually a human. However, with the development of generative AI, this general biometric authentication can be insecure. Sybil users can generate fake faces or palms to cheat the cameras, thus controlling multiple Sybil wallets.

Specialized Biometric Authentication. WorldCoin is a PoP solution co-founded by OpenAI's Sam Altman [6]. They have developed a specialized iris-scanning device called Orb, which scans users' irises and links them to a wallet. Additionally, they have integrated a UBI project that sends free tokens to motivate users. The iris data is stored in a centralized database protected by zero-knowledge proof systems. However, WorldCoin faces privacy issues and high development costs. The operation within their databases lacks transparency and the deployment of iris-scanning devices on a large scale can be very expensive.

Empirical Analysis.Gitcoin Passport and many token airdrop activities use empirical analysis-based strategies to detect and ban Sybil wallets [7]. These strategies require wallets to meet specific criteria, such as being attached to social media accounts, frequently interacting with certain smart contracts, and having sufficient balances. This approach is practical, as it only requires on-chain data, but it may be bypassed by advanced bots. For instance, some bots facilitate automatic 'farming' by having Sybil wallets interact with other wallets, giving the impression that they are controlled by a non-Sybil user.

Graph-based Analysis. Many research works use graph analysis techniques and machine learning models such as GNNs to classify and detect Sybil wallets [23] [86] [37]. These models aim to identify hidden relationships and patterns of wallets, thereby detecting possible Sybil wallets. However, the accuracy of these approaches is highly dependent on the availability and quality of data, which is difficult to guarantee in the blockchain world. Most addresses are anonymous and lack any label that indicates whether the wallet is a Sybil or not.

Physical Activity Verification. The encounter protocol requires physical attendance at periodic meetings in specific regions to ensure that each user can appear in only one location, thus guaranteeing personhood [18]. Similarly, BrightID uses QR codes on phones to enable users to verify and link each other's identity offline. Therefore, users with many links are considered real humans [10]. Physical verification is relatively privacy-preserving and secure, but it is inconvenient and impractical for users, as it requires frequent physical meetings.

2.3 Asset Management in Digital twins

Centralized/Committee-based Protocols. To enable efficient interoperability, centralized designs are widely adopted by native protocols. Notary schemes directly host clients' tokens in custodial wallets and designate an authority (such as crypto exchanges) to facilitate their exchange efficiently [15, 25]. Similarly, committee-based protocols, such as MPC bridges and vote-oracle bridges[66, 80], appoint a small group of off-chain committees to verify and vote on cross-chain transactions, offering more decentralized features compared to notary schemes. Despite their convenience and efficiency, both solutions rely on trusting off-chain entities, which are usually not transparent and permissioned, making them vulnerable to internal corruption and attacks [66].

Chain-based Protocols. To further reduce the needed trust, chain-based protocols are developed, which process cross-chain transactions fully on-chain, thus making the protocols trustless. However, these protocols typically suffer from expensive verification and chain heterogeneity. Hash-Time Lock Contracts (HTLCs) are pioneering peer-to-peer protocols that allow users to deploy paired contracts on two chains to control asset release. However, HTLCs lack efficiency [14] because they require manual peer matching, enforcing users to wait for another user with the same token swap demand. As a result, HTLCs are rarely used to support large-scale cross-chain applications. Polkadot and Cosmos (Blockchain of Blockchain, BoB) employ hubs to process cross-chain transactions efficiently [96, 48], but these hubs only support their own specific homogeneous chains. HyperService [59] proposes a cross-chain programming framework, but it still requires significant modifications to the underlying components of heterogeneous chains, which is not feasible for in-production chains. LC-based bridges [50] are currently the mainstream protocols that deploy light clients (LCs) on each chain to verify cross-chain transactions. However, the internal verification workload of on-chain LCs is extremely expensive. ZKLC-based bridges [99, 95] attempt to reduce on-chain cost by moving verification to off-chain provers using zk-SNARKs. Unfortunately, this requires intensive computing power and multiple distributed servers due to the large circuit size of signature verification. Additionally, all LC-based protocols face high scaling cost due to redundant LCs.

Cross-Shard. Another related line of work involves blockchain sharding techniques [76, 71, 67, 40, 100, 53]. In these works, cross-shard processing techniques are developed to retrieve transactions from different shards. While these works share some similarities with cross-chain transaction processing, one key difference is that they only consider single blockchain scenarios, where all nodes trust each other and only simple transaction verification based on Authenticated Data Structure (ADS) is required, such as Merkle proof verification. In contrast, in cross-chain scenarios, blockchains that do not trust each other, and require complicated verification like block header verification.

Chapter 3

Projecting from Physical to Virtual

3.1 Overview

The metaverse represents a paradigm shift towards interconnected, persistent virtual environments where users interact within a shared digital world, blurring the lines between physical and virtual realities [43]. This vision promises immersive and engaging experiences, driving significant interest from both industry and academia. Consequently, numerous metaverse platforms are emerging, targeting diverse applications such as gaming, social interaction, remote collaboration, and complex dynamics simulation.

A cornerstone of metaverse immersiveness is the effective integration of the physical world into the virtual realm. **Physical-to-Virtual Projection (P2V)**, the process of incorporating real-world objects, states, and information into the metaverse, is crucial for enhancing this sense of presence, engagement, and authenticity (see Figure 3.1). P2V enables users to connect with and experience external physical activities without direct physical participation. For instance, educational metaverse platforms aim to replicate campus life for students with disabilities by projecting real-time scenes and activities, fostering a greater sense of inclusion [28].

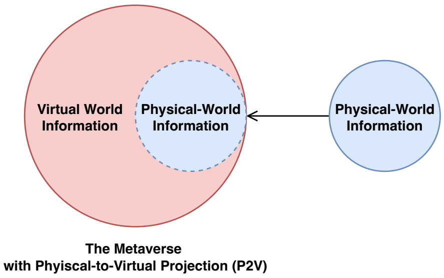

Virtual World Information
Physical-World Information
Physical-World Information
The Metaverse
with Phyiscal-to-Virtual Projection (P2V)

Figure 3.1: A conceptual illustration of Physical-to-Virtual (P2V) projection in the metaverse. Incorporating real-world information significantly enhances the immersive experience.

3.1.1 The Problem and Motivations

Despite the potential of P2V, practical, efficient, and scalable solutions remain elusive. Current metaverse platforms often rely on user-generated 3D models [28] which lack real-time fidelity, or expensive, specialized Virtual Reality (VR) systems for incorporating physical objects [92, 78]. These approaches struggle with the vastness, dynamism, and distributed nature of real-world objects. Capturing the attributes of numerous pedestrians in a busy scene using VR equipment, for example, is impractical. As a result, many platforms lack robust P2V capabilities, limiting their immersiveness and applicability.

Furthermore, the concept of P2V naturally evolves towards the use of **Digital Twins (DTs)** – dynamic virtual representations of physical objects or systems [98]. Adopting DTs allows the metaverse not only to represent physical objects but also to simulate their behavior and characteristics, enabling richer interactions [22, 60]. Examples range from representing real-world goods as Non-Fungible Tokens (NFTs) for virtual trading[1] to real-time rendering and prediction of nearby vehicles in autonomous driving simulations.

However, integrating DTs introduces a critical challenge: **trustworthiness**. Ensur-

[1] https://www.forbes.com/sites/bernardmarr/2022/06/01/the-amazing-ways-nike-is-using-the-metaverse-web3-and-nfts/

ing that a DT accurately reflects its physical counterpart (*correctness*) and that its state remains synchronized with the physical object's state (*consistency*) is paramount, especially in high-stakes applications or value-based interactions like NFT trading (see Figure 3.2). Inconsistency, whether accidental (e.g., network issues) or malicious (e.g., tampering), can undermine user experience, safety, and the perceived value within the metaverse [57, 11]. Existing solutions often rely on centralized trusted parties, which conflicts with the decentralized ethos of the metaverse and raises privacy concerns [21], or naively use blockchain without verifying the link between on-chain data and off-chain reality [103].

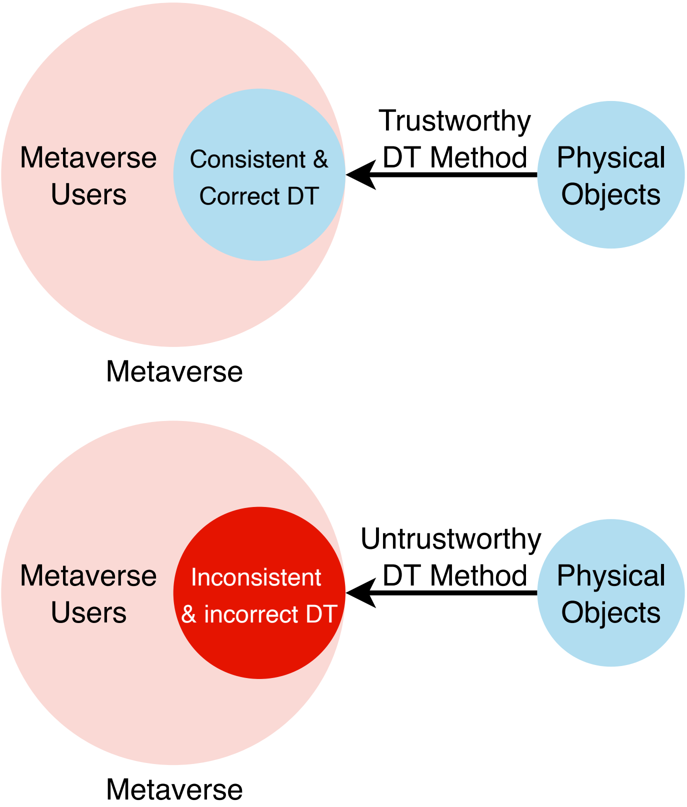


Figure 3.2: A conceptual illustration of the trustworthy Digital Twin problem in the metaverse. Digital twins must be correct (accurately reflecting physical attributes) and consistent (synchronized state) with their physical counterparts.

3.1.2 Research Challenges

Enabling practical, trustworthy P2V and DT integration in the metaverse poses significant research challenges:

1. **Real-time Performance and Scalability:** P2V/DT necessitates the real-time collection, processing, and analysis of potentially massive volumes of data from distributed physical sensors [102]. This demands substantial computation, storage, and networking resources, often unavailable without high-end infrastructure or efficient resource management. Achieving low latency for a large number of dynamic objects is critical for immersion.

2. **State Consistency Across Providers:** Metaverse platforms are often hosted by multiple Metaverse Service Providers (MSPs). Network issues (packet drops, delays) or malicious actions by MSPs can lead to divergent views of the metaverse state among users, compromising the shared experience [91, 32, 39]. Mechanisms are needed to ensure all users perceive a consistent virtual world state.

3. **Digital Twin Trustworthiness (Correctness and Consistency):** Beyond provider-level consistency, the DTs themselves must be trustworthy representations of physical reality. This involves ensuring the DT generation process correctly captures physical attributes and that mechanisms exist to validate the ongoing consistency between the DT's state and the physical object's state, even under potential adversarial manipulation or model inaccuracies [41, 72].

4. **Privacy Preservation:** While incorporating real-world data enhances immersion, it also raises privacy concerns, especially when dealing with sensitive information like human attributes or identities. Solutions must balance data utility with user privacy, potentially leveraging techniques that process data locally or anonymize it before broader use.

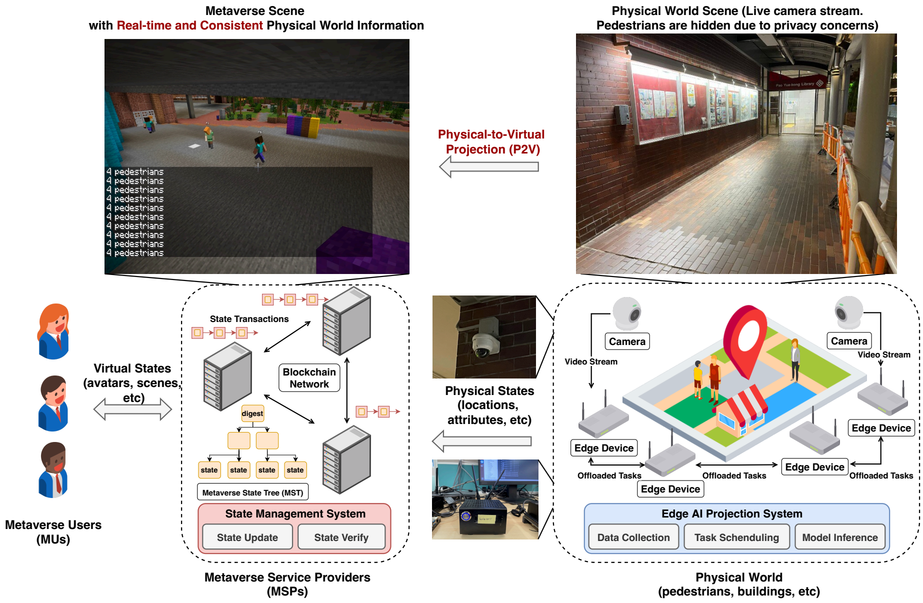


Figure 3.3: PolyVerse Overview: An edge computing-empowered metaverse platform facilitating Physical-to-Virtual projection (P2V). Edge devices with AI models collect and process physical world information in real-time. A blockchain-based state management system ensures consistency among MSPs.

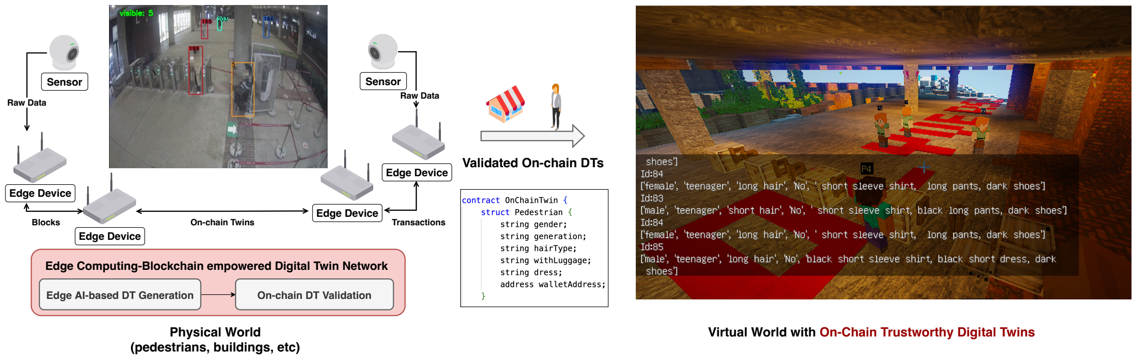
Existing works and platforms often fall short in addressing these challenges comprehensively [101, 69, 91, 33, 38].

3.1.3 Our Approach and Contributions

This chapter presents a framework based on the synergistic integration of two systems, PolyVerse and PolyTwin, designed to address the challenges of incorporating trustworthy digital twins into the metaverse efficiently and scalably. Our approach leverages edge computing for real-time processing and blockchain technology for ensuring consistency and trustworthiness.

As illustrated conceptually in Figure 3.3 (PolyVerse) and Figure 3.4 (PolyTwin), our combined approach features:

- **Edge AI for Real-time P2V/DT Generation:** We deploy clusters of edge



```
contract OnChainTwin {
    struct Pedestrian {
        string gender;
        string generation;
        string hairType;
        string withLuggage;
        string dress;
        address walletAddress;
    }
```

```
 shoes']
Id:84
['female', 'teenager', 'long hair', 'No', ' short sleeve shirt,  long pants, dark shoes']
Id:83
['male', 'teenager', 'short hair', 'No', ' short sleeve shirt, black long pants, dark shoes']
Id:84
['female', 'teenager', 'long hair', 'No', ' short sleeve shirt,  long pants, dark shoes']
Id:85
['male', 'teenager', 'long hair', 'No', 'black short sleeve shirt, black short dress, dark
 shoes']
```

Figure 3.4: PolyTwin Overview: An edge blockchain-enabled trustworthy Digital Twin network. Edge devices generate DTs from physical objects using AI. A blockchain network with Proof-of-Consistency validates the consistency between DTs and physical objects.

computing devices equipped with sensors (e.g., cameras) and AI models directly in the physical world. These edge clusters collect, process, and analyze physical data locally, extracting relevant attributes (e.g., location, appearance, actions) to generate DTs with low latency. This addresses the real-time performance and scalability challenge while also enhancing privacy by processing raw data near the source.

- **Efficient Edge Resource Management:** To handle varying workloads and resource-constrained edge devices, we employ techniques like collaborative edge clustering (distributing model inference tasks across devices) and optimized task scheduling algorithms (e.g., PolyHeuristic) to minimize latency and maximize throughput.

- **Blockchain for Consistency and Trustworthiness:** We utilize blockchain technology in two complementary ways:

  – *MSP State Consistency (PolyVerse):* A blockchain network maintained by MSPs, combined with an efficient data structure (Metaverse State Tree - MST), ensures that all MSPs maintain a consistent view of the overall metaverse state, mitigating issues from network problems or malicious providers.

- *DT Validation (PolyTwin):* A Proof-of-Consistency (PoC) mechanism, executed by edge devices within a cluster and potentially anchored to a blockchain, cross-verifies the generated DT attributes against physical reality (as inferred by multiple models/devices), ensuring the correctness and consistency of the DT itself.

The main contributions presented in this chapter, integrating the work of PolyVerse and PolyTwin, are:

1. **Edge AI-Powered Real-time P2V/DT Framework:** We propose a comprehensive framework utilizing edge computing clusters and AI for efficient, scalable, low-latency generation and projection of digital twins from the physical world into the metaverse.

2. **Optimized Edge Task Scheduling:** We introduce the PolyHeuristic algorithm for scheduling computation tasks (e.g., AI model inference) across heterogeneous edge devices to optimize latency under dynamic workloads and resource constraints.

3. **Blockchain-based State Consistency for MSPs:** We develop a novel state management system using the Metaverse State Tree (MST) data structure and a BFT blockchain consensus mechanism to guarantee a consistent view of metaverse states among potentially untrusted MSPs, enabling efficient state updates and verification.

4. **On-Chain Digital Twin Validation (Proof-of-Consistency):** We propose the PoC mechanism to ensure the trustworthiness (correctness and consistency) of the generated digital twins by leveraging collaborative cross-verification among edge devices within a cluster, secured by blockchain principles.

5. **Prototype Implementation and Evaluation:** We demonstrate the feasibility and effectiveness of our approach through prototype implementations (PolyCampus, PolyExchange) and extensive experimental evaluations, showcasing low-latency performance, scalability, and the ability to maintain consistency and trustworthiness.

To the best of our knowledge, this combined approach represents a novel and comprehensive solution addressing the critical challenges of real-time performance, scalability, consistency, and trustworthiness for incorporating digital twins into the metaverse.

3.2 Background and Related Work

3.2.1 Digital Twins in the Metaverse

A Digital Twin (DT) is a dynamic virtual representation of a physical entity, process, or system. It mirrors the physical counterpart by integrating real-time data, simulation models, and analytics, enabling monitoring, analysis, prediction, and optimization [98]. Initially prominent in manufacturing and engineering (e.g., simulating aircraft engines), the DT concept is increasingly being adopted in the metaverse [22, 60].

In the metaverse context, DTs serve as the bridge connecting the physical and virtual worlds. They allow real-world objects, environments, and even dynamic processes to be represented with high fidelity within the virtual space. This enables users to interact with virtual counterparts that accurately reflect the state and behavior of physical reality, significantly enhancing immersiveness. Examples include:

- **NFT-backed Physical Goods:** Tokenizing real items (like sneakers) as NFTs creates a DT where ownership and potentially state information are linked.

Trading the NFT in the metaverse corresponds to a change in ownership of the physical item.

- **Smart Cities and Environments:** Creating DTs of cities or buildings allows for virtual exploration, simulation of traffic or environmental conditions, and remote management [26].

- **Real-time Simulation:** Autonomous driving systems render nearby vehicles and environments as DTs to predict movements and aid decision-making [102].

- **Immersive Social Interaction:** Projecting real-time activities or environments (like a university campus) allows remote users to participate virtually [28].

3.2.2 Trust Issues in Metaverse Digital Twins

Despite the promise, the integration of DTs into the metaverse surfaces significant trust issues [91]. The value and utility of a DT hinge on its faithfulness to its physical counterpart. We identify two key aspects of trustworthiness:

1. **Correctness:** The DT must be generated based on actual physical data and accurately reflect the relevant attributes of the physical object. An incorrectly generated DT (e.g., wrong attributes assigned) misrepresents reality.

2. **Consistency:** The state of the DT must remain synchronized with the state of the physical object over time. A stale or maliciously altered DT state breaks the link with reality, potentially leading to safety issues (e.g., in autonomous driving) or economic losses (e.g., in NFT trading).

Ensuring both correctness and consistency is challenging. Physical environments are dynamic, data collection can be noisy or incomplete, AI models used for attribute extraction may have inaccuracies, and systems can be subject to network failures or malicious attacks [57, 11]. Malicious actors might tamper with sensor data, model outputs, or communication channels to create inconsistent or incorrect DTs for illicit gain.

3.2.3 Limitations of Existing Approaches

Current approaches to P2V and DT integration often fail to adequately address trustworthiness, scalability, and real-time requirements simultaneously:

- **Centralized Systems:** Relying on a single trusted entity (e.g., a cloud server or a specific provider) simplifies management but creates a single point of failure and trust [21]. Such systems may not align with the decentralized nature of the metaverse and can raise privacy concerns.
- **Specialized Hardware:** Systems based on expensive VR/AR equipment offer high fidelity for limited objects but lack scalability for large, dynamic environments [92, 78].
- **Basic Blockchain Integration:** Many proposals simply store DT data or hashes on a blockchain [41, 72, 38]. However, the blockchain itself typically cannot verify if this data corresponds to physical reality without a trusted oracle or a robust validation mechanism [103]. This does not prevent the injection of incorrect or inconsistent data onto the chain.
- **Synchronization Protocols without Adversarial Tolerance:** Some protocols exist for state synchronization, particularly in manufacturing DTs [83, 77], but they often assume non-adversarial environments and may not be suitable for open, potentially malicious metaverse settings.

