A Sample Model for Integrating Decentralized Payment Systems Through Common Gateway for E-commerce

VADIMS ZILNIEKS, INGARS ERINS
Faculty of Computer Science and Information Technology
Riga Technical University
10 Zunda Embankment, Riga, LV-1048
LATVIA

Abstract: - E-commerce is constantly exploring opportunities to streamline payment service integration, particularly in terms of purchase channels and settlement methods. Traditionally, such integrations were provided through bank-acquirers. New initiatives such as Open Banking present a novel approach by offering a centralized gateway for third-party access to banking services. The purpose of this paper is to propose a sample model of a merchant gateway that leverages distributed ledger technology to enable seamless integrations with both purchase and settlement systems. The model holds the potential for accelerating purchase and withdrawal processing but also introduces new challenges that need to be addressed.

Key-Words: - distributed ledger, blockchain, finality, e-commerce, refund, cross-chain bridge

Received: March 14, 2023. Revised: September 4, 2023. Accepted: September 14, 2023. Published: September 22, 2023.

1 Introduction

Modern payment systems are relocating to a more abstract level from the perspective of end-users. Users pay by mobile banking applications or near-field communication (NFC) like ApplePay or GooglePay. They are not interested much in lower levels of operations and wish to press a button to buy a product in a real-time period. Meanwhile, merchants also seek rapid payment processing to receive funds as quickly as possible.

Today merchants are faced with heterogeneous payment platforms. They are often required to participate in a significant number of them, depending on the target groups. Additionally, integrations with internal software can be a crucial concern. That is why payment institutions are very interested in initiatives like Open Banking, [1].

The primary objective for merchants is to identify channels for receiving incoming payments and processing outgoing payments, e.g. fund withdrawal. While the banking system solved many connectivity issues, the process of establishing these channels can belong due to onboarding and acquiring workflows, the subjects of banking rules and regulatory laws.

A new player in the payment area is decentralized finances (DeFi), [2], with numerous solutions and high market capitalization as well. While DeFi is not well-regulated, it is developing at a rapid pace. As a result, it may quickly capture business niches and show greater flexibility to external factors. Although blockchain-based projects look attractive for quick solutions, they also include significant risks.

There are numerous projects of distributed financial systems, beginning with pioneering Bitcoin, [3], and continuing with Ethereum, [4], as a platform for programming financial logic through smart contracts. Public blockchain projects gained some popularity, and potential e-customers may already have an account in these networks.

The main disadvantage of these projects is the unstable financial state of the funds and the lack of financial regulation, which, is hard to attach compared with traditional finances. While the legal status of these projects is beyond the scope of this paper, merchants are entitled to take the risk of crypto fluctuations and restrictions, just as customers take the risk of having no deposit insurance.

Decentralized solutions have both negative and positive aspects. Focusing on the technical side, the positive aspects include easier account opening and faster transactions by removing intermediary actors. Negative aspects include high transaction fees due to expensive data storage in the public blockchain, [5].

One potential solution is to remove or reduce some data from the blockchain using smart contracts:

- Using off-chain data with state channels or payment channels. Refers to a technique of removing part of transactions from a blockchain to reduce fees and increase scalability;
- Side-chains or Layer 2 (L2) solutions. These are separate blockchains on a base of the main one, defined transactions are offloaded from the main to secondary chains;
- Cross-chain bridges, as mechanisms that allow transfers between separate blockchains. By transferring funds to an alternative platform, transaction fees can be potentially reduced.

L2 solutions are promising as they are secured by the underlying Layer 1 (L1). Transactions are anchored to L1, and any malicious actions can be investigated even if the side-chain is closed. There are two main methods of L1 and L2 connection: optimistic, [6], and zero-knowledge, [7]. The optimistic approach involves considering an operation as trusted and correct until someone reports it's not. This is achieved through special proofs or challenges and is characterized by a delay period, known as the challenge period, for each such operation. Another area is zero-knowledge (ZK) proofs, which provide mathematical proof of the correctness of an operation. ZK solutions are currently in an active area of research, [7], [8].

State channels slowed down after Lightning Network's, [9], popularity in the 2010s. Some argue, [10], [11], [12], that the main idea behind Lightning was not fully achieved. Only large investors and well-managed Bitcoin node groups can hold direct payment channels and utilize them for e-commerce and trade-off security issues.

This paper will investigate the cross-chain bridge option as a potential solution for the effective utilization of various distributed platforms. As will be described later, bridge technologies can provide effective use of separate platforms.

The scenario proposed in this paper involves the implementation of a private blockchain by the merchant, which features an elastic mechanism for interaction with external payment platforms depending on risk considerations.

A sample model involves a store with an automatic service (Figure 1), where customers can purchase goods by scanning a QR code. On scanning, the application adds an item to a virtual shopping cart. After the customers have added the desired goods to the shopping cart, they proceed to exit with a payment zone. There, they can choose a payment method and process the payment using the wireless terminals available. The same technology can be applied to the online store and internal or external merchant accounting networks.

