ASTERIZM

An enterprise-oriented blockchain interoperability solution that allows to perform fast and confidential cross-chain transactions between various public and private chains.

Build cross-chain dApps, while saving money on transactions, staying compliant with regulations and avoiding risks of data leaks.

White Paper v. 1.2

Artem Avdeev, Alex Gotovets, Denis Polulyakhov –
created at 30.10.2022, updated at 14.06.2023

Contents

1. Introduction

Abstract

The rapidly developing blockchain industry attracts the attention of many developers, companies and investors who seek to bring their expertise to the market and create additional value. Technologies are developing faster with the influx of funds and specialists, so over the past couple of years, many new blockchains have appeared, using technologies that are designed to solve issues of already existing popular networks, as well as imperfections in many real-world industries.

Such a variety of different blockchains entails a strict separation of the users' liquidity and the interaction of projects with each other between networks. Each chain has its own ecosystem of products with liquidity and community not connected with the ecosystems of other blockchains.

This market situation is a great opportunity for cross-chain communication projects, especially in the direction of large business, which is becoming more and more integrated into the world of cryptocurrencies every month.

Over 150 networks with 2300+ protocols

•

Over 500 enterprises have already implemented blockchain in business processes

This Paper introduces Asterizm, the first enterprise-oriented blockchain interoperability protocol providing a trustless and confidential communication channel for companies building cross-chain dApps.

Asterizm implementation by Web3 protocols or enterprises with private/public blockchains makes seamless and confidential cross-chain operations (transmission of arbitrary messages or assets) in dApps possible without high transaction latency and overpaying for the intermediate blockchain as a guarantee of cross-chain transaction validity.

Thanks to encryption at the core of Asterizm architecture and a unique soft consensus model based on 2 independent off-chain entities, dApp developers can bring together liquidity, data, and users from different chains to provide better service in a trustless way for the project customers building cross-chain lendings/DEXes/marketplaces/bridges and omnichain corporate solutions (sensitive data exchange or performing transactions in blockchains by other companies) without legal risks of personal or corporate data leaks.

All cross-chain transactions processed by the Asterizm protocol are encrypted on a client (developer) side using a preferred encryption method and validated on the destination chain with an on-chain Asterizm Connector.

It is important to note that Asterizm protocol allows building cross-chain smart contracts that implement complex intercorporate enterprise data systems running simultaneously in different networks and securely synchronizing with each other without any data leaks (for example, medical data exchange, CBDCs payment services, scientific data exchange, finance data exchange, etc.). **[Figure 1]**

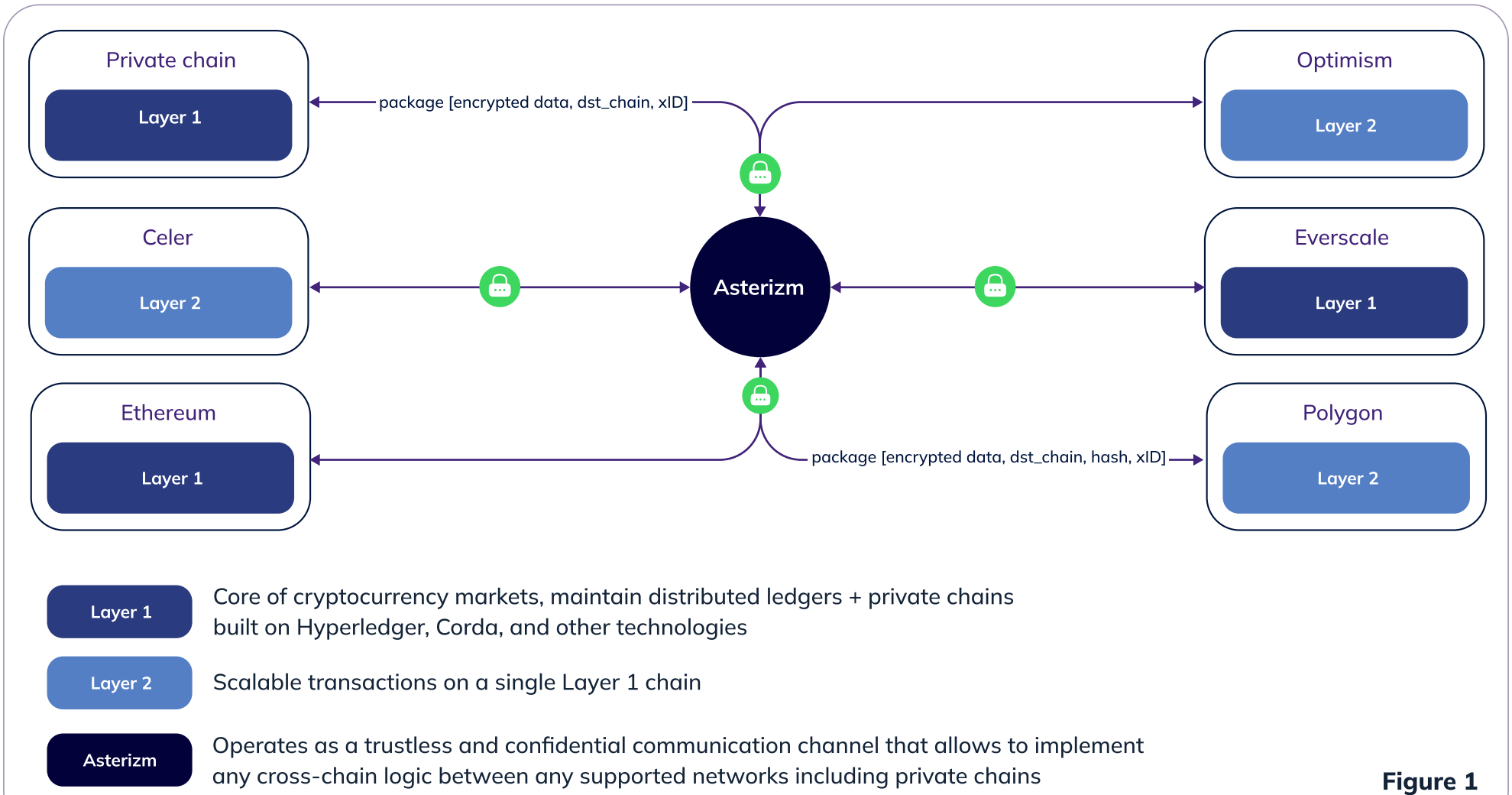

Figure 1

1.1 Introduction to the subject area

1.1.1 Introduction

The main concept of the blockchain rests on three pillars: decentralization, transparency and immutability.

Basic principles:

- No single organization controls the blockchain;
- Actions in the blockchain are verifiable and irreversible

Thus, new blockchains with their own product ecosystems began to appear, and the problem of communication between such blockchains came to the forefront.

Users and developers are forced to allocate time, resources, and liquidity between chains, which significantly hinders the onset of mass adoption of cryptocurrencies and the growth of the market as a whole.