Therefore, there is a clear need for a holistic solution that combines the efficiency of edge computing with the trust guarantees of blockchain, specifically tailored to the challenges of real-time, consistent, and trustworthy DTs in the metaverse.

3.3 System Architecture and Design Goals

3.3.1 System Architecture

We propose a layered architecture integrating edge computing and blockchain to support trustworthy DTs in the metaverse. The key entities are:

- *Physical Objects ($\mathcal{PO}$):* Entities in the real world ($po_i \in \mathcal{PO}$) targeted for projection into the metaverse. Each object possesses physical attributes (PA_i). These objects are typically passive sources of information (e.g., pedestrians, vehicles, buildings).

- *Edge Infrastructure:* Consists of sensors (e.g., cameras, LiDAR) deployed in the physical environment and Edge Computing Clusters (ECCs). Each ECC comprises multiple geographically proximate edge devices (e_j) equipped with processing capabilities (CPU/GPU/NPU) and AI models. The edge infrastructure is responsible for:

  - Sensing the physical environment and capturing raw data.
  - Processing raw data using AI models to extract physical attributes (PA_i) and states.
  - Generating intermediate digital twin representations (vo_{int}).
  - Participating in validation protocols (e.g., PoC).
  - Transmitting validated state information to MSPs.

- *Metaverse Service Providers (MSPs):* Entities (m_k) responsible for hosting the metaverse environment ($\mathcal{V}$), managing virtual states, and facilitating user interactions. They:
  - Receive validated physical state information or DT updates from the edge infrastructure.
  - Maintain the state of virtual objects (Digital Twins, VO) corresponding to physical objects ($vo_i \in \mathcal{VO}$, with virtual attributes VA_i).
  - Participate in a blockchain network to ensure consistency of the overall metaverse state among themselves.
  - Serve metaverse content and state information to users.
- *Metaverse Users (MUs):* End-users interacting with the metaverse through client software. They connect to MSPs to send inputs and receive the virtual world state, which their client renders locally. They may also perform verification checks on the received state information.

This architecture distributes the workload: the edge handles real-time sensing and initial processing/validation, while MSPs manage the persistent virtual world state and user interactions, with blockchain providing trust anchors at both levels.

3.3.2 Design Goals

Our integrated system aims to achieve the following key design goals for incorporating DTs into the metaverse:

1. **Real-time Projection:** The system must be able to capture changes in the physical world and reflect them in the corresponding DTs within the metaverse with minimal delay. This requires low latency from sensing to rendering.

Definition (Real-time Projection): Let $P = \{po_1, ..., po_n\}$ be the set of physical objects and V be the virtual environment containing corresponding virtual objects $VO = \{vo_1, ..., vo_n\}$. A projection function $f : P \times T \to V$ maps a physical object po_i at time t to its virtual representation vo_i in V. The projection is real-time if the time Δt required to update vo_i reflecting a change in po_i is less than a predefined threshold ϵ, i.e., $\Delta t < \epsilon$.

2. **State Consistency (among MSPs):** All honest MSPs must maintain an identical view of the metaverse state, including the states of all DTs. Users connecting to different honest MSPs should perceive the same virtual world.

   Definition (MSP State Consistency): Let M be the set of MSPs maintaining the virtual state VO. The system ensures state consistency if: (a) For any two honest MSPs $m, m' \in M$, their representation of any virtual object vo_i at a logical time is identical. (b) If a malicious MSP m^* provides an inconsistent state, honest MUs can detect this inconsistency, potentially via cryptographic proofs.

3. **Digital Twin Trustworthiness (Correctness & Consistency):** The DTs within the metaverse must be faithful representations of their physical counterparts.

   Definition (Trustworthy Digital Twin): A DT vo_i corresponding to po_i is trustworthy if: (a) **Correctness:** Its attributes VA_i are generated based on actual measured physical attributes PA_i from the physical world $\mathcal{P}$. (b) **Consistency:** The state/attributes VA_i remain synchronized with the actual state/attributes PA_i of po_i over time, such that $VA_i \neq PA_i$ only with a negligible probability η, even in the presence of potential adversaries or system faults.

4. **Scalability:** The system should be able to handle a large number of physical objects and users efficiently, scaling resources as needed.

5. **Privacy:** Sensitive information gathered from the physical world should be handled appropriately, minimizing exposure and protecting user privacy, potentially through edge processing and anonymization.

Our proposed techniques, combining edge AI with blockchain mechanisms (MST and PoC), are designed to collectively achieve these goals.

3.4 Edge AI for Real-time Digital Twin Generation and Projection

To meet the goals of real-time projection and scalability, we leverage edge computing infrastructure equipped with AI models. This section details the techniques used for efficient DT generation and projection at the edge.

3.4.1 General Principle

The core idea is to perform the computationally intensive tasks of data acquisition, preprocessing, and AI inference as close to the physical source as possible, using edge devices. This minimizes the latency associated with transmitting large volumes of raw sensor data (e.g., video streams) to a central cloud and allows for quicker updates to the DT states. Furthermore, processing data locally enhances privacy, as sensitive raw data may not need to leave the local edge environment.

3.4.2 Collaborative Edge Clustering (from PolyTwin)

Handling complex AI models (e.g., for simultaneous object detection, attribute recognition, and action analysis) on resource-constrained single edge devices can still lead to high latency. To address this, we employ *collaborative edge clustering*.

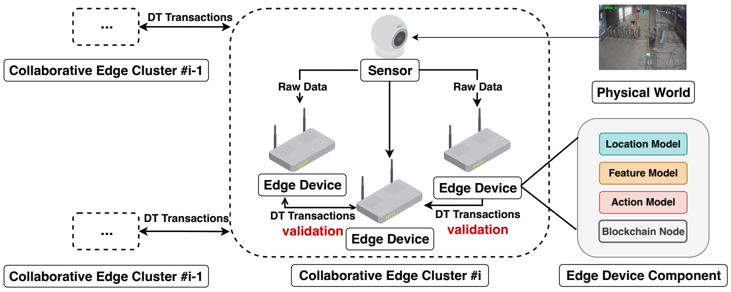


Figure 3.5: Collaborative Edge Clustering: AI model workloads are split and distributed across multiple edge devices within a cluster (e_{lc}, e_f, e_a) to optimize inference efficiency. Devices within the cluster can also cross-verify results via PoC before finalizing the DT.

As shown in Figure 3.5, instead of running a monolithic AI model on one device, we:

1. **Decouple AI Tasks:** Break down the overall task into sub-tasks (e.g., location detection, feature extraction, action recognition).

2. **Deploy Sub-Models:** Deploy specialized, potentially lighter-weight AI sub-models for each sub-task onto different edge devices within a geographically co-located cluster (EC). For instance, e_{lc} handles location, e_f handles features, and e_a handles actions.

3. **Synchronize Inputs:** Raw sensor data (e.g., video from camera cam) is streamed simultaneously to relevant devices in the cluster.

4. **Parallel Inference:** Each device performs inference for its assigned sub-task in parallel.

5. **Aggregate Results:** The outputs (e.g., location lc, features f, action a) are collected and aggregated (e.g., by a designated device $\hat{e}$ or through a shared mechanism) to form the intermediate digital twin state $vo_{int} = (lc, f, a)$ for the physical object po.

This approach leverages the combined computational power of the cluster, reduces the burden on individual devices, and enables parallel processing, leading to lower overall latency for DT attribute generation. The number and specialization of devices in a cluster can be adapted based on application requirements and available hardware.

3.4.3 Optimized Task Scheduling (PolyHeuristic from PolyVerse)

Even with clustering, managing dynamic workloads (e.g., varying numbers of objects in a camera's view) across heterogeneous edge devices requires intelligent task scheduling. We adapt the PolyHeuristic algorithm, originally proposed in PolyVerse, for this purpose.

Edge Network and Application Model

We model the edge network within or across clusters as a graph $G = (V, E)$, where V is the set of M edge devices and E represents network links. Each device $i \in V$ has computational capacity PS_i and maximum available resources R^i_{max}. Let K be the number of concurrent tasks (e.g., processing video streams or tracking objects).

Assume an application requires sequential dependent tasks, e.g., Task 1 (Detection) with workload T^k_d and resource requirement R^{det}_{req}, followed by Task 2 (Attribute Recognition) with workload T^k_r, resource requirement R^{rec}_{req}, and inter-task data dependency $D^k_{d,r}$. Let T_{ik} be the data transmission latency from source k to device i, and R_{ij} be the data rate between devices i and j.

Problem Formulation

We aim to minimize the total completion time L_k for all tasks k, subject to resource constraints. Let $x_{ik} = 1$ if task 1 for source k runs on device i, and $y_{jk} = 1$ if task 2 for source k runs on device j.

The total resource request on device i is:

$$R^i_{req} = \sum_{k=1}^{K} (x_{ik} \cdot R^{det}_{req} + y_{ik} \cdot R^{rec}_{req}) \leq R^i_{max} \tag{3.1}$$

The processing time L_k for task k involves data transmission, task execution, and inter-device communication if tasks are split ($i \neq j$):

$$\begin{aligned} L_k = \sum_{i=1}^{M} x_{ik} \cdot T_{ik} + \sum_{i=1}^{M} x_{ik} \cdot \frac{T^k_d}{PS_i} + \sum_{j=1}^{M} y_{jk} \cdot \frac{T^k_r}{PS_j} \\ + \sum_{i=1}^{M} \sum_{j=1}^{M} x_{ik} \cdot y_{jk} \cdot \frac{D^k_{d,r}}{R_{i,j}} \cdot \sigma(i-j) \end{aligned} \tag{3.2}$$

where $\sigma(x) = 0$ if $x = 0$ and $\sigma(x) = 1$ if $x \neq 0$. (Note: PolyVerse formula had $\sigma(i-j)$ logic reversed, corrected here assuming $\sigma = 0$ for local transfer).

The objective is to minimize the average or total latency:

$$\min_{x_{ik}, y_{ik}} \sum_{k=1}^{K} L_k \tag{3.3}$$

Subject to:

$$R^i_{req} \leq R^i_{max}, \quad \forall i \in V \tag{3.4}$$

$$\sum_{i=1}^{M} x_{ik} = 1, \quad \forall k \in \{1, ..., K\} \tag{3.5}$$

$$\sum_{j=1}^{M} y_{jk} = 1, \quad \forall k \in \{1, ..., K\} \tag{3.6}$$

$$x_{ik}, y_{jk} \in \{0, 1\}, \quad \forall i, j \in V, k \in \{1, ..., K\} \tag{3.7}$$

PolyHeuristic Algorithm

This optimization problem is NP-hard. PolyHeuristic provides a greedy, two-stage approach (Algorithm 1) based on:

- *High workload first:* Prioritize scheduling tasks with larger computational demands to avoid them being delayed by resource scarcity later.
- *Model Reuse / Task Placement:* Consider placing dependent tasks (like attribute recognition, which might not need frequent updates) strategically, potentially reusing instances or placing them on devices that minimize data transfer latency from the preceding task.

Algorithm 1 outlines the process. Stage 1 places the initial task (e.g., detection) considering data transmission and execution time. Stage 2 places the dependent task (e.g., attribute recognition) considering inter-device communication cost (if placed on a different device) and execution time, again subject to resource availability. The algorithm can be extended to consider reusing already running model instances (e.g., for attribute recognition) if resources are scarce, as mentioned in the original PolyVerse description.

By combining collaborative clustering and optimized scheduling, our edge AI approach aims to generate DT attributes efficiently and with low latency, addressing the real-time and scalability requirements.

Algorithm 1 PolyHeuristic Task Scheduling (Adapted from PolyVerse)

```
Input: Tasks (streams) $K$, Device capacities $\{PS_i\}$, resources $\{R^i_{avail}\}$, Task workloads $\{T^k_d, T^k_r\}$, Resource needs $\{R^{det}_{req}, R^{rec}_{req}\}$, Data dependencies $\{D^k_{d,r}\}$, Network info $\{T_{ik}, R_{ij}\}$
Output: Task allocation policy $x_{ik}$ (Task 1) and $y_{jk}$ (Task 2)
Create task priority list $I$ (e.g., descending order of $T^k_d + T^k_r$)
Initialize $x_{ik} = 0, y_{jk} = 0$ for all $i, j, k$. $R^i_{avail} = R^i_{max}$.
Stage 1: Schedule Task 1 (e.g., Detection)
for each task $k$ in priority list $I$ do
    Find candidate devices $C_k = \{i \mid R^i_{avail} \geq R^{det}_{req}\}$
    if $C_k \neq \emptyset$ then
        Calculate execution cost for each $i \in C_k$: $cost_i = T_{ik} + \frac{T^k_d}{PS_i}$
        Select device $i^* = \arg\min_{i \in C_k}\{cost_i\}$
        Set $x_{i^*,k} = 1$
        Update $R^{i^*}_{avail} = R^{i^*}_{avail} - R^{det}_{req}$
    else
                                                  ▷ Handle task rejection or queuing
    end if
end for
Stage 2: Schedule Task 2 (e.g., Attribute Recognition)
for each task $k$ in priority list $I$ do
    Let $i_k$ be the device where Task 1 for $k$ is placed ($x_{i_k,k} = 1$).
    Find candidate devices $C'_k = \{j \mid R^j_{avail} \geq R^{rec}_{req}\}$
    if $C'_k \neq \emptyset$ then
        Calculate execution cost for each $j \in C'_k$: $cost_j = \frac{D^k_{d,r}}{R_{i_k,j}} \cdot \sigma(i_k - j) + \frac{T^k_r}{PS_j}$
        Select device $j^* = \arg\min_{j \in C'_k}\{cost_j\}$
        Set $y_{j^*,k} = 1$
        Update $R^{j^*}_{avail} = R^{j^*}_{avail} - R^{rec}_{req}$
    else
                                                  ▷ Handle task rejection or queuing
    end if
end for
return $x_{ik}$, $y_{jk}$
```

3.5 Blockchain-based State Management and Validation

While edge AI provides efficiency, ensuring consistency and trustworthiness requires mechanisms that can operate in potentially adversarial environments. We employ blockchain technology at two levels: ensuring consistency among MSPs (based on PolyVerse) and validating the trustworthiness of DTs generated at the edge (based on PolyTwin).

3.5.1 Threat Model

We assume a standard Byzantine Fault Tolerant (BFT) adversarial model:

- **MSPs:** A fraction (e.g., up to $f < 1/3$) of MSPs may be malicious. They can deviate arbitrarily from protocols, attempting to corrupt state, cause inconsistencies, censor updates, or deny service. Honest MSPs follow the protocol.

- **Edge Devices:** Similarly, a fraction of edge devices within a cluster (e.g., up to $f' < 1/3$) could be compromised or malicious. They might try to generate incorrect DT attributes or provide false validations.

- **Network:** The network may experience delays or temporary partitions, but messages between honest parties are eventually delivered.

- **Cryptography:** Standard cryptographic primitives (hash functions, digital signatures) are assumed to be secure (unbreakable by PPT adversaries).

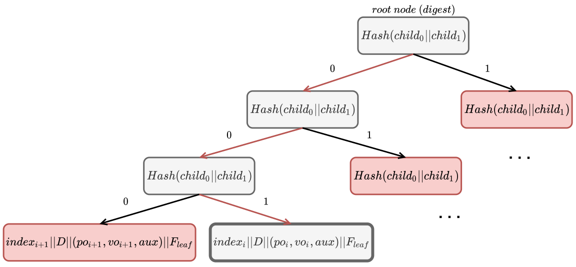


Figure 3.6: Metaverse State Tree (MST): State tuples (po_i, vo_i, aux) are stored in leaf nodes, indexed by a combination of mapping rule (MR) and state ID (sid). Interior nodes aggregate hashes, culminating in a root digest. Red nodes indicate a membership proof path for state $index_i = 001$.

3.5.2 MSP State Consistency via Blockchain and MST (PolyVerse)

To ensure all users see a consistent metaverse state, regardless of which honest MSP they connect to, PolyVerse proposed a blockchain-based state management system.

Data Model and Metaverse State Tree (MST)

Metaverse state, including physical states (PO), corresponding virtual states (VO), mapping rules (MR), and auxiliary data (aux like timestamps), needs to be managed consistently. Storing full state directly on a blockchain is often infeasible due to performance limitations.

PolyVerse introduced the **Metaverse State Tree (MST)**, a Merkle Tree-like authenticated data structure (ADS), to efficiently manage and verify state (Figure 3.6).

- **Leaf Nodes:** Store the actual state tuple (po_i, vo_i, aux) at a fixed depth D. Each leaf has a unique $index$, typically formed by concatenating the mapping rule identifier MR (representing a scene or context) and the object's state ID sid. The hash includes the index, depth, data, and a leaf flag F_{leaf}. $N_{leaf} =$

$Hash(index_i||D||(po_i, vo_i, aux)||F_{leaf})$

- **Empty Nodes:** Represent unused index ranges, containing the MR prefix and an empty flag F_{empty}. $N_{empty} = Hash(MR||D||F_{empty})$

- **Interior Nodes:** Computed recursively by hashing the concatenation of their two children's hashes. $N_{interior} = Hash(child_0||child_1)$

- **Root Node:** The top node, whose hash (the MST digest) represents an authenticated summary of all states included in the tree.

MST enables: 1) Efficient state accumulation off-chain. 2) Generation of compact membership proofs (Merkle paths) for state verification. 3) Fast state retrieval using the index structure.

State Update and Verification Operations

MSPs use the MST and a BFT blockchain (e.g., a consortium chain running PBFT or PoA) to manage state:

State Update (by MSPs):

1. `getPO(ECC)` → `(sid, PO, aux)`: MSP receives physical state updates from the edge.

2. `calculateVO(PO, MR)` → `VO`: MSP computes the corresponding virtual state based on mapping rules.

3. `generateADS(PO, VO, aux)` → `digest`: MSP inserts the state tuple (PO, VO, aux) into its local MST based on the calculated index, updating the tree and obtaining a new root digest. This is done off-chain.

4. `submitTX(state_batch, digest)` → `receipt`: Periodically, or upon trigger, the MSP submits a batch of recent state updates and the corresponding MST digest to the blockchain network for consensus.

State Verification (by MUs):

1. `getVO(MSP)` → `VO`: MU requests virtual object state from an MSP.
2. `reqVerify(VO)` → `(state_tuple, digest, proof, receipt)`: MU requests proof for the received state. The MSP provides the full state tuple, the MST digest relevant to that state, the Merkle proof (authentication path in MST), and optionally the blockchain receipt confirming the digest.
3. `verifyVO(digest, proof)` → `bool`: MU locally verifies: (a) The state tuple and proof reconstruct the provided digest. (b) The digest matches a digest confirmed on the blockchain (using the receipt). This confirms the state was included consistently by the MSP consensus.

Throughput Optimization (Lazy Trigger): To avoid blockchain bottlenecks, MST digests are committed lazily. MSPs can serve state based on their local MST immediately after `generateADS`. The `submitTX` is triggered only when a batch size/time limit is reached, or when a verification request necessitates confirming a specific state on-chain. This allows high throughput for state updates while still providing on-demand, verifiable consistency via the blockchain anchor.

This mechanism ensures that all honest MSPs eventually converge to the same state history, verifiable by users.

3.5.3 DT Trustworthiness Validation via PoC (PolyTwin)

While MST ensures MSPs agree on *some* state, it doesn't guarantee that state accurately reflects physical reality. PolyTwin's Proof-of-Consistency (PoC) addresses this by validating the DT attributes *before* they are finalized and potentially submitted to the MSP layer. PoC operates within the edge cluster.

Proof-of-Consistency (PoC) Mechanism

PoC leverages the redundancy inherent in the collaborative edge cluster (Section 3.4) for cross-verification.

PoC Protocol (within Edge Cluster EC):

1. **Initial Inference:** As described in Sec 3.4, devices e_{lc}, e_f, e_a infer attributes lc, f, a. A designated device $\hat{e}$ aggregates these into $vo_{int} = (lc, f, a)$.

2. **Transaction Proposal:** $\hat{e}$ creates a proposed DT transaction $tx_{dt} = \{vo_{int}, \sigma_{\hat{e}}\}$ containing the intermediate DT and its signature, and broadcasts it within EC.

3. **Cross-Verification:** Upon receiving tx_{dt}, other devices $e_j \in EC$ $(j \neq \hat{e})$ perform verification. They use their idle AI models (or re-run models) on the same input data (raw) to infer attributes they didn't initially compute. For example, e_{lc} might run the feature and action models.

4. **Voting:** If e_j's independently inferred attributes are consistent (match within a tolerance δ) with those in vo_{int} from tx_{dt}, e_j generates a vote (e.g., a signature on tx_{dt}). $Vote_j = Sign_{e_j}(tx_{dt}) \quad \text{if } \|\text{infer}(po, e_j, raw) - vo_{int}\| \leq \delta$

5. **Consensus:** Votes are collected (e.g., by $\hat{e}$ or broadcast). If tx_{dt} receives votes from a threshold number of devices (e.g., $> 2/3$ of $|EC|$), it is considered validated by PoC.

6. **Finalization:** The validated tx_{dt} (now possibly including collected votes/signatures) represents a trustworthy DT update. This validated update is then sent to the MSPs for incorporation into the global state (potentially via the MST mechanism).

Algorithm 2 summarizes the DT generation and PoC validation flow.

PoC ensures that a DT update is only accepted if multiple independent inferences within the edge cluster agree, significantly increasing confidence in its correctness and consistency with the physical object, even if some edge devices are faulty or malicious (up to the BFT threshold). This validated DT update can then be reliably used by the MSP layer, managed via the MST and blockchain for global consistency.

By combining MST for MSP-level consistency and PoC for edge-level DT validation, our framework provides end-to-end trust guarantees for incorporating digital twins into the metaverse.

