Building Interoperability: A Decentralized Bridge Connecting Polkadot and Cosmos Ecosystems

Kristián Košťál, Dušan Morháč, Juraj Mečír
Faculty of Informatics and Information Technologies
Slovak University of Technology
Bratislava, Slovakia
{kristian.kostal,dusan.morhac,xmecir}@stuba.sk

***Abstract*—Cross-chain interoperability remains a critical challenge in blockchain technology, especially as the number of decentralized networks continues to rise. Each blockchain operates in isolation, creating an urgent need for solutions that enable seamless interactions across these networks. This paper presents a Proof-of-Concept (PoC) solution for bridging interoperability between the Polkadot and Cosmos multi-chain ecosystems. Currently, no existing solution facilitates secure cross-chain interactions between these platforms. Our approach addresses this gap by utilizing a hashed timelock mechanism, introducing a modular and reusable framework that can be deployed in fully operational networks. Built on smart contracts and the Substrate framework, the proposed solution relies on relayers - trusted, incentivized, and fairly selected API servers - to manage communication and proof exchange between chains. This framework serves as a foundational step toward developing robust cross-chain solutions for integrating Polkadot, Cosmos, and similar networks.**

***Index Terms*—Blockchain, Ledger, Interoperability, Cross-chain, Hashed timelock mechanism**

I. Introduction

The term blockchain is often associated with Bitcoin, introduced in 2008 by Satoshi Nakamoto [1]. Blockchain is, however, far broader than just Bitcoin. Blockchain has many helpful use cases in various industries - such food industry [2], [3], [4], [5], healthcare [6], [7] or finance [8], [9]. While blockchain use cases are broad, the industry still leans towards blockchains for cryptocurrency or as an investment. As the industry creates more specific blockchains covering various needs, the question" How can we make these heterogeneous blockchains communicate with each other?" appears more frequently. The answer to this question is often very complex and lies in blockchain interoperability, a broad topic summed into two words.

Let's first begin by explaining what the term "blockchain" means. Blockchain has become very popular thanks to its benefits, such as immutable storage, so once something is stored on the blockchain, it remains on it for the remainder of the chain's lifetime. Blockchain stores data in blocks, chained using cryptography, hence the name "blockchain." Every block except for the first block, called "Genesis block," stores the previous block's hash, the timestamp of block creation, and a list of transactions along with details. There are only two ways to change data in blockchain:

- **Updating data** - This method does not remove original data from the chain only updates the pointer to the latest data. The original data can still be easily found.
- **Consensual agreement of majority** - The majority of the network must agree to change the data. This method is used for network updates or other governance decisions.

Another benefit of blockchain is decentralization. Decentralization mitigates the need for a central authority to decide what is and is not truthful. Blockchain can determine truthful information automatically by using consensual algorithms. The network's consensus about whether a particular block belongs to it is reached when most of it agrees. There are multiple variations of block validation consensus algorithms, such as:

- **Proof of Work (PoW)** - The most popular consensus algorithm where entities called miners have to solve cryptographic puzzles.
- **Proof of Stake (PoS)** - More environmentally friendly solution compared to PoW. Instead of miners, this algorithm incorporates block validators that are chosen based on certain parameters.
- **Nominated Proof of Stake (NPoS)** - Modified version of PoS algorithm where validators are nominated to be fairly chosen by the community. The community is incentivized to nominate non-malicious validators to receive nomination rewards.

An additional term used in the context of the blockchain is the term "smart contract." Smart contracts are virtual machines that execute code and return output based on input parameters and conditions set in code. Because smart contracts are stored on the blockchain, anyone (even their owner) cannot remove or modify them without network block consensus. There are smart contracts that do not have owners and are self-sufficient. These smart contracts can be considered trusted third parties between non-trusting chains [10].

As the number of sovereign blockchains grows, the need for communication arises. That is why chain developers have been focusing on creating interoperability between different chains for some time now. Having seamless interoperability between multiple chains enables various helpful use cases.

Multi-chain networks such as Polkadot [11] or Cosmos [12] have their interoperability protocols. But can we connect these two multi-chain networks reliably and trustlessly?

Currently, there is no complete solution that addresses this matter. This raises an important question: Is achieving chain-agnostic interoperability within multi-chain ecosystems feasible? Connecting two sovereign chains is relatively straightforward; however, bridging multi-chain ecosystems introduces significant challenges due to the need to address a wide range of differences—some of which are architectural—making implementing chain-agnostic solutions more complex. For example, Polkadot does not natively support the Ethereum Virtual Machine (EVM), relying instead on the Substrate framework, complicating connections. Conversely, Cosmos supports EVM-compatible smart contracts, facilitating easier integration with other EVM-compatible chains.

The following paper focuses on this problem and tries to address it with a new complex, robust, secure, and chain-agnostic solution featuring a state-of-the-art relaying service design that preserves decentralization.