Main objectives:

- high availability of the internal network (debit terminals);
- seamless integration with external payment platforms without any major architectural modifications;
- support for refunds/reversals;
- effective processing of transactions for fast purchases and fast payouts.

The use of distributed ledger technology (DLT) and trustless systems in e-commerce, addressing technical and legal issues, has been explored in the research, [13]. The regulatory landscape for DLT in e-commerce, taking into account factors such as effectiveness, compliance, and customer protection, is discussed in the work by, [14]. An overview of cashless payments in Europe and Japan is provided in the study by, [15]. The topics of Decentralized Finances, Open Banking, and Open Finances are researched by, [2].

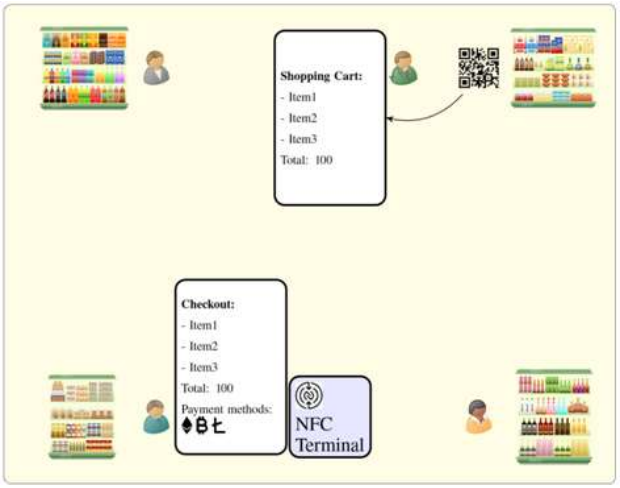


Fig. 1: Store with Wireless Payments

Structure of the paper: Section 2 provides an overview of DLT and Open Banking. Section 3 is about the integration of the external platform through a bridge. The topic of the finality problem is explored in Section 4. Section 5 presents a workflow scheme. Section 6 takes points of refunds. Finally, Conclusions with the main findings.

2 Open Banking and Distributed Ledger

Since 2018, the European Union has implemented an updated version of the single gateway concept for third-party providers (TPP) to control banking account access, known as Payment Services Directive 2 (PSD2), [1]. This allows customers to manage all their accounts through a single application, using a protocol defined by the PSD2 interface with financial institutions. It produced a new topology type for customer and bank interaction known as Open Banking (Figure 2). Some possible use cases:

- a customer wants to make an instant payment for a service and chooses a bank in his mobile application that provides such payments in the region of use. However, this process may not be user-friendly, as the customer needs to know additional information about instant payment routing;
- a customer wants to reduce banking fees and pays his bill from a different bank account.

Distributed ledger technology is an alternate concept, where participants are connected in a peer-to-peer network, and all transactions are ordered and validated through a consensus algorithm. DLT projects can be either public like Bitcoin and Ethereum, or private. In private networks, participation is restricted by predefined rules, resulting in higher trust in the network. This approach allows a selection of a consensus algorithm that provides better finality (see Section. 4).

If PSD2 enables the connection of multiple banks in a single point of access, it is worth exploring the potential of applying a similar concept for DLT projects.

The leading challenge here is the lack of standardization in DLT projects. Public projects are developing their infrastructure and are focused on attracting new participants, while private projects like R3 Corda, [16], or Consensys Quorum, [17], are working in independent groups of developers. Hence, in the absence of an equivalent to PSD for DLT platforms, it is crucial to find methods for establishing connections and managing these platforms effectively.

3 Methods to Integrate Multiple Platforms

DLT systems' integration within the same platform is commonly referred to as a Layer 2 (L2) solution. The concept originated as a scalability solution for large blockchain platforms. L2 serves as a superstructure on the top of the main blockchain, enabling the transfer of data to another chain, or side-chain. The sidechains operate similarly to the main blockchain, utilizing consensus mechanisms and blocks of transactions. Blocks can be regularly posted as proof of consistency to the main chain using smart contracts like in Plasma, [18], or in full amount to a log chain, in the case of Ethereum rollups, [6], [7].

State channels such as Lightning Network, [9], or, [19], operate by removing a portion of data from the main chain, referred to as off-chain data, as it is not replicated on the main chain. Using specific Hash Time-Lock Contracts (HTLC), two or more participants make initial funding through the main chain for their subsequent common use. Afterward, they can conduct off-chain transactions limited to this deposit. The final result will be posted to the main chain. To simplify this process for merchants, commercial hubs like Bitpay, [20], provide an API to manage off-chain payments.

One of the primary challenges in these two methods is the ability to implement a large number of channels without being strongly tied to a specific DLT technology.

Cross-chain bridge is a platform for connecting two or more independent blockchains. There are various projects, including but not limited to Polkadot, [21], Cosmos, [22], DeBridge, [23], and Rainbow Bridge, [24].