3.5.4 Latency Requirements in Digital Twin Synchronization

The effectiveness of digital twin integration in the metaverse fundamentally depends on achieving stringent latency requirements that enable seamless synchronization between physical and virtual realms. In immersive metaverse applications, the human perceptual system demands near-instantaneous feedback to maintain the illusion of presence and prevent motion sickness or disorientation. Research in virtual reality and human-computer interaction has established that end-to-end latency must remain below 20 milliseconds for haptic feedback, under 50 milliseconds for visual updates to feel instantaneous, and within 100-150 milliseconds for general interactive responsiveness [91]. These constraints become even more critical when digital twins represent dynamic physical entities such as moving vehicles, human gestures, or rapidly changing environmental conditions, where temporal misalignment between physical state changes and their virtual representations can lead to safety hazards or degraded user experience.

The challenge of meeting these latency requirements is compounded by the distributed nature of the digital twin ecosystem. The synchronization pipeline involves multiple stages: sensor data acquisition at the edge (typically 1-5ms), AI model inference for

Algorithm 2 Digital Twin Generation and PoC Validation (PolyTwin)

Input: Physical object po, Edge Cluster $EC = \{e_1, ..., e_n\}$, Sensor data raw
Output: Trustworthy digital twin update $tx_{dt\_validated}$

```
procedure DTGen($po, EC, raw$)                                  ▷ Run by devices in EC
    // Parallel inference by specialized devices (e.g., $e_{lc}, e_f, e_a$)
    $po.lc \leftarrow inferLocation(po, e_{lc}, raw)$
    $po.f \leftarrow inferFeature(po, e_f, raw)$
    $po.a \leftarrow inferAction(po, e_a, raw)$
    $vo_{int} \leftarrow (po.lc, po.f, po.a)$
    Select proposer $\hat{e} \in EC$ (e.g., randomly or round-robin)
    $tx_{dt} \leftarrow \{vo_{int}, \sigma_{\hat{e}}\}$
    Broadcast $tx_{dt}$ within $EC$
    return $tx_{dt}$
end procedure
procedure PoCValidate($tx_{dt}, EC, raw$)                       ▷ Run by devices in EC
    $votes \leftarrow \{\}$
    for each device $e_j \in EC$ do
        if $e_j \neq \hat{e}$ then                                  ▷ Verifier node
            Verify $\sigma_{\hat{e}}$ on $tx_{dt}$
            Infer attributes using $e_j$'s other models: $vo'_{int}$ = inferOther($po, e_j, raw$)
            if $vo'_{int} \simeq tx_{dt}.vo_{int}$ then               ▷ Check consistency within tolerance
                $vote_j \leftarrow Sign_{e_j}(tx_{dt})$
                Add $vote_j$ to $votes$
                Broadcast $vote_j$ (optional, depends on consensus)
            end if
        end if
    end for
    if $|votes| \geq \lfloor \frac{2}{3}|EC| \rfloor$ then                ▷ Consensus threshold
        $tx_{dt\_validated} \leftarrow tx_{dt}$ with collected votes
        return $tx_{dt\_validated}$
    else
        return Failure or retry
    end if
end procedure
```

attribute extraction (10-50ms depending on model complexity), network transmission between edge clusters and MSPs (5-30ms for local networks, potentially higher for wide-area deployments), blockchain consensus for trustworthiness validation (ranging from hundreds of milliseconds to seconds), and finally rendering updates in the metaverse client (5-15ms). Our collaborative edge clustering approach specifically addresses the inference bottleneck by parallelizing AI workloads across multiple devices, while the lazy trigger mechanism in MST allows immediate state propagation to users before blockchain confirmation, effectively decoupling the critical path of user experience from the consensus latency.

Furthermore, different categories of digital twin applications exhibit varying latency sensitivities that must be considered in system design. Safety-critical applications like autonomous vehicle simulation or industrial process monitoring require ultra-low latency (sub-10ms) with minimal jitter to ensure reliable decision-making. Social and collaborative applications, such as virtual meetings or shared workspaces, can tolerate moderate latency (50-200ms) but demand consistency across all participants to maintain coherent interactions. Entertainment and gaming applications prioritize smooth, predictable updates over absolute minimal latency, often employing prediction and interpolation techniques to mask network delays. Our framework's flexible architecture, combining real-time edge processing with eventual consistency guarantees through blockchain, enables application-specific optimization of this latency-consistency trade-off, allowing developers to tune system parameters based on their specific use case requirements while maintaining the fundamental trustworthiness properties essential for valuable digital twin interactions.

3.6 Implementation and Evaluation

To demonstrate the feasibility and evaluate the performance of the proposed framework integrating concepts from PolyVerse and PolyTwin, we developed prototypes and conducted experimental evaluations.

3.6.1 Prototype Implementations

We implemented two main prototypes showcasing different aspects of the framework:

PolyCampus: This prototype, drawing inspiration from both original papers, realizes a metaverse campus environment where real-world campus elements, primarily pedestrians, are projected as digital twins in real-time.

- **Functionality:** Edge devices equipped with cameras capture video feeds near campus locations. AI models (running collaboratively on edge clusters) detect pedestrians, track their locations, and recognize attributes (gender, clothing, etc.). These attributes update corresponding avatars (DTs) in a virtual campus environment (e.g., hosted on a Minecraft server [2]).
- **Technology Used:** Edge devices (NVIDIA Jetson series), AI models (MobileNetV2 for detection, ResNet-50 for attributes, Kalman filter for tracking), blockchain network (private Ethereum with PoA/PBFT for state consistency/validation), virtual world platform (Minecraft).
- **Goal Demonstration:** Showcases real-time P2V projection, edge AI efficiency (PolyHeuristic scheduling), DT attribute generation, and potentially PoC validation for pedestrian attributes and MST for state management among simulated MSPs. Figure 3.7 illustrates the concept.

PolyExchange: This prototype focuses on demonstrating trustworthy DTs for physical interactions with economic implications, specifically trading.

- **Functionality:** Edge devices with cameras monitor a physical space where users might exchange goods. AI models (e.g., YOLOv8 for pose estimation) detect specific actions indicative of a trade (e.g., handshake, item transfer). Upon

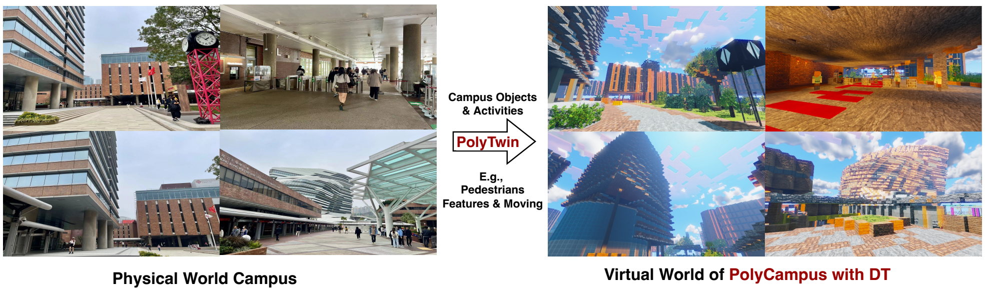

Campus Objects & Activities
PolyTwin
E.g., Pedestrians Features & Moving
Physical World Campus
Virtual World of PolyCampus with DT

Figure 3.7: PolyCampus Prototype: Real-world campus activities (pedestrian movements and attributes) are captured by edge AI and projected as digital twins into a virtual campus, enhancing immersion for metaverse users.

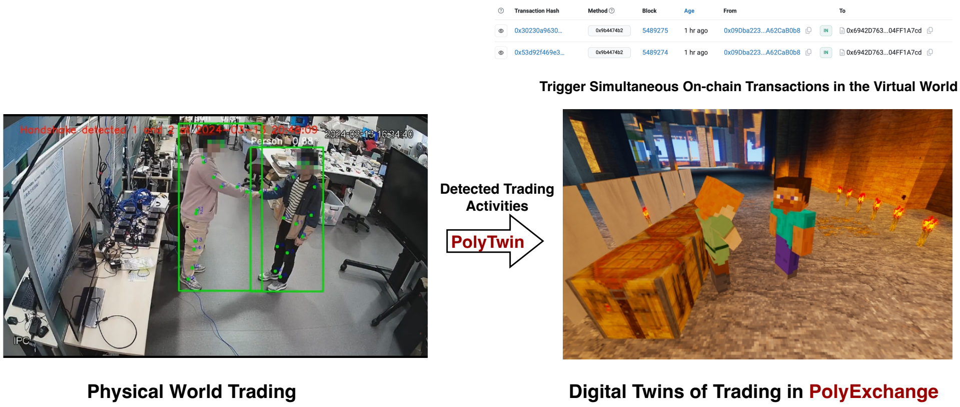
detecting a completed physical trade action, the system triggers a corresponding transaction (e.g., transfer of a virtual good or token) between the users' associated accounts on a blockchain, representing the DT of the trade.

- **Technology Used:** Edge devices (NVIDIA Jetson), AI models (YOLOv8 for pose estimation, MLP for action classification), blockchain network (for recording validated trades).

- **Goal Demonstration:** Showcases action recognition using edge AI, the concept of bonded operations (physical action triggering virtual transaction), and the use of PoC to validate the detected action before committing the virtual transaction, ensuring consistency between physical and virtual events. Figure 3.8 illustrates the concept.

3.6.2 Experimental Setup

- **Hardware:** Experiments utilized a mix of edge devices including NVIDIA Jetson TX2 (256-core Pascal GPU, 8GB RAM), Jetson Xavier NX (384-core Volta GPU, 8GB RAM), Jetson Orin NX (1024-core Ampere GPU, 16GB RAM), and Jetson AGX Orin (1792-core Ampere GPU, 32GB RAM). A workstation (Intel Core i9, 64GB RAM) hosted the metaverse server (Minecraft) and potentially



	Transaction Hash	Method	Block	Age	From		To
	0x30230a9630...	0x9b4474b2	5489275	1 hr ago	0x09Dba223...A62CaB0b8	IN	0x6942D763...04FF1A7cd
	0x53d92f469e3...	0x9b4474b2	5489274	1 hr ago	0x09Dba223...A62CaB0b8	IN	0x6942D763...04FF1A7cd

Trigger Simultaneous On-chain Transactions in the Virtual World
Handshake detected 1 and 2 at 2024-03-13 20:48:09
Person 0.84
Person 0.88
2024-03-13 16:34:40
Detected Trading Activities
PolyTwin

Physical World Trading

Digital Twins of Trading in PolyExchange

Figure 3.8: PolyExchange Prototype: Physical trading activities detected by edge AI are mapped to validated blockchain transactions in the metaverse, enabling trustworthy linkage between physical actions and virtual asset transfers.

blockchain nodes. Multiple cameras (e.g., 1920x1080 @ 25/30fps) provided input feeds.

- **Software:** AI models were implemented using standard frameworks (e.g., PyTorch, TensorFlow). The blockchain network used Geth (Go Ethereum) client in a private PoA configuration or a custom PBFT implementation for PoC. Communication between components used standard networking protocols (e.g., TCP/IP, UDP).

- **AI Models:**

  - Pedestrian Detection: MobileNetV2.

  - Pedestrian Tracking: Kalman Filter based tracker.

  - Attribute Recognition: ResNet-50 trained on Market-1501/DukeMTMC-reID datasets [56].

  - Pose Estimation / Action Recognition: YOLOv8 for pose, MLP classifier for actions (e.g., trading gestures).

3.6.3 Evaluation Results

Edge AI Performance and Scalability (PolyVerse & PolyTwin)

We evaluated the latency of the edge AI pipeline under varying conditions.

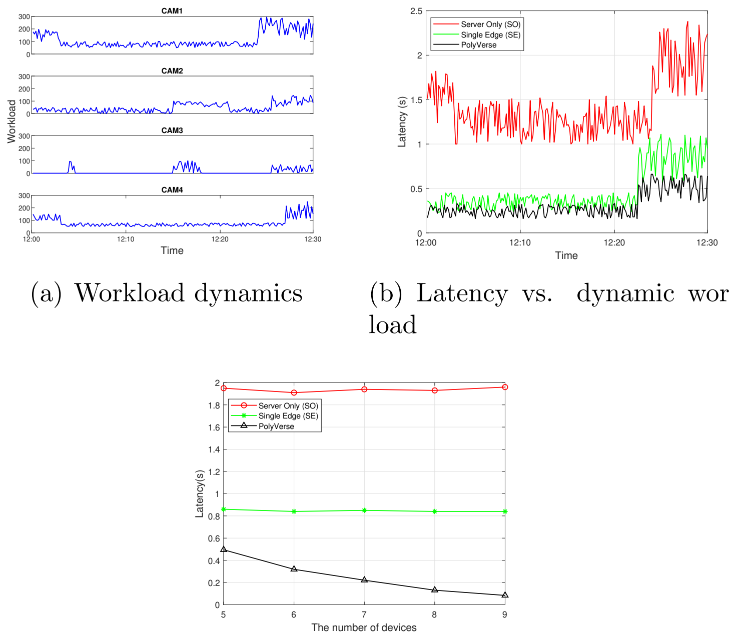
Latency vs. Workload: Figure 3.9(a) shows typical dynamic workloads (number of pedestrians) from different cameras in the PolyCampus setup. Figure 3.9(b) compares the end-to-end latency (from camera capture to attribute availability) for different processing strategies: Server Only (SO - all processing on central server), Single Edge (SE - each stream processed entirely on one edge device), and our collaborative approach (PolyVerse/PolyHeuristic).

- Edge processing (SE, PolyVerse) significantly outperforms SO due to reduced data transmission.

- PolyVerse (using PolyHeuristic scheduling and implicitly allowing collaboration) maintains lower latency than SE, especially under high workloads, by distributing tasks effectively.

Scalability with Devices: Figure 3.9(c) shows that as more edge devices are added to the collaborative pool, the overall latency decreases for the PolyVerse approach, demonstrating effective resource utilization and scalability.

Collaborative Inference Latency (PolyTwin): Figure 3.10 shows the latency distribution for attribute inference on edge devices (Jetson Orin NX/AGX Orin). The results indicate stable low-latency performance even with varying numbers of pedestrians per frame, supporting the real-time goal. Offloading parts of the model (like feature recognition) to other nodes in the cluster improved processing frame rates by ≈20% compared to running all tasks synchronously on one device.



(a) Workload dynamics

(b) Latency vs. dynamic workload

(c) Latency vs. number of devices

Figure 3.9: Edge AI Performance Evaluation (from PolyVerse): (a) Dynamic workload (pedestrians) over time. (b) End-to-end latency comparison under dynamic workload. (c) Latency reduction with increasing number of collaborative edge devices using PolyVerse scheduling.

Accuracy of DT Generation (PolyTwin)

- **Pedestrian Attributes (PolyCampus):** Tests on 800 pedestrian instances yielded a 96% detection rate. Among detected pedestrians, 85% were correctly re-identified (maintaining consistent ID) within the camera view. Attribute recognition accuracy depends heavily on the underlying model and training data (specific accuracy percentages for attributes like gender, clothing were likely detailed in the original PolyTwin paper but are summarized here).

- **Action Recognition (PolyExchange):** For recognizing trading hand gestures in static scenes without occlusions, tests achieved a 93% success rate in correctly identifying the action and triggering the corresponding virtual transaction.

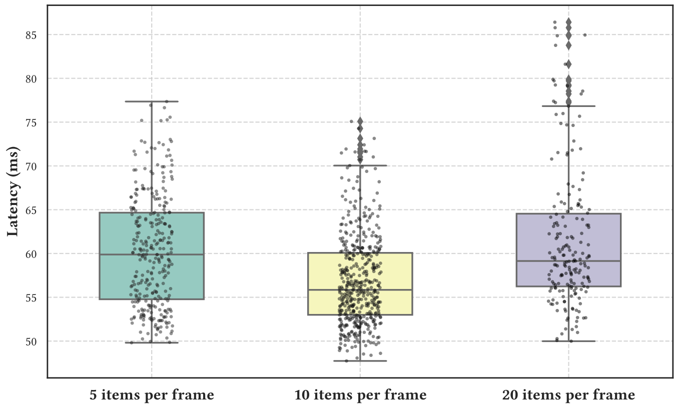


Figure 3.10: Edge AI Inference Latency Distribution (from PolyTwin): Shows relatively stable low latency for pedestrian attribute inference across different scenes and pedestrian densities on Jetson Orin devices.

These results suggest the edge AI models are capable of generating DT attributes and recognizing actions with reasonable accuracy for the prototype scenarios.

Blockchain Performance and Trustworthiness

MST Performance (PolyVerse): The MST operations (`generateADS`) are performed off-chain, primarily involving hash computations and tree updates, which are very fast (sub-millisecond). The throughput is mainly limited by how often digests are submitted to the blockchain (`submitTX`). The lazy trigger mechanism allows high throughput for state updates while providing eventual consistency anchored by the blockchain. Verification (`verifyVO`) involves reconstructing a Merkle path (logarithmic in the number of states) and potentially checking a blockchain receipt.

PoC Efficiency (PolyTwin): Figure 3.11 shows the latency distribution for different procedures in the PoC mechanism implemented with PBFT consensus among edge devices.

- *Voting:* The cross-verification and voting step within the edge cluster has relatively low latency (median around 0.57s in the tests). This involves running inference on idle models and cryptographic signing.

- *Consensus/Finalization:* Reaching consensus (PBFT) on the validated transaction $tx_{dt\_validated}$ among the edge nodes takes longer (median around 12 seconds in the tests). This latency depends on the number of nodes in the cluster and network conditions.

While the final PoC consensus adds latency compared to purely local processing, it provides a strong guarantee of DT trustworthiness before the state propagates further. The overall latency needs to be acceptable for the target application's real-time requirements.

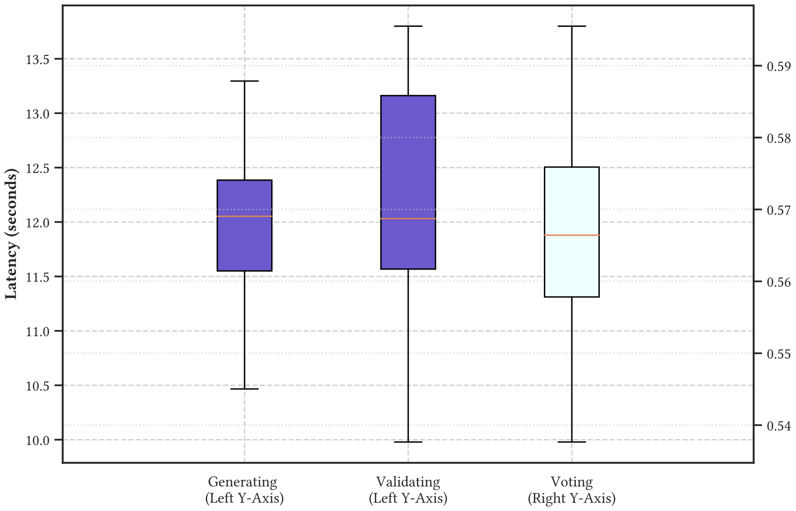


Figure 3.11: Latency Distribution of Proof-of-Consistency (PoC) Procedures (from PolyTwin): Shows latency for voting within the cluster and for achieving final PBFT consensus on the validated DT transaction.

Security Analysis: The combined framework provides trustworthiness satisfying the definitions in Section 3.3.2:

- **Correctness:** DTs are generated directly from physical sensor data using AI models deployed on-site at the edge. The PoC mechanism further enhances correctness by requiring consensus among multiple inferences before validation.

- **DT Consistency (PoC):** PoC ensures that the generated DT state is consistent with the consensus view of physical reality within the edge cluster, tolerating up to $f' < 1/3$ malicious edge devices. Maliciously generated inconsistent DTs are unlikely to pass PoC validation.

- **MSP State Consistency (MST + Blockchain):** The use of MST and BFT consensus among MSPs guarantees that all honest MSPs maintain a consistent view of the (PoC-validated) DT states, tolerating up to $f < 1/3$ malicious MSPs. Users can verify the consistency of the state they receive using MST proofs and blockchain receipts.

The evaluation results indicate that the proposed framework, integrating edge AI (with clustering and scheduling) and blockchain mechanisms (MST and PoC), offers a viable path towards achieving real-time, scalable, consistent, and trustworthy digital twins in the metaverse.

3.7 Conclusion and Future Discussion

This chapter addressed the critical challenge of integrating trustworthy Digital Twins (DTs) into the metaverse to enhance immersion and enable novel interactions bridging the physical and virtual worlds. We presented a comprehensive framework, synthesizing the contributions of the PolyVerse and PolyTwin systems, that tackles the key issues of real-time performance, scalability, state consistency, and DT trustworthiness.

Our core approach relies on a synergistic combination of edge computing and blockchain technology. We utilize **Edge AI clusters** deployed in the physical environment to perform low-latency data acquisition, processing, and AI-driven attribute extraction for DT generation. Techniques like **collaborative edge clustering** and the **Poly-Heuristic task scheduling algorithm** optimize resource utilization and minimize latency on resource-constrained edge devices, ensuring real-time performance even under dynamic workloads.

To establish trust and consistency, we employ blockchain mechanisms at two levels. First, the **Metaverse State Tree (MST)** combined with **BFT consensus among MSPs** ensures that all service providers maintain a globally consistent view of the metaverse state, preventing divergence caused by network issues or malicious actors. Users can efficiently verify the integrity of the state they receive. Second, the **Proof-of-Consistency (PoC) protocol**, executed within edge clusters, provides strong guarantees about the trustworthiness (correctness and consistency) of the DTs themselves by requiring cross-validation and consensus among multiple edge devices before a DT update is finalized.

We demonstrated the practicality and effectiveness of our framework through the development and evaluation of prototypes like **PolyCampus** (real-time pedestrian projection) and **PolyExchange** (trustworthy mapping of physical trades to virtual transactions). Experimental results confirmed the low-latency performance of edge AI processing, the benefits of optimized scheduling, the feasibility of PoC validation, and the overall system's ability to provide consistent and trustworthy DT representations under standard BFT assumptions.

