xPayments: Cross-Domain End-to-End Payment Protocol using Payment Channel Network

1st Anupa De Silva
Data Science Institute
University of Galway, Ireland
anupa.shyamlal@insight-centre.org

2nd Subhasis Thakur
Data Science Institute
University of Galway, Ireland
subhasis.thakur@insight-centre.org

3rd John Breslin
Data Science Institute
University of Galway, Ireland
john.breslin@universityofgalway.ie

***Abstract*—The rapid integration of digital technologies has precipitated a substantial transition from traditional in-person commerce to digital trade, even encompassing physical goods. This transformation not only simplifies the purchasing process but also broadens the market reach for vendors, enabling global connectivity with consumers. Presently, there is a shift towards Metaverse-based applications, with blockchains serving as the foundational infrastructure for data management, thereby enhancing the overall user experience. Given this context, integrating payment methods becomes pivotal, particularly in light of the widespread adoption of blockchain technology. Our proposal introduces xPayment, a decentralised payment protocol designed to facilitate end-to-end and cross-domain transactions for digital goods via payment channel networks (PCNs). Initially, we introduce the concept of cyclic exchange and formally demonstrate its atomicity, on which we develop xPayment protocol to facilitate an independent and interoperable payment mechanism integrated into blockchain-based ownership transfers. We provide both theoretical and empirical evidence of its atomicity, minimal merchant opportunity cost, and customisable privacy features.**

***Index Terms*—Cryptocurrency, Blockchain, Payment, Metaverse, E-commerce, Atomic Swap, Payment Channel Network, Interoperability, NFT**

I. Introduction

Blockchain technology has emerged as a crucial enabler, supporting the tracking and tracing of products from their origin to subsequent owners, irrespective of whether they are physical or virtual goods. This role of blockchain instils a sense of security and confidence in digital transactions in a decentralised setting. Financial transactions are pivotal in many blockchain applications, and their seamless integration is paramount. All blockchain architectures inherently support integrated data and finance management. However, in practical terms, no single all-in-one blockchain effectively meets the diverse needs, challenges, and opportunities presented by different use cases. As a result, there has been a proliferation of blockchain implementations customised to meet the specific requirements of individual use cases. Further, the blockchain trilemma poses a theoretical limitation, making it challenging to simultaneously achieve high levels of decentralisation, scalability, and security. Consequently, developers and organisations often prioritise certain attributes based on their specific use case requirements, goals, and constraints. For example, payment applications prioritise transaction throughput and resistance to double spending, while data applications focus on securely storing and managing large volumes of information. These considerations have led to the development of multiple blockchains optimised for different combinations of attributes. Given the proliferation of blockchain applications, a pertinent question arises: Can consumers keep their financial assets in every blockchain of the diverse blockchain applications?

Decoupling financial operations from data management offers a promising solution to the trilemma. This approach curbs the proliferation of payment methods that lead to currency fragmentation, consequently causing liquidity challenges, price volatility, and uncertainty [1]. Furthermore, it enhances consumer convenience by allowing them to utilise their default financial source across various applications and scenarios. When coupled with interoperability, in which blockchain plays a pivotal role, decoupling can extend across connected contexts, including different blockchains, physical and virtual worlds, and interactions between virtual environments. The potential of blockchain interoperability, as emphasised by Kannengiesser et al. [2], is significant in upholding robust security and fostering trust within the blockchain ecosystem. It enables the payment system in the physical world to transact in digital environments such as blockchain, Metaverse, or gaming, allowing for seamless financial interactions and eliminating the necessity to switch or store assets across multiple platforms. This approach also empowers developers to focus on achieving the optimal balance between scalability and security without encountering a dilemma.

The objective of this paper is to propose a cross-domain payment method. It aims to create an end-to-end trading protocol in a decentralised setting where the product exchange and the associated payment are intertwined. The traded product can be digital, such as a Metaverse avatar, or physical with a corresponding digital representation, such as a tokenised version. Initially, we introduce the concept of cyclic exchange and formally demonstrate its atomicity. Subsequently, we develop the xPayment protocol by utilising the PCN concept to facilitate an independent payment mechanism that is interoperable with data management blockchains. Based on our development efforts, we have successfully created a proof of concept utilising Hyperledger as the blockchain for data management. This implementation showcases the expedited exchange process, reducing the impact on the merchant's opportunities. Additionally, we present potential use cases, emphasising the growing significance of Metaverse applications and xPayment's role.