Cross-chain bridge work principle is shown in Figure 3.

On the source DLT network, a smart contract is implemented as a light client for the target DLT network. A light client in this context refers to executable smart contract code that validates the consistency of a blockchain by verifying the root hashes, without requiring to store all transaction data. The same model is implemented on the target network.

The middle agent, also known as a relayer, is responsible for transferring blocks from one network to another through the use of smart contract addresses. It depends on the block generation time. Receiving the contract then verifies the validity of the block through the light client.

The cross-chain bridge architecture demonstrates a high level of compatibility with the target model of the research.

One important issue of cross-chain solutions is the occurrence of *hard forks*, which often arise as a result of protocol or blockchain changes within a bridged DLT platform. Integration then may require the modifications of existing smart contracts or alternative solutions such as data migration to another platform to ensure compatibility. Hard forks can lead to inconsistencies in transaction history and potential vulnerability in the security of the blockchain. A possible solution here is proxy contracts, [25].

Steps of a fund transfer from one platform to another as an e-commerce purchase are enumerated in Figure 3. The external platform in the context refers to the DLT platform used by the purchaser, while the internal refers to the DLT platform of the merchant.

1. Using a mobile application (MA) Transaction *eTx_CustomerToMerchant* for the amount *eAmount_X* is initiated in the external platform *eDLT* (from the customer account *eAcc_Customer* to smart contract for locking the funds *eSmrtCntr_Locker*);
2. *eTx_CustomerToMerchant* is included in a block *eB_Y* (committed);
3. *eB_Y* header is linked to *eTx_CustomerToMerchant* in the MA to track it;
4. *eAmount_X* is locked in eDLT;
5. Relayer automatically copies *eB_Y* to *iSmrtCnt_eClient*;
6. *iSmrtCnt_eClient* verifies *eB_Y* and stores it;
7. the MA updates the transaction state of *eTx_CustomerToMerchant* by checking *iSmrtCnt_eClient* for *eB_Y header*;
8. The user through MA (or Oracle node) sends a transaction *iTx_UnlockForMerchant* to *iSmrtCnt_Minter* with a cryptographic proof of *eTx_CustomerToMerchant* locked;
9. *iSmrtCnt_Minter* checks the proof with *iSmrtCnt_eClient* and mints amount of iAmount_X* = eAmount_X to *iDLT* in a new transaction *iTx_SendToMerchant*;
10. *iTx_SendToMerchant* is proposed to the internal blockchain *iDLT*;
11. *iTx_UnlockForMerchant* is included in the internal block;
12. Merchants can be notified about successful purchases in the back office application.

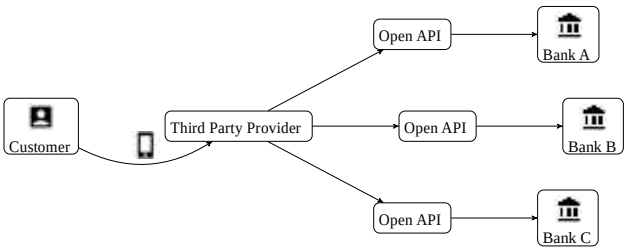


Fig. 2: Open Banking flowchart

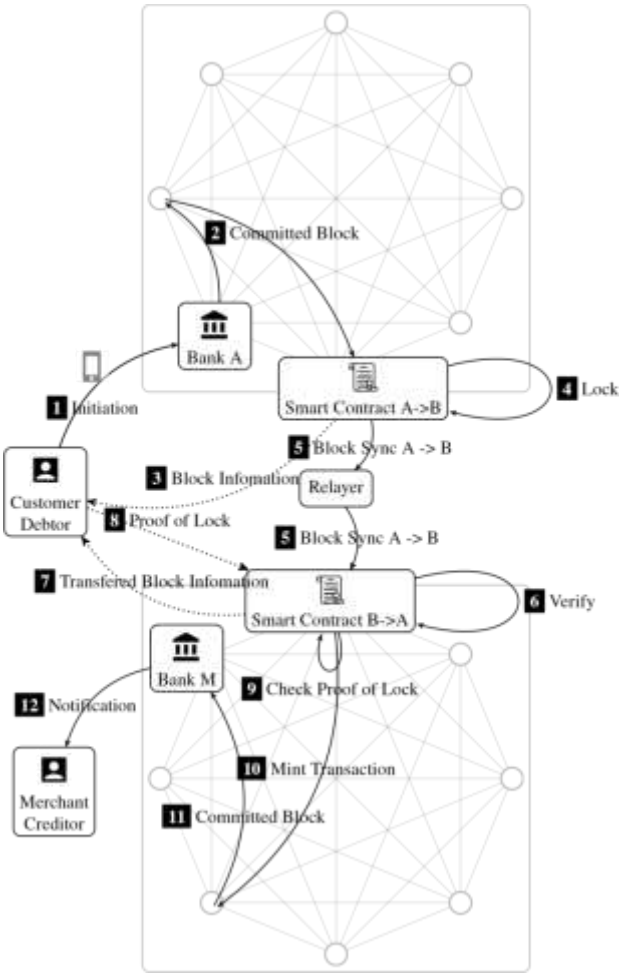


Fig. 3: Bridge scheme

Minted tokens from a public blockchain are held as equivalents of their original value (wrapped tokens). These tokens are not utilized within the internal private network, in most cases they are just burned to unlock the corresponding value on the source network in the event of withdrawal.