Looking forward, the integration of trustworthy DTs holds immense potential for enriching the metaverse. A primary direction is reducing the latency of object validation. While the Proof-of-Consistency (PoC) protocol provides strong guarantees, our measurements show that the final Byzantine agreement step can add seconds of delay to the critical path for committing updates. Future work will explore optimistic validation modes that provisionally expose DT updates to users and MSPs, with retroactive correction anchored by PoC; fast-path or hardware-assisted consensus using TEEs/cryptographic acceleration; and hierarchical validation in which safety-critical attributes receive full PoC while non-critical fields use lightweight checks. Complementary techniques such as anomaly detectors to pre-filter evidently correct updates, batching and pipelining of validations, and sharded edge clusters that validate in parallel can further compress end-to-end latency. These designs must carefully balance responsiveness with security, providing tunable policies so applications can choose the appropriate trust–latency trade-off.

Another important avenue concerns coordination between digital twins. Many metaverse interactions are inherently multi-object (e.g., collision avoidance among pedestrians and vehicles, collaborative manipulation, or trades and contracts). Coordination introduces cross-DT dependencies and causal constraints that can amplify latency and generate consistency anomalies if handled naively. We envision spatial–temporal partitioning into coordination zones hosted at the edge, with local leaders responsible for ordering causally related updates; prediction/intent sharing to mask communication delays; and conflict-free replicated data types or reservation-based protocols to resolve concurrent actions. Extending MST with group-state commitments and adapting PoC to validate joint events (multi-DT PoC) would provide verifiable guarantees for coordinated behaviors while keeping coordination latency bounded.

Beyond these, scaling trustworthy DTs at city or campus scale will require operational methods for deploying and managing large fleets of heterogeneous edge clusters and sensors, as well as robust fusion of diverse modalities (e.g., LiDAR, audio, thermal) for richer and more reliable representations. Trustworthy DTs also enable advanced simulation and prediction in the metaverse, which calls for models that leverage validated states while respecting privacy. Finally, interoperability remains crucial: common schemas and verification formats for DT updates and proofs would allow different MSPs and platforms to exchange trustworthy information seamlessly.

In conclusion, by effectively combining edge intelligence with blockchain-based trust mechanisms, the framework presented in this chapter offers a robust foundation for realizing the vision of a deeply immersive and reliable metaverse seamlessly integrated with the physical world through trustworthy digital twins.

Chapter 4

Eusuring Identity Uniqueness

4.1 Overview

The Metaverse is a virtual world that enables physical-world users to interact through avatars, bridging the gap between virtuality and reality. In recent years, metaverse-related concepts have gained significant attention and development from both academics and industry giants such as Meta and Apple [70]. Numerous metaverse projects and products are now available on the market.

An important pillar for such a metaverse to successfully emerge after years of stagnation is the *blockchain and cryptocurrency integration* since 2018. Compared with traditional metaverse platforms [73] [51], blockchain-based metaverse enables decentralized ownership of digital assets and data, such as cryptocurrencies, non-fungible tokens (NFTs), digital artworks, and digital estates, which often carry considerable real-world value [28]. These features are extremely attractive to metaverse users; therefore, almost all large metaverse platforms actively support blockchain-based digital assets, including Decentraland, Sandbox, and Somnium Space [36] [3].

Background. Metaverse identity management is the key mechanism to secure and support decentralized ownership in a blockchain-based metaverse. Typically, it employs blockchain wallets for users to receive and hold assets. When users wish to transfer or trade their held assets, they sign transactions with their private keys and submit these transactions to the blockchain network within the metaverse.

However, existing wallets are vulnerable to *Sybil attack*. Sybil attackers often generate a multitude of Sybil wallets through bots and scripts to gain illegal benefits or disrupt the operation of the metaverse. For instance, Token Airdrops (TAs) are currently a popular promotional activity in the metaverse, typically involving the distribution of free tokens (or NFTs) to engaged users. Due to the potential benefits, Sybil attackers will attempt to create as many wallets as possible for the TA, eventually acquiring a large portion of the tokens. According to statistics[1], such Sybil attacks have caused millions of pounds in financial losses to metaverse projects.

Motivations and Challenges. The Proof-of-Personhood (PoP) protocol attempts to solve the Sybil problem by ensuring that a wallet is controlled by only one unique human. Unfortunately, very few works discuss or present detailed and practical solutions for PoP, either from academia or industry. Moreover, these solutions suffer from an anti-Sybil trilemma: they either cannot provide strong security against generative AI (such as general face-based authentication schemes), or they cannot effectively distinguish new honest users from Sybil attackers (such as through social-graph analysis), or they must rely on highly specialized and expensive devices [58] [79] [63] [89].

Our Solutions. In this paper, we propose the Eden protocol, an edge computing-empowered Proof-of-Personhood protocol for anti-Sybil measures in the metaverse. The Eden protocol aims to solve the anti-Sybil trilemma through a hybrid approach, combining economic video-based human recognition with on-chain transactional activity analysis. Eden confirms the unique binding between a user and a wallet address from the perspectives of both the physical and virtual worlds.

[1] https://medium.com/holonym/Sybil-resistant-airdrops-023710717413

Specifically, we develop *Eden devices*, customized edge computing boxes with cameras to recognize users' humanity through videos and on-device recognition models, and deploy them in the physical world. Users who pass the recognition can claim their unique wallet addresses. To further prevent generative AI-based Sybil attacks on Eden devices (e.g., malicious users may generate videos to bypass recognition and create multiple Sybil wallets), we analyze the recognized users' transactional activity and use a scorecard-based regression model to predict the probability that a wallet is Sybil. Lastly, to make Eden decentralized and resilient to attackers who may compromise Eden devices in the physical world, we integrate the aforementioned human recognition and transaction analysis procedures into a consensus, Proof-of-Trustworthiness (PoT), which is run by multiple Eden devices. PoT continually accumulates personhood claims, summarizing them as a personhood score, and eventually provides a quantitative measure of a user's personhood.

Our Contributions. The contributions of this paper are summarized as follows:

- **Eden Protocol**. Eden represents an innovative Proof-of-Personhood protocol that employs a hybrid approach, combining physical-world human recognition with on-chain transactional analysis to affirm the personhood of a given wallet address, thereby significantly enhancing anti-Sybil measures. The Eden protocol consists of three components:
  - **Eden Device**. The Eden device is a customized edge computing device equipped with a camera, designed for efficient identity recognition of physical-world users while preserving user privacy by retaining all raw data on the device. This device is considerably more cost-effective compared to specialized devices by utilizing general hardware.
  - **Transactional Activity Analysis**. Eden further analyzes the transactional patterns of a user's wallet once recognized by the Eden device and calculates a probability score to identify Sybil wallets. This analysis en-

sures that Eden is resilient to AI-based Sybil attacks.

- **Proof-of-Trustworthiness**. Eden integrates the aforementioned human recognition and transactional activity analysis procedures into a decentralized consensus, Proof-of-Trustworthiness (PoT). PoT assigns a reliable personhood score to a given wallet address, ensuring that Eden maintains consistency and robustness in determining user personhood, even when some Eden devices are compromised by attacks.

- **Prototype and Evaluation**. We have developed a hardware prototype for the Eden devices and deployed multiple devices in practice. We also integrate Eden into our metaverse campus platform, demonstrating that the PoP protocol can facilitate immersive interaction experiences in the metaverse. We also conduct intensive experiments on Eden, measuring the recognition efficiency of the Eden device and the overall latency and accuracy of PoT. The results demonstrate that Eden is viable in providing a practical PoP solution.

4.2 Background of Blockchain-based Metaverse

4.2.1 The Emergence

In recent years, particularly since 2018, numerous blockchain-based metaverse platforms have been developed. Unlike traditional metaverse that mainly focus on providing high-quality 3D models and interactions through augmented reality (AR), virtual reality (VR), and mixed reality (MR), blockchain-based metaverse platforms emphasize decentralized ownership of digital assets, such as non-fungible tokens (NFTs), tokens, and digital artworks [87]. Around these digital assets, many wonderful metaverse activities are organized, including token airdrops, NFT-based User Generated Content (UGC) creation, and virtual land and estate trading. Essentially, these activities offer immersive experiences similar to reality and provide rich opportunities for users with various purposes, thereby attracting them to join the metaverse. For instance, many celebrations involving famous artists, singers, musicians, and even professional traders and brokers actively participate in such metaverse activities, thus forming a prosperous ecosystem.

4.2.2 Identity Management and Sybil Attacks

To own digital assets on a blockchain, metaverse users must generate a private key and its corresponding public key. The public key is then encoded as wallet addresses, serving as both the asset account and the identity of a metaverse user. Users can sign transactions to transfer assets using their private key [82].

One major security concern regarding digital assets in the metaverse is the Sybil Attack on wallets. For example, as illustrated in Fig. 4.1, during token airdrop activities, organizers distribute a fixed amount of free tokens to each eligible wallet, thereby gaining community attention and promotional effects. However, due to the incentive motivation, malicious attackers may attempt to generate and control numerous Sybil wallets via bots to illegally maximize their receipt of the airdrops. As a result, such fraudulent behaviour significantly increases the costs of an airdrop and diminishes the effects of community attention and promotion [31].

Beyond financial attacks, Sybil wallets can also be utilized in other attacks to severely exploit other components' security within blockchains, such as P2P networking, consensus, data privacy, etc. [108] [44].

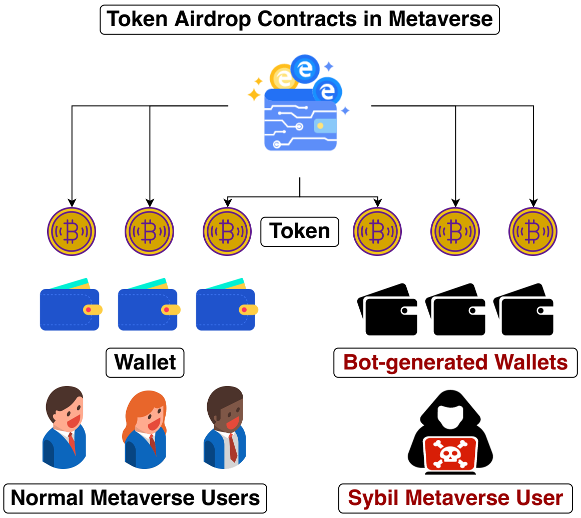


Figure 4.1: Sybil attack on token airdrop activities.

4.3 Problem Definition

4.3.1 Proof-of-Personhood

Under blockchain-based metaverse context, a PoP protocol Π_{pop} aims to make only one wallet address w_u controlled by each unique person u, thus forming a unique binding (w_u, u) in blockchain BC, where

$$(w_u, u) \leftarrow \Pi_{pop}(u, BC), \ \ ||w_u|| = 1$$

A PoP protocol Π_{pop} should simultaneously satisfy the following properties:

Definition 1 (Person-bound)**.** *A PoP protocol* Π_{pop} *is **person-bound** if and only if it takes only a human* u *as input, instead of other objects.*

Remark. *Other objects, such as robots and generative AI models (like DeepFake), may easily cheat a biometric-based PoP protocol, thus creating non-human identities on a blockchain.*

Definition 2 (Non-transferable)**.** *A PoP protocol* Π_{pop} *is **non-transferable** if any* *user u cannot transfer the wallet control right w_u to another user u', modifying the binding (w_u, u) to (w_u, u').*

Remark. *Using a hardware wallet or password-protected wallet as a PoP protocol is not non-transferable as the device and passwords can be sold or lent to other users, thus breaking the unique binding.*

Definition 3 (Sybil-resistant)**.** *A PoP protocol Π_{pop} is* ***Sybil-resistant*** *if it resists Sybil users. A Sybil user u_{Sybil} generates and keeps multiple key pairs $\{pk, sk\}^k$, deriving multiple blockchain wallet addresses $W_{Sybil} = \{w_0, w_1, w_2, \cdots, w_k\}$ in the metaverse.*

$$(W_{Sybil}, u_{Sybil}) \leftarrow \Pi_{Sybil}(u_{Sybil}, BC), \ \ ||W_{Sybil}|| > 1$$

Remark. *In practice, such Sybil attacks on blockchain can be implemented in multiple ways, such as malicious bots/scripts that automatically generate multiple wallets for one user.*

4.4 Eden Protocol Specification

4.4.1 Human Recognition on Eden Device

The Eden device is an edge computing box with cameras, on which we deploy a video-based on-device human recognition model to recognize and verify the users' humanity, as shown in Fig. 4.2.

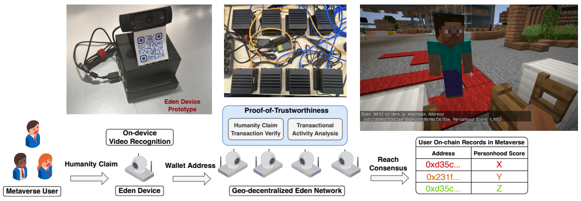

Eden Device Prototype
Proof-of-Trustworthiness
Humanity Claim Transaction Verify
Transactional Activity Analysis
[Eden INFO:] Hi! Here is *Hachimi*, Address: 0xE0218DA0d7D36E1aaF3Bd8426409B0f867367E6e, Personhood Score: 0.9053
On-device Video Recognition
Metaverse User
Humanity Claim
Eden Device
Wallet Address
Geo-decentralized Eden Network
Reach Consensus
User On-chain Records in Metaverse
Address
Personhood Score
0xd35c...
X
0x231f...
Y
0xd35c...
Z

Figure 4.2: Eden device architecture.

Specifically, a user u first visits a random Eden device e and scans a connection QR code on e to establish a temporary communication channel ch with e to exchange messages. Then the camera on e will detect and capture a video frame of u for model inference. The model inside of e then extracts the attribute encoding $attr_u$ of u locally based on the video frame. Finally, e stores $attr_u$ in the local database e, and sends an *initialization message* m_{ini}

$$m_{ini} = (hash(attr_u), ts, lc, \sigma_e) \tag{4.1}$$

where ts and lc are the timestamp and the geolocation of e when generating m_{ini}. σ_e is the signature of e in $(hash(attr_u), ts, lc)$.

After receiving m_{ini} and verifying σ_e, u generates a *humanity claim transaction* tx_c

$$tx_c = \{id_u = hash(attr_u), PK_u, ts, lc, \sigma_e, \sigma_u\} \tag{4.2}$$

where PK_u is the wallet address of u, and σ_u is the signature of u on $(id_u = hash(attr_u), PK_u, ts, lc)$. Then tx_c is sent back to e through channel ch, and tx_c together with $attr_u$ is broadcast to other Eden devices and wait for verification.

In summary, the above procedures establish an initial binding between a human and a wallet through the humanity claim transaction tx_{claim}.

4.4.2 Transactional Activity Analysis

Although the Eden device provides physical-world humanity verification, it is still potentially vulnerable to generative AI-based Sybil users. For example, a Sybil user may use DeepFake or other image-generating models to generate an arbitrary number of fake faces, and claim the binding of multiple Sybil addresses to these faces.

To solve this problem, our idea is to find and introduce other metrics that are hard to be simulated by generative AI, thus detecting generative AI-based Sybil users.

Transactional activities of a wallet have this feature, which are token transfers and smart contract calls on blockchains. Due to the real value of blockchain tokens, Sybil users can hardly let Sybil wallets simulate transactions of normal wallets. This fact makes the pattern of transactional activity a useful tool to detect Sybil wallets.

Sybil Patterns Normal users' transactional activities usually have a regular pattern in terms of balance, transaction quantity, transfer amount, and the relevant addresses. In contrast, Sybil addresses may have relatively different patterns. For example, many Sybil wallets only have low balances and records since they are only used for receiving airdrop tokens.

In the Eden Protocol, according to related works (such as money-laundering detection) and our observations, we pick four essential criteria to classify Sybil wallets and normal wallets:

- *Balance.* The balance status of a wallet address, whose value can be measured in a unified token like ETH or USDT.
- *Flow amount.* The total flow amount of a wallet address in a certain period.
- *Number of transactions.* The quantity of historical committed transactions of a wallet address.
- *Number of relative wallets.* The quantity of unique wallets that a wallet has interacted with, e.g., transfer tokens.

Transactional Activity Scoring. We conducted the transactional activity scoring using a simulated dataset (Table 4.1), in which behaviours such as short-time holders, inactive accounts, fixed asset transfer targets, and an account frequently transacting with a specific group of users were modelled. These behaviors are considered indicative of potential Sybil wallets. In the simulated dataset, we generated 0 to 50 different transactions for each simulated user. We utilize the above Sybil patterns to score the on-chain behaviors as shown in Equation 4.3 and use the probability output of the model as the user's transactional activity score.

$$S_{ta} = P(\text{Trustworthy Behaviour}|\mathbf{P_u}) = \frac{1}{1 + e^{-\mathbf{w}^T \mathbf{P_u}}} \tag{4.3}$$

where $\mathbf{w}$ represents the weight vector, and $\mathbf{P_u}$ includes the Sybil patterns features of user sample u.

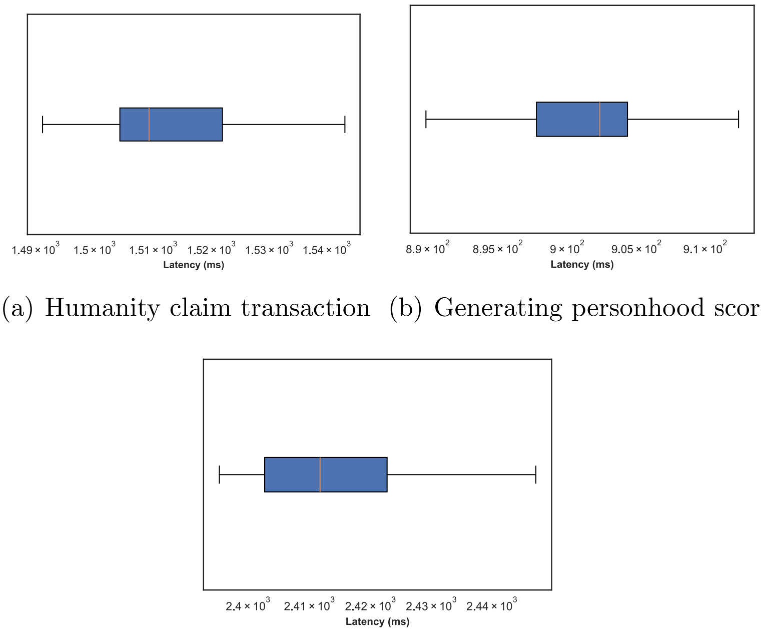
The weights $\mathbf{w}$ are trained using the logistic regression model on a data set with known labels (trustworthy or untrustworthy), optimizing the following likelihood function:

$$\mathcal{L}(\mathbf{w}) = \prod_{i=1}^{n} P(\text{label} = y_i|\mathbf{p}_i)^{y_i}(1 - P(\text{label} = y_i|\mathbf{p}_i))^{1-y_i} \tag{4.4}$$

where y_i are the known user labels (0 or 1), and $\mathbf{P_i}$ are the patterns features for the ith known user sample. Once the learning of the weights $\mathbf{w}$ is complete, they will be synchronized across all Eden devices. In this paper, we employ a relatively simple logistic regression model; however, in a more general workflow, the first step will obtain Sybil patterns features $\mathcal{F}_{\text{Sybil}} = \{f_1, f_2, \ldots, f_n\} = \mathcal{M}_{\text{FE}}(\text{on-chain behaviours})$ where $\mathcal{M}_{\text{FE}}$ is a feature extraction model. Subsequently, the scoring model is employed to evaluate the features of users $\text{Score} = \mathcal{M}_{\text{LR}}(\mathcal{F}_{\text{Sybil}})$. Depending on specific requirements, the feature extraction and scoring models could be replaced by any model, such as a GNN-based model to aggregate neighbor Address/User's information.

4.4.3 Proof-of-Trustworthiness

Unlike centralized Proof-of-Personhood (PoP) protocols such as WorldCoin, which rely on a single trusted entity to assess user personhood, the Eden protocol introduces a decentralized consensus mechanism, Proof-of-Trustworthiness (PoT) to evaluate the personhood of a wallet.



(a) Humanity claim transaction (b) Generating personhood score

(c) Overall PoT process

Figure 4.3: Eden Protocol Latency Evaluation: (a) Latency of humanity claim transaction. (b) Latency of generating personhood score. (c) Overall latency of the Proof-of-Trustworthiness (PoT) process.

The primary objective of PoT is to amalgamate the assessment outcomes from human recognition and transactional activity analysis, ultimately yielding a personhood score that indicates the likelihood of a wallet being non-Sybil.

This section describes the two stages of PoT. The first stage involves defining the rules for voting on and finalizing a humanity claim transaction. The second stage details the computation of the final personhood score for a finalized humanity claim transaction, tailored to various user scenarios.

Voting and Consensus. Upon receiving a claim transaction tx_c from the Eden network EN (consist of multiple Eden devices), each Eden device e employs a decision-making procedure to ascertain whether tx_c constitutes a legitimate humanity claim transaction. For each tx_c, each e checks if it satisfies the following conditions:

- *Correctness.* Both signatures σ_u and σ_e are correct.

- *Uniqueness.* The binding (id_u, PK_u) within tx_c must be unique across the Eden network, ensuring no other versions of bindings such as (id'_u, PK_u) or (id_u, PK'_u) exist within EN.

- *Location Reasonable.* The discrepancy in location lc and timestamp ts between consecutive claim transactions tx_c^i and tx_c^{i-1} is deemed unreasonable (e.g., two claims are committed in 1 hour from the same u, but the claimed location changes from the US to the EU, which is suspicious).

These three conditions are essential for confirming whether tx_c is a legitimate humanity claim transaction. If all of the above conditions are true, then each e attaches a vote v for tx_c. Additionally, each e also calculates the transaction activity scores S_{ta} for the PK_u.

Remark. *Historical transaction records needed when calculating S_{ta} can be sourced from external blockchain oracles such as blockchain explorers.*

Once the claim transaction tx_c garners more than two-thirds of the votes from the Eden network EN, it is considered finalized within EN.

Calculating Personhood Score. At this stage, each finalized tx_c has enough votes and S_{ta} (ranging from 0 to 1). Despite these two results both implicitly representing a wallet's degree of personhood, summarizing them into a unified personhood score will be a better choice for application's reference.