This is especially important in the case of enterprise blockchains, which currently have to exist autonomously, making it completely impossible to implement business cases where two companies interact with each other on-chain, exchanging data or assets, and taking the digital economy and business processes to a whole new level, moving the entire industry forward.

The goal of the Asterizm protocol is to solve this acute problem by providing a trustless infrastructure solution to create cross-chain dApps without the risk of compromising transmitted data and overpaying for fast cross-chain transactions.

Asterizm allows enterprises and Web3 protocols to:

- create NFTs and assets bridges,
- develop cross-chain smart contracts with arbitrary logic,
- create cross-chain dApps that exchange sensitive information: user data, financial data, and corporate data.

The implementation of cross-chain communications is only possible with an additional mechanism that goes beyond the usual blockchain cryptosystem, which must be trustless*.

1.1.2 What is cross-chain

Cross-chain communications refers to the information transferring between one or more blockchains. Cross-chain communications are motivated by two common requirements for distributed systems: accessing data and accessing functionality which is available in other systems.

Cross-chain Consensus is the technique by which nodes or entities on a destination blockchain know that nodes or entities on a source blockchain have come to agreement on some fact. It allows information from a source blockchain to be trusted on a destination blockchain. Understanding how cross-chain consensus is achieved and the underlying trust assumptions of the cross-chain communications protocol is important when evaluating the appropriateness of a protocol for use with permissionless and permissioned blockchains.**[1]**

---

*Trustless in this context is used in the good sense of "a method where you don't have to trust any third party

1.1.3 Why privacy matters in cross-chain communications?

Today, the cross-chain solutions market has established an approach to designing the protocol architecture as a public decentralized network of validators responsible for the security, integrity, and validity of cross-chain transactions made through them in the form of messages or digital assets.

Under the concept of a public decentralized validator network, all data that passes through these validators is transmitted in the open, which dramatically narrows the options for utilization of cross-chain communication technology, being limited exclusively to Web3 protocols that transmit non-sensitive information.

The presence of privacy at the level of cross-chain communication protocol opens up completely new horizons for the use of technology and also makes communication much safer even within the existing web3 protocols.

For example, data privacy is critical when cross-chain communications are used to exchange sensitive user, corporate and financial data. This is especially relevant when it comes to interoperability between private blockchains, where the data is inherently private and should not be exposed to third parties, including the cross-chain communication provider.

Today, developers have to trust the cross-chain communications provider with their data, whether it's financial information, which a validator could capture and use for its purposes before it's delivered to the destination network or other arbitrary data.

Cross-chain transaction privacy is relevant for cross-chain DEXes/farming strategies, GameFi, NFT marketplaces, as well as large Web2 market companies that have integrated blockchain technology into their business processes.

Asterizm protocol has implemented a unique approach to secure cross-chain transactions based on data encryption, which allows you to conduct (process) fast transactions confidentially for any third party, including Asterizm itself, without overpaying for infrastructure validators.

Legal part

It is important to mention that nowadays, in terms of legal compliance in the cryptocurrency market, everyone is mostly concerned about compliance with SEC rules and regulations, but no one pays attention to personal and corporate data protection laws (GDPR), which didn't go anywhere and are just as relevant for blockchain technology projects as for Web2 companies.

This is particularly relevant for large companies that implement blockchain in their business processes. The lack of solutions that ensure cross-chain communications compliance with their security regulations, as well as with the legislation, strongly hinders the development of the entire crypto industry. After all, it is large companies that will primarily contribute to the global mass adoption of cryptocurrencies and blockchain technology.

P.S. In Section 2.1 you can find more information on how Asterizm's architecture and approach make cross-chain transactions confidential and fast compared to existing solutions, and allow to avoid high infrastructure maintenance costs.

1.1.4 Market and demand for solutions

As cryptocurrencies' adoption increases, the demand for cross-chain transactions has grown tremendously. The total value locked (TVL) in the major Ethereum bridge protocols has increased by 2.1 times from $13,7B to $30,2B between May and October 2021.

- New smart contract blockchains such as Avalanche, Terra, and Binance Smart Chain (BSC) are gradually thwarting Ethereum’s monopoly (Ethereum’s market share reduced from 98% in January to 66% in October 2021).
- BSC showed the highest TVL with $13.7 billion, followed by Polygon, Avalanche, Fantom Anyswap, and Arbitrum at the end of October 2021.

$168 Billion

Total value bridged

68% ✧ $115 Billion

Past 12 monthes

Sources: Estimated performed based on Nansen & Dune Analytics data

Cross-chain solutions are grouped into two different taxonomies:

- By their purpose: asset-specific, chain-specific, application-specific, and generalised “Internet of Blockchains” bridges; and by their method of validation: externally verified, natively verified, and locally verified.
- Based on TVL data for bridges connected to the Ethereum network, chain-specific bridges have historically bridged the lion’s share (90-95%) of cryptocurrencies since April 2021. Application-specific bridges have started to gain popularity among users, bridging 27% of TVL in Ethereum in October 2021, skyrocketing four times from September.

More comprehensive blockchain solutions (Cosmos, Polkadot, and Avalanche) have emerged to solve the lack of interoperability at a lower infrastructure level and introduce scalability where simple bridges cannot. Their popularities are also reflected in the changes of their market values:

- The market capitalisation of their native coins (Cosmos's ATOM, Polkadot's DOT and Avalanche's AVAX) rose by 130%, 31% and 270%, respectively, between May and October 2021.**[2]**

1.1.5 Market participants

Today (September 2022), there are many projects in the cross-chain industry that offer their technological solutions for protocols and end users in order to solve the problem of interoperability.

- **Blockchain bridges**
- **Interoperability-Focused Blockchains**
- **Cross-chain messaging protocols**

At this point, it is important to mention that none of the solutions listed below provide the ability to transfer data/messages from one network to another in a confidential format.

Blockchain bridges

The most popular cross-chain solutions on the market today are blockchain bridges, which can be divided into the following categories **[2]**:

Asset-Specific	Chain-Specific	Application-Specific	Generalised
Interlay	Avalanche Bridge	AnySwap	Chainlink
WBTC	Binance Bridge	cBridge	Cosmos IBC
tBTC	Harmony Bridge	Celer Network	Polkadot
WRAPPED	Polygon Bridge	Thorchain	
	Rainbow Bridge	Wanchain	
	Terra Shuttle		
	Solana Wormhole		

Based on the data of the major bridge solutions, chain-specific bridges dominated the market before September 2021. However, application-specific bridges were under the spotlight recently as their TVL skyrocketed 4 times in October.

Asset-Specific Bridges

Asset-specific bridges are built to transfer specific cryptocurrencies. The most well-known example is the Wrapped Bitcoin (WBTC) operated by BitGo. WBTC is an ERC-20 token that matches the value of Bitcoin due to 1:1 backing of Bitcoin. WBTC allows users to unlock the equity potential of their previously dormant capital in the Bitcoin network to participate in DeFi. Minting WBTC in the wrapped framework is initiated by a merchant and performed by a custodian without involving users.