II. Background

A. Payment Channel Network

PCN is a second-layer scaling solution for blockchain networks that facilitates fast, low-cost, and secure transactions. PCNs allow parties to conduct multiple transactions off-chain,

settling them on the blockchain only when necessary, thus alleviating on-chain congestion and reducing fees. To begin using a PCN, two parties open a payment channel, which involves creating a contract on the blockchain which locks up a certain amount of funds from both parties whose consent is required to move the funds, ensuring mutual control. This transaction is recorded on the blockchain, establishing the initial balance of the channel. Once the channel is open, the two parties can transact with each other off-chain by adjusting the balance. Each time a new transaction is made, the previous state is invalidated by exchanging revocation keys. This prevents either party from unilaterally broadcasting an outdated state to the blockchain to reclaim funds. A key feature of PCNs is their ability to route payments through a network of channels. If two parties do not have a direct channel, payments can still be routed through multiple intermediate nodes (other participants in the network) with established channels. The sender must find a path with sufficient liquidity to facilitate the payment. Payments are securely routed through the network using cryptographic techniques like Hashed Time-Lock Contract (HTLC) or Anonymous Multi-Hop Lock (AMHL) [3]. They ensure that the payment is either fully completed or not at all, preventing loss of funds in transit.

B. Atomic Swap

When goods are exchanged, atomicity ensures that the exchange happens as a single, indivisible transaction, where all involved parties either receive their desired goods or none at all. Achieving atomicity becomes challenging when the parties are from different environments. For instance, when a merchant and a consumer operate in different environments, realising the atomicity of the transaction is complex. An example of this challenge is when two parties from different geographical locations engage in a financial transaction for a product purchase. Achieving atomicity in such scenarios can involve one party taking the risk and initiating the transaction or employing an escrow service to mediate. However, this complexity does not apply to blockchains due to the atomic swap technique introduced by Tier Nolan in 2013.

Algorithm 1 $\mathsf{TLC}_{i,j}(a_i, a_j, t_i, t_j)$

1: a_j and a_i are locked until t_j and t_i times elapse.
2: **if** v_j releases a_j to v_i before t_j time **then**
3: v_j receives a_i.
4: **end if**
5: **if** t_j/t_i elapses **then**
6: v_j/v_i retains a_j/a_i, respectively.
7: **end if**

The atomic swap involves the exchange of assets a_i and a_j held by two parties, v_i and v_j, on separate blockchains B_i and B_j. This exchange ensures neither party can receive their respective assets without first transferring their own. Initially, both parties consent to the terms outlined in Algorithm II-B, referred to as the Time Lock Contract (TLC). The atomic swap process begins with the *locking* assets, rendering them non-transferable until specific conditions are met. First, v_j locks asset a_j for a duration of t_j. Then, v_i does the same for asset a_i with a timeout of t_i or takes no action, ultimately resulting in the return of a_j to v_j after the timeout. Importantly, v_i must establish the timeout before t_j to allow a time window to claim the product. Then, v_j initiates the *release* phase, during which v_j can either release the product before t_i or abstain from any action, resulting in the return of the assets to the respective party after the timeouts have elapsed. If the asset is released, v_i is entitled to claim a_j before t_j as per the contract terms. Both parties must verify the counterpart's action before taking any action, with the verification timeline depending on the settlement time of the respective blockchain.

Griefing and Opportunity Cost: When users deviate from the atomic swap, griefing can occur on both sides. Deviations may occur due to underlying reasons such as potential loss avoidance from asset value fluctuation, change of intention, or deliberate griefing attacks. Griefing vulnerabilities are observable in cryptocurrency exchanges based on atomic swaps, particularly when confronted with price fluctuations. In trading, the incentives to execute the swap rest predominantly with the merchant. However, consumers may rescind their commitment, and malicious entities may impersonate consumers to disrupt transactions, necessitating the critical implementation of tight time locks to mitigate potential opportunity costs merchants face. It is crucial to note that the effectiveness of time locks is contingent on the settlement time of the underlying blockchain.