Therefore, in the second stage, we mainly perform mathematical smoothing and adjusting on these two results to output a unified personhood score. Our target of smoothing and adjusting is to ensure different types of users gain different levels of difficulty in achieving a personhood score. Specifically, the personhood score calculation rule should satisfy:

- Sybil users who attempt to bind multiple keys receive the lowest scores (such as zero), as their tx_c will not be finalized.

- Generative AI-based Sybil users only obtain very limited scores, as their S_{ta} are relatively low.

- Regular users who consistently claim humanity through an Eden device and maintain a high S_{ta} will be quickly assigned a high personhood score.

- Proactive users who frequently claim humanity through an Eden device and maintain a high S_{ta} will be assigned a high personhood score, albeit not as efficiently as regular users. This design serves as the last defense to prevent advanced Sybil users who have successfully simulated regular users, by increasing the difficulty of obtaining an extremely high personhood score.

The detailed calculation rule `PScoreCal()` for a wallet w_u's personhood score S^up is defined as follows:

We first adjust the scale and distribution of S_{ta} through an intermediary personhood score S^*p, which is composed of 1 basic score for finalizing the tx_c, and a sigmoid curved score of S_{ta}.

$$S^*p = \frac{1}{1+e^{-(S_{ta}-\frac{1}{2})}} + \frac{1}{2} \tag{4.5}$$

The range of S_p^* is $[\frac{1}{2}, \frac{3}{2}]$ when S_{ta} is [0,1]. $S_{ta} = \frac{1}{2}$ is the threshold for distinguishing non-Sybil and Sybil wallets.

For each round of PoT, we define a set that includes all voted devices $V = \{e_1, e_2, \ldots, e_v\}$. Then the personhood score of w_u acquired in round r is

$$S^up[r] = \frac{1}{|V|}\sum_{e\in V} S_p^* \tag{4.6}$$

Finally, we accumulate all the personhood scores from round 0 to round r $(S^up[0], S^up[1], ..., S_p^u[r]$

We leverage a modified Exponential Moving Average as follows:

$$Sp^u = \sqrt{1-\alpha}\cdot Sp^u[r-1] + \sqrt{\alpha}\cdot S_p^u[r]. \tag{4.7}$$

Specifically, $S_p^u[0] = 0, \alpha = 0.2$.

With a smoothing factor α, this adjustment smooths the impact of each new score, it assigns different weights to the historical score $Sp^u[r-1]$ and the claim score $Sp^u[r]$ for the current round r. The personhood score becomes less sensitive to disturbances in the $Sp^u[r]$ as the number of consensus rounds increases. In the initial rounds of user claims, the personhood score increases rapidly, which motivates users to interact with the Eden network (regular users); however, as the number of claims grows, the rate of score increase slows down, preventing users from obtaining excessively high scores through frequent interactions with the Eden network (proactive users).

For Sybil users, it is difficult for them to gain a high personhood score in PoT. Specifically, Sybil users who want to bind multiple keys to one user will not be accepted by PoT, thus gaining zero personhood score. For generative AI-based Sybil users, as they find it hard to maintain sufficient transaction activities, they only gain a limited personhood score from PoT. In summary, through PoT, the Eden network will ultimately form a list of stable personhood scores. Such scores can be easily ranked and provide an informative reference for applications.

Algorithm 3 Proof-of-Trustworthiness (PoT)

```
Input: $u$                                   ▷ User
Input: $e$                                   ▷ An Eden device interacted with $u$
Input: $PK_e$                                ▷ Eden device $e$'s public key
Input: $EN$                                  ▷ Eden network with $n$ nodes
Input: $tx_c$                                ▷ The humanity claim transaction
Output: $PK_u, S_p$                          ▷ Personhood score of $PK_u$
$round \leftarrow 0$
procedure PoT($tx_c, S_{ta}, EN$)
    for each $e \in EN$ do
        if signVerify($tx_c$) then
            if $tx_c.PK_u \notin EN.tx$ then
                $tx_c.vote \leftarrow tx_c.vote + 1$
            else
                if $tx_c.PK_u \in EN.tx$ and $lcVerify(tx_c)$ then
                    $tx_c.vote \leftarrow tx_c.vote + 1$
                end if
            end if
        end if
    end for
end procedure
procedure finalizePoT($tx_c$)
    if $tx_c.vote > \frac{2 \cdot \text{size}(EN)}{3}$ then
        $tx_c.S_p \leftarrow \text{calPScore}(tx_c.S_{ta})$
        $round \leftarrow round + 1$
        return $tx_c.PK_u, S_p$
    end if
    return $S_{ta}$
end procedure
```

4.5 Prototype and Evaluation

4.5.1 Prototype Development

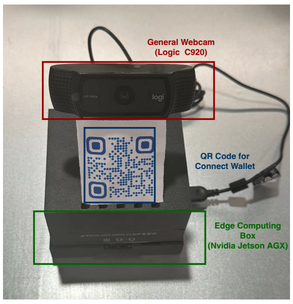


Figure 4.4: Hardware specification of the prototype of Eden device.

Eden Device Prototype. We implement a prototype version of the Eden device. As shown in Fig. 4.4, it consists of an edge computing box (Nvidia Jetson AGX, 1792-core NVIDIA Ampere architecture GPU with 32GB RAM), a general webcam (Logitech C920, 1080p), and a QR code as the interfaces to connect the user's wallet and the Eden device.

Face Recognition. For devices such as the Eden device, which essentially qualify as edge devices, we prioritize the selection of models that are simple, reliable, and lightweight. The Dlib[46] and face recognition[4] deep learning toolkit's face recognition module, which is implemented based on the Histogram of Oriented Gradients (HOG) and Convolutional Neural Network (CNN) methods, was utilized on Eden devices to recognize and generate face encoding. To enhance the model's resistance to fraud, we employed a straightforward liveness detection logic. Only users who pass this liveness test will proceed to facial feature encoding and further steps within the Eden network. The logic includes blinking, mouth opening, nodding, and shaking the head, where passing any two of these conditions allows progression to subsequent steps. In future work, we will also employ specific anti-spoofing deep learning models for identification.

Table 4.1: Sybil wallet dataset

Type	Description	Amount
Unused Address	Have no balance and no transactions on-chain	10
Human-like Address	Have a certain balance and be active on-chain	50
Bot-like Address	Send tokens to fixed addresses frequently	15
	Token holding time is short	25
	Send fixed tokens frequently like a script	10
Total		**110**

Proof-of-Trustworthiness. We deploy a layer-2 blockchain network (Polygon zkEVM) on the Eden network to implement the PoT consensus. L2 blockchain networks are usually more efficient than general blockchain networks since they use a sequencer to boost consensus and do not have heavy workloads. We implement the PoT in this L2 blockchain in the form of voting smart contracts, which can equally simulate the consensus and reduce development burdens. We set up eight Eden devices to join this network to verify and generate the final personhood score.

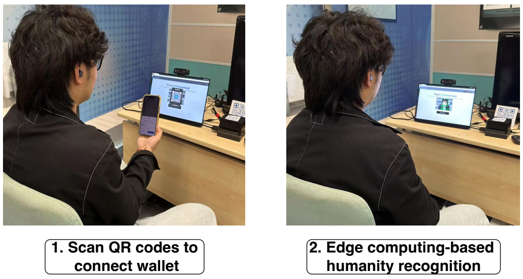

1. Scan QR codes to connect wallet
2. Edge computing-based humanity recognition

Figure 4.5: Interaction with Eden devices. This procedure ensures a wallet is actually controlled by a human. All the sensitive data are processed on-device without uploading to a centralized party.

4.5.2 Latency Measurement

The latency of the Eden protocol mainly affects the user experience when claiming their humanity, as they may need to stay with the Eden device in the physical world for a while and wait for the update of the personhood score, as shown in Fig. 4.5.

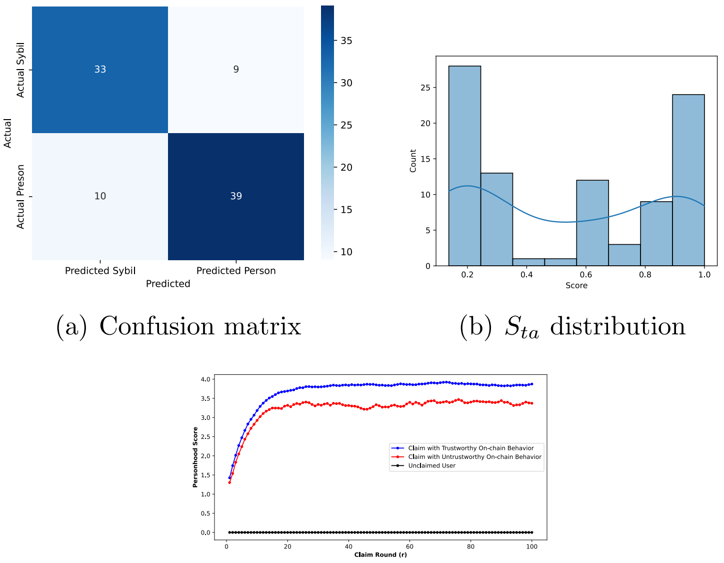
To measure the latency, we conduct 100 humanity claims and input these addresses into the Eden protocol to evaluate the latency. As shown in Fig. 4.3, results indicate that humanity claims usually take around 1.5 seconds to recognize the user and broadcast the humanity claim transaction to the Eden network. For the personhood score generation in PoT, the consensus confirmation time takes about 1 second with eight Eden devices. These results initially prove that using the Eden device is feasible and practical to be deployed in the physical world, providing users with efficient claim procedures.

4.5.3 Security Analysis

The security of the Eden protocol lies in its Sybil wallet detection and classification ability. To evaluate such ability in practice, we construct a *Sybil wallet dataset* based on the Sybil pattern in section 4.4.2. This dataset has 100 wallet addresses which are labelled as three types, as shown in Table 4.1. We jointly input these addresses with humanity claims on the Eden device to simulate different users like regular users and Sybil users.

As shown in Fig. 4.6(a) and Fig. 4.6(b), our regression model presents an initial Sybil detection ability based on purely transaction analysis. Most Sybil wallets are successfully assigned with a low transaction analysis score $S_{ta} < 0.5$.

We further record the accumulation of different users' personhood scores S_p. As shown in Fig. 4.6(c), black dots represent a Sybil user keeping trying to claim multiple wallets for one person. However, this Sybil user is unable to make the illegal humanity claim accepted by PoT, thus the personhood score remains zero (highly suspicious Sybil wallet). Red dots represent one of the generative AI-based Sybil user's wallets. The Sybil user successfully bypasses the humanity claim by generating multiple fake videos and thus has multiple wallets. However, this wallet can still be detected according to its untrustworthy transaction behaviours, it only gains a relatively low



(a) Confusion matrix

(b) S_{ta} distribution

(c) Personhood score accumulation

Figure 4.6: Security analysis of Eden Protocol: (a) Confusion matrix of the regression model for detecting Sybil wallets by transaction records. (b) S_{ta} distribution of our Sybil wallet dataset where lower values indicate higher likelihood of being a Sybil wallet. (c) Personhood score S_p accumulation of different wallet types after multiple rounds of PoT consensus.

personhood score. Blue dots represent a regular user's wallet, which successfully claims its identity on Edge devices and maintains good transaction records, thus gaining the highest personhood records.

4.6 Conclusion

This paper presents the Eden protocol, a novel Proof-of-Personhood protocol for anti-Sybil in the metaverse. The protocol employs edge computing and transaction analysis techniques to detect Sybil users effectively. We have developed initial version prototypes and conducted essential experiments to demonstrate the feasibility and practicability of the Eden protocol.

In the future, we envision that the Eden protocol will become a promising technical routine for solving the Proof-of-Personhood problem between the virtual and physical worlds, thus ensuring the metaverse ecosystem secure and prestigious.

Chapter 5

Enabling Asset Interoperability

5.1 Background

Blockchain is a decentralized ledger technology that uses cryptographic techniques and consensus mechanisms to achieve Byzantine Fault Tolerance (BFT), enabling decentralized trust and secure data sharing. Leveraging the philosophy of blockchain, the next generation of the web, known as Web 3.0, is being built. In recent years, a wide range of Web 3.0 applications are emerging, including cryptocurrencies, which revolutionize digital money, Decentralized Finance (DeFi) protocols that disrupt traditional financial systems, immersive virtual environments in the Metaverse, and various Decentralized Applications (DApps) [47, 52].

The Problem. With the rapid development of Web 3.0, on-chain data and assets are increasingly being distributed across multiple blockchains. According to statistics, there are already over 1,000 public blockchains in the market, hosting more than 10,000 types of on-chain assets [93]. This extensive distribution creates a critical need for blockchain interoperability protocols, which enable the retrieval and transfer of on-chain data and assets between source and destination chains through cross-chain transactions [75, 90]. With interoperability, conventional DApps could leverage data and assets from multiple chains simultaneously, thereby supporting a wider range of applications. For example, cross-chain DeFi services can increase liquidity and offer diversified financial services by depositing and exchanging assets from different chains, such as cryptocurrencies, Non-Fungible Tokens (NFTs), and Real-world Assets (RWAs)[94]. Likewise, an interoperable Metaverse platform could enable users to access various virtual worlds, thus enriching their experiences [54].

However, there are three major technical challenges when making chains interoperable: *trust requirement*, *expensive verification*, and *chain heterogeneity*.

Trust Requirement. When processing cross-chain transactions, the interoperability protocol must maintain the same level of BFT security as typical public blockchains to avoid compromising overall security. This implies that the protocol should be decentralized and trustless. However, achieving this level of security is challenging, as the protocol must handle complex tasks such as cross-chain transaction retrieval, processing, and verification, while maintaining consistency and liveness. As a result, many solutions are centralized or semi-centralized, such as notary schemes and committee-based protocols [66, 80]. These are widely used by cryptocurrency exchanges but are vulnerable to internal corruption and attacks due to their reliance on trust. For example, one of the largest multi-party computation (MPC)-based cross-chain bridges, Multichain, was severely exploited due to allegedly compromised secret keys, leading to a financial loss of over 120 million USD [85, 84, 107].

Expensive Verification. As different blockchains do not trust each other, they must verify incoming cross-chain transactions to ensure they are valid and confirmed on the source chains. However, this verification process is expensive and inefficient, particularly when it is performed on-chain, as it involves numerous complex cryptographic operations and the storage of block headers. For example, verifying an Ethereum Virtual Machine (EVM)-compatible transaction through an on-chain Light Client (LC) consumes approximately 18 million gas, which is equivalent to about 60 USD on Ethereum at the time of writing [50]. Such high cost is mainly due to the storage of public keys and the signature verification process. Although cutting-edge solutions aim to reduce on-chain cost by zk-SNARKs, they still require significant off-chain computational resources for proof generation [50, 99, 95].

Chain Heterogeneity. Connecting heterogeneous chains via interoperability protocols presents additional challenges. Heterogeneous chains differ in their underlying components, such as smart contract engines, supported cryptographic primitives, parameters, and transaction formats. As a result, they cannot directly verify and confirm transactions from one another. For instance, an EVM chain like Ethereum cannot directly verify transactions from Solana because the EVM lacks support for the multi-signature scheme used in Solana transactions. Therefore, existing solutions either only support specific chain types [96, 48], or require significant modifications on the underlying components of chains to achieve compatibility [59], which are both not feasible for in-production public chains. LC-based bridges may suffer less from compatibility issues, but still need to redundantlh deploy LC contracts on each chain [50, 104, 99], as shown in Figure 5.1. This approach incurs quadratic complexity $O(N^2)$ when extending to additional chains, thus posing huge gas consumption and development burdens.

Our Approach. In this paper, we introduce `MAP`, a scalable and trustless blockchain interoperability protocol. At a high level, `MAP` aims to minimize the computational cost when scaling to new chains while maintaining decentralized security, without any underlying modifications on chains. Specifically, `MAP` employs a novel relay chain architecture as the intermediary to relay cross-chain transactions from source chains to destination chains. This architecture eliminates the need of deploying pairwise chain-to-chain light clients. Moreover, to reduce both the on-chain and off-chain cost when verifying transactions, we propose an optimized zk-based light client scheme, *hybrid light client*, which adaptively decouples the signature verification workloads [16] according to their diverse performance in on-chain smart contracts and off-chain circuits.

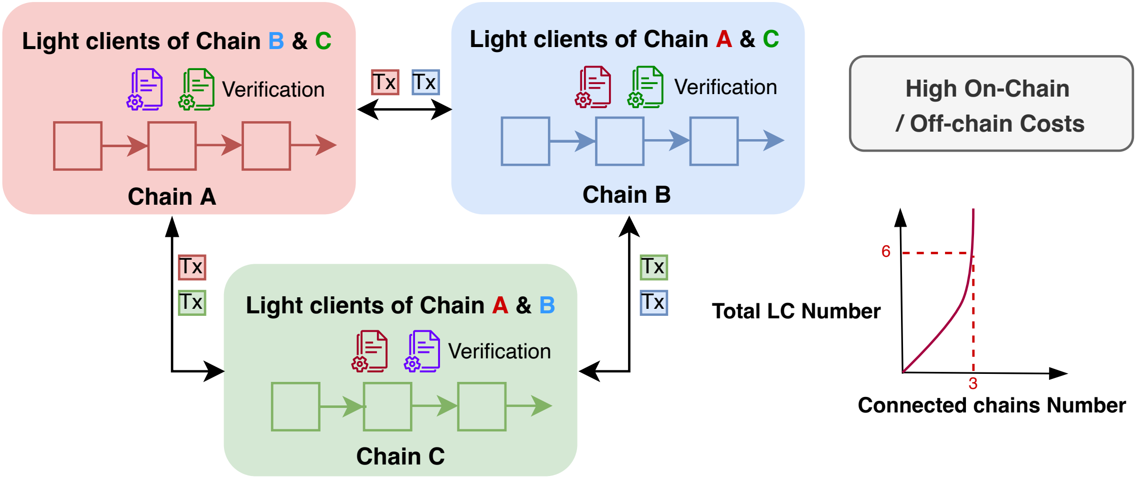

Light clients of Chain B & C
Verification
Chain A
Tx
Tx
Light clients of Chain A & C
Verification
Chain B
High On-Chain / Off-chain Costs
Tx
Tx
Light clients of Chain A & B
Verification
Chain C
Tx
Tx
6
Total LC Number
3
Connected chains Number

Figure 5.1: The redundant number of on-chain light clients in LC-based solutions. To connect three chains A, B, and C, LC-based protocols must deploy the chain-to-chain LCs of chains B and C on chain A to allow chain A to verify transactions from those chains (and same for chains B and C), resulting in total $3*2 = 6$ ($O(N^2)$) LCs needed and posing heavy on-chain or off-chain workloads

Contributions. In summary, `MAP` makes the following technical contributions:

- `MAP` introduces a unified relay chain to facilitate cross-chain transactions between heterogeneous chains, achieving decentralized security while reducing the required number of on-chain LCs from $O(N^2)$ to $O(N)$. Furthermore, the relay chain renders `MAP` chain-agnostic. When extending to new chains, only corresponding on-chain light clients are required to deploy.

- We develop a hybrid light client scheme based on zk-SNARKs that reduces both the on-chain and off-chain cost of verifying cross-chain transactions. We adaptively decouple the verification workloads of BLS signatures and proof generation based their performance in on-chain smart contracts and off-chain circuits. This scheme achieves a reduction in on-chain cost by 35% and off-chain cost by 25% compared to the existing state-of-the-art works.

- We evaluate the performance and security of `MAP`. Specifically, for performance, we are the first to perform large-scale measurements on existing interoperability protocols. For security, besides the cross-chain liveness and consistency proof, we identify and discuss a new security issue named *inter-chain security degradation* between interoperable chains.

- We deployed `MAP` on six public chains and support over 50 cross-chain applications, relaying over 200K real-world cross-chain transactions, worth over 640 million USD. Base on such practical experiences, we construct the first cross-chain dataset, *BlockMAP*, containing over 150k cross-chain transactions across six chains. We also open-sourced all the codes of `MAP`, accompanied by detailed documentations.

5.2 System Model and Goals

Interoperability Model. In `MAP`, we consider the most general interoperability model that exists in most cross-chain applications. Within this model, there are typically two types of chains to communicate with each other: the source chain $\mathbb{SC}$ and the destination chain $\mathbb{DC}$. $\mathbb{SC}$ is the initiating entity, which first receives and confirms cross-chain transactions ctx from users and DApps. Then, a *blockchain interoperability protocol* is deployed between $\mathbb{SC}$ and $\mathbb{DC}$, responsible for transmitting ctx from $\mathbb{SC}$ to $\mathbb{DC}$. Different from interoperability in traditional databases or networking protocols, blockchain interoperability especially focuses on ensuring the verifiability and trustworthiness of ctx because of the trustless nature of blockchains.

Transaction Model. Interoperability between blockchains is implemented in the form of cross-chain transactions ctx in `MAP`. A ctx is a blockchain transaction from $\mathbb{SC}$ to $\mathbb{DC}$ containing the message or asset to be transferred. Formally it is defined as $ctx = \{\mathbb{DC}, payload\}$. The $\mathbb{DC}$ field is the chain id of $\mathbb{DC}$, which identifies the destination of ctx. $payload$ field represents the types of ctx. When a ctx is an asset transaction, its *payload* contains the specific asset type, the amount, and the asset operation instructions; when a *ctx* is a message transaction, its *payload* contains the smart contract calls. In MAP, different types of *ctx* are handled in the identical way.

Design Goals. MAP has the following design goals:

1. **Trustless**. Maintaining the same level of BFT security as typical public blockchains.

2. **Scalability**. Gas-efficient and computationally efficient when processing cross-chain transactions and scaling to new chains.

3. **Chain-agnostic**. When extending to new chains, no underlying modifications needed except deploying new smart contracts.

5.3 MAP Protocol

5.3.1 Overview

As shown in Figure 5.2, there are two pipelined phases of cross-chain relay in MAP:

(Phase 1. $\mathbb{SC}$ - $\mathbb{RC}$ relay). First, cross-chain transactions *ctx* are firstly committed by users or DApps and confirmed on the source chain $\mathbb{SC}$ (❶). Then, an off-chain server *prover* will proactively monitor this confirmation event and retrieve the *ctx* with its associated proofs issued by $\mathbb{SC}$, such as headers and Merkle proofs (❷). Then the *ctx* and its proofs are sent to the unified relay chain $\mathbb{RC}$ by *prover* for generating proofs (❸).