Steps of fund transfer from the merchant's platform to any other platform as a withdrawal:

1. Using a back-office application of merchant *Transaction iTx_MerchantToMainAccount* is proposed in the internal blockchain *iDLT* for a specified amount *iAmount_X* of minted tokens (from *iAcc_Merchant* to *iSmrtCntr_Burner*);

2. *iTx_MerchantToMainAccount* is included in the internal block *iB_Y* (committed);
3. Tokens of amount *iAmount_X* are burned in *iSmrtCntr_Burner*;
4. Relayer automatically copies block *iB_Y* to *eDLT*;
5. Merchant through back office application or automatically by oracle node sends transaction *iTx_UnlockForMainAccount* to *eSmrtCntr_Locker* with a cryptographic proof of burn *iTx_MerchantToMainAccount*;
6. *eSmrtCntr_Locker* checks the proof with *eSmrtCnt_iClient* and unlocks amount of *eAmount_X* = iAmount_X* to *eDLT* in a new transaction *eTx_SendToMainAccount*;
7. *eTx_SendToMainAccount* is proposed in *eDLT*;
8. *eTx_SendToMainAccount* is included in an *eDLT* block (committed).

The specific steps may vary depending on the bridge platform used, but the main principles remain consistent.

Up to this point, a proposed gateway serves as a connection point between separate DLT platforms, ensuring the availability of internal payment processing agents, such as terminals. The speed of transactions within this scheme is correlated with the issue of finality in DLT networks.

4 Finality in Distributed Ledger and Bridges

The term ”finality” refers to a state where all parties involved in a transaction have reached a consensus that the transaction is complete and cannot be reversed or modified. In the financial world, the issue of finality is not as pressing because of many checkpoints. As shown in Figure 4 at least three parties are involved in the transaction process: The debtor agent, the Certified authority, and the Creditor agent. Any error or rejection at any stage will decline the whole transaction. After settlement, when balances are compared with external reports, the state of all transactions can be considered as fully agreed upon.

DLT finality is of great importance due to the occurrence of forks, where multiple competing blocks are generated simultaneously. This situation creates uncertainty on which chain of blocks should be considered valid. As a result, the majority of proof-of-work-based public blockchains and their derivatives are designed to offer probabilistic finality, [26]. There is a probability that a confirmed block will become a part of the longest or most valuable chain, and still, a probability that it will be replaced by a different chain. Such forks traditionally are named *soft forks*, compared to above mentioned hard forks. Hard forks are not planned by DLT algorithms and mostly make destructive changes in a blockchain. It can be a rollback to some previous block or a change of the algorithm.

The issue of probabilistic finality can be mitigated by waiting for the next *N* blocks after the given transaction block. As the number of blocks *N* after the original transaction was committed grows, the probability of achieving finality also increases, [27], [28].

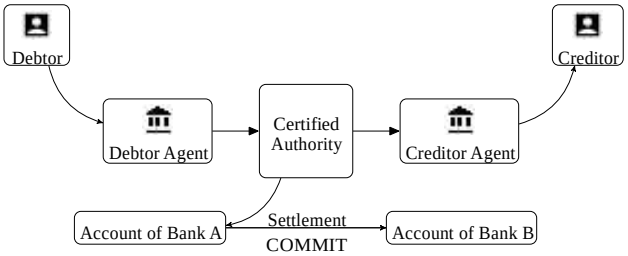


Fig. 4: Finality in Centralized Model

Figure 5 shows the number of blocks after the included transaction. The probability of an alternative fork, that can challenge the current blockchain, becomes lower with every new block, but cannot be fully declined.

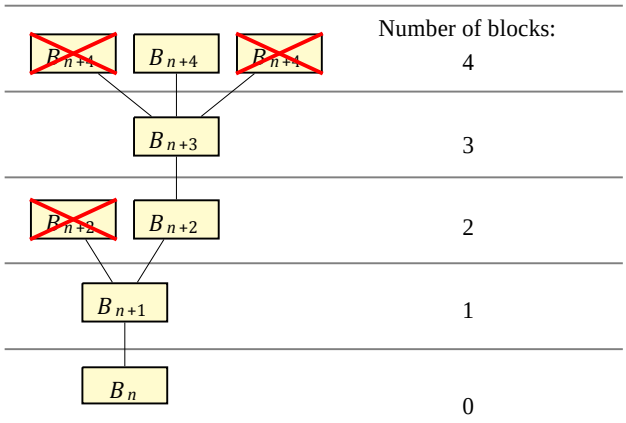


Fig. 5: DLT eventual finality

In contrast to PoW-based blockchains, other consensus algorithms, like proof-of-stake (PoS), [29], or delegated proof-of-stake (DPoS) provide deterministic or strong finality (Figure 6). Fork situations are typically resolved through voting or another form of governance, where the decision-

making is delegated to a group of validators or stalkers, [29].