C. Related Works

Extensive research within the academic community has focused on the interoperability of blockchain, employing techniques such as atomic swaps, side chains, and the utilisation of third-party services [4], [5]. The straightforward solution is to use third-party services to address the trust deficit between transacting parties. Another approach is the implementation of sidechains, which enables the interoperability of two existing blockchains by maintaining an asset ledger on the mainchain and connecting it to the sidechain through a cross-chain communication protocol [4]. Additionally, industrial solutions, such as Cosmos and Polkadot, boast economical consensus mechanisms to facilitate interoperability.

Unlike other methods, atomic swaps allow for decentralised transactions and can be easily implemented without major architectural changes. Game theory evaluations shed light on the implementation of atomic swaps. Belotti et al.'s game theoretical model [6] explores swaps involving more than two parties and establishes the Nash equilibrium for concurrent publishing of exchange contracts and immediate asset transfers. Additionally, Xu et al. [7] demonstrate that an agent may withdraw from a swap to maximise their utility as asset prices fluctuate, indicating that the success of atomic swaps is tied to crypto asset volatility. Several proposals to enhance the success of atomic swaps are also presented, including the introduction of premium [8], third-party mediation, and adjustments to the exchange rate in response to price fluctuations. These insights are crucial for understanding and improving the efficacy of atomic swaps.

In interoperability applications, there is a notable emphasis on facilitating cross-chain transactions involving financial assets [9] and data transmission protocols [10]. Our particular focus lies in the realm of asymmetrical transactions, which involve the exchange of both goods and financial assets. Noteworthy work in this area can be found in [11], which introduces an NFT scheme enabling trades to be settled through

a single Bitcoin transaction, as opposed to the execution of complex smart contracts.

III. Cyclic Exchange

The following excerpt outlines the concept of *cyclic exchange*, which refers to a transaction that returns to the originator through intermediary entities. We formally define this process and substantiate its atomicity through formal verification. This is a critical stride in building end-to-end payment protocol, xPayments. The preliminary definitions are as follows. A graph $\mathbb{G}$ comprises a set of vertices $\mathbb{V}$ and a set of edges $\mathbb{E}$, where each edge $(v_i, v_j) \in \mathbb{E}$ connects two vertices. In a graph, a path is a sequence of vertices $(v_0, v_1, ..., v_{N-1})$ connected by edges, where $\{(v_0, v_1), ..., (v_i, v_{i+1}), ..., (v_{n-1}, v_{N-1})\} \subseteq \mathbb{E}$. A path is cyclic if the first and last vertices have an edge between them. Formally, a subgraph of $\mathbb{G}$ is a cyclic graph when it consists of a path of vertices $v_0, ..., v_{N-1}$ with an edge (v_0, v_{N-1}).

A cyclic exchange is a cyclic graph in which each vertex represents a party involved in the overall transaction and an edge, say for $(v_i, v_{i'})$ where $i' = (i+1)\%N$, represents the exchange of assets $a_{i,i'}$ and $a_{i',i}$ from v_i to $v_{i'}$ and vice versa, respectively. We refer to $a_{i,i'}$ as a *forward-flow asset* that flows clockwise and $a_{i',i}$ as a *backward-flow asset* that flows in the opposite direction. At the initiation of the exchange, both parties are subject to the TLC, outlined in Algorithm II-B. This exchange process is extended to other nodes, enabling parties to sequentially exchange assets in a clockwise and anti-clockwise manner as defined in Definition III.1.