A common criticism of wrapped assets is that they are fundamentally managed by a centralized entity that oversees the gateway and rules by which assets are locked and minted like WBTC and HBTC.

Chain-Specific Bridges

A bridge between two blockchains usually supports simple operations like locking and unlocking tokens on the source chain and minting new assets on the destination chain. One good example is Polygon, a protocol and a framework for building and connecting Ethereum compatible blockchain networks. Although such a bridge can be scalable and faster in transaction speed, the limited blockchains access is the main bottleneck.

Application-Specific Bridges

As the name suggests, these bridges focus on specific applications. For example, THORChain is a blockchain that aggregates liquidity across multiple chains through its multichain THORSwap DEX.

Generalised Bridges

Protocols in this category design a large-scale comprehensive solution to facilitate general data transfer across multiple blockchains. The data can be tokens, smart contracts, network states, and so on. The representatives are Cosmos IBC and Polkadot, which we'll elaborate a little bit later in this document.

Interoperability-Focused Blockchains

More comprehensive blockchain solutions have emerged to solve the lack of interoperability at a lower infrastructure level and introduce scalability where simple bridges cannot.

We describe three major projects working to facilitate interoperability cross-chain communication: Cosmos, Polkadot, and Avalanche. The following chart summarizes their features:

Feature	Avalanche	Cosmos	Polkadot
Genesis Block Date	21 Sep 2020	13 Mar 2019	27 May 2020
Consensus	Proof-of-Stake	Proof-of-Stake	Nominated Proof-of-Stake
Number of Projects	~343	~255	~499
Token	AVAX	ATOM	DOT
Transactions per second	4.500 - 10.000 per Subnet	1.000 TPS per Hub/Spoke	1.500 per Parachain
Time-to-Finality	<2 seconds	6 seconds	60 seconds

Cosmos

Cosmos' key protocol that bridges together its ecosystem is the Inter-Blockchain Communication (IBC) that is built with a Hub & Spoke design architecture.

Marketing itself as the "Internet of Blockchains", Cosmos is a decentralized network of independent blockchains that aims to bridge blockchains in a trustless and permissionless manner that does not require the trusting intermediaries like Wrapped assets or chain-specific bridges.

Polkadot

Compared to Cosmos, Polkadot's institutional setup is slightly more centralized. This is due to its mandated "shared-security" federation model that revolves around a common set of shared validators on its central "Relay Chain".

Although commonly mistaken for a Layer-1 blockchain, Polkadot is closer to a "Layer-0" meta-protocol that serves to connect Layer-1 blockchains.

Developers can launch their own side-blockchains (known as "parachains") that connect to the Relay Chain on Polkadot at a much faster speed. Its first parachain auctions, which granted developers rights to develop a chain integrated to its main Relay Chain, were held on 11 November 2021, marking the project's very first steps towards multichain interoperability.

Avalanche

Avalanche is a "platform of platforms" network where thousands of heterogeneous, interconnected individual blockchains (known as subnets) can be built on top of it.

Anyone can create their own customised applications on a subnet with the power to issue and design their own tokenomics or customise their own validation requirements, consensus mechanisms and entry barriers. Avalanche's unique proposition is its novel "Avalanche Consensus" protocol that uses repeated random sub-sampled voting. This consensus mechanism works by querying a few validators for approval and only further queries more validators when approval is conflicted. This way, Avalanche achieves consensus with minimal overhead per node. Avalanche is capable of scaling up to 10,000 validators per subnet.

Cross-chain messaging protocols

There is a whole ecosystem of platforms working to expand the scope of cross-chain communication. As the name suggests, these protocols(mostly bridges) allow for any piece of data, including tokens, the state of a chain, a contract call, an NFT, or governance votes, to be moved from chain A to chain B.

This section will explore the design of seven data messaging bridges: LayerZero, Wormhole, Nomad, Celer Inter-chain Message (IM), Multichain's anyCall, Hyperlane (previously Abacus), and Axelar **[3]**.

✧ Axelar

Axelar Network describes itself as a full-stack decentralized transport layer delivering secure cross-chain messages across Web3.

It provides a uniform cross-chain messaging solution for both developers and users. Developers can use Axelar gateway contracts and connect to any EVM contract on any chain without having to make any changes to their chains or UIs.

Axelar's main selling points revolve around its extensive developer kit and its connection with Cosmos-based chains like Osmosis and Juno. Moreover, Axelar is a Cosmos-based chain itself and uses its own blockchain for validation. This feature is key in Axelar's design and is the reason for many of its strengths and some trade-offs. **[3]**

✧ LayerZero

LayerZero is a generalized data messaging protocol that describes itself as an "omni-chain" solution. It is designed to carry lightweight messages across a bevy of chains via gas-efficient, non-upgradeable smart contracts.

The most basic component of LayerZero are the "Endpoints" found on supported chains. These endpoints are implemented as a series of smart contracts that allow domains to communicate with each other, with each chain having its own "Library" in the LayerZero system. Each Endpoint comes

with a messaging library native to the domain the Endpoint sits on, along with a proxy, which makes sure the Endpoint uses the correct library version. Once deployed, the Endpoints are like smart contracts that cannot be shut down, allowing for an immutable flow of messages.

From there, LayerZero relies upon two off-chain entities, an Oracle and a Relayer, to pass messages between the endpoints found on different domains. In this setup, an oracle (like Chainlink) forwards a block header from domain A to domain B, while a separate relayer passes a transaction proof from domain A to domain B. If the two match and the proof is validated by the block header, then the cross-chain message is sent to the destination address. **[3]**

P.S. The Asterizm protocol partly resembles Layer Zero in its concept, but an important point and advantage is to provide encryption of the transmitted data in parallel with cross-chain transfer validation on the user application side by running the server with Asterizm software, which actually makes the system trustless from the very beginning and without the use of third-party Chainlink-type services, since encrypted data already passes through the Relayer Server and the only reason for decentralizing Relayers to bring servers uptime to 99.(9).

We will analyze the implementation of the Asterizm solution further in section 2.1 of the document.

✧ Celer Interchain Message (Celer IM)

Celer Interchain Message (Celer IM) is designed as a “plug and play” cross-chain composability solution for building cross-chain dApps to promote efficient liquidity utilization, coherent application logic, and shared state across tens of chains. Essentially, Celery IM offers devs an easy way to instantly create a cross-chain dApp.