The paper is organized into the following sections:

- **Background** - The following section discusses the problem more thoroughly. The section also addresses Polkadot, Cosmos, interoperability, and potential use cases of interoperability between them.
- **Solution design** - The design section addresses decisions and features introduced to the solution and the solution architecture overview.
- **Solution evaluation** - The following section evaluates solution design and performance.
- **Conclusion** - The conclusion section discusses results, summarizes the solution contribution and concludes the article.

II. Background

The previous section summed up blockchain technology in simple terms. The following section discusses the interoperability principles relevant to the proposed solution and provides a detailed introduction to the Polkadot and Cosmos ecosystems.

A. Interoperability

Term interoperability can be translated as exchanging information or transactions between two or more blockchains. Achieving interoperability between blockchains that differ in core design principles, such as consensus, can be a tough challenge. Origin chains cannot access storage on destination chains and vice versa. This limits the full potential of cross-chain communication that could otherwise be much more advanced between them. As study [13] points out, the lack of interoperability proposes a series of constraints that prevent users from having a seamless experience. According to sources such as [14], [15] or [16] there are multiple use cases for interoperability for example:

- **Asset transfers** - Fungible or non-fungible asset transfers between different chains. If something on chain A costs currency B from chain B, the user can easily route currency B to chain A.
- **Cross-chain oracles** - Smart contract on chain A reads a new event for smart contract on chain B. It executes the same action or action according to input parameters from reading the event.
- **Cross-chain asset encumbrance** - The assets for a specific account on chain A are locked, and the same conditions are applied to chain B

According to studies [17] and [15] cross-chain solutions can be classified into following categories:

- **Notary schemes** - The most straightforward technological method for facilitating cross-chain operations is notary mechanisms. In a notary mechanism, transactions rely heavily on a trusted entity or a group of trusted entities. When the parties involved in a transaction do not trust each other, a trusted intermediary is introduced. This intermediary ensures that one ledger's state reflects another's corresponding actions. If a group of entities serves as the intermediary, they reach decisions through Byzantine Fault Tolerance (BFT) consensus [15].
- **Sidechains and Relays** - According to a formal definition [21], a sidechain is a blockchain that validates data on other blockchains. Typically, a smart contract can access data from another ledger because a partial copy of that ledger is stored on the ledger where the smart contract resides. Notably, there is no need for a third-party interface, as the two ledgers communicate directly. There are two types of relays: one-way relays, where Ledger A can read from Ledger B but Ledger B cannot read from Ledger A, and two-way relays, where both Ledger A and Ledger B can read from each other [16].
- **Hash-locking** - a widely recognized technique for facilitating cross-chain atomic operations requiring minimal inter-chain awareness. In this approach, it is sufficient for the blockchains to exchange only a single hash [15]. A basic example of this mechanism is a cross-chain asset exchange between parties A and B [16]:
  1) Party A generates a random secret s and computes its hash, $\text{hash}(s) = h$. Party A then sends h to Party B.
  2) Both Party A and Party B lock their assets into smart contracts under the following conditions: Party A locks their asset first, followed by Party B after verifying that Party A's asset has been successfully locked. On Party A's side, if the secret is provided within $2X$ seconds, the asset is transferred to Party B; otherwise, it is returned to Party A. On Party B's side, if the correct secret (i.e., the value whose hash is h) is provided within X seconds, the asset is transferred to Party A; otherwise, it is returned to Party B.
  3) Party A reveals the secret within X seconds to claim the asset from Party B's contract. Consequently, Party B learns the secret and can claim the asset from Party A's contract.

B. Polkadot

Polkadot represents a scalable, heterogeneous multi-chain framework featuring a shared security model [17]. It provides a foundational Relay Chain, which supports numerous validatable, globally coherent dynamic data structures concurrently. These data structures are referred to as ”parallelized” chains or Parachains. The native currency of Polkadot is called DOT, and it is utilized for transaction fees and staking validators on the Relay Chain [11].

The Relay Chain serves as the central chain within the Polkadot network. All validators within Polkadot are staked on the Relay Chain in DOT and perform validation tasks for the Relay Chain. The Relay Chain encompasses a relatively limited set of transaction types, including interactions with the governance mechanism, Parachain auctions, and participation in the Nominated Proof-of-Stake (NPoS) system. Its functionality is intentionally minimal; for example, it does not support smart contracts. The Relay Chain's primary role is coordinating the overall system, including Parachains, while specific tasks are delegated to the Parachains, which feature diverse implementations and capabilities [11].

Parachain is a data structure that maintains global coherence and can be validated by Relay Chain validators. While Parachains typically take the form of blockchains, they are not required to be blockchain-based. Due to their parallel nature, Parachains facilitate concurrent transaction processing, enhancing the Polkadot network's scalability. Parachains participate in the network's shared security and communicate with one another via the Cross-Chain Message Passing (XCMP) protocol [18], [19].

Parathread allows blockchains to connect to the Polkadot network without acquiring a dedicated Parachain slot. This is achieved by sharing a Parachain slot among multiple Parathreads. This arrangement enables blockchains that cannot afford a Parachain slot or find i t e conomically u nfeasible to connect to the Polkadot network [19]. The Polkadot network's architecture can be summed up in Fig. 1.