Definition III.1. *(Cyclic Exchange) Cyclic exchange $\mathbb{C}_N$ is carried out by the parties who form a cyclic graph with initiator v_0 and agree on* TLC *s.t. $\forall i \in [0, ..., N-1]$, $\mathsf{TLC}_{i,i'}(a_{i,i'}, a_{i',i}, t_i, t_{i'})$ is established where $i' = (i+1)\%N$, $a_{i,j}$, $a_{j,i}$, t_i, and t_i are their respective assets and time-locks and $t_i > t_{i'}$.*

Theorem 1. *(Atomicity of Cyclic Exchange) The exchanges associated with the edges in $\mathbb{C}_N$ are either entirely successful or entirely unsuccessful if all parties act rationally, $\forall i \in [0, ..., N]$, release of $a_{i,(i-1)\%N}$ depends on $a_{i,(i+1)\%N}$ and initiator v_0 releases $a_{0,1}$ after the locking phase's completion.*

Proof. To prove the theorem, we utilise Uppaal [12], a comprehensive software environment tailored for modelling, validating, and verifying real-time systems represented as networks of timed automata. Each party or vertex is depicted as a finite automaton, and the initiator is modelled separately due to its distinct behaviour. The initiator instigates the process at time unit $t = 0$, and each party commences in the *retained* state, transitioning to the *swapped* state based on the party's behaviour. Interactions among automata are facilitated through three signals: lock_fwd for locking, lock_bck for confirming the lock, and release for releasing. A timeout is introduced for each vertex to denote the maximum time it can remain in a given state, employing time locks as defined by a sequence of timeouts T of size N such that $\forall i \in [0, ..., N-1], T[i] > T[(i+1)\%N]$. Furthermore, the model encompasses a release dependency, where a vertex releases only after the locking phase. We also introduce a committed state (no wait between two actions) during the release phase to indicate that parties act rationally, ensuring they do not deviate from releasing backwards when they receive the successor's asset.

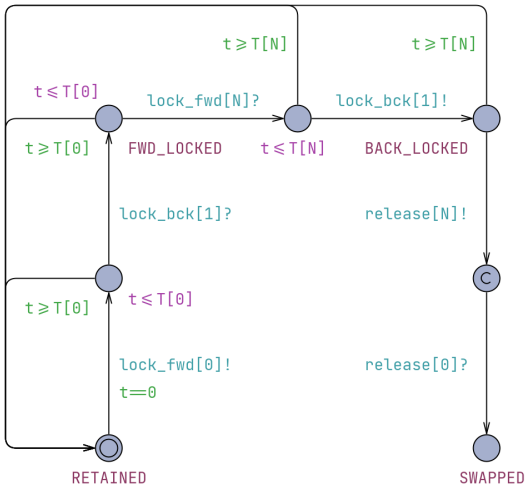

Fig. 1. Finite automaton of the initiator v_0 denoted by $V(0)$.

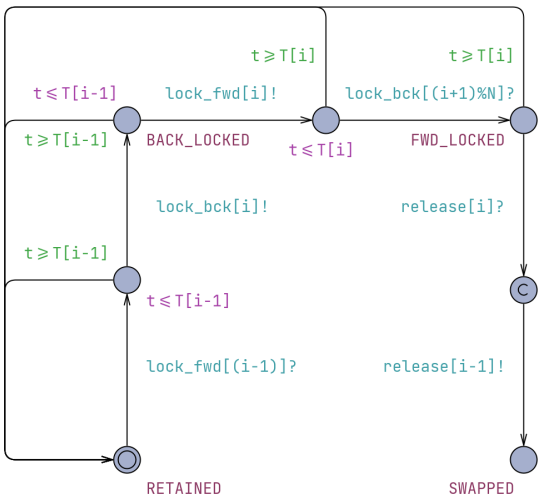

Fig. 2. Finite automaton of the vertex v_i denoted by $V(i)$ ($i \in [1, ..., N]$).

TABLE I
Verification properties and results.

	Property	Result
1	`V(4).SWAPPED → not V(3).SWAPPED`	False
2	`V(3).SWAPPED → not V(2).SWAPPED`	False
3	`V(2).SWAPPED → not V(1).SWAPPED`	False
4	`V(1).SWAPPED → not V(0).SWAPPED`	False
5	`E <> forall(i,int[0,4]) V(i).RETAINED or forall(i,int[0,4]) V(i).SWAPPED`	True

In assessing the system's properties, as presented in Table I, the Computation Tree Logic of Uppaal's verification module is employed. We have set N to 5 for simplicity, although this can be scaled to any $N \geq 2$. Our initial focus is on the response property when the predecessor reaches a state of successful swap. The outcome indicates that successors are exclusively in a swapped state, thereby ensuring successful cyclic exchange. Moreover, the verification confirms that all

vertices in the system eventually transition to a retained or swapped state, thereby establishing the atomicity of the model. □