The Celer IM architecture is powered by a combination of on-chain smart contracts that receive and send messages and the Celer State Guardian Network, a proof-of-stake blockchain built on Tendermint specializing in authenticating cross-chain messages. cBridge, a fungible token and NFT bridging application, is built with this architecture as a “built-in” cross-chain dApp. With the combination, Celer IM enables a robust set of use cases for dApps like cross-chain DEXes, yield aggregators, lending protocols, multi-chain NFTs, and more. **[3]**

Summary

As you can see, any cross-chain solution requires an intermediate layer(off-chain), which can be expressed either as a bundle of its own blockchain and classic software on servers, or simple software running on the servers of a cross-chain project.

Thus, it becomes clear that the main goal of cross-chain communication solution providers has been to implement stable, secure, and cost-effective technology solutions used as an intermediate layer to broadcast messages from one chain to another.

Key parameters of the intermediate layer, which the existing solutions focus on:

- Security
- Economic efficiency
- Scalability

This covers simple cross-chain Web3 cases only

Later in this document, we will consider how Asterizm takes into account in its concept of the maximum performance achievement in all the parameters specified above.

The current approach to cross-chain communication no longer covers all market needs and consumes a significant amount of resources and time to complete one cross-chain transaction.

The lack of privacy at the communication channel level makes the transfer of sensitive information highly risky and in the case of corporations implementing blockchain even illegal, which seriously hinders the development of the entire crypto industry and opens up huge potential for projects that can solve this problem.

- Security
- Economic efficiency
- Scalability

\+

- Privacy

This covers all possible use cases, including enterprise and government projects

In the next chapter of the Paper, we will explain how **Asterizm managed to meet all 4 key parameters of a cross-chain communication solution** while reducing cross-chain transaction cost and latency.

2. Asterizm protocol

2.1 Overview

Asterizm is an infrastructure solution that provides a trustless and confidential communication channel for companies building cross-chain dApps, allowing them to save money and avoid data transfer risks.

Asterizm secures cross-chain transactions with an on-chain module called Asterizm Connector and two independent off-chain entities: Asterizm Relayers and a Client off-chain module.

Perform fast and confidential cross-chain transactions without overpaying

When developing the Asterizm protocol, all best practices of cross-chain projects were taken into account to create the most balanced solution for market participants.

Data privacy, security, scalability, and economic efficiency were brought to the fore in Asterizm protocol for enterprises entering the crypto world and major Web3 protocols.

2.2 Components

Asterizm Technology Overview

Asterizm technology is based on encryption and on-chain validation of cross-chain transactions on the destination network.

Asterizm secures cross-chain transactions with Asterizm Connector as an on-chain module and two independent off-chain entities: Asterizm Relayers and Client off-chain module.

All data (messages or instructions) transferred through the Asterizm protocol is encrypted with the Client off-chain module and validated on the destination chain in the Asterizm Connector with ***hash*** based on transaction data and a unique set of parameters that prove the transaction validity.

This approach allows to transfer data confidentially between networks, and ensures the integrity and validity of the cross-chain transaction even if the Client off-chain module or Asterizm relays have been hacked.

Security and operational stability thanks to encryption and on-chain validation

Since the information transmitted through Asterizm servers is encrypted, there is no point in hacking the server data, as it is impossible to do anything with the transmitted data: neither change nor tamper with it. The encryption key is located on the client's server only.

If the client's server is compromised, the attacker will not be able to spoof the outgoing transaction or create a fraudulent transaction, as the validation of the cross-chain transaction takes place on-chain in the destination network using the ***hash*** and the ***xID*** generated on the smart contract in the source network.

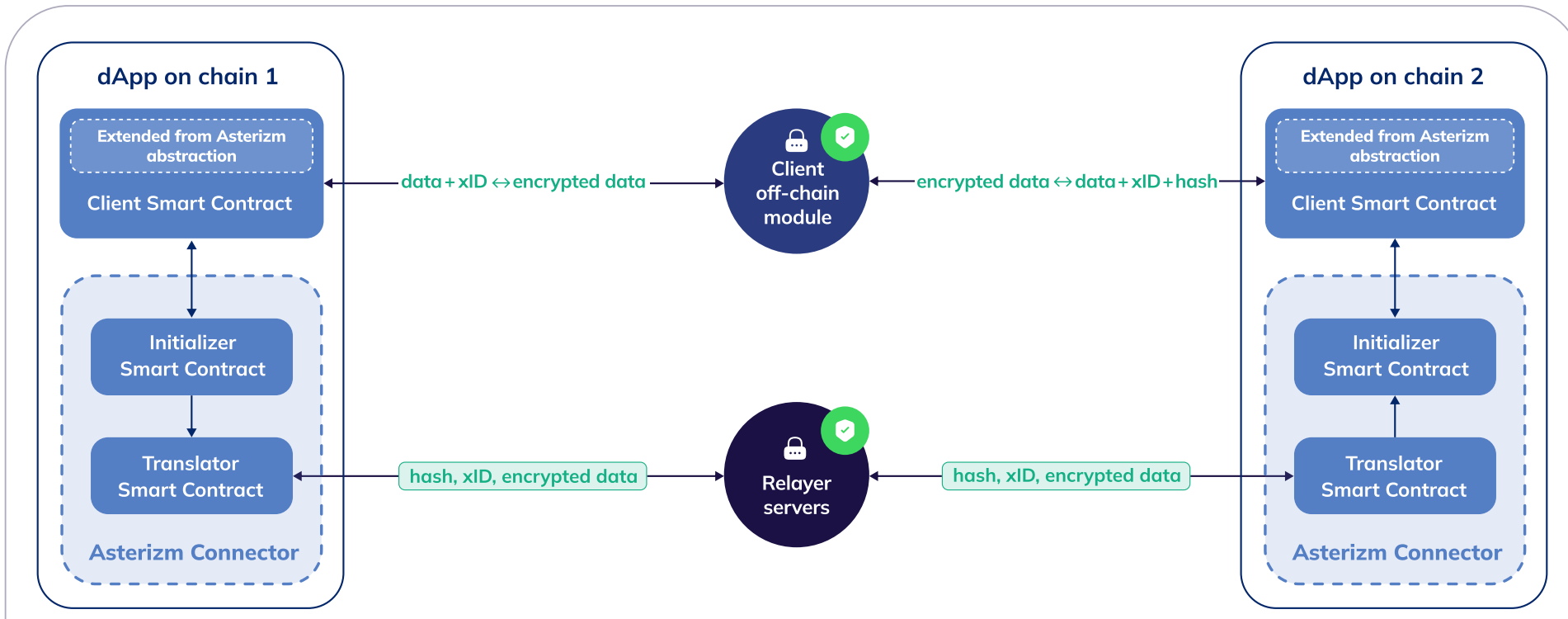

Figure 2

Our solution consists of the following components:

Smart Contract Abstraction	Abstract smart-contract class, developed by the Asterizm team, which the client should extend by implementing its cross-chain logic. The abstraction contains methods for communicating with the Client off-chain module, sending and receiving cross-chain messages, as well as several methods for securing the transaction, including ***xID*** and ***hash*** generation.
Translator Smart Contract	This smart contract works as a transmitter in every network supported by Asterizm. The Translator sends and receives encrypted data packets by communicating with Asterizm Relayer Servers.