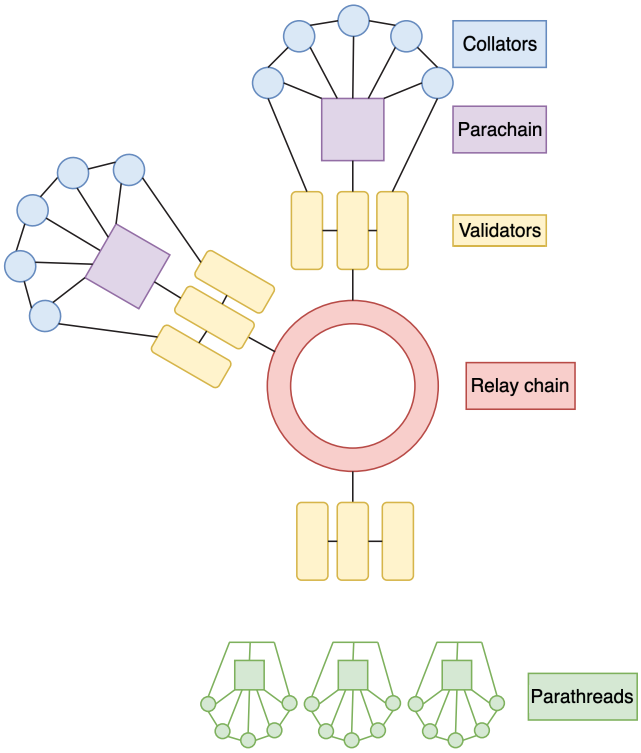

Fig. 1. High-level Polkadot architecture overview

C. Cosmos

Cosmos is a decentralized network consisting of independent parallel blockchains known as zones. Eaczonene employs a Byzantine Fault Tolerance (BFT) consensus algorithm. The central zone in the Cosmos network is the Cosmos Hub, which connects various other blockchains (zones) using the Inter-Blockchain Communication (IBC) protocol [20]. Tendermint BFT powers zones [21], offering a high-performance, consistent, secure consensus mechanism similar to Practical Byzantine Fault Tolerance (PBFT). Token transactions between zones always pass through the Cosmos Hub, which records each zone's total number and types of tokens. This setup isolates zones from failures in other zones. The native currency within Cosmos is Atom, which is used for paying fees and staking validators [12].

The Cosmos Hub manages numerous independent blockchains, also referred to as zones. It continuously receives block commits from these zones, enabling it to track their current statuses. Similarly, each zone monitors the status of the Hub but does not track the statuses of other zones. Information packets are transferred between zones using Merkle proofs to verify that the information has been sent and received. This transfer mechanism is called Inter-Blockchain Communication (IBC) [20]. The architecture of the Cosmos network can be observed in Fig. 2.

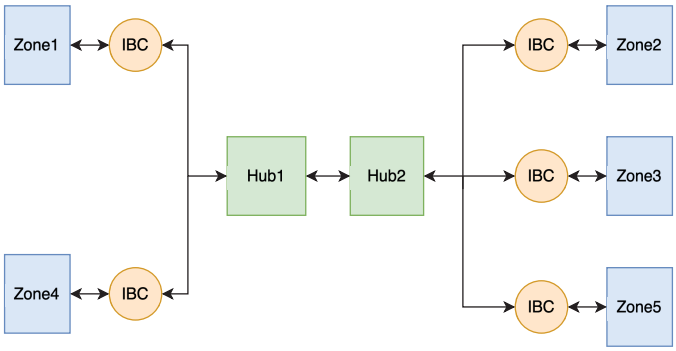

Fig. 2. High-level Cosmos architecture overview

D. Axelar

While there is currently no direct, seamless solution for connecting Polkadot and Cosmos, specific workarounds allow users to transfer assets between these ecosystems. For instance, Cosmos's support for EVM and Polkadot's EVM-compatible Parachain, Moonbeam, enable asset exchanges through protocols such as Axelar [22]. This approach leverages smart

contracts deployed on each chain, which function as gateways to route assets through a central network and forward them to the intended destination. While this method is well-suited for EVM-compatible chains, it does not extend support to Substrate, the foundational framework of Polkadot. Consequently, options for connecting Polkadot's Substrate-based chains with other ecosystems remain limited. Axelar's high-level architecture overview can be seen in Fig. 3