The *unified relay chain* $\mathbb{RC}$ is an intermediary blockchain that processes cross-chain transactions between source and destination chains in a unified manner. More specifically, $\mathbb{RC}$ integrates multiple *hybrid on-chain LCs* of each $\mathbb{SC}$ (our zk-SNARKs-based optimized version of LCs, details in §5.3.4), which receive *ctx*s from *prover* and verify

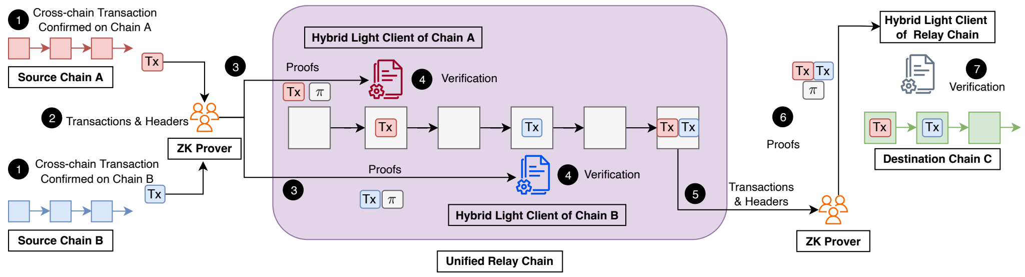


Figure 5.2: Overview of `MAP`: We introduce a unified relay chain architecture to facilitate cross-chain communications, which continually retrieves and verifies cross-chain transactions from source blockchains. Transactions are verified by hybrid light clients, which are implemented by smart contracts and off-chain provers.

whether they are legal and already confirmed on $\mathbb{SC}$ (❹). After the verification, the ctxs are temporarily confirmed and appended to $\mathbb{RC}$.

(Phase 2. $\mathbb{RC}$ - $\mathbb{DC}$ relay). Similar with phase 1, there is another off-chain server $prover$ retrieving ctxs from $\mathbb{RC}$ (❺). $prover$ generates the proofs of ctxs for verification on the destination chain $\mathbb{DC}$ (❻). On each $\mathbb{DC}$, an *identical* hybrid on-chain LC of $\mathbb{RC}$ is deployed, which verifies whether ctxs confirmed on $\mathbb{RC}$.

Recalling the overall procedures, the ctxs initially committed to $\mathbb{SC}$ are eventually confirmed on $\mathbb{DC}$, thus finalizing the entire cross-chain transaction relay (❼).

Note that there could be multiple $\mathbb{SC}$ and $\mathbb{DC}$ pairs in `MAP`, and the relay process is executed in the same way for each pair. Besides, $\mathbb{SC}$ and $\mathbb{DC}$ are relative, and they could be reversed during relay processes.

5.3.2 Unified Relay Chain

Insights. To address the challenges of trust and heterogeneity, we present two key insights for designing the architecture of blockchain interoperability protocols: (1) Only a BFT system can maintain the same security level with connected blockchains, thus avoiding degradation of overall security. Therefore, the overall architecture must be BFT-secure, such as a blockchain. (2) For decentralized protocols such as (ZK)LC-based bridges, the number of LCs on each chain is actually *overlapping* and *redundant*. That is, each chain only considers how to verify other chains from its own perspective (i.e., deploying other chains' LCs linearly), ignoring that the same type of LC can be deployed multiple times from a global perspective. For example, as shown in Figure 5.1, each type of LC is actually deployed twice. Therefore, if a new entity is able to verify transactions regardless of whether they come from different heterogeneous chains, the heterogeneity challenge could be resolved.

Architecture. To consolidate the above insights, we introduce the relay chain $\mathbb{RC}$ as the cross-chain intermediary in `MAP`. First, $\mathbb{RC}$ is also a blockchain that primarily responsible for receiving transactions from the source chain, verifying them, and forwarding verified transactions to the destination chain. This relay chain fundamentally ensures that `MAP` maintains decentralized security and trustworthiness.

Moreover, to address the challenge of chain heterogeneity, we adopt a *unified processing* strategy that enables $\mathbb{RC}$ to efficiently verify *ctx* from different heterogeneous chains, thus minimizing the number of LCs on $\mathbb{SC}$ and $\mathbb{DC}$. Specifically, `MAP` uses the on-chain LCs for cross-chain transaction verification. However, unlike existing LC-based bridges that require each of the LCs to be deployed on every other chain, we instead integrate the LCs of different chains into a single $\mathbb{RC}$. Consequently, all on-chain LCs $\Pi_{hlc}^{sc} = \langle \Pi_{hlc}^{sc_1}, \Pi_{hlc}^{sc_2}, \ldots, \Pi_{hlc}^{sc_i} \rangle$ are built on $\mathbb{RC}$ (the internal process of Π_{hlc}^{sc} will be introduced in §5.3.4).

Cross-Chain Relay. The general process of relaying *ctx* from source chain $\mathbb{SC}_i$ to $\mathbb{DC}$ works as follows. As shown in the Algorithm 4, there are two pipelined phases.

First, for the $\mathbb{SC}_i$–$\mathbb{RC}$ phase, after *ctx* is committed and confirmed on $\mathbb{SC}_i$, it will emit a confirmation event by outputting the block header bh^{sc_i} with the Merkle tree root $r_{mkl}^{sc_i}$ (line 2). Then a *prover* between $\mathbb{SC}_i$ and $\mathbb{RC}$ will monitor this confirmation event and proactively retrieve the *ctx* and generate the proofs $\langle ctx, bh^{sc_i}, \pi_{mkl}^{sc_i}, \pi_{zk}^{sc_i} \rangle$ (line 4-

Algorithm 4 Unified Relay Chain in `MAP`

Input: A cross-chain transaction ctx from $\mathbb{SC}_i$ to $\mathbb{DC}$

Output: Updated $\mathbb{DC}$ by ctx

```
procedure SourceChain($ctx$)
    $(bh^{sc_i}, r^{sc_i}_{mkl}) \leftarrow \text{confirm}(ctx, \mathbb{SC}_i)$   ▷ $ctx$ is firstly committed and confirmed on $\mathbb{SC}_i$
    for $prover$ between $\mathbb{SC}_i$ and $\mathbb{RC}$ do
        retrieves $(bh^{sc_i}, r^{sc_i}_{mkl})$ emitted by $ctx$ from $\mathbb{SC}_i$
        $\pi^{sc_i}_{mkl}, \pi^{sc_i}_{zk} \leftarrow \text{genProof}(bh^{sc_i}, r^{sc_i}_{mkl}, ctx)$
        return transmit$(ctx, bh^{sc_i}, \pi^{sc_i}_{mkl}, \pi^{sc_i}_{zk}, \mathbb{RC})$
    end for
end procedure
procedure RelayChain($ctx, bh^{sc_i}, \pi^{sc_i}_{mkl}, \pi^{sc_i}_{zk}$)
    if $\Pi^{sc_i}_{hlc}(ctx, bh^{sc_i}, \pi^{sc_i}_{mkl}, \pi^{sc_i}_{zk}) = \text{True}$ then
        $(bh^{rc}, r^{rc}_{mkl}) \leftarrow \text{confirm}(ctx, \mathbb{RC})$   ▷ $ctx$ is verified and confirmed on $\mathbb{RC}$
        for $prover$ between $\mathbb{RC}$ and $\mathbb{DC}$ do
            retrieves $(bh^{rc}, r^{rc}_{mkl})$ emitted by $ctx$ from $\mathbb{RC}$
            $\pi^{rc}_{mkl}, \pi^{rc}_{zk} \leftarrow \text{genProof}(bh^{rc}, r^{rc}_{mkl}, ctx)$
            return transmit$(\widehat{ctx}, bh^{rc}, \pi^{rc}_{mkl}, \pi^{rc}_{zk}, \mathbb{DC})$
        end for
    end if
end procedure
procedure DestinationChain($\widehat{ctx}, bh^{rc}, \pi^{rc}_{mkl}, \pi^{rc}_{zk}$)
    if $\Pi^{rc}_{hlc}(\widehat{ctx}, bh^{rc}, \pi^{rc}_{mkl}, \pi^{rc}_{zk}) = \text{True}$ then
        $(bh^{dc}, r^{dc}_{mkl}) \leftarrow \text{confirm}(\widehat{ctx}, \mathbb{DC})$   ▷ $ctx$ is verified on $\mathbb{DC}$
        return $\mathbb{DC}$
    end if
end procedure
```

5) from $\mathbb{SC}_i$ and transmit them to $\mathbb{RC}$ (line 6). Then $\mathbb{RC}$ verifies these transactions against the corresponding $\Pi_{hlc}^{sc_i}$ of $\mathbb{SC}_i$ built on $\mathbb{RC}$. After verification, the ctx are confirmed on $\mathbb{RC}$ as *intermediary cross-chain transactions* $\widehat{ctx}$.

Then, in the second $\mathbb{RC} - \mathbb{DC}$ phase, $\widehat{ctx}$ will also emit a confirmation event to $\mathbb{RC}$ by outputting the block header bh^{rc} with the Merkle tree root r_{mkl}^{rc} (line 9). Then a *prover* between $\mathbb{RC}$ and $\mathbb{DC}$ will get the $\widehat{ctx}$ and generate its proofs $\langle \widehat{ctx}, bh^{rc}, \pi_{mkl}^{rc}, \pi_{zk}^{rc} \rangle$ (line 11-12). These proofs are transmitted to $\mathbb{DC}$ for further verification (line 13). The key difference here is that only one identical type of Π_{hlc}^{rc} needs to be deployed on each $\mathbb{DC}$ (line 15) to verify $\widehat{ctx}$. This is because all $\widehat{ctx}$ are now from $\mathbb{RC}$, even though they were originally from different $\mathbb{SC}_i$. After passing the verification of Π_{hlc}^{rc}, the $\widehat{ctx}$ are confirmed on $\mathbb{DC}$ as the finalized cross-chain transactions ctx.

Consensus. To ensure the decentralized security of the relay process on $\mathbb{RC}$, we make $\mathbb{RC}$ run a BFT consensus (e.g., a Proo-of-Stake(PoS) BFT consensus like IBFT [64]). Generally, it enforces honest nodes with economic incentives, while punishing malicious behavior by slashing staked tokens. Validators will be motivated to participate and behave honestly because of token rewards from staking and processing cross-chain transactions.

5.3.3 Usage and Purpose of Zero-Knowledge Proofs

Zero-knowledge proofs (ZKPs) are central to achieving trustless and scalable interoperability in MAP. In cross-chain settings, destination chains cannot natively execute the verification logic of foreign chains due to heterogeneous cryptographic primitives and execution environments. Reimplementing full verification on-chain is prohibitively expensive because it requires elliptic-curve operations and persistent storage of validator sets. ZKPs provide a succinct attestation that a complex verification procedure was executed correctly off-chain, allowing any chain to check only a small proof on-chain. This maintains decentralization (no trusted relayers), bounds on-chain verification cost, and mitigates data exposure by avoiding the need to publish all low-level intermediate values. In short, ZKPs enable verifiable computation for cross-chain validation while preserving the security level of Byzantine fault tolerant blockchains [35, 99, 88].

Within `MAP`, ZKPs are used to prove the correctness of light-client logic without executing all details on-chain. Off-chain provers generate zk-SNARKs that attest to two kinds of statements: (i) validator-set transition and aggregate-signature validity when headers indicate an epoch change (the `Update` logic), and (ii) sufficient validator votes for a block that contains a given receipt root together with the membership of the target receipt (the core of `Verify`). On-chain, the relay chain and destination chains verify only the succinct proofs via verifier contracts, incurring near-constant gas regardless of the size of validator sets or signature batches. Combined with our hybrid design that keeps hash-heavy steps outside circuits and proves arithmetic-heavy steps inside, ZKPs substantially reduce both on-chain gas and off-chain proving cost while preserving soundness and completeness guarantees.

ZKPs also improve portability across heterogeneous chains: a single verifier for a chosen proving system (e.g., Groth16) can accept proofs of correctness for diverse source-chain rules, avoiding bespoke cryptographic support on every destination chain. This abstraction simplifies interoperability, enabling `MAP` to validate transactions originating from multiple ecosystems through uniform, succinct verification.

5.3.4 Hybrid Light Client

Although introducing the relay chain can effectively reduce the required number of on-chain LCs through unified processing, the heavy on-chain LC verification workload remains a bottleneck [104, 99].

On-chain Verification. To explore potential optimization spaces, we analyze the cost of each procedure in normal EVM-PoS light clients. After a transaction tx is

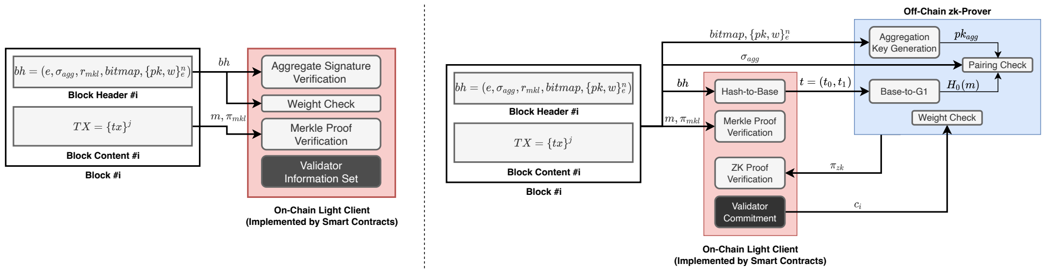

$bh = (e, \sigma_{agg}, r_{mkl}, bitmap, \{pk, w\}^n_e)$
Block Header #i
$TX = \{tx\}^j$
Block Content #i
Block #i
bh
m, π_{mkl}
Aggregate Signature Verification
Weight Check
Merkle Proof Verification
Validator Information Set
On-Chain Light Client (Implemented by Smart Contracts)
$bh = (e, \sigma_{agg}, r_{mkl}, bitmap, \{pk, w\}^n_e)$
Block Header #i
$TX = \{tx\}^j$
Block Content #i
Block #i
Off-Chain zk-Prover
$bitmap, \{pk, w\}^n_e$
σ_{agg}
bh
m, π_{mkl}
Aggregation Key Generation
pk_{agg}
Pairing Check
Hash-to-Base
$t = (t_0, t_1)$
Base-to-G1
$H_0(m)$
Weight Check
Merkle Proof Verification
ZK Proof Verification
π_{zk}
Validator Commitment
c_i
On-Chain Light Client (Implemented by Smart Contracts)

Figure 5.3: Our hybrid light client overperforms conventional light clients by adoptive offloading. We move the on-chain verification workloads to off-chain provers through zk-SNARKs. Meanwhile, we keep the hash operations on-chain to minimize the circuits size and proof generation time.

committed and finalized by consensus, a block B and its header bh will be produced and appended on chain [29]. To prove that such tx is included in B, the following major content needs to be inputted to normal light client Π_{lc}:

- a receipt message m emitted by tx inside B.
- a Merkle proof π_{mkl} for m extracted from B, which is usually provided by full nodes.
- a header $bh = (\{pk, w\}^n, \sigma_{agg}, bitmap, r_{mkl})$ that consists of:
  - an epoch number e.
  - a current validator information set $vs_e = \{pk, w\}^n_e$ that contains n validator public keys and corresponding voting weights corresponding to e. When consensus entering a new epoch, a new validator information set will be updated.
  - an aggregate signature σ_{agg} from validators signing B.
  - a mapping value $bitmap$ that indicates which validator actually signed B.
  - a root hash of receipt trie r_{mkl} that is computed from m.
- other auxiliary information such as timestamp and epoch size E

With above input content, the normal Π_{lc} is defined as three algorithms ($\texttt{Setup}, \texttt{Update}, \texttt{Verify}$), as shown in Figure 5.3 (left):

- $vs_g \leftarrow \texttt{Setup}(para)$: given system parameters, $para$, initialize Π_{lc} in terms of the epoch size, E, the vote threshold, T, and the initial validator information, $\{pk, w\}^n_g$. Then output a validator set, $vs_g = \{pk, w\}^n_g$, that indicates the current validator set stored in Π_{lc}.

- $vs_{e+1} \leftarrow \texttt{Update}(e, vs_e, bh)$: given a header bh with an epoch change, verify the aggregate signature σ_{agg} inside bh and update the current validator set vs_e to a new validator set $vs_{e+1} = \{pk, w\}^n_{e+1}$.

- $\{0, 1\} \leftarrow \texttt{Verify}(vs_e, m, bh, \pi_{mkl})$: given a message, m, emitted from tx and its header, bh, check whether tx is successfully included in B through its aggregate signature σ_{agg}, vote weights, and its Merkle proof π_{mkl}. Output $\{0, 1\}$ as the result. The incremental increase in the epoch number, e, is also verified during the signature verification.

Efficiency Optimization Space. Computation and storage are the main overheads when triggering `Update` and `Verify`. For computation, aggregate signature verification is frequently performed, which is essentially operations on elliptic curves, including hashing (i.e., the Hash-to-Curve algorithm), equation evaluations, and pairing checks[19, 13, 34]. These operations are inefficient in EVM due to their relatively high complexity when calculating underlying fields via curve equations. For instance, currently verifying one EVM cross-chain transaction with full BLS signatures can cost up to 1×10^6 gas[104] (approximately 30 USD on ETH). For storage, Π_{lc} needs to store $vs_e = \{pk, w\}^n_e$ persistently and frequently read them. Since most of PoS blockchains have more than 100 validators, storing and updating these data at the end of each epoch on smart contracts requires a large amount of storage space, thus consuming a expensive gas fees. Storing one validator information set requires 0.1×10^6 gas.

Hybrid Verification. To estimate the high gas fee consumption, we develop a hybrid verification scheme Π_{hlc} to reduce on-chain cost using off-chain zk-SNARKs.

First, we aim to efficiently prove the two functions `Update` and `Verify` using zk-SNARKs. We compress the validator information vs into a single commitment: $vs = \text{commitment}(\{(pk_0, w_0), (pk_1, w_1), \ldots, (pk_n, w_n)\})$ to reduce the on-chain storage overhead. In this way, the validator aggregate signatures of bh and the corresponding voting weights must satisfy this commitment value to pass verification. One native approach to implementing zk-SNARKs for proving is to program and compile all verification procedures into circuits, i.e., input the entire block header into the circuit along with all signature verification algorithms [99, 88]. Then deploy an off-chain prover to generate the zk-proofs based on this circuit and submits them to Π_{hlc} for verification.

However, we observe that despite Π_{hlc} improving efficiency by shifting on-chain workloads to off-chain provers, generating zk-proofs for verifying the entire aggregate signature instead requires substantial off-chain storage and computational resources for the prover. Specifically, in this way, the circuit size for an aggregate signature verification is extremely large due to multiple complex operations such as *Hash-to-Curve* and pairing checks (e.g., typically exceeding 2×10^7 gates and 100 GB for a block with eight signatures[30]). These factors also increase the proof generation time.

To optimize the off-chain cost of generating zk-proofs, we try to decouple the aggregate signature verification process and handle it separately. Specifically, the *Hash-to-Curve* algorithm in the BLS scheme hashes the message m to curve points in $\mathbb{G}$, which typically consists of two steps in practical implementations:

1. *Hash-to-Base.* Input a a message m and map it to possible coordinates (base field elements) through hash functions. This returns a field element t.

2. *Base-to-G.* Input a field element t and calculate the curve point (x, y) through the curve equations.

Since *Hash-to-Base* mainly consists of multiple hash operations, it can be efficiently computed through smart contract but inefficiently compiled into circuits due to its large size. In contrast, *Base-to-G* performs arithmetic operations in the finite field through elliptic curve equations, which can be relatively briefly and efficiently expressed into circuits. In this way, we improve the off-chain efficiency of zk-SNARKs based aggregate signature verification, further speed up the entire Π_{hlc}.

With the above optimization, the Π_{hlc} is defined as the following algorithms, as shown in Figure 5.3(right):

- $vs_g \leftarrow$`Setup`$(para)$: given the system parameters, $para$, initialize Π_{hlc} with the hard-coded epoch size, E, the vote threshold, T, and the initial validator information commitment, $vs_g = C(\{pk, w\}_g^n)$. Then, output a validator set, $vs_g = \{pk, w\}_g^n$, that indicates the current validator set stored in Π_{hlc}.

- $vs_{e+1} \leftarrow$`Update`(vs_e, h, π_{zk}): given header bh during an epoch change, verify the aggregate signature, σ_{agg}, of bh. First, compute the base field elements $t = (t_0, t_1)$ in G_1 by hash function $H_0(bh)$, and send t to the prover. After receiving π_{zk} that satisfied c, update the current validator set, vs_e, with the new validator set, $vs_{e+1} = C(\{pk, w\}_{e+1}^n)$.

- $\pi_{zk} \leftarrow$`GenZK`$(bitmap, vs_e, \sigma_{agg}, t,)$: given extracted $bitmap$, $vs_e = \{pk, w\}_e^n$, σ_{agg}, validator set commitment c from vs_e and t from `Update`, run a zk-SNARKs system and generate a zk-proof, π_{zk}, for c.

- $\{0, 1\} \leftarrow$`Verify`(vs_e, m, h, π_{mkl}): given message m emitted from tx and its header bh, verify whether tx is successfully included in B through its aggregate signature, σ_{agg}, and there are sufficient weights according to the stored vs_e and its Merkle proof π_{mkl}. Then output $\{0, 1\}$ as the result.