A. Applications of Cyclic Exchange

The concept of cyclic exchanges is a generic protocol applicable under different conditions. One necessity is an environment to enforce TLC contract and create the dependency among the pairwise exchange. Cyclic exchange can be applied when multiple parties are involved in different blockchains and want to exchange their assets, such as NFTs, tokens, and cryptocurrencies, directly with each other. In this case, each party in the cycle sends their asset in exchange for another asset. To make this possible, TLC requires an additional lock to create dependency among the asset transfers. An example is the HTLC, which consists of a collision-resistant hash as the lock and its pre-image as the key. Similar to on-chain operations, cyclic exchange can be applied to off-chain transactions, such as PCN. This represents a specialised case where the keys for the locks become the backward-flow assets, creating a dependency. The lock can be either hash-based or signature-based, but the basic principle is that money is transferred in exchange for the key to the lock. In practice, cyclic exchange can be observed in PCN, where nodes engage in cyclic transfers to rebalance their channels. The Off-Chain Cyclic Exchange is a hybrid model utilising the blockchain to resolve disputes between paired entities. However, in this segment, we focus on interoperability between pairs, where pairs are based on both on-chain and off-chain to execute the cyclic exchange. We present this as xPayments.

IV. xPayments Design

The cyclic exchange is ideally suited for facilitating end-to-end and mediated transfer processes. Its end-to-end nature is especially advantageous for integrating embedded payments, particularly in digital trading. Digital trading typically involves digitalised products, services, or digital representation of physical items. In this context, we highlight the following points regarding digital trading. Atomicity of product and payment exchange is paramount in a decentralised setting where merchant and consumer need not trust each other. Various domains based on blockchains possess inherent security and trust level expectations. Mediated transfers can effectively reconcile such disparities without compromising inherent strengths or necessitating structural modifications within the domain. Merchants expect to minimise opportunity costs caused by the temporary product assignment to a consumer. If the transaction fails, the merchant must retain and make the product available to others. Consumers expect privacy such that their identities cannot be linked to the purchasing action.

Figure 3 presents the overall architecture of xPayments where the consumer maintains custodial wallets for asset management (Asset Manager) connected to the blockchain and payments (Consumer Wallet) connected to PCN. Merchant connects to the same blockchain through Asset Store and maintains a wallet (Merchant Wallet) to receive payments. In a generic merchant-consumer interaction context, a merchant operates a blockchain-based digital store, and consumers engage with the same blockchain for their transactions. However, consumers use a payment method independent of the blockchain, typically through PCN. Additionally, consumers are assumed to connect through their custodial wallets to simplify client operations.

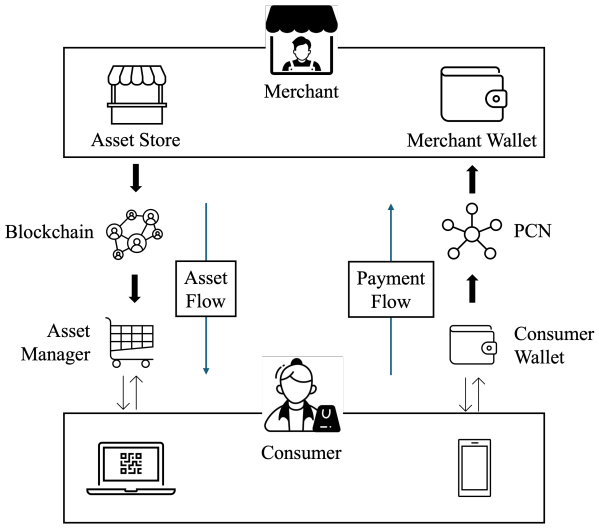

Fig. 3. Generalised merchant-consumer exchange based on xPayments.