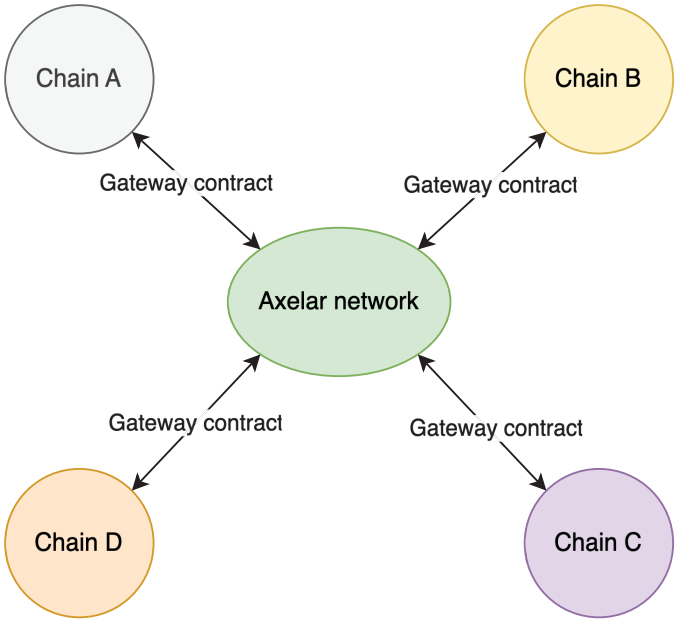

Fig. 3. High-level Axelar architecture overview

To our knowledge, there is currently no way to transfer assets between Cosmos and Polkadot.

III. Design

As mentioned in previous sections, this paper proposes a solution for a trustless and decentralized bridge between two multi-chain networks called Polkadot and Cosmos. There are certain design principles the proposed solution needs to follow:

- Solution needs to be simple to implement into existing and running chains without interfering with how the network works.
- Solution must allow for secure and decentralized exchange of fungible and non-fungible assets.

A. Solution architecture overview

The system architecture is composed of two main components:

- **Relayer component** - A component that monitors events on both networks and facilitates communication between them.
- **RPC client component** - Front-end web application that provides a user interface for seamless interaction with the system.

These components run simultaneously and control state on both networks. The network configuration in this case is:

- **Polkadot network** - Polkadot Network can include multiple chains; for the sake of simplicity, our configuration consists of a Relay Chain and a single Parachain, referred to as BridgeChain. The Relay Chain's primary responsibility is to maintain and secure the Polkadot Network, while BridgeChain is designed to implement the logic necessary for bridging asset transfers.
- **Cosmos** network - Cosmos network consists of a similar architecture to Polkadot, functioning as a Cosmos hub with one Cosmos zone that implements innovative contract capabilities.

The high-level architecture overview of the solution can be seen in Fig. 4.

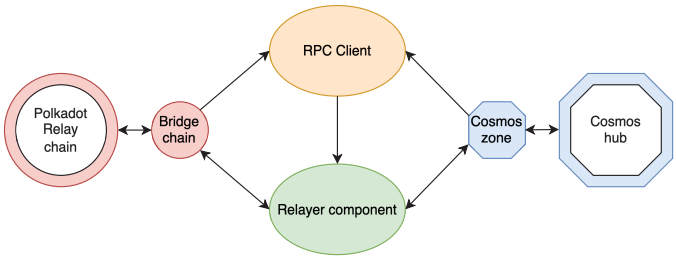

Fig. 4. High-level bridge architecture overview

B. Polkadot Network implementation

As previously mentioned, the Polkadot network consists of the Relay Chain and the BridgeChain. The solution utilized in implementing BridgeChain is the Substrate framework. Substrate, developed by Parity Technologies, is a blockchain framework that offers a comprehensive toolset for creating custom blockchains from scratch. The Substrate provides a comprehensive template pallet to establish the basic logic easily extended for cross-chain transfer. This approach allows for easy integration of our solution into existing blockchains as quickly as adding a pallet. The Relay Chain ensures the proper functioning and security of the Polkadot network, while all the asset transfer logic is implemented on BridgeChain within a Substrate pallet. The decision to implement the logic in a Substrate pallet was made due to the reusability of pallets, which can be seamlessly integrated into any Substrate-based chain.

On BridgeChain, there are two types of accounts:

- **Regular User Accounts** - These accounts can trade assets within the Polkadot network and transfer assets to the Cosmos network.
- **Vaults** - Accounts that facilitate cross-chain asset transfers. Vaults offer their assets for trade with assets on other networks. For each successful cross-chain token transfer they enable, vaults are rewarded with tokens on the Polkadot network. Vaults must ensure they have sufficient assets for trade, as offering more tokens than they possess will result in penalties. Any regular user can become a vault, and vaults can revert to regular users. Additionally, vaults are expected to operate a Relaying service node, as it is economically beneficial.

The logic implemented in the pallet on BridgeChain provides the following functionality:

- **Vault Functionality** - This allows a regular user to become a vault and vice versa. Users can initiate a transaction to become a vault by providing their Cosmos chain address and specifying the number of tokens available for cross-chain transfers on both networks. The vault owner guarantees the availability of the declared tokens on both chains. If this guarantee is not met, the vault is penalized (slashed), and the account is reverted to a regular user. Similarly, a user can issue a transaction to stop being a vault, provided there is no active cross-chain swap, and the account will revert to a regular user. Evervaultlt gets a certain percentage of the transaction and rewards for providing liquidity. Vaults are incentivized to invest more to get better rewards. The higher liquidity they bring, the higher their earnings are.
- **Cross-Chain Asset Transfer** - This functionality enables asset transfers between the Cosmos and Polkadot networks using hashed timelocks. Transferring assets between these chains and the associated transactions are detailed later in the paper. The properties of hashed timelocks ensure the atomicity of these transfers.