For bridged blockchains, the total time of cross-chain finality will be:

$$T = t_{sp} + N_s * t_{sc} + t_r + t_v + t_p + t_{tp} + N_t * t_{tc} \quad (1)$$

where:

t_{sp} - time of a pending state of a new transaction,

N_s - number of blocks on source blockchain for eventual finality, for strong finality, equals 1,

tsc - time until source blockchain block commit,

t_r - time until the next synchronization of the relayer, t_v - time of target smart contract verification,

t_p - time of getting proof of lock or proof of burn,

t_{tp} - time of the pending state of a new transaction (mint),

N_t - number of blocks on target blockchain for eventual finality, for strong finality equals 1,

t_{tc} - time until target blockchain block commit.

The formula highlights that even in good conditions, such as having strong finality on both platforms, the process can still experience some delays. The main factor of these delays is the bridge by itself, particularly when the relayer must find an optimal frequency for block synchronization. The frequency is constrained by the time it takes for finality to be achieved on both platforms, as well as the fees associated with executing transactions. If these fees are high relative to the purchase value, it is necessary to find a balance between the minimum frequency needed for block validation and an acceptable waiting time for the transfer. Some DLT platforms require validation of every block like Ethereum, [29], while others like NEAR only require validation of one block per epoch, [30].

Table 1 (Appendix) provides an overview of the time to finality, block generation time, and average fee per transaction for some notable public blockchain projects. Most projects are using variations of PoW and PoS algorithms, with Solana being an exception as using the proof-of-history consensus, [31]. A quick analysis of the data shows variation in the time to finality compared to the block generation time.

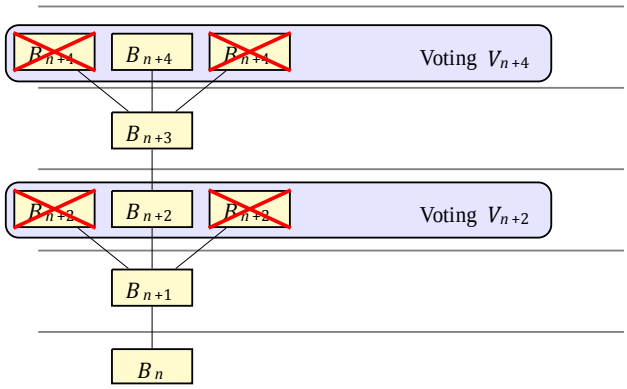


Fig. 6: DLT strong finality for voting protocols

The selection of DLT platforms for the merchant becomes crucial, as it must prioritize platforms with fast finality and low transaction fees. Otherwise, it may result in long waiting periods for transaction confirmation, potentially lasting for hours. Therefore, choosing DLT platforms and corresponding parameters can provide the desired combination of fast finality and cost-effective transactions. Deciding between integrating a popular platform with longer finality or a faster one with a smaller target group of customers can be a challenging trade-off.

5 Workflow of Bridged Gateway Solution and Security Issues

The customer selects an item from a shelf that is equipped with a QR code. Scanning the code opens a mobile application and adds the item to the customer's shopping cart.

When the customer is in the payment zone, he communicates his smartphone with a terminal, chooses the payment source, and payment is initiated. Then he waits on approval.

If an external blockchain is selected for debit, the system will wait for a relayer and a predetermined period before the next step of processing. The customer's transaction will be sent to a bridge contract address with the parameter of the merchant's account. After the source block is committed, information about the transaction will be relayed to the target DL along with appropriate block information and a proof of lock. The target transaction will also undergo a final state, depending on its network mechanism.

If the customer's and merchant's payment systems are the same, the bridge mechanism is not activated, and transactions proceed according to the appropriate DL protocol.

Challenges may arise for the transaction to be refunded after a customer has left the store due to product return, or in a case of timeout for faster platforms or later disputes. Then a refund mechanism can be activated to ensure customer satisfaction with the service (See Section. 6).

A relayer is responsible for monitoring new blocks on both DLT platforms. Upon receiving a new block, he pushes it to a smart contract deployed on a linked blockchain. The smart contract verifies the validity of the new block and stores it for future checks of transaction proofs.

The security of the proposed model heavily relies on the relayer that bridges two platforms. Since the relayer resends cryptographic secrets of locked tokens, it theoretically can access and potentially steal them. Therefore, the level of trust placed in the relayer is critical for the transaction process.