The detailed process is as follows. The merchant provides asset information to the consumer's asset manager, including the merchant's payment end-point (wallet address), price, and a user and product-specific identifier. The consumer's asset manager then generates a unique lock with a key, only known to the asset manager, and prompts the consumer to capture it through their mobile wallet. With the customer's consent, the mobile client facilitates a mediated transfer to the provided address, integrating the provided lock. Like PCN, we assume an anonymous communication to the merchant whereby the customer can also include the user and product-specific identifier alongside the payment information. The merchant cross-verifies the payment received through PCN and the product cost using a unique identifier. If the verification is successful, the merchant performs a conditional transfer of the relevant products while maintaining the same lock on the relevant blockchain. The consumer's asset manager verifies the transaction, provides the key to the lock and obtains the asset. The consumer is then notified regarding the successful completion of the asset transaction, and the merchant can observe the key for the lock. Upon obtaining the key from the blockchain, the merchant can release the payment and claim the funds, thereby enabling subsequent nodes to claim the funds. Any failure during the above steps leads to returning the assets to the respective parties.

A. Implementation

We implement xPayments for generic consumer-merchant engagement. The implementation of PCN and the use of blockchain involves two critical design decisions. Hyperledger Fabric, which utilises a PBFT consensus mechanism, was chosen for the blockchain due to its cost-effectiveness and high-scale data management capabilities. To prevent wormhole attacks, PCN is implemented using a Schnorr-based AMHL instead of hash-based locks. The user interface of the merchant's store can be either a traditional web interface or a Metaverse-based application. Selling items are modelled as non-fungible

TABLE II
AVERAGE TIME TAKEN OF EACH PHASE OF THE CYCLIC EXCHANGE.

Action	Timeline
Payment Locking	0.316s
Asset Locking	2.134s
Asset Release	1.032s
Payment Release	1.508s
Asset Received	3.482s
Payment Received	4.130s
Overall Time	5.014s

tokens in the blockchain, based on ERC271 specifications for NFT[1] and developing swapping contracts. PCN is emulated through a mobile app with a mobile client for the consumer and server-based intermediaries. Merchants run the PCN wallet and asset manager on the server side.

The process begins with the user submitting a purchase request, which then redirects to the consumer's asset manager with product and merchant information. The asset manager generates and returns a QR code, including the lock and payment data, whereby the consumer can use the PCN mobile wallet to initiate the transaction towards the merchant. We utilise a standard two-user setting for the PCN. Upon receiving the payment, the merchant initiates the swap protocol with the same lock, which the consumer (i.e., asset manager) has to reveal the key to claim the product. The consumer listens to the blockchain events and claims the product by revealing the key and completing the transfer. Similarly, the merchant listens to the claiming event to extract the key and claim the received payment through PCN, finalising the exchange. However, the PCN intermediaries' settlement process is yet to be completed by releasing the locks and claiming the fees. The proof of concept can be found here[2].

B. Results

In the proof of concept, we use a 100ms delay for communication between PCN peers to emulate the realistic environment. Further, Hyperledger Fabric and PCN collate and process transactions every second. We measure the time taken at each step of the overall process. Table II presents the average time taken. The process commences with the consumer's PCN client initiating the payment locking process, which takes approximately 0.316 seconds to reach the merchant. Subsequently, the merchant's asset store initiates the asset locking process, which involves authorising the *swapping* contract to transfer the asset, followed by initiating the swap protocol with the lock received from the PCN payment. The consumer receives the asset after an average of 3.482 seconds, enabling the merchant to claim the payment. Regarding the overall exchange, the process is completed after an average of 4.130 seconds. It is important to note that settlement concludes when the merchant receives the payment, although the overall process continues until all intermediaries are compensated. On average, the overall process takes 5.014 seconds to complete.

V. EVALUATION AND DISCUSSION

The payment process mirrors the card payment procedure in e-commerce, with data management occurring in centralised databases and payments executed through SET/3D Secure payment protocols. Compared to traditional methods, we emphasise the unique aspects of our approach, including atomic exchange, controllable privacy, merchant settlement, and characteristics derived from our decentralised nature. Furthermore, xPayments demonstrate a reduction in opportunity costs, addressing issues commonly associated with blockchain-based solutions. Table V summarises comparisons between xPayments and other methods.