C. Cosmos Network

The Cosmos network setup comprises a single Hub and Zone with innovative contract capabilities. The solution utilizes Gaia and the Cosmos SDK to create a local testnet of the Cosmos Hub and Zone with smart contracts enabled. The bridge smart contract is written in Rust using CosmWasm. This smart contract can be deployed on any Cosmos chain that supports CosmWasm smart contracts. A smart contract is deployed ozonene and takes care of hashed time-locking logic. To initialize the smart contract, the caller locks tokens into the contract using a secret for a specified time window, determining the tokens' recipient. The recipient must reveal the secret within the given time window to claim the tokens. If the recipient does not claim the tokens, the contract can be canceled after the time window expires.

Unlike BridgeChain, the Cosmos network user must have only a regular user account. These accounts can trade assets within the Cosmos network and transfer assets to the Polkadot network.

D. Relayer Component

The Relayer component is an API server that intermediates communication between the Cosmos and Polkadot networks. A vault is expected to operate this component, as it acts as a vault when transferring funds between these networks. The Relayer component listens for events on both chains. It attempts to complete as many cross-chain transfers as possible since completing a cross-chain transfer is economically advantageous for the vault.

E. Asset transfer from Polkadot to Cosmos

The asset transfer from Polkadot to Cosmos chain happens in the following order:

1) The user selects the number of tokens they want to transfer and sends it along with their Cosmos address and hash h to BridgeChain. Hash h is created from secret s using the BlakeTwo256 hash function. This is triggered by initiating the Polkadot-to-Cosmos transfer transaction on BridgeChain. Once initiated, a suitable vault is found, the specified amount of the user's tokens is locked, and an event about this transaction is emitted.
2) After the event is emitted, the vault locks its tokens with hash h on the Cosmos chain in a smart contract. Thvaultlt also designates the user's Cosmos address as the recipient of the locked tokens. Once this transaction is completed, the Cosmos chain emits an event.
3) After the vault locks its tokens in the smart contract, the user can claim the tokens on the Cosmos chain by revealing secret s.
4) After secret s is revealed, the vault claims the user's locked tokens on BridgeChain by issuing a claim tokens transaction. This transaction also updates the vault's available token balance for cross-chain swaps.

F. Asset Transfer from Cosmos to Polkadot

To initiate a transfer from Cosmos to Polkadot, the user has to do the following steps:

1) The user selects the number of tokens they want to transfer from the Cosmos chain. This is initiated by the Cosmos-to-Polkadot transfer transaction on BridgeChain. A suitable vault is found, and an event about this transaction is emitted.
2) After the event is emitted, the vault locks its tokens with hash h on BridgeChain through the lock tokens transaction. Hash h is created from secret s using the BlakeTwo256 hash function. This transaction emits an event containing the vault's address on the Cosmos chain.
3) After the event is emitted, the user locks their tokens with hash h on the Cosmos chain in a smart contract and designates the vault's Cosmos address as the recipient of the locked tokens. Once this transaction is completed, the Cosmos chain emits an event.
4) After the user locks their tokens in the smart contract, the vault claims the user's tokens on the Cosmos chain by revealing secret s.
5) After secret s is revealed, the user claims the vault's locked tokens on BridgeChain by issuing a claim tokens transaction. This transaction also updates the vault's available token balance for cross-chain swaps.

G. Security Considerations

Security is a paramount concern in designing and implementing blockchain bridges, particularly given the history of hacking incidents that have plagued similar systems. While our proposed solution emphasizes secure cross-chain interactions, conducting a comprehensive security evaluation to substantiate these claims is crucial. This evaluation addresses potential vulnerabilities that could be exploited in cross-chain systems.

One of the primary threats to the proposed bridge arises from the relayer component. If the relayer malfunctions or is compromised, it could lead to significant security risks, including the potential for asset loss or unauthorized access to user funds. We recommend implementing robust monitoring and failover mechanisms for the relayer to mitigate this risk. The bridge includes redundancy measures, where multiple relayers operate in parallel, ensuring that others can take over without disrupting service if one fails.

Additionally, while beneficial for ensuring atomicity in cross-chain transactions, hashed timelocks may introduce vulnerabilities under network congestion conditions. In scenarios where the network is heavily loaded, the time constraints imposed by hashed timelocks could lead to transaction failures, resulting in a poor user experience and potential asset lock-up. To address this issue, the bridge incorporates dynamic time windows that adjust based on network conditions, allowing for greater flexibility and reducing the likelihood of transaction failures.

IV. Security Analysis

Ensuring the security of cross-chain interoperability solutions is paramount, as blockchain bridges have historically been targeted by adversaries. This section provides a comprehensive analysis of the security aspects of our proposed Cosmos-to-Polkadot bridge, identifies potential attack vectors, and proposes mitigation strategies to enhance its robustness.