Scenarios of security improvement:

- Fraud detectors can identify any mistakes or malicious behavior in the connected platforms. In the event of proven fraudulent activity, punishment mechanisms can be activated, such as the burning of deposit tokens, a ban on the relayer, delegated blocking of the agent;
- The financial motivation of the relayer through operation fees;
- Use of commercial contracts with negotiated insurance of operations.

Fraud detectors also can be motivated by bonuses for discovering malicious actions. The first challenge is that there is no guarantee for fraud detectors to be online. The second one for a given scheme is that the target network is private, and detectors must be able to verify transactions as participants of a closed group without the benefit of fee concurrency, as seen in public DL networks. Fraud detectors perform their role for financial gain. In that case, it can be achieved using other (centralized) methods of financial motivation.

6 Refunds

Any e-commerce model needs to include a refund option. As the DLT process unites operations of purchase and settlement, reversal and refund here mean the same, and the term refund will be used further. Three possible categories of refunds within the model:

- automatic refund triggered by the system when a purchase exceeds a predetermined timeout;
- automatic refund on technical issue;
- refund by a request, performed manually.

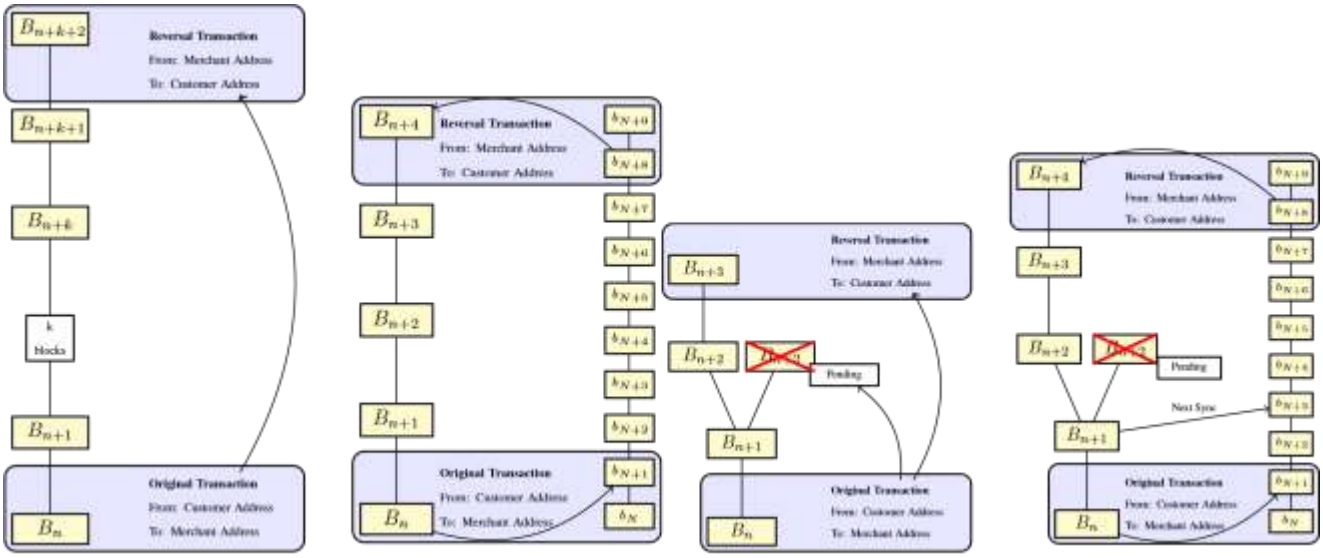
At first glance, a refund in the DLT environment is easier to process compared to traditional acquiring methods. Instead of making special requests depending on the situation, for DLT it is sufficient to change two addresses of the original transaction, initiating a new reverted transaction in the case of a single blockchain (Figure 7a). In the case of bridged systems, the new transaction involves calling a contract that links to the appropriate destination network, providing the customer's account number from the original purchase transaction (Figure 7b) *B* and *b* represent blocks of two systems.

Additionally, any fees associated with the original transaction can be included in the refund amount.

However, DLT refunds can face situations of data inconsistency. See examples in Figure 7c for a single blockchain and Figure 7d for a bridge. In the first case, the transaction is returned, but the original one was removed from a proposed block due to fork solving, and it is existing in a pending state. In theory, it can be in that state till the refund is processed and added to the block. In the second case, a fork situation in a source platform as well, but the target platform synchronized the wrong block, and a refund also can potentially get to the customer’s address faster than the original.

Incorrect block synchronization is a valid concern. As mentioned above, the relayer is playing a crucial role and can be considered as a potential weak point. One approach to addressing this issue is the optimistic model when it is initially trusted by default but can be punished by watching agents. Another approach is to establish a form of central governance to ensure the reliability of relayers.

To improve the situation with refund consistency on a single DLT platform, the use of specific smart contracts can provide a solution. It can be a gateway that handles all transactions for the merchant. When a refund is requested, the smart contract can verify the original transaction in its records. If the transaction is found, then it can be processed.