A. Atomicity

With respect to the definition of cyclic transfer, the consumer, PCN intermediary nodes, and merchant constitute a cyclic transfer where the consumer is the initiator. Let us consider the secrets of the individual locks as clockwise assets and let the transaction fees for the PCN intermediary nodes, the merchant's fee, and the merchant's product be the anti-clockwise assets (Even though intermediaries pass more than the transaction fees, the final lump sum is the transaction fee). All parties are incentivised to ensure the expiration time condition and rational assumptions. Further, only the payment initiator knows the keys to the intermediate locks, and AMHL guarantees that all entities must participate in the release process. Therefore, we argue that xPayments are according to the definition III.1 and, hence, satisfy the atomicity of the process, defined in 1.

B. Controllable Privacy

Traditional payment methods offer limited control over user privacy and are largely governed by regulations set by central authorities. In contrast, users can control the desired level of privacy in PCN-based payments by adjusting the number of intermediaries. The isolation of product and payment handling further enhances user privacy.

C. Minimised loss

In accordance with Section II-B, it is essential to acknowledge that all forms of atomic swaps are vulnerable to Griefing attacks, wherein preceding nodes are impacted when subsequent nodes deviate from the protocol. The proposed xPayment involves multiple swaps, prompting a need to highlight the area most susceptible to disruption during cyclic transfers. In the merchant-consumer domain, merchants and payment intermediaries are driven by incentives to maximise profits through product sales and transaction mediation, respectively. As a result, it is unlikely for subsequent nodes representing consumers to deviate from the protocol. However, merchants may experience opportunity costs as a result of consumer divergence. For instance, when a merchant locks a product for a consumer to claim, the consumer's change of mind or potential malicious intent can lead to non-receipt of the product, resulting in extended lock periods (i.e., lock timeouts) and preventing other consumers from making purchases. Considering the non-realtime nature of blockchain settlements, which can take minutes to complete in certain blockchains, longer timeouts are necessary. Consequently, it becomes imperative to set extended expiration times. Our design decisions play a critical role in minimising opportunity costs. We utilise PBFT-based Hyperledger for data management and PCN for payments, both of which offer near-realtime settlement. As a result, consumers are able to set a minimum timeout for the

[1] bitcoinwiki.org/wiki/erc-721
[2] https://github.com/anupasm/la_net/tree/main/xworld

TABLE III
COMPARISON OF XPAYMENTS WITH OTHER TECHNIQUES AND SETTINGS.

Implementation		High Throughput	Transaction Cost	Scalability	Payment Security	Controllable Privacy
Data Management	Payment					
Ethereum		Low	High	Low	High	High
Hyperledger		High	Low	High	Low	High
Database	SET/3D Secure	Average	Average	Average	High	Average
Hyperledger	PCN	High	Low	High	High	High

product lock, thereby preventing extended delays in releasing the product. We demonstrate this in our PoC in Section IV-B.

D. Use cases

The proposed protocol is suitable for application involving two distinct environments, one of which operates as a monetary system facilitating the exchange of goods in the other environment. A prime example is supply chain management facilitated by blockchain technology for traceability. In this context, the purchase record is critical for future resale of the product in a secondary market, warranty claims, and awareness of any deficiencies. Therefore, developers can implement a scalable blockchain infrastructure to handle large volumes of data effectively. Instead of integrating the payment mechanism directly into the system, they can enable interoperability with an external monetary system through the proposed protocol. The Metaverse ecosystem also encompasses various financial activities involving creating, exchanging, and consuming virtual goods, services, and experiences. These activities reflect many facets of real-world economics but are exclusively conducted within virtual realms. Therefore, it is crucial to establish a seamless mechanism for making payments from the physical world that is compatible with the blockchain underpinning the Metaverse. Maintaining separate monetary systems for the virtual and physical worlds, with money in each blockchain, may prove less convenient for consumers. While consumers may trust the blockchain for the goods they purchase, they may be hesitant to keep their money within it. The integration of the proposed design can facilitate inter-world transactions and support atomicity. The gaming industry is also a prime example of financial activities within the Metaverse involving in-game transactions. This includes players making in-app purchases to acquire virtual items or other content within a game. Our proposed protocol can streamline these transactions, enhancing the gaming experience.