A. Threat Model and Adversarial Conditions

Our threat model assumes the presence of various adversarial conditions, including but not limited to the following:

- **Malicious Relayers:** Since the proposed design relies on relayers for cross-chain communication, an adversary could attempt to manipulate or withhold messages, resulting in transaction failures or fund lock-ups.
- **Front-running Attacks:** Attackers could observe transactions and execute them with higher priority, potentially gaining unfair advantages or causing transaction failures.
- **Replay Attacks:** A previously valid message could be reused maliciously, leading to unintended double-spending or incorrect state updates.
- **Time-lock Manipulation:** Attackers could exploit network congestion to delay time-sensitive transactions, leading to asset losses.
- **Smart Contract Vulnerabilities:** Bugs in the bridge's smart contracts could allow unauthorized withdrawals or fund leakage.
- **Economic Attacks:** Adversaries could attempt to manipulate vault incentives, creating instability in the bridge mechanism.

B. Mitigation Strategies

To counter these threats, our solution integrates several security enhancements inspired by established best practices in blockchain interoperability:

1) Relayer Redundancy and Decentralization: To mitigate risks associated with malicious or compromised relayers, our system adopts a decentralized relayer model:

- Multiple relayers operate in parallel to prevent a single point of failure.
- A cryptographic proof mechanism ensures that relayers cannot modify transaction data.
- Economic incentives penalize malicious behavior by slashing misbehaving relayers' stakes.

2) Front-running Prevention: To mitigate front-running attacks, our approach implements the following safeguards:

- Transactions utilize commit-reveal schemes to obscure crucial details until execution.
- Fee structures and gas bidding strategies minimize front-running opportunities.

3) Replay Protection: To prevent replay attacks, we incorporate:

- Nonces for each cross-chain transaction to ensure uniqueness.
- Chain-specific identifiers to prevent transactions from being reused across different networks.

4) Dynamic Time-lock Adjustments: Since hashed time-lock contracts (HTLCs) can be vulnerable to network congestion issues, we implement:

- Dynamic expiration windows that adjust based on network conditions.
- A fallback mechanism allowing users to reclaim funds if a time-lock expires.

5) Smart Contract Security: All smart contracts deployed in our solution undergo rigorous security checks, including:

- Formal verification and static analysis tools to detect vulnerabilities.
- Bug bounties and third-party audits to identify and address weaknesses before deployment.

6) Economic Security and Vault Stability: To prevent economic attacks targeting vaults, our design includes:

- Collateral requirements for vault operators to ensure honest participation.
- Dynamic fee adjustments that respond to network congestion and vault availability.
- Reputation-based scoring for vaults, rewarding those with a history of secure and efficient operation.

V. Evaluation

The system was developed and tested on a local network consisting of 100 nodes, simulating real-world performance. Test cases focused on determining the overall performance of both chains and the overall performance of the Relayer component.

A. BridgeChain benchmarking

BridgeChain supports various types of transactions, ranging from vault logic to transactions that enable the cross-chain transfer of tokens. To assess the performance of this chain,

the focus was on testing transactions that initiate cross-chain swaps of assets. In total, the test triggered 200 transactions that initiated a cross-chain swap. The measured time results can be seen in Fig. 5.

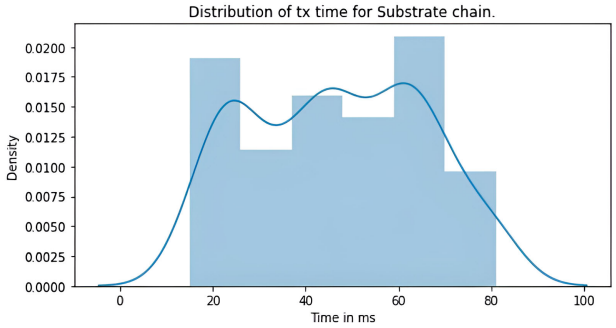

Fig. 5. Time distribution for transaction time on the Polkadot network.

The displayed time includes the time taken to issue transactions from the API server. To communicate with BridgeChain, the test interacted with the JavaScript library called ”Polkadot.js.” The transaction time on BridgeChain ranged from 15 ms to 81 ms, with an average execution time of approximately 44 ms. This is slightly slower compared to other Substrate-based chains. This slower performance is due to the latency introduced by the API from which the transactions were issued, as well as the limited power of the test machine.

B. Cosmos chain benchmarking

The test analyzed the same metrics as observed in BridgeChain on Polkadot. The transactions were called from a JavaScript API using CosmJs.

Fig. 6 shows the execution times of 200 separate transactions.