(a) Refund in DLT platform (b) Cross-platform Refund (c) Wrong Refund in DLT platform (d) Wrong Cross-platform Refund

Fig. 7: Refunds

Sample smart contract logic for a refund (*SmrtCnt_Storage*):

1. Purchase transaction goes to address of *SmrtCnt_Storage*;
2. After being added to the block, the transaction is added to *SmrtCnt_Storage* memory by emitting a unique identifier to the customer's application and the merchant's back-office system;
3. The transaction goes to the merchant's account;
4. In the case of refund, the customer through the application executes a method of *SmrtCnt_Storage* with the identifier of the original transaction to ensure its inclusion and finalization;
5. If the transaction is found, then *SmrtCnt_Storage* produces reversal operation;
   If not found, then there can be additional logic for saving this identifier for cases the transaction is not committed yet and will be refunded automatically, or to report that it's not found.

In the case of a bridge scheme, this logic can be integrated into the existing sequence of smart contracts for registering the original transaction. After registering, the transaction continues processing and transferring to a target platform. If a refund is requested and the original transaction is found in the contract transaction list, the funds can be unlocked and returned. This verification ensures that only valid transactions are used for the refunds within the bridge scheme, as transaction inclusion in the list is possible only after block commitment. For blockchain forks, consistency is maintained through smart contract verification, which addresses finality issues discussed in Section 4.

7 Conclusions

Today's advancements in technology enable the creation of innovative business concepts, one such concept is the use of cross-blockchain bridging as an alternative to Open Banking. Bridging can serve as a gateway to connect with external distributed ledger platforms that provide essential functions for e-commerce requirements including purchases, refunds, and settlement in a single atomic process.

By utilizing bridges, businesses can establish a link between their e-commerce platform and external DLT platforms, allowing seamless integrations through modifying only bridges, not a system.

There are several open challenges that need to be addressed in this solution. One of them is achieving deterministic finality for popular public blockchains, as PoW based blockchains provide probabilistic finality. Additionally, the time it takes to achieve finality and associated transaction fees play a significant role in the overall model's efficiency.

Trust in the bridge relayers is a fundamental issue since the relayers have access to the cryptographic secrets of locked funds. There are several potential methods to enhance security, including the implementation of fraud detectors, providing financial motivation for the relayers, and

introducing insurance for operations. While insurance may increase centralization within, it is not an objective of the proposed model.

However, these issues must be considered when designing new e-commerce systems with bridge technology. The provided schema is limited by the need for manual analysis of payment systems to be integrated. Particularly, the long processing networks need to be thoroughly examined to ensure that the total time of bridged transactions remains acceptable for merchants.

Standardization can have a significant impact in this context, akin to the impact of PSD2, which facilitated streamlined integrations between financial institutions and service providers. The presence of standardized protocols in DLT solutions, particularly in their intercommunication, could give comparable advantages, analogous to those observed in the domain of Open Banking.

Even in the current situation, we consider the concept of private blockchain with a bridge to external DLT platforms hold great promise for the future. That process is not dependent on the limitations of public blockchains, as the bridge can be updated to any new DLT platform, whether public or private. As technology continues to advance, such solutions are likely to become more robust and reliable for a new era of secure and efficient e-commerce transactions.

References:

[1] European Parliament. Consolidated text: Directive (eu) 2015/2366 of the european parliament and of the council of 25 november 2015 on payment services in the internal market, amending directives 2002/65/ec, 2009/110/ec and 2013/36/eu and regulation (eu) no 1093/2010, and repealing directive 2007/64/ec (text with eea relevance), 2015.

[2] Dirk A. Zetzsche, Douglas W. Arner, and Ross P. Buckley. Decentralized finance. *Journal of Financial Regulation*, 6, p.172–203, 9 2020.

[3] Satoshi Nakamoto. Bitcoin: A peer-to-peer electronic cash system. 2009.

[4] Gavin Wood. Ethereum: a secure decentralised generalised transaction ledger. *Ethereum Project Yellow Paper*, p.1–32, 2014.

[5] Giuseppe Antonio Pierro and Henrique Rocha. The influence factors on ethereum transaction fees. pages 24–31. Institute of Electrical and Electronics Engineers Inc., 5 2019.

[6] Ethereum Foundation. Optimistic rollups, 2023. Available online: https://ethereum.org/en/developers/docs/scaling/optimistic-rollups/

[7] Ethereum Foundation. Zero-knowledge rollups, 2023. Available online: https://ethereum.org/en/developers/docs/scaling/zk-rollups/

[8] Yuri Bespalov, Alberto Garoffolo, Lyudmila Kovalchuk, Hanna Nelasa, and Roman Oliynykov. Probability models of distributed proof generation for zk-snark-based blockchains. *Mathematics*, 9, 12 2021.

[9] Joseph Poon and Thaddeus Dryja. The bitcoin lightning network: Scalable off-chain instant payments, 2016.