5.4 Performance Evaluation

Metrics	**Central.**	**Commit.**	**HTLC**	**BoB**	**LC**	**ZKLC**	**MAP**
Solutions	Binance[15] CoinBase[25]	Celer[27] Multichain[66]	EthHTLC Lighting	Polkadot Cosmos	LayerZero Horizon	zkBridge zkRelay	`MAP`
Type	Notary	MPC	HTLC	BoB	LC	ZKLC	**ZKLC+ Relay**
Security	Trusted	Semi-Trusted	Trustless	Trustless	Trustless	Trustless	**Trustless**
On-chain (gas)	N/A	0.5M	1.5M	0.08M	1M	0.3M	**0.65M (35%↓)**
Off-chain (gates)	N/A	N/A	N/A	N/A	N/A	20M	**15.7M (25%↓)**
Latency	1s	310s	N/A	13s	227s	153s	**210s**
Complexity	$O(N)$	$O(N^2)$	$O(N^2)$	$O(N)$	$O(N^2)$	$O(N^2)$	**O(N)**

Table 5.1: Performance comparison of blockchain interoperability protocols. Results are mainly from Polygon to Ethereum transaction workload (cost per transaction).

Experiment Setup. We set up two Google Compute Engine machine type c2d-highcpu-32 instance (32 vCPUs with 64GB RAM, ˜800 USD per month) as provers. For the relay chain, the hardware configuration for validator is similar with e2-standard-4 (4 vCPUs with 16 GB RAM).

Baselines and Workloads. Since very few works provide quantitative performance evaluation results, it is difficult to find an available and common baseline to ensure fairness [81, 75, 90]. To this end, we perform the first comprehensive measurement and comparison of existing blockchain interoperability protocols. As shown in Table 5.1, we measure five key security and scalability metrics across six representative types of protocols. We set the cross-chain transactions from *Polygon to Ethereum* as workloads for comparison, which is mostly supported by existing works. For protocols that do not support such workloads (such as Polkadot), we select their popular source-destination chain pair for evaluation. For each type of workload, we measure 100 transactions and record the average result or cost per cross-chain transaction.

Table 5.2: Circuit size of provers for verifying different number of validator signatures

Number of Sig. per ctx	Circuit Size
4	0.9×10^6 gates
8	15.7×10^6 gates
16	25.2×10^6 gates
32	49.3×10^6 gates

5.4.1 Evaluation Results

On-chain Costs. For on-chain cost, we mainly refer to the LC-based bridges as baselines, because they are the most common decentralized solutions [104]. For each cross-chain transaction verification, on-chain LCs require ~1×10^6, while MAP requires only ~0.65M, saving ~35%. These cost are deterministic in repeated tests on smart contracts [29, 97].

Off-chain Costs. For off-chain cost caused by zk-SNARKs, we refer to the standard implementation using snarkjs Groth16 to prove the signature verification scheme in transaction verification as the baseline [30, 99, 88]. As shown in Table 5.2, for eight signatures, the circuit size of the `MAP` prover is ~1.57×10^7 gates, which is reduced by ~25% compared to the aforementioned baselines (2×10^7 gates). Correspondingly, the proof generation time is also reduced by ~25% due to its linear relationship with circuit size. Moreover, `MAP` only needs a single sever to generate proofs rather than multiple server settings [99], which further reduces the off-chain cost in deploying and maintaining processes.

Number of On-chain Light Clients (Scaling Up Cost). According to statistics[1], a PoS-BFT EVM light client requires approximately ~100K gas per validator information storage. Assuming the number of validators is 100 for each chain, then for connecting N chains, LC-based bridges need to spend $10^7 \times N(N-1)$ gas to deploy LCs. In contrast, for `MAP`, it is ~100K gas fixed per LC for validator information set

[1] https://github.com/shresthagrawal/poc-superlight-client

commitment storage (no matter how many validators), which means only $2 \times 10^5 \times N$ gas is needed. Moreover, it avoids establishing communication channels with every other chain. Instead, each chain only needs to ensure the communications with the relay chain, which further makes MAP more practical.

Cross-chain Latency. We measure the end-to-end latency of cross-chain transactions relayed in MAP, from the confirmation timestamp on source chains until the confirmation timestamp on destination chains, including transaction transmission between chains, proof generation, and on-chain LC verification. As shown in Table 5.1, the results indicate that MAP's cross-chain latency is ˜210 seconds. Compared to existing works, these results suggest that despite introducing provers and relay chain will slightly increase latency. However, the overall impact is negligible, as the latency for cross-chain applications is not prioritized like conventional chains in practice.

5.5 Security Analysis

We thoroughly analyze the security of MAP. Particularly, as previous works have extensively proved the transaction liveness and consistency within a single PoS-BFT chain [74, 68], we focus on demonstrating the newly introduced components (i.e., provers and relay chain) in MAP will maintain the cross-chain liveness and consistency under various attacks.

5.5.1 Assumptions

MAP works under several basic and common security assumptions [81] [99].

Assumption 1. ***(PoS-BFT Threshold)***. *For $\mathbb{RC}$, more than $\tau = \frac{2S}{3}$ of the stakes are controlled by honest validators, where S is the total stakes. This group of honest validators is always live, i.e., they will confirm ctx in a timely manner.*

Assumption 2. ***(Secure Cryptographic Primitives)***. *The cryptographic primitives used in* `MAP`*, including the BLS signature, the Groth16 zk-SNARKs, and the hash functions, are secure against probabilistic polynomial-time (PPT) adversaries. That is, no PPT adversary can generate incorrect proofs or signatures that would be accepted.*

Assumption 3. ***(Minimal Prover and Reachable Communication)***. *At least one prover is available and honest in* `MAP`*, i.e., they will correctly generate the proofs* π_{mkl} *and* π_{zk} *and transmit cross-chain transactions ctx between chains, i.e.,* $\mathbb{SC}$, $\mathbb{RC}$, *and* $\mathbb{DC}$. *Additionally, we assume that the communication channels between the provers and the chains are reachable (i.e., no network partitions, though they may be insecure).*

Remark. *If Assumption 3 does not hold, chains will be isolated and not interoperable in any sense.*

5.5.2 Liveness and Consistency

Theorem 1. ***(Cross-chain Liveness)***. *If a valid ctx is committed to and confirmed on* $\mathbb{SC}$*, then it will eventually be confirmed on* $\mathbb{DC}$ *via* `MAP`*, assuming the above assumptions hold.*

Proof. Given a committed *ctx* from $\mathbb{SC}$, there are two potential cases that could prevent it from being confirmed on $\mathbb{DC}$: *Case 1*: A faulty or compromised *prover* refuses to generate proofs and transmit *ctx* between SC-RC or RC-DC. *Case 2*: Sufficient validators of RC are corrupted to force $\mathbb{RC}$ to withhold *ctx*, preventing it from being sent to $\mathbb{DC}$. For Case 1, by Assumption 3, at least one *prover* will transmit *ctx* to $\mathbb{RC}$ and $\mathbb{DC}$ (a single *prover* is sufficient for processing transactions from any number of chains). Therefore, even if other *provers* are faulty or compromised (e.g., via DDoS attacks), $\mathbb{RC}$ and $\mathbb{DC}$ can still receive and verify *ctx* from the reliable *prover*.

For Case 2, previous works have proven that any liveness attacks on PoS-BFT chains involving the refusal to verify transactions require at least $\frac{1S}{3}$ stakes [74, 68], which is prevented by Assumption 1. Even in the case of DDoS attacks on partial the $\mathbb{RC}$ validators, since the honest validators are live and control over $\frac{2S}{3}$, they will always confirm the *ctx* in time. As a result, Π_{hlc} run by the validators will eventually verify *ctx* and confirm it on both $\mathbb{RC}$ and $\mathbb{DC}$, thereby guaranteeing the overall cross-chain liveness. □

Theorem 2. ***(Cross-chain Consistency)****. If a valid ctx is committed and confirmed on* $\mathbb{SC}$ *and a* $\underline{ctx}$ *is finally confirmed on* $\mathbb{DC}$ *via* `MAP`*, then* $ctx = \underline{ctx}$*, assuming the above assumptions hold.*

Proof. Given a *ctx* from $\mathbb{SC}$, there are two potential cases for consistency attacks: *Case 1*: A malicious $\underline{prover}$ generates a tampered $\underline{ctx}$ with its proofs and tries to get them accepted by $\mathbb{RC}$. *Case 2*: Adversaries directly corrupt $\mathbb{RC}$ to force it to accept a tampered $\underline{ctx}$. For Case 1, in order to pass Π_{hlc} verification, the malicious $\underline{prover}$ would need to forge block headers (including the corresponding signatures and Merkle proofs) to generate incorrect proofs. However, by Assumption 2, this is highly unlikely to succeed. Therefore, Π_{hlc} will not accept $\underline{ctx}$ as a valid cross-chain transaction on $\mathbb{RC}$. For Case 2, corrupting $\mathbb{RC}$ to accept a tampered $\underline{ctx}$ requires controlling at least $\frac{2S}{3}$ of the validators, which is prevented by Assumption 1. Therefore, any tampered $\underline{ctx}$ will not be accepted on $\mathbb{RC}$, thus ensuring cross-chain consistency. □

5.5.3 Inter-Chain Security

Despite the analysis in §5.5.2 proving that cross-chain transaction verification is secure under Assumptions 1, 2, and 3, it does not fully match cross-chain scenarios. Specifically, within a single chain, the profit-from-corruption can hardly be higher than cost-to-corruption because they are calculated by the relative token value. That is, within a chain A with security threshold $\tau_A = \frac{2S_A}{3}$, it is unlikely to see a transaction with value over τ_A.

We identify a new security issue when connecting multiple chains with interoperability protocols that may converse the above situation, which also applies to other chain-based protocols but not well discussed before. We name this issue *Inter-Chain Security Degradation*. This issue indicates that the overall security of interoperable multi-chain networks is as strong as the least secure chain. For example, given three interoperable PoS-BFT chains A, B, and C, with their BFT security boundaries as $\tau_A = \frac{2S_A}{3}$, $\tau_B = \frac{2S_B}{3}$, and $\tau_C = \frac{2S_C}{3}$, the security of the entire network is $\min(\tau_A, \tau_B, \tau_C)$. This can be justified by considering the following situation: assume $\tau_B = \min(\tau_A, \tau_B, \tau_C)$. If a *ctx* from chain A to chain B has an extremely large value $V_{extreme} > \tau_B$, the validators of chain B will be sufficiently motivated to manipulate $ctx_{extreme}$ (such as double-spending), even if they were honest before (Assumption 1) and run the risk of being slashed by all the staked. This is because their profit-from-corruption is now explicitly higher than cost-to-corruption. In other words, the security of chains A, B, and C is *degraded* to $V_{extreme} < \tau_B$ due to the existence of interoperability.

Discussion Regarding `MAP`, this degradation requires the security of the relay chain to be strong enough (i.e., high staked value) to support cross-chain transactions. To examine the risk, we provide real-world statistics from a period of `MAP`. As shown in Figure 5.4, the most valuable cross-chain transaction was a 100K USDC transfer from NEAR in March 2023[2], worth 1.3% of the total `MAP` stakes (7M USD), far away from the security threshold. This also indicates `MAP` could still support transactions worth up to 4.67M USD.

In summary, although inter-chain security degradation is unavoidable due to the mismatched economic security level of different connected chains, `MAP` is still highly secure in practical scenarios.

[2] `https://maposcan.io/cross-chains/565`

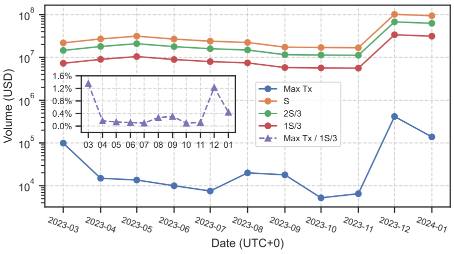


Figure 5.4: Historical statistics of `MAP`: The maximum value of any single cross-chain transaction is significantly smaller than the security boundary of the relay chain

5.6 Conclusion

This paper introduces `MAP`, a trustless and scalable blockchain interoperability protocol with practical implementations. `MAP` strikes a balance between security and scalability by introducing a unified relay chain architecture and optimized zk-based hybrid light clients (LCs). We conducted extensive experiments to comprehensively evaluate its performance and analyze its security. We envision `MAP` as a practical solution for interoperable data and networking infrastructure in the Web 3.0 era.

5.7 Preliminaries

PoS-BFT Consensus. Proof of Stake with Byzantine-Fault Tolerance (PoS-BFT) consensus has become a best practice for blockchain development due to its high energy-efficiency and security in recent years. It requires nodes (validators) to deposit funds as stakes to be qualified to participate in the consensus and to guarantee security. PoS-BFT consensus procedures typically operate and iterate in epochs. At the beginning and end of each epoch, validators are rotated and elected as committees by the PoS mechanism. During the epoch, there will be a fixed period of time for the committees to validate, agree and finalize proposed blocks according to BFT algorithms and PoS mechanism [62, 29].

Light Client. The light client is designed for resource-constrained devices such as mobile phones running blockchain nodes. It only syncs and stores block headers to reduce storage and computation overheads. Therefore, only partial functions of full nodes are available, such as transaction query and verification, while the costly consensus and mining procedures are usually excluded [24].

Aggregate Signature. Aggregate signature refers to the signature scheme that supports batching signatures to reduce overheads[16, 17]. In aggregate signature schemes, multiple signatures are aggregated as one signature, which can be further verified by an aggregated public key. Nowadays, BLS signature and its variants currently are widely used in PoS-BFT chains due to their high efficiency.

Zero-knowledge Proof. The Zero-Knowledge Proof (ZKP) is a type of cryptographic system that allows a prover to convince a verifier that a given statement is true or false without disclosing any other information. ZKP systems typically need to express and compile the statement proof procedures into circuits with constraints (gates) to generate proofs, which is computationally expensive [81, 65, 55].

5.8 Implementations details

For the relay chain, we develop a client software of the unified relay chain node[3]. To overcome the heterogeneity, we integrate the most commonly adopted cryptographic primitives and parameters in existing chains into the smart contract engine of the relay chain. Specifically, supported hashing algorithms include SHA-3, SHA-256, keccak256, and blake2b, while signature algorithms (or elliptic curves) include ed25519, secp256k1, sr25519, and BN256, which covers most public chains. We adopt IBFT in the relay chain, which is also well tested and widely adapted in many chains.

[3] https://github.com/mapprotocol/atlas

We also implement six hybrid LCs for six chains[4] in the form of multiple smart contract pairs. For off-chain provers, we use Groth16[35] to express the BLS signature verification (except *Hash-to-Base*) through Circom, alongside with our optimizations to reduce the size of the circuit[5]. First, we make BLS public keys in $\mathbb{G}_2$, while the signatures are in $\mathbb{G}_1$ to reduce the signature size. Second, as mentioned before, we move two *Hash-to-Base* functions out of the circuit to simplify the constraints in the circuit.

5.9 Supported Chains and Cross-chain Applications

`MAP` supports six major public chains: including EVM chains such as Ethereum, BNB chains, Polygon, and Conflux, and Non-EVM chains such as Klaytn and Near. By 2024, there are over 640M USD assets relayed by over 5M cross-chain transactions with `MAP`[6]. Over 50 industrial cross-chain applications and layer-2 projects are built[7]. Representative cross-chain applications range from cross-chain swap (ButterSwap), crypto payment (AlchemyPay), liquidity aggregation (Openliq), DePINs (ConsensusCore), DeFi solutions development (Unify) [8].

[4] `https://github.com/mapprotocol/atlas`, `https://github.com/mapprotocol/map-contracts/tree/main/mapclients/zkLightClient`, and `https://github.com/zkCloak/zkMapo`

[5] `https://github.com/zkCloak/zkMapo`

[6] https://www.maposcan.io

[7] A full list at `https://www.mapprotocol.io/en/ecosystem`

[8] `https://www.butterswap.io/swap`, `https://alchemypay.org`, `https://www.consensuscore.com`,`https://openliq.com`, `https://unifiprotocol.com`

5.10 Real-world Cross-chain Dataset

Based on the experiments and our deployment statistics, we prune and provide the first public, real-world blockchain interoperability dataset, `BlockMAP`[9], which consists of 150k cross-chain transactions from six popular public chains. The dataset includes several essential attributes, such as transaction direction, start and end timestamps, token types, and amounts. This dataset presents practical measurement of real-world cross-chain transactions, aiming to offer new insights and understandings for future blockchain research.

[9] https://zenodo.org/records/13928962

Chapter 6

Conclusion

6.1 Conclusion

In this thesis, we have proposed a comprehensive framework for trustworthy digital twins in the metaverse, addressing the challenges of projection, identity management, and asset sharing. We have developed an edge computing architecture that efficiently collects physical object attributes through deployed sensors and AI models. To ensure digital twin consistency, we introduced a Proof of Consistency (PoC) mechanism that cross-verifies digital twins against physical object inference results on the blockchain, along with an optimized data structure for storing digital twin states. Additionally, we designed Eden, an edge computing-empowered Proof-of-Personhood protocol to address Sybil attacks in the metaverse, and MAP, a scalable and trustless blockchain interoperability protocol for secure digital asset relay among heterogeneous chains. Our experimental results demonstrate the practical efficiency and security of our proposed solutions.

6.2 Future Research

We close this thesis by providing some suggestions for future research. We vision that the following two directions are worth further exploration for trustworthy digital twins in the metaverse:

- **Edge AI and Resource Sharing for Digital Twins** The generation of digital twins in DTNs requires integrating multi-dimensional attributes and states from diverse physical data sources. Traditional approaches, such as digital signal processing or specialized machine learning models, struggle with limited generalization capabilities in these tasks. Foundation Models (FMs), particularly Large Language Models (LLMs), demonstrate superior capabilities in handling complex relationships between physical objects and their attributes, making them promising candidates for digital twin data processing.

  However, deploying FMs on resource-constrained edge devices remains challenging due to significant mismatches in memory and computational requirements. Therefore, designing efficient resource sharing and scheduling mechanisms for edge computing is essential. Such mechanisms would enable computational workloads to be effectively distributed across multiple edge devices while maintaining optimal performance.

- **Content-aware Networking for Digital Twins** Networking performance is critical for DTNs due to frequent data uploading and synchronization requirements. Conventional networking protocols like TCP/IP are ill-suited for real-time DTN communication due to their slow connection establishment processes and lack of content awareness, resulting in inefficient handling of heterogeneous data.

  Content-aware networking, particularly semantic networking, offers a promising solution for efficient digital twin communication. These approaches facilitate efficient data transmission while capturing complex knowledge relationships, potentially reducing data volume and latency in digital twin operations. However, existing protocols require optimization for digital twin-specific requirements. First, digital twins demand more stringent real-time throughput and latency guarantees than traditional content delivery systems. Second, they require high resilience against network failures while maintaining data consistency. Finally, the heterogeneity of sensing data types and formats introduces additional challenges for content-aware routing and caching mechanisms.

- **Collaborative On-Chain and Off-Chain Computing** To ensure reliable consistency and tokenization, computational tasks associated with digital twin generation—including data collection, processing, and model inference—must be integrated with blockchain networks. However, existing blockchains primarily support basic mathematical operations and cryptographic primitives, making computationally intensive tasks impractical. For instance, executing deep neural network inference on-chain can consume over 2.88G gas (exceeding 200k USD), rendering it cost-prohibitive for practical digital twin applications.

  A collaborative on-chain and off-chain computing framework presents a promising direction for digital twin implementations. While existing works provide features such as model inference or data storage, executing end-to-end digital twin generation processes on-chain remains challenging due to task heterogeneity and computational intensity. Moreover, ensuring the trustworthiness of physical world data through blockchain oracles poses additional challenges.

- **Digital Twin Interoperability** Digital twin interoperability remains a significant challenge in current implementations. For example, while NVIDIA Omniverse adopts OpenUSD as its primary framework for extensible 3D content creation and collaboration, other platforms like Apple ARKit and Google ARCore lack OpenUSD support, creating barriers to cross-platform sharing and collaboration. This fragmentation has resulted in isolated digital twin ecosystems.

Designing interoperability protocols for digital twins is particularly challenging due to heterogeneous data formats and operation definitions across platforms. Additionally, performance requirements and trust issues in cross-platform data transmission introduce further complexity, as data format conversion between platforms can lead to information loss and increased latency.

References

[1] Decentraland. https://decentraland.org/. Accessed: March 10, 2025.

[2] Minecraft. https://www.minecraft.net. Accessed: March 10, 2025.

[3] The sandbox. https://www.sandbox.game/en/. Accessed: March 10, 2025.

[4] Modern face recognition with deep learning. https://github.com/ageitgey/face_recognition, Apr 2017. Accessed: Apr 13, 2017.

[5] Meta horizon workrooms — virtual workroom — work with meta. https://www.meta.com/work/workrooms/?utm_content=95648, Mar 2023. Accessed: March 10, 2025.

[6] For every human. https://worldcoin.org, Apr 2024. Accessed: Apr 10, 2024.

[7] Gitcoin passport. https://support.gitcoin.co/gitcoin-knowledge-base/gitcoin-passport/what-is-gitcoin-passport, Apr 2024. Accessed: Apr 10, 2024.

[8] Humanity protocol. https://cfh.xyz, Apr 2024. Accessed: Apr 10, 2024.

[9] The internet of humans. https://proofofhumanity.id, Apr 2024. Accessed: Apr 10, 2024.

[10] Proof of uniqueness. https://www.brightid.org, Apr 2024. Accessed: Apr 10, 2024.

[11] Francis Akowuah and Fanxin Kong. Physical invariant based attack detection for autonomous vehicles: Survey, vision, and challenges. In *2021 Fourth International conference on connected and autonomous driving (MetroCAD)*, pages 31–40. IEEE, 2021.

[12] Moayad Aloqaily, Ouns Bouachir, Fakhri Karray, Ismaeel Al Ridhawi, and Abdulmotaleb El Saddik. Integrating digital twin and advanced intelligent technologies to realize the metaverse. *IEEE Consumer Electronics Magazine*, 2022.

[13] Zachary Williamson Antonio Salazar Cardozo. Eip-1108: Reduce alt-bn128 precompile gas costs, 2018. Last accessed 14 Oct 2024.

[14] Andreas M Antonopoulos, Olaoluwa Osuntokun, and René Pickhardt. *Mastering the Lightning Network*. " O'Reilly Media, Inc.", 2021.

[15] Binance. Binance - cryptocurrency exchange for bitcoin, ethereum, 2023. Last accessed 14 Oct 2024.