Initializer Smart Contract	This smart contract acts as a validator in the destination network, checking the integrity and authenticity of the transaction using the ***hash*** and ***xID***, and as a validator (checker) in the source network, verifying the sequence and several other parameters of the transaction.
Asterizm Relayer Server	Asterizm or partner servers that act as a transport layer. These servers transfer encrypted messages from one network to another without any consensus between them, but with rules on waiting for the right number of blocks to validate the transaction in each supported network, which significantly speeds up cross-chain transactions.
Client off-chain module	Module for client-server infrastructure - a Docker image, functioning as an encryptor and Oracle for cross-chain transactions. This module performs encryption and decryption of transmitted data, as well as verification of their validity using ***xID*** and ***hash*** after transmission through the relay servers.
Asterizm Connector	Lightweight on-chain client consisting of Initializer and Translator smart contracts deployed on each Asterizm supported chain to perform cross-chain operations.
Client Smart Contract	A smart contract deployed by a client that extends the Asterizm smart contract abstraction. This contract interacts with the Client off-chain module and the Initializer smart contract on the source and destination networks to initialize and validate the cross-chain transaction, respectively.

2.3 Asterizm solution logic

Before describing the logic, it is worth it to explain several concepts used below:

- ***xID*** is a unique cross-chain transaction ID generated after initializing the cross-chain transaction using the ***_initAsterizmTransferEvent()*** method on the client smart contract

- ***Hash*** is a unique set of bytes of the following data generated by the hash function: source chain id, destination chain id, client smart contract address in the source chain, client smart contract address in the destination chain, payload and ***xID***.

- Cross-chain transaction validation is performed with ***xID*** and ***hash*** thanks to two independent off-chain entities and an on-chain Asterizm Connector.

- The off-chain part consists of the Client off-chain module, which operates as an encryptor and Oracle, and the Asterizm Relayer, which functions as a transport layer and transfers encrypted data.

- It is necessary for the Client off-chain module and Relayer server to be different entities, as this ensures the reliability of the cross-chain transaction validation approach.

- Encryption and decryption of transferred data are performed on the Client off-chain module with an arbitrary method (AES-256 by default).

  Encryption is executed after initialization of the ***_initAsterizmTransferEvent()*** method, and decryption is performed after receiving encrypted data from the client contract in the destination network and ***xID*** and ***hash*** verification.

- Asterizm Connector and Asterizm Relayers don't have access to the encryption key that encrypts transmitted data on the Client off-chain module. Thanks to this concept, there is no need to maintain a consensus mechanism at the level of Asterizm Relayers.

- In theory, anyone can be a relay, so in the future, we plan to add transport layer support for the Client off-chain module with the condition that this module can be a transport layer only for transactions of other protocols. This approach will significantly increase system fault tolerance in terms of uptime.

- The Client off-chain module and Asterizm Relayers wait for a specific number of blocks for each network before accepting the Translator or Client event emitted by the smart contract. Each client can configure the required number of blocks for each network.

- Consensus in cross-chain transaction validation occurs in the destination network by on-chain verification of the transaction using ***xID*** and ***hash*** from transmitted data received from Asterizm Relayers and Client off-chain module. This constitutes a soft consensus model that does not require unnecessary calculations, a complex economic model, and a large number of participants.

- Payload of the cross-chain transaction can be arbitrary.

- Asterizm functions in both EVM networks and non-EVM networks. The number of contracts in Asterizm Connector can vary depending on the features of the non-EVM network.

- For enterprise projects, it is allowed to make changes to the Docker image supplied as a Client off-chain module, due to the peculiarities of on-chain business processes in private blockchains.

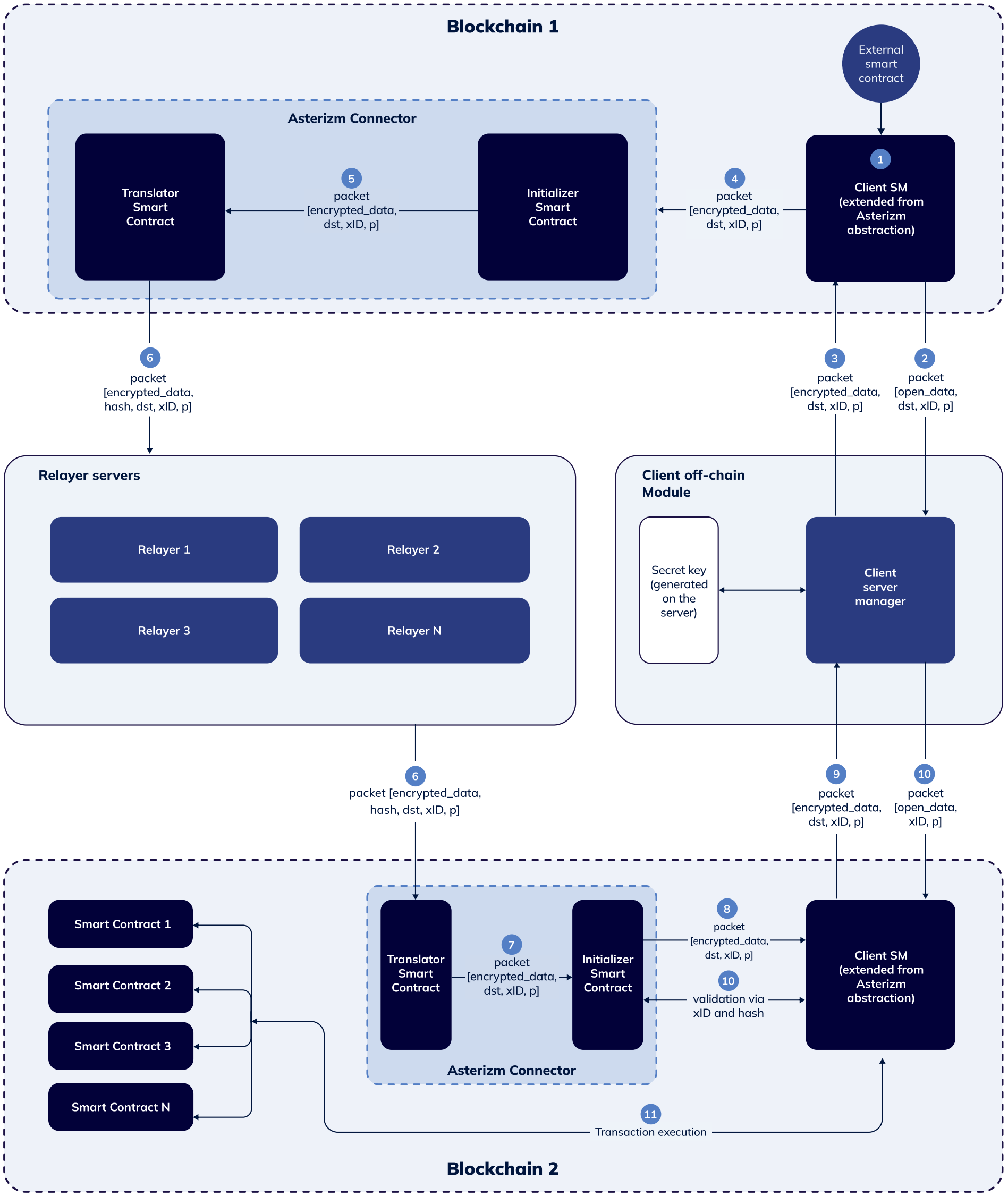
Blockchain 1
External smart contract
Asterizm Connector
Translator Smart Contract
5 packet [encrypted_data, dst, xID, p]
Initializer Smart Contract
4 packet [encrypted_data, dst, xID, p]
1 Client SM (extended from Asterizm abstraction)
6 packet [encrypted_data, hash, dst, xID, p]
3 packet [encrypted_data, dst, xID, p]
2 packet [open_data, dst, xID, p]
Relayer servers
Relayer 1
Relayer 2
Relayer 3
Relayer N
Client off-chain Module
Secret key (generated on the server)
Client server manager
6 packet [encrypted_data, hash, dst, xID, p]
9 packet [encrypted_data, dst, xID, p]
10 packet [open_data, xID, p]
Smart Contract 1
Smart Contract 2
Smart Contract 3
Smart Contract N
Translator Smart Contract
7 packet [encrypted_data, dst, xID, p]
Initializer Smart Contract
8 packet [encrypted_data, dst, xID, p]
10 validation via xID and hash
Client SM (extended from Asterizm abstraction)
Asterizm Connector
11 Transaction execution
Blockchain 2