[10] Cosimo Sguanci and Anastasios Sidiropoulos. Mass exit attacks on the lightning network. 8 2022. Available online: http://arxiv.org/abs/2208.01908

[11] Antoine Riard and Gleb Naumenko. Timedilation attacks on the lightning network. 6 2020. Available online: http://arxiv.org/abs/2006.01418

[12] Joshua Henslee and Zach Resnick. Why the lightning network doesn't work, 2022. Available online: https://unboundedcapital.com/blog/why-lightning-doesnt-work

[13] Horst Treiblmaier and Christian Sillaber. The impact of blockchain on e-commerce: A framework for salient research topics. *Electronic Commerce Research and Applications*, 48, 7 2021.

[14] Gustavo Guarín Duque and Julián David Zuluaga Torres. Enhancing e-commerce through blockchain (dlts): The regulatory paradox for digital governance. *Global Jurist*, 20, 7 2020.

[15] Markus Heckel and Franz Waldenberger. *The Future of Financial Systems in the Digital Age. Perspectives from Europe and Japan*. 2021.

[16] R3. Capitalize on the new digital economy—transact openly and securely, at scale, 2023. Available online: https://r3.com/products/corda/

[17] Consensys. A trusted, open source foundation for your blockchain solution., 2023. Available online: https://consensys.net/quorum/

[18] Joseph Poon and Vitalik Buterin. Plasma: Scalable autonomous smart contracts. *Whitepaper*, p.1-47, 2017.

[19] Raiden. What is the raiden network? 2020. Available online: https://raiden.network/101.html
[20] bitpay. Bitpay supports lightning network payments: Accept and make cheap, scalable transactions, 2022. Available online: https://bitpay.com/blog/bitpay-supports-lightning-network-payments/
[21] Gavin Wood. Polkadot: Vision for a heterogeneous multi-chain framework, 2016.
[22] Jae Kwon and Ethan Buchman. A network of distributed ledgers, 2019.
[23] deBridge. dedocs: Protocol overview. 2023. Available online: https://docs.debridge.finance/the-core-protocol/protocol-overview
[24] Aurora. How the rainbow bridge works, 2021. Available online: https://aurora.dev/blog/2021-how-the-rainbow-bridge-works
[25] Mobin Hajizadeh. Upgrading smart contracts, 2023. Available online: https://ethereum.org/en/developers/docs/smart-contracts/upgrading/
[26] Christian Decker and Roger Wattenhofer. Information propagation in the bitcoin network. *13th IEEE International Conference on Peer-toPeer Computing, IEEE P2P 2013 - Proceedings*, 2013.
[27] Emmanuelle Anceaume, Antonella Del Pozzo, Thibault Rieutord, and Sara Tucci-Piergiovanni. On finality in blockchains. volume 217. Schloss Dagstuhl- Leibniz-Zentrum fur Informatik GmbH, Dagstuhl Publishing, 2 2022.
[28] Vitalik Buterin and Virgil Griffith. Casper the friendly finality gadget. 10 2017.
[29] Luca Pennella. Proof-of-stake (pos), 2023. Available online: https://ethereum.org/en/developers/docs/consensus-mechanisms/pos/
[30] Near Foundation. Eth-near rainbow bridge, 2020. Available online: https://near.org/blog/eth-near-rainbow-bridge/
[31] Anatoly Yakovenko. Solana: A new architecture for a high performance blockchain v0.8.13, 2018. Available online: https://solana.com/solana-whitepaper.pdf

APPENDIX

Table 1. DLT Projects. Parameters for Finality and Bridging

Project	Consensus Type	Block generation time	Average time to finality	Transaction Fee
NEAR	PoS	1 s	3 s[a]	low
Avalanche	PoS	2 s	1.3 - 3.4 s	low
Polygon PoS Chain	PoS	2.3 s	5 min[b]	low
Polkadot	PoS	6 s	12 s - 60 s	low
Bitcoin	PoW	10 min	60 min[c]	high
Ethereum	PoW	15 s	90 s[d]	high
Ethereum 2.0	PoS	12 s	6 min - 12 min[e]	high[f]
Algorand	PoS	1 s	4 s - 5 s	low
Solana	PoH	0.4 s	2.3 s - 46 s	low

[a]three confirmations [b]128 confirmations [c]six confirmations [d]six confirmations epoch-based finality [f]planned lower than in the Ethereum (PoW)

Contribution of Individual Authors to the Creation of a Scientific Article (Ghostwriting Policy)

- Vadims Zilnieks: Conceptualization, Data curation, Investigation, Methodology, Writing - original draft preparation.
- Ingars Erins: Supervision, Conceptualization, Validation, Reviewing, and Editing.

Sources of Funding for Research Presented in a Scientific Article or Scientific Article Itself

No funding was received for conducting this study.

Conflicts of Interest

The authors have no conflicts of interest to declare.

Creative Commons Attribution License 4.0 (Attribution 4.0 International , CC BY 4.0)

This article is published under the terms of the Creative Commons Attribution License 4.0
https://creativecommons.org/licenses/by/4.0/deed.en_US