[16] Dan Boneh, Craig Gentry, Ben Lynn, and Hovav Shacham. Aggregate and verifiably encrypted signatures from bilinear maps. In *Advances in Cryptology—EUROCRYPT 2003: International Conference on the Theory and Applications of Cryptographic Techniques, Warsaw, Poland, May 4–8, 2003 Proceedings 22*, pages 416–432. Springer, 2003.

[17] Dan Boneh, Craig Gentry, Ben Lynn, Hovav Shacham, et al. A survey of two signature aggregation techniques, 2003.

[18] Alain Brenzikofer. encointer–local community cryptocurrencies with universal basic income. *arXiv preprint arXiv:1912.12141*, 2019.

[19] Vitalik Buterin. Eip-198: Big integer modular exponentiation, 2017. Last accessed 14 Oct 2024.

[20] Yang Cai, Jaime Llorca, Antonia M Tulino, and Andreas F Molisch. Compute- and data-intensive networks: The key to the metaverse. In *2022 1st International Conference on 6G Networking (6GNet)*, pages 1–8. IEEE, 2022.

[21] Yuxi Cai, Georgios Fragkos, Eirini Eleni Tsiropoulou, and Andreas Veneris. A truth-inducing sybil resistant decentralized blockchain oracle. In *2020 2nd Conference on Blockchain Research & Applications for Innovative Networks and Services (BRAINS)*, pages 128–135, 2020.

[22] Yinfeng Cao, Jiannong Cao, Dongbin Bai, Zhiyuan Hu, Kaile Wang, and Mingjin Zhang. Polyverse: An edge computing-empowered metaverse with physical-to-virtual projection. In *2023 International Conference on Intelligent Metaverse Technologies & Applications (iMETA)*, pages 1–8. IEEE, 2023.

[23] Wren Chan and Aspen Olmsted. Ethereum transaction graph analysis. In *2017 12th international conference for internet technology and secured transactions (ICITST)*, pages 498–500. IEEE, 2017.

[24] Panagiotis Chatzigiannis, Foteini Baldimtsi, and Konstantinos Chalkias. Sok: Blockchain light clients. In *International Conference on Financial Cryptography and Data Security*, pages 615–641. Springer, 2022.

[25] Coinbase. Coinbase - buy and sell bitcoin, ethereum, and more with trust, 2023. Last accessed 14 Oct 2024.

[26] Sahraoui Dhelim, Tahar Kechadi, Liming Chen, Nyothiri Aung, Huansheng Ning, and Luigi Atzori. Edge-enabled metaverse: The convergence of metaverse and mobile edge computing. *arXiv preprint arXiv:2205.02764*, 2022.

[27] Mo Dong, Qingkai Liang, Xiaozhou Li, and Junda Liu. Celer network: Bring internet scale to every blockchain. *arXiv preprint arXiv:1810.00037*, 2018.

[28] Haihan Duan, Jiaye Li, Sizheng Fan, Zhonghao Lin, Xiao Wu, and Wei Cai. Metaverse for social good: A university campus prototype. In *Proceedings of the 29th ACM international conference on multimedia*, pages 153–161, 2021.

[29] Ben Edgington. Upgrading ethereum, a technical handbook on ethereum's move to proof of stake and beyond, 2023. Last accessed 14 Oct 2024.

[30] Electron-Labs. Ed25519 implementation in circom, 2023. Last accessed 14 Oct 2024.

[31] Michael Fröwis and Rainer Böhme. The operational cost of ethereum airdrops. In *Data Privacy Management, Cryptocurrencies and Blockchain Technology: ESORICS 2019 International Workshops, DPM 2019 and CBT 2019, Luxembourg, September 26–27, 2019, Proceedings 14*, pages 255–270. Springer, 2019.

[32] Christian Gehrmann and Martin Gunnarsson. A digital twin based industrial automation and control system security architecture. *IEEE Transactions on Industrial Informatics*, 16(1):669–680, 2019.

[33] Edd Gent. A cryptocurrency for the masses or a universal id?: Worldcoin aims to scan all the world's eyeballs. *IEEE Spectrum*, 60(1):42–57, 2023.

[34] Christopher Gorman. Eip-3068: Precompile for bn256 hashtocurve algorithms, 2020. Last accessed 14 Oct 2024.

[35] Jens Groth. On the size of pairing-based non-interactive arguments. In *Advances in Cryptology–EUROCRYPT 2016: 35th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Vienna, Austria, May 8-12, 2016, Proceedings, Part II 35*, pages 305–326. Springer, 2016.

[36] Barbara Guidi and Andrea Michienzi. Social games and blockchain: exploring the metaverse of decentraland. In *2022 IEEE 42nd International Conference on Distributed Computing Systems Workshops (ICDCSW)*, pages 199–204. IEEE, 2022.

[37] Beibei Han, Yingmei Wei, Qingyong Wang, Francesco Maria De Collibus, and Claudio J Tessone. Mt 2 ad: multi-layer temporal transaction anomaly detection in ethereum networks with gnn. *Complex & Intelligent Systems*, 10(1):613–626, 2024.

[38] Haya R Hasan, Khaled Salah, Raja Jayaraman, Mohammed Omar, Ibrar Yaqoob, Saša Pesic, Todd Taylor, and Dragan Boscovic. A blockchain-based approach for the creation of digital twins. *IEEE Access*, 8:34113–34126, 2020.

[39] Omar Hashash, Christina Chaccour, Walid Saad, Kei Sakaguchi, and Tao Yu. Towards a decentralized metaverse: Synchronized orchestration of digital twins and sub-metaverses. *arXiv preprint arXiv:2211.14686*, 2022.

[40] Zicong Hong, Song Guo, Enyuan Zhou, Wuhui Chen, Huawei Huang, and Albert Zomaya. Gridb: Scaling blockchain database via sharding and off-chain cross-shard mechanism.

[41] Sihan Huang, Guoxin Wang, Yan Yan, and Xiongbing Fang. Blockchain-based data management for digital twin of product. *Journal of Manufacturing Systems*, 54:361–371, 2020.

[42] Mathias Hummel and Kees van Kooten. Leveraging nvidia omniverse for in situ visualization. In *High Performance Computing: ISC High Performance 2019 International Workshops, Frankfurt, Germany, June 16-20, 2019, Revised Selected Papers 34*, pages 634–642. Springer, 2019.

[43] INTERFACING CYBER AND PHYSICAL WORLD WORKING GROUP. IEEE 2888 standards. `https://sagroups.ieee.org/2888/`, accessed 2023. Accessed on March 9, 2023.

[44] Mubashar Iqbal and Raimundas Matulevičius. Exploring sybil and double-spending risks in blockchain systems. *IEEE Access*, 9:76153–76177, 2021.

[45] Supun Karunarathna, Shalitha Wijethilaka, Pasika Ranaweera, Kasun T Hemachandra, Tharaka Samarasinghe, and Madhusanka Liyanage. The role of network slicing and edge computing in the metaverse realization. *IEEE Access*, 11:25502–25530, 2023.

[46] Davis E. King. Dlib-ml: A machine learning toolkit. *Journal of Machine Learning Research*, 10:1755–1758, 2009.

[47] Gaurish Korpal and Drew Scott. Decentralization and web3 technologies. *Authorea Preprints*, 2023.

[48] Jae Kwon and Ethan Buchman. Cosmos whitepaper. *A Netw. Distrib. Ledgers*, 27, 2019.

[49] Kit Yung Lam, Liang Yang, Ahmad Alhilal, Lik-Hang Lee, Gareth Tyson, and Pan Hui. Human-avatar interaction in metaverse: Framework for full-body interaction. In *Proceedings of the 4th ACM International Conference on Multimedia in Asia*, pages 1–7, 2022.

[50] Rongjian Lan, Ganesha Upadhyaya, Stephen Tse, and Mahdi Zamani. Horizon: A gas-efficient, trustless bridge for cross-chain transactions. *arXiv preprint arXiv:2101.06000*, 2021.

[51] Mehdi Letafati and Safa Otoum. Global differential privacy for distributed metaverse healthcare systems. In *2023 International Conference on Intelligent Metaverse Technologies I& Applications (iMETA)*, pages 01–08, 2023.

[52] Ricky Leung. Leveraging ai and blockchain for streamlining healthcare payments. In *Proceedings of the 2023 5th Blockchain and Internet of Things Conference*, BIOTC '23, page 58–62, New York, NY, USA, 2023. Association for Computing Machinery.

[53] Pengze Li, Mingxuan Song, Mingzhe Xing, Zhen Xiao, Qiuyu Ding, Shengjie Guan, and Jieyi Long. Spring: Improving the throughput of sharding blockchain

via deep reinforcement learning based state placement. In *Proceedings of the ACM Web Conference 2024*, WWW '24, page 2836–2846, New York, NY, USA, 2024. Association for Computing Machinery.

[54] Taotao Li, Changlin Yang, Qinglin Yang, Shizhan Lan, Siqi Zhou, Xiaofei Luo, Huawei Huang, and Zibin Zheng. Metaopera: A cross-metaverse interoperability protocol. *IEEE Wireless Communications*, 30(5):136–143, 2023.

[55] Wanxin Li, Collin Meese, Mark Nejad, and Hao Guo. Zk-bft: A zero-knowledge and byzantine fault tolerant consensus for permissioned blockchain networks. In *Proceedings of the 2023 6th International Conference on Blockchain Technology and Applications*, ICBTA '23, page 70–77, New York, NY, USA, 2024. Association for Computing Machinery.

[56] Yutian Lin, Liang Zheng, Zhedong Zheng, Yu Wu, Zhilan Hu, Chenggang Yan, and Yi Yang. Improving person re-identification by attribute and identity learning. *Pattern Recognition*, 95:151–161, 2019.

[57] Jinshan Liu and Jung-Min Park. "seeing is not always believing": Detecting perception error attacks against autonomous vehicles. *IEEE Transactions on Dependable and Secure Computing*, 18(5):2209–2223, 2021.

[58] Zheng Liu and Hongyang Zhu. Fighting sybils in airdrops. *arXiv preprint arXiv:2209.04603*, 2022.

[59] Zhuotao Liu, Yangxi Xiang, Jian Shi, Peng Gao, Haoyu Wang, Xusheng Xiao, Bihan Wen, and Yih-Chun Hu. Hyperservice: Interoperability and programmability across heterogeneous blockchains. In *Proceedings of the 2019 ACM SIGSAC conference on computer and communications security*, pages 549–566, 2019.

[60] Zhihan Lv, Liang Qiao, Yuxi Li, Yong Yuan, and Fei-Yue Wang. Blocknet: Beyond reliable spatial digital twins to parallel metaverse. *Patterns*, 3(5), 2022.

[61] Zhihan Lv, Shuxuan Xie, Yuxi Li, M Shamim Hossain, and Abdulmotaleb El Saddik. Building the metaverse using digital twins at all scales, states, and relations. *Virtual Reality & Intelligent Hardware*, 4(6):459–470, 2022.

[62] Damiano Di Francesco Maesa and Paolo Mori. Blockchain 3.0 applications survey. *Journal of Parallel and Distributed Computing*, 138:99–114, 2020.

[63] Daniel Mebrahtom, Siem Hadish, Aron Sbhatu, Moayad Aloqaily, and Mohsen Guizani. Trust but verify - blockchain-empowered decentralized authentication schema on the metaverse: A self-sovereign identity approach. In *2023 International Conference on Intelligent Metaverse Technologies I& Applications (iMETA)*, pages 1–8, 2023.

[64] Henrique Moniz. The istanbul bft consensus algorithm. *arXiv preprint arXiv:2002.03613*, 2020.

[65] Eduardo Morais, Tommy Koens, Cees Van Wijk, and Aleksei Koren. A survey on zero knowledge range proofs and applications. *SN Applied Sciences*, 1:1–17, 2019.

[66] Multichain. Cross-chain router protocol, 2023. Last accessed 14 Oct 2024.

[67] Faisal Nawab and Mohammad Sadoghi. 2023.

[68] Michael Neuder, Daniel J Moroz, Rithvik Rao, and David C Parkes. Low-cost attacks on ethereum 2.0 by sub-1/3 stakeholders. *arXiv preprint arXiv:2102.02247*, 2021.

[69] Huansheng Ning, Hang Wang, Yujia Lin, Wenxi Wang, Sahraoui Dhelim, Fadi Farha, Jianguo Ding, and Mahmoud Daneshmand. A survey on metaverse: the state-of-the-art, technologies, applications, and challenges. *arXiv preprint arXiv:2111.09673*, 2021.

[70] Huansheng Ning, Hang Wang, Yujia Lin, Wenxi Wang, Sahraoui Dhelim, Fadi Farha, Jianguo Ding, and Mahmoud Daneshmand. A survey on the metaverse: The state-of-the-art, technologies, applications, and challenges. *IEEE Internet of Things Journal*, 2023.

[71] Yanqing Peng, Min Du, Feifei Li, Raymond Cheng, and Dawn Song. Falcondb: Blockchain-based collaborative database. In *Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data*, pages 637–652, 2020.

[72] Benedikt Putz, Marietheres Dietz, Philip Empl, and Günther Pernul. Ethertwin: Blockchain-based secure digital twin information management. *Information Processing & Management*, 58(1):102425, 2021.

[73] Masud Rana, Ulrich Norbisrath, Eero Vainikko, and Bruno Rossi. Transforming affordable virtual reality headsets into effective learning environments. In *2023 International Conference on Intelligent Metaverse Technologies I& Applications (iMETA)*, pages 1–8, 2023.

[74] Gabriel Antonio F Rebello, Gustavo F Camilo, Lucas CB Guimaraes, Lucas Airam C de Souza, Guilherme A Thomaz, and Otto Carlos MB Duarte. A security and performance analysis of proof-based consensus protocols. *Annals of Telecommunications*, pages 1–21, 2021.

[75] Kunpeng Ren, Nhut-Minh Ho, Dumitrel Loghin, Thanh-Toan Nguyen, Beng Chin Ooi, Quang-Trung Ta, and Feida Zhu. Interoperability in blockchain: A survey. *IEEE Transactions on Knowledge and Data Engineering*, 2023.

[76] Pingcheng Ruan, Tien Tuan Anh Dinh, Dumitrel Loghin, Meihui Zhang, Gang Chen, Qian Lin, and Beng Chin Ooi. Blockchains vs. distributed databases: Dichotomy and fusion. In *Proceedings of the 2021 International Conference on Management of Data*, pages 1504–1517, 2021.

[77] Moon Gi Seok, Wen Jun Tan, Wentong Cai, and Daejin Park. Digital-twin consistency checking based on observed timed events with unobservable transitions in smart manufacturing. *IEEE Transactions on Industrial Informatics*, 19(4):6208–6219, 2022.

[78] Tianyu Shen, Shi-Sheng Huang, Deqi Li, Zhiyuan Lu, Fei-Yue Wang, and Hua Huang. Virtualclassroom: A lecturer-centered consumer-grade immersive teaching system in cyber–physical–social space. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 2022.

[79] Gurinder Singh, Garima Bhardwaj, S Vikram Singh, and Vikas Garg. Biometric identification system: security and privacy concern. *Artificial intelligence for a sustainable industry 4.0*, pages 245–264, 2021.

[80] Michael Sober, Giulia Scaffino, Christof Spanring, and Stefan Schulte. A voting-based blockchain interoperability oracle. In *2021 IEEE International Conference on Blockchain (Blockchain)*, pages 160–169. IEEE, 2021.

[81] Xiaoqiang Sun, F Richard Yu, Peng Zhang, Zhiwei Sun, Weixin Xie, and Xiang Peng. A survey on zero-knowledge proof in blockchain. *IEEE network*, 35(4):198–205, 2021.

[82] Saurabh Suratkar, Mahesh Shirole, and Sunil Bhirud. Cryptocurrency wallet: A review. In *2020 4th international conference on computer, communication and signal processing (ICCCSP)*, pages 1–7. IEEE, 2020.

[83] Behrang Ashtari Talkhestani, Nasser Jazdi, Wolfgang Schloegl, and Michael Weyrich. Consistency check to synchronize the digital twin of manufacturing automation based on anchor points. *Procedia Cirp*, 72:159–164, 2018.

[84] Bitcoin Taxes. 5 biggest crypto cross-chain bridge hacks in 2022, 2023. Last accessed 14 Oct 2024.

[85] CHAINALYSIS TEAM. Multichain protocol experiences mysterious withdrawals, suggesting multi-million dollar hack or rug pull [updated 7/19/23], 2023. Last accessed 14 Oct 2024.

[86] Jeyakumar Samantha Tharani, Eugene Yougarajah Andrew Charles, Zhé Hóu, Marimuthu Palaniswami, and Vallipuram Muthukkumarasamy. Graph based visualisation techniques for analysis of blockchain transactions. In *2021 IEEE 46th Conference on Local Computer Networks (LCN)*, pages 427–430. IEEE, 2021.

[87] Vu Tuan Truong, Long Le, and Dusit Niyato. Blockchain meets metaverse and digital asset management: A comprehensive survey. *Ieee Access*, 11:26258–26288, 2023.

[88] Psi Vesely, Kobi Gurkan, Michael Straka, Ariel Gabizon, Philipp Jovanovic, Georgios Konstantopoulos, Asa Oines, Marek Olszewski, and Eran Tromer. Plumo: An ultralight blockchain client. In *International Conference on Financial Cryptography and Data Security*, pages 597–614. Springer, 2022.

[89] Chen Wang, Yan Wang, Yingying Chen, Hongbo Liu, and Jian Liu. User authentication on mobile devices: Approaches, threats and trends. *Computer Networks*, 170:107118, 2020.

[90] Gang Wang, Qin Wang, and Shiping Chen. Exploring blockchains interoperability: A systematic survey. *ACM Computing Surveys*, 2023.

[91] Yuntao Wang, Zhou Su, Ning Zhang, Rui Xing, Dongxiao Liu, Tom H Luan, and Xuemin Shen. A survey on metaverse: Fundamentals, security, and privacy. *IEEE Communications Surveys & Tutorials*, 2022.

[92] Yuyang Wang, Lik-Hang Lee, Tristan Braud, and Pan Hui. Re-shaping post-covid-19 teaching and learning: A blueprint of virtual-physical blended classrooms in the metaverse era. In *2022 IEEE 42nd International Conference on*

Distributed Computing Systems Workshops (ICDCSW), pages 241–247. IEEE, 2022.

[93] Ziwei Wang, Jiashi Gao, and Xuetao Wei. Do nfts' owners really possess their assets? a first look at the nft-to-asset connection fragility. In *Proceedings of the ACM Web Conference 2023*, WWW '23, page 2099–2109, New York, NY, USA, 2023. Association for Computing Machinery.

[94] Sam Werner, Daniel Perez, Lewis Gudgeon, Ariah Klages-Mundt, Dominik Harz, and William Knottenbelt. Sok: Decentralized finance (defi). In *Proceedings of the 4th ACM Conference on Advances in Financial Technologies*, pages 30–46, 2022.

[95] Martin Westerkamp and Jacob Eberhardt. zkrelay: Facilitating sidechains using zksnark-based chain-relays. In *2020 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW)*, pages 378–386. IEEE, 2020.

[96] Gavin Wood. Polkadot: Vision for a heterogeneous multi-chain framework. *White paper*, 21(2327):4662, 2016.

[97] Gavin Wood et al. Ethereum: A secure decentralised generalised transaction ledger. *Ethereum project yellow paper*, 151(2014):1–32, 2014.

[98] Yiwen Wu, Ke Zhang, and Yan Zhang. Digital twin networks: A survey. *IEEE Internet of Things Journal*, 8(18):13789–13804, 2021.

[99] Tiancheng Xie, Jiaheng Zhang, Zerui Cheng, Fan Zhang, Yupeng Zhang, Yongzheng Jia, Dan Boneh, and Dawn Song. zkbridge: Trustless cross-chain bridges made practical. In *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security*, pages 3003–3017, 2022.

[100] Cheng Xu, Ce Zhang, Jianliang Xu, and Jian Pei. Slimchain: Scaling blockchain transactions through off-chain storage and parallel processing. *Proceedings of the VLDB Endowment*, 14(11):2314–2326, 2021.

[101] Minrui Xu, Wei Chong Ng, Wei Yang Bryan Lim, Jiawen Kang, Zehui Xiong, Dusit Niyato, Qiang Yang, Xuemin Sherman Shen, and Chunyan Miao. A full dive into realizing the edge-enabled metaverse: Visions, enabling technologies, and challenges. *IEEE Communications Surveys & Tutorials*, 2022.

[102] Minrui Xu, Dusit Niyato, Benjamin Wright, Hongliang Zhang, Jiawen Kang, Zehui Xiong, Shiwen Mao, and Zhu Han. Epvisa: Efficient auction design for real-time physical-virtual synchronization in the metaverse. *arXiv preprint arXiv:2211.06838*, 2022.

[103] Ibrar Yaqoob, Khaled Salah, Mueen Uddin, Raja Jayaraman, Mohammed Omar, and Muhammad Imran. Blockchain for digital twins: Recent advances and future research challenges. *Ieee Network*, 34(5):290–298, 2020.

[104] Ryan Zarick, Bryan Pellegrino, and Caleb Banister. Layerzero: Trustless omnichain interoperability protocol. *arXiv preprint arXiv:2110.13871*, 2021.

[105] He Zhang, Qinglin Qi, Wei Ji, and Fei Tao. An update method for digital twin multi-dimension models. *Robotics and Computer-Integrated Manufacturing*, 80:102481, 2023.

[106] He Zhang, Qinglin Qi, and Fei Tao. A consistency evaluation method for digital twin models. *Journal of Manufacturing Systems*, 65:158–168, 2022.

[107] Jiashuo Zhang, Jianbo Gao, Yue Li, Ziming Chen, Zhi Guan, and Zhong Chen. Xscope: Hunting for cross-chain bridge attacks. In *Proceedings of the 37th IEEE/ACM International Conference on Automated Software Engineering*, pages 1–4, 2022.

[108] Shijie Zhang and Jong-Hyouk Lee. Double-spending with a sybil attack in the bitcoin decentralized network. *IEEE transactions on Industrial Informatics*, 15(10):5715–5722, 2019.