Figure 3

packet - data packet containing cross-chain transaction content and related parameters;

encrypted_data - message/data transmitted as part of a cross-chain transaction;

dst - destination network ID;

hash - a unique set of bytes of the following data generated by the hash function: source chain id, destination chain id, client smart contract address in the source chain, client smart contract address in the destination chain, payload and ***xID.***

xID - a unique cross-chain transaction ID generated when the cross-chain transaction is initialized in the ***_initAsterizmTransferEvent()*** method;

p - additional parameters of the transaction: address of the source contract, destination contract, timestamp, and other data;

tokens - assets that are transferred with the message as part of a cross-chain transaction.
For example, to implement a cross-chain farming strategy, assets are transferred and then the methods of the required smart contracts are called to allocate the funds in the farming protocol.

Logic [Figure 3]:

1. The client implements a smart contract using an abstract class from Asterizm and deploys it to the required networks.

   The _initAsterizmTransferEvent() method is used to initiate a cross-chain transaction on the contract. The data and parameters of the cross-chain transaction are transmitted to it.

   The transaction can contain arbitrary information and one or more actions (instructions) to be performed on the destination networks.

2. The client smart contract sends the received data and ***hash*** based on the payload generated at that moment ***xID***, source chain id, destination chain id, client smart contract address in the source and destination chains.

   Note: At this step, the key parameters are formed, which will further ensure the security of the cross-chain transaction, namely to confirm its validity and integrity in the destination chain.

3. The Client off-chain module encrypts the cross-chain transaction contents and sends it back to the client's smart contract in the source initializer network, calling the cross-chain transfer initialization method ***initAsterizmTransfer().***

4. Before sending the encrypted data received from the client off-chain module to the Initializer smart contract, the Client smart contract checks the received data using ***hash*** matching, thus preventing spam and spoofed transactions.

   Note: This step verifies that this exact transaction was initiated on the client's smart contract, which eliminates the possibility of spam from the client's off-chain module.

5. The Initializer smart contract checks the transaction nonce to preserve the cross-chain transaction execution sequence and transmits the encrypted data with the unique parameters to the Translator smart contract.

   Note: When a data packet is received on the Initializer smart contract, the client is identified based on the destination chain id, the Client smart contract address in the destination chain, and the client smart contract address in the source chain. After the client (sender) of the cross-chain transaction is determined, the nonce value is incremented for it.

6. Asterizm Relayer servers pull the encrypted data with the parameters from the Translator contract and send it to the Translator smart contract in the destination network for further processing.

7. The Translator smart contract on the destination network accepts the encrypted data with the parameters from the Relayers and passes it to the Initializer smart contract for validation.

8. After receiving the encrypted data and the parameters, the Initializer smart contract checks the nonce to comply with the transaction sequence, stores the transaction ***xID*** to validate the transaction at step 10, and transmits the data to the client's smart contract.

9. The client's smart contract in the destination network receives the encrypted data with the parameters, creates an event, and transmits the data to the client's off-chain module for verification and decryption.

   Note: This is where the first step of cross-chain transaction validation is performed by matching the ***hash*** from the decrypted data, original data and ***xID*** generated in the source network at the transaction initialization, and by checking the source network and address. The client's server verifies that the data is complete and consistent, thus eliminating the risk of compromise at the Relayer server level.

10. After data verification and decryption, the client's server initiates the transaction on the client smart contract by calling the ***asterizmClReceive()*** method in the destination network, sending the decrypted data and ***xID*** to the contract, and validating the transaction on the Initializer smart contract by checking the ***hash.***

    Note: At this step, the integrity of the data and the trusted addresses is checked before the transaction is executed. This check eliminates the risk of hacking the client's server.

    Deep tech: The verification is performed by checking the ***hash*** from the decrypted data and the ***xID*** that was previously stored on the Initializer contract. If the verification succeeds, it means that the client's contract executes exactly the transaction it sent, and does so for the first time. This algorithm eliminates the possibility of spamming transactions and spoofing data on the client's server (if, for example, the server is compromised).

11. After receiving the confirmation of the validity and integrity of the cross-chain transaction, the client's smart contract executes the instructions or data sent in the cross-chain transaction, calling methods of other contracts or performing calculations on the client's contract. The logic of this step depends solely on the client's business logic implemented in the cross-chain transaction.

2.4 Server infrastructure

Asterizm has a few types of technical solutions:

Asterizm Relayer Server is a server software that operates as a transport layer for encrypted messages sent by clients from one chain to another.