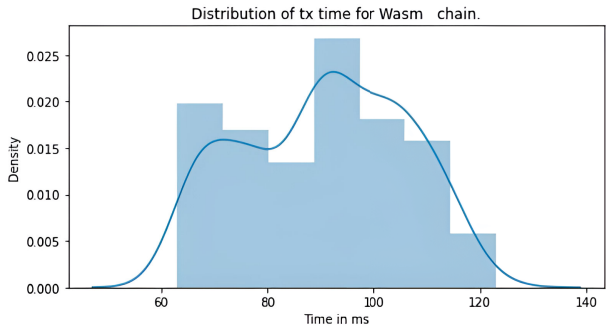

Fig. 6. Time distribution for transaction times on the Cosmos network.

The displayed time includes the time taken to issue transactions from the API server. As shown in the figure, transaction times ranged from 66 ms to 121 ms, with an average execution time of just over 90 ms. This processing time is bound to the machine’s performance and should not be noticeable in real-world applications. However, there is always room for improvement.

C. Fee feasibility test

The transfer fee depends on bridge demand. As the number of vaults grows, the bridge fee is driven down, and the opposite is true if the number of transactions grows. To simulate real-world performance, various numbers of messages were fired at the same time, and the fee was compared among them. The result of this test can be observed in Fig. 7.

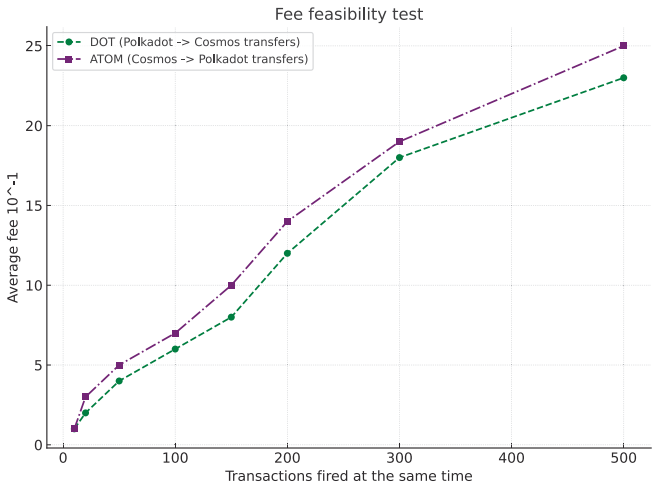

Fig. 7. Fee feasibility test results.

The result is linear, with occasional spikes but nothing too high. Fees proved to be predictable and stable during lower or higher loads.

D. Comparison to other solutions

Currently, Axelar is the sole solution known to enable this type of transfer, albeit indirectly. To assess its performance in relation to our solution, we conducted a detailed comparison centered on two critical factors:

- The cost associated with cross-chain transactions
- The speed of cross-chain transactions

The findings of this comparative analysis are summarized in Table I.

VI. Conclusion

The paper proposes a solution that bridges the interoperability gap between Polkadot and Cosmos networks.

The introduction section quickly covered blockchain basics, while the background section dove deeper into the topic and outlined a clear interoperability gap. This gap was then addressed in the design section, which contains the solution prototype architecture overview and a profound explanation of the solution implementation details. The design section also covers the cross-chain transaction flow from both chains.

The evaluation section focused on benchmarking the bridge transaction execution times on chains and fee feasibility. As pointed out in the evaluation section, the average transaction time on BridgeChain was 44 ms, while the average on the Cosmos chain was 90 ms. Since transferring tokens between

TABLE I. COMPARISON BETWEEN OUR SOLUTION AND AXELAR

Chain/Test	Transfer time solution Mean (Batch of 10 transactions)	Transfer time Axelar	Fees solution Mean (Batch of 10 transactions)	Fees Axelar
To Polkadot 1	58 seconds	68 seconds	1,05$	1,60$
To Polkadot 2	73 seconds	104 seconds	1,05$	1,60$
To Polkadot 3	65 seconds	93 seconds	1,05$	0,85$
To Polkadot 4	113 seconds	71 seconds	1,05$	1,04$
To Polkadot 5	67 seconds	96 seconds	1,05$	1,08$
To Cosmos 1	100 seconds	137 seconds	1,05$	0,65$
To Cosmos 2	103 seconds	90 seconds	1,05$	1,24$
To Cosmos 3	84 seconds	121 seconds	1,05$	0,61$
To Cosmos 4	79 seconds	102 seconds	1,05$	0,60$
To Cosmos 5	81 seconds	84 seconds	1,05$	1,60$

networks requires the execution of four or five transactions, this theoretically implies that a cross-chain transfer can be completed in under 30 seconds, considering block times. However, in practice, cross-chain transfers take approximately 1 minute due to delays in user input. This is still impressive compared to other traditional bridges, which can take up to 30 minutes or a few hours to complete. The bridge can facilitate fast block processing times on both networks, allowing cross-chain transfers much quicker. The fee feasibility also proved to have positive results, where the result was nicely linear without high spikes.

Some future work is needed to make this solution feasible for real-world applications. The solution also needs extensive auditing, as many bridge projects have been victims of multiple hacks in the past, resulting in losses of millions of dollars. The solution can also benefit from being dockerized for easier launch of the Relayer component for investors interested in becoming Relayers. Another interesting advancement could be improving the fee mechanism to help users and vaults without any party having to sacrifice much on any occasion - vaults with lower fees when there is not demand and users with higher fees when there is high demand.