VI. CONCLUSION

This paper presents an end-to-end trading protocol in a decentralised setting where the product exchange and the associated payment are intertwined. The traded product can be digital, such as a Metaverse avatar, or physical with a corresponding digital representation, such as a tokenised version. Initially, we introduce the concept of cyclic exchange and formally demonstrate its atomicity. Subsequently, we develop the xPayment protocol by utilising the PCN concept to facilitate an independent and interoperable payment mechanism that is interoperable with data management blockchains. Based on our development efforts, we have successfully created a proof of concept utilising Hyperledger as the blockchain for data management. This implementation showcases the expedited exchange process, reducing the impact on the merchant's opportunities. The proposed xPayment protocol is particularly suited to emerging Metaverse applications, given the proliferation of blockchains that the Metaverse relies on for data management. It allows users to experience seamless payments with their preferred wallet application without reliance on centralised entities.

ACKNOWLEDGMENT

This publication has emanated from research supported in part by a grant from Science Foundation Ireland and the Department of Agriculture, Food and the Marine on behalf of the Government of Ireland under Grant Number 16/RC/3835 (VistaMilk), and also by a grant from SFI under Grant Number 12/RC/2289_P2 (Insight). For the purpose of Open Access, the author has applied a CC BY public copyright licence to any Author Accepted Manuscript version arising from this submission.

REFERENCES

[1] S. Claessens, "Fragmentation in global financial markets: good or bad for financial stability?" 2019.

[2] N. Kannengießer, S. Lins, T. Dehling, and A. Sunyaev, "Trade-offs between distributed ledger technology characteristics," *ACM Computing Surveys (CSUR)*, vol. 53, no. 2, pp. 1–37, 2020.

[3] G. Malavolta, P. Moreno-Sanchez, C. Schneidewind, A. Kate, and M. Maffei, "Anonymous multi-hop locks for blockchain scalability and interoperability," *Cryptology ePrint Archive*, 2018.

[4] G. Wang, "Sok: Exploring blockchains interoperability," *Cryptology ePrint Archive*, 2021.

[5] A. Zamyatin, M. Al-Bassam, D. Zindros, E. Kokoris-Kogias, P. Moreno-Sanchez, A. Kiayias, and W. J. Knottenbelt, "Sok: Communication across distributed ledgers," in *Financial Cryptography and Data Security: 25th International Conference, FC 2021, Virtual Event, March 1–5, 2021, Revised Selected Papers, Part II 25*. Springer, 2021, pp. 3–36.

[6] M. Belotti, S. Moretti, M. Potop-Butucaru, and S. Secci, "Game theoretical analysis of cross-chain swaps," in *2020 IEEE 40th International Conference on Distributed Computing Systems (ICDCS)*. IEEE, 2020, pp. 485–495.

[7] J. Xu, D. Ackerer, and A. Dubovitskaya, "A game-theoretic analysis of cross-chain atomic swaps with htlcs," in *2021 IEEE 41st international conference on distributed computing systems (ICDCS)*. IEEE, 2021, pp. 584–594.

[8] R. Han, H. Lin, and J. Yu, "On the optionality and fairness of atomic swaps," in *Proceedings of the 1st ACM Conference on Advances in Financial Technologies*, 2019, pp. 62–75.

[9] X. Jia, Z. Yu, J. Shao, R. Lu, G. Wei, and Z. Liu, "Cross-chain virtual payment channels," *IEEE Transactions on Information Forensics and Security*, 2023.

[10] S. Zaman, R. Dantu, S. Badruddoja, S. Talapuru, and K. Upadhyay, "Layerwise interoperability in metaverse: Key to next-generation electronic commerce," in *2023 IEEE International Conference on Metaverse Computing, Networking and Applications (MetaCom)*. IEEE, 2023, pp. 9–16.

[11] M. S. Kiraz, E. Larraia, and O. Vaughan, "Nft trades in bitcoin with off-chain receipts," in *International Conference on Applied Cryptography and Network Security*. Springer, 2023, pp. 100–117.

[12] G. Behrmann, A. David, and K. G. Larsen, "A tutorial on uppaal," *Formal methods for the design of real-time systems*, pp. 200–236, 2004.