Asterizm Client off-chain module is an open server software that operates as an encryptor and Oracle to protect transmitted data and their validation in the destination network. It is a Docker image that has to be deployed on the Client's server (usually a frontend server for Web3 projects and a secure dedicated server for enterprises)

2.5 Minimization of fees

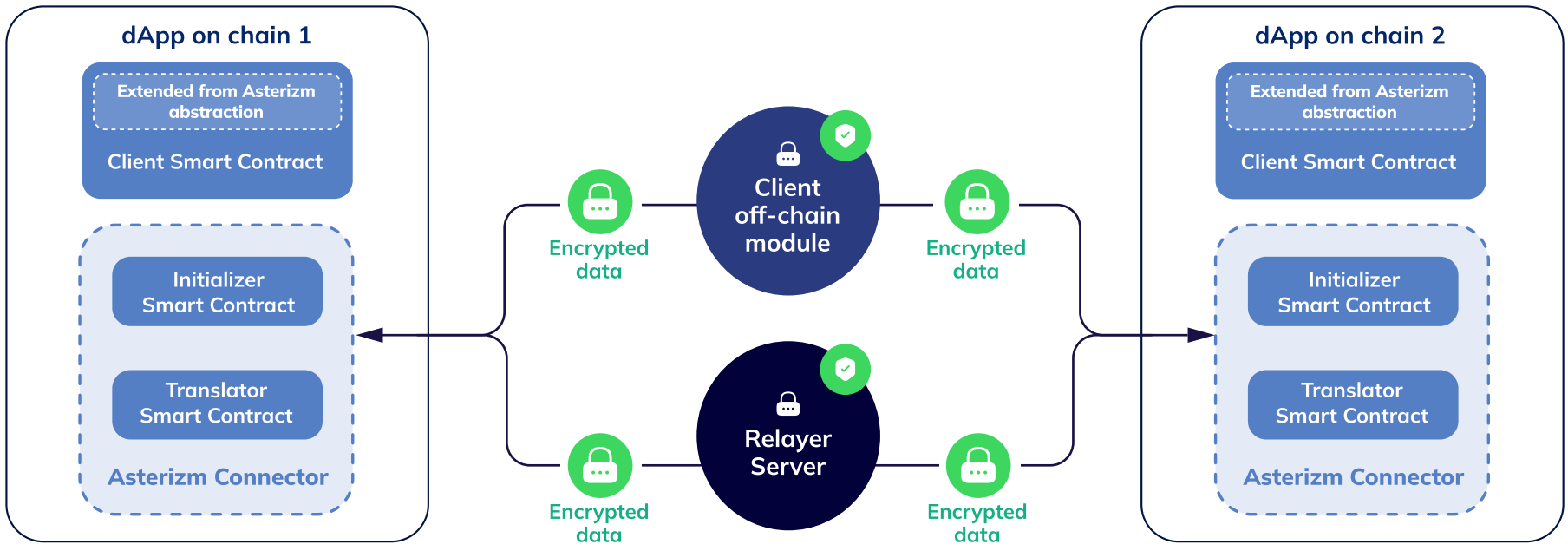

Figure 4

Gas fees are an integral part of blockchain technology, which ensures data security and uninterrupted network operation, motivating validators to ensure the continuation of the chain.

As you probably know, the operation of smart contracts on Layer 1 chains (in first-level networks) can be extremely expensive, especially if the amount of information stored constantly grows.

Previous trustless cross-chain validation solution based on cross-chain state machine replication (SMR), such as Golden Gate **[4]**, could cost millions of dollars per day to run on popular Layer 1 chains like Ethereum.

The implementation of Asterizm Connector takes into account this problem by reducing the amount of data stored, as well as changing the approach to validation and monitoring of changes in the state of the blockchain in a key way.

To solve this problem, we set out to design the most lightweight client possible. Our key observation is that replicating and storing block headers within the client is not necessary. Rather, we delegate the task of fetching the necessary cross-chain headers, transaction proofs and data encryption to off-chain entities: the Client off-chain module and Relayer Server.

This results in Asterizm Connector being incredibly lightweight, making it cost-effective even on notoriously expensive **[5]** chains like Ethereum.

3. Privacy, validity, security, and reliability

When developing the Asterizm protocol, the top priorities were to provide a confidential, secure, and reliable channel of valid data from one network to another.

Next, we will consider in detail the importance of the aforementioned protocol characteristics using the example of a description of potential problems and their solutions that were found by the Asterizm team.

3.1 Privacy and security of transmitted data

We can consider two potential attack vectors on the transmitted data:

- External
- Internal

External vector

For example, a hacker can get access to sensitive data such as personal or corporate data. A breach of such data could result in large fines for the project, lawsuits, or instances of extortion.

If we talk about DeFi/GameFi/DAO cross-chain transactions, data leaks can lead to front running, violation of business logic of the protocol, or even loss of investment opportunities (if cross-chain transactions are used to implement farming strategies in different networks). **[Figure 5]**