ACKNOWLEDGMENT

This work was supported by the Slovak Science Grant Agency under project VEGA 1/0300/25.

REFERENCES

[1] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," *Satoshi Nakamoto*, 2008.

[2] N. Patel, A. Shukla, S. Tanwar, and D. Singh, "Kranti: Blockchain-based farmer's credit scheme for agriculture-food supply chain," *Transactions on Emerging Telecommunications Technologies*, vol. 35, no. 4, e4286, 2024.

[3] H.-W. Jang, H. S. Jung, and M. Cho, "Blockchain adoption in the food and beverage industry from a behavioral reasoning perspective: Moderating roles of supply chain partnerships," *Journal of Hospitality and Tourism Technology*, vol. 15, no. 1, pp. 138–155, 2024.

[4] I. Beveridge, J. Angelis, and M. Mihajlov, "Benefits and challenges with blockchain technology in global food supply chains: Views from the practice," *British Food Journal*, vol. 126, no. 7, pp. 2769–2786, 2024.

[5] P. Giganti, M. Borrello, P. M. Falcone, and L. Cembalo, "The impact of blockchain technology on enhancing sustainability in the agri-food sector: A scoping review," *Journal of Cleaner Production*, p. 142 379, 2024.

[6] A. Atadoga, O. A. Elufioye, T. T. Omaghomi, O. Akomolafe, I. P. Odilibe, O. R. Owolabi, *et al.*, "Blockchain in healthcare: A comprehensive review of applications and security concerns," *International Journal of Science and Research Archive*, vol. 11, no. 1, pp. 1605–1613, 2024.

[7] W. A. Al-Nbhany, A. T. Zahary, and A. A. Al-Shargabi, "Blockchain-iot healthcare applications and trends: A review," *IEEE Access*, 2024.

[8] H. Wu, Q. Yao, Z. Liu, *et al.*, "Blockchain for finance: A survey," *IET Blockchain*, 2024.

[9] O. Adisa, B. S. Ilugbusi, O. C. Obi, *et al.*, "Decentralized finance (defi) in the us economy: A review: Assessing the rise, challenges, and implications of blockchain-driven financial systems.," *World Journal of Advanced Research and Reviews*, vol. 21, no. 1, pp. 2313–2328, 2024.

[10] M. Alharby, A. Aldweesh, and A. Van Moorsel, "Blockchain-based smart contracts: A systematic mapping study of academic research (2018)," in *2018 International Conference on Cloud Computing, Big Data and Blockchain (ICCBB)*, IEEE, 2018, pp. 1–6.

[11] G. Wood, "Polkadot: Vision for a heterogeneous multi-chain framework," *White paper*, vol. 21, no. 2327, p. 4662, 2016.

[12] J. Kwon and E. Buchman, "Cosmos whitepaper," *A Netw. Distrib. Ledgers*, vol. 27, pp. 1–32, 2019.

[13] T. Thamrin, Y. Arifin, and S. W. H. L. Hendric, "Trends in blockchain applications: Current and future perspectives," in *2024 2nd International Conference on Software Engineering and Information Technology (ICoSEIT)*, IEEE, 2024, pp. 187–191.

[14] A. Bigiotti, L. Mostarda, A. Navarra, A. Pinna, R. Tonelli, and M. Vaccargiu, "Interoperability between evm-based blockchains," in *International Conference on Advanced Information Networking and Applications*, Springer, 2024, pp. 98–109.

[15] V. Buterin, "Chain interoperability," *R3 research paper*, vol. 9, pp. 1–25, 2016.

[16] T. Koens and E. Poll, "Assessing interoperability solutions for distributed ledgers," *Pervasive and Mobile Computing*, vol. 59, p. 101 079, 2019.

[17] R. Belchior, A. Vasconcelos, S. Guerreiro, and M. Correia, "A survey on blockchain interoperability: Past, present, and future trends," *Acm Computing Surveys (CSUR)*, vol. 54, no. 8, pp. 1–41, 2021.

[18] Polkadot, *Polkadot xcm documentation*. [Online]. Available: https://wiki.polkadot.network/docs/learn-xcm (visited on 01/12/2024).

[19] Polkadot, *Polkadot architecture overview*. [Online]. Available: https://wiki.polkadot.network/docs/learn-architecture (visited on 01/12/2024).

[20] C. Goes, "The interblockchain communication protocol: An overview," *arXiv preprint arXiv:2006.15918*, 2020.

[21] A. Amoordon and H. Rocha, "Presenting tendermint: Idiosyncrasies, weaknesses, and good practices," in *2019 IEEE International Workshop on Blockchain Oriented Software Engineering (IWBOSE)*, IEEE, 2019, pp. 44–49.

[22] Axelar, *Axelar whitepaper*. [Online]. Available: https://www.axelar.network/whitepaper (visited on 01/12/2024).