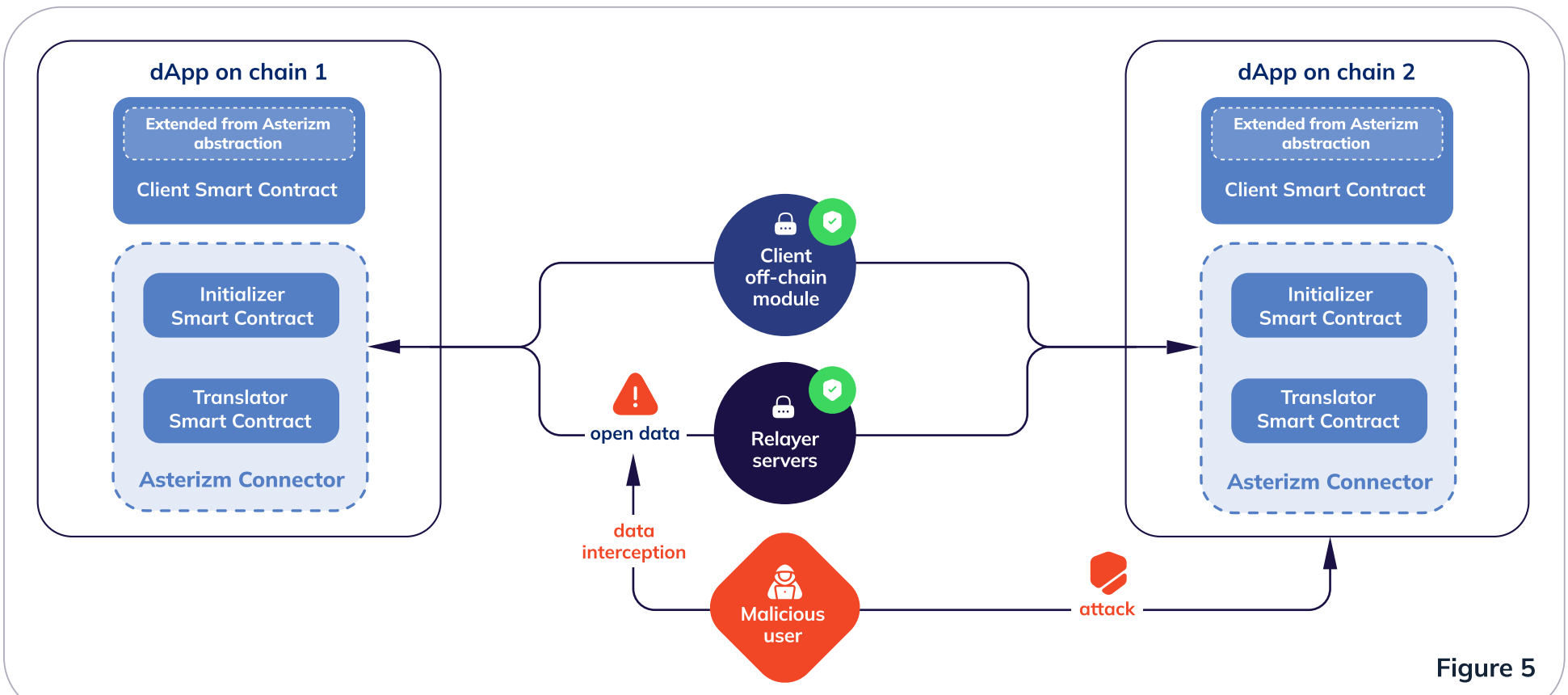

Figure 5

Internal vector

In this case, we assume that a malicious Relayer Server operator reads the transmitted data and makes adjustments to them, or creates new transactions that benefit from the information received. **[Figure 6]**

In this case, in addition to malicious actions, the hacker gets access to sensitive data, which can lead to criminal liability, as the laws on the protection of personal and corporate data also apply to projects in the blockchain industry.

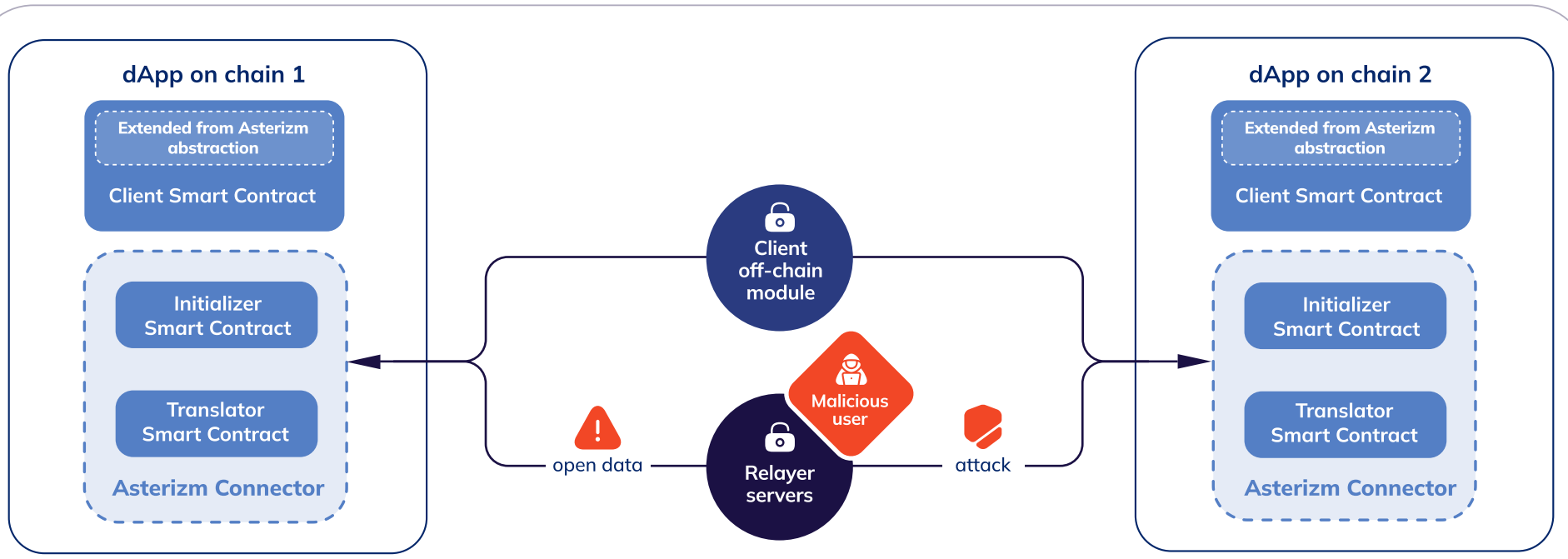

Figure 6

3.2 Reliability issues

It is important to note that in addition to protecting the transmitted data from being read and modified by the attacking party, it is necessary to ensure the smooth operation of the data transmission channel from one network to another.

Since such data is actually transmitted by a classic server, **it is necessary to minimize external risks:**

- Hoster failure
- Legal restrictions in the country where the software server is rented.
- Operational risks: server failure for any reason.

In Asterizm this issue is solved by distributing Relayer servers across different hosts in different countries. In the near future, we are planning to implement a transport module within the client module, which will allow transferring encrypted data of other clients, thus forming a network of servers, where client A acts as a transport layer for client B, and data of client A is transferred by the client C, etc.

3.3 Cross-chain transaction validity problem

The key idea of validity is **in two theses:**

- Each message transmitted between the networks appears as a result of a change in the state of the sending blockchain. In fact, this means that **each message is associated with a transaction in the sending network.**
- Each message can be delivered and executed in the receiving network only **if the transaction providing this message was successfully carried out in the sending network and has an identifier in the form of a hash.**

The absence of a mechanism verifying compliance with the above conditions when performing a cross-chain operation entails a number of vulnerabilities for the communication system.

3.4 Asterizm Solution

The Asterizm solution is designed to address the issues described above so that dApp developers are confident that they are using a reliable and secure solution to integrate confidential cross-chain operations into their products.

The off-chain part of the Asterizm protocol consists of two components, which together with the Asterizm Connector provide a solution to the voiced problems **[Figure 7]**:

- Client off-chain module
- Relayer Server

The validity of the cross-chain transaction is ensured by sending the identifier of the cross-chain message (***xID***) assigned at the time of calling the client smart contract on the source blockchain to the Client off-chain module and the Relayer server, which allows the Initializer smart contract on the destination network to compare the ***xID*** received from the Relayer Server with the one stored on the Client off-chain module. Client off-chain module and Relayer server await the number of blocks required for each network to eliminate the risk of receiving a fake blockchain state.

Thus, before decrypting and executing the received message on the destination chain, the ***xID*** is checked to make sure that the transaction actually took place on the source network and wasn't modified at the Relayer or by a malicious third party.

The security of the transmitted message is ensured by its encryption on the Client off-chain module side. A developer of the Client's dApp deploys a Docker image on the server and creates a private encryption key to encrypt the transmitted messages. Neither Asterizm nor any other third party has access to the private key at any point in time.

This key is used to encrypt transmitted data that the Client off-chain module receives from the client smart contract on the source chain, and later this data is decrypted in the destination chain only if xID matches the one received from the Relayer server.

Uninterrupted transmission of data from one network to another is provided by a network of Relayer servers without any consensus at the transport layer, and in the near future, the Client off-chain module could become the transport layer for other Asterizm clients in order to maximize uptime closer to 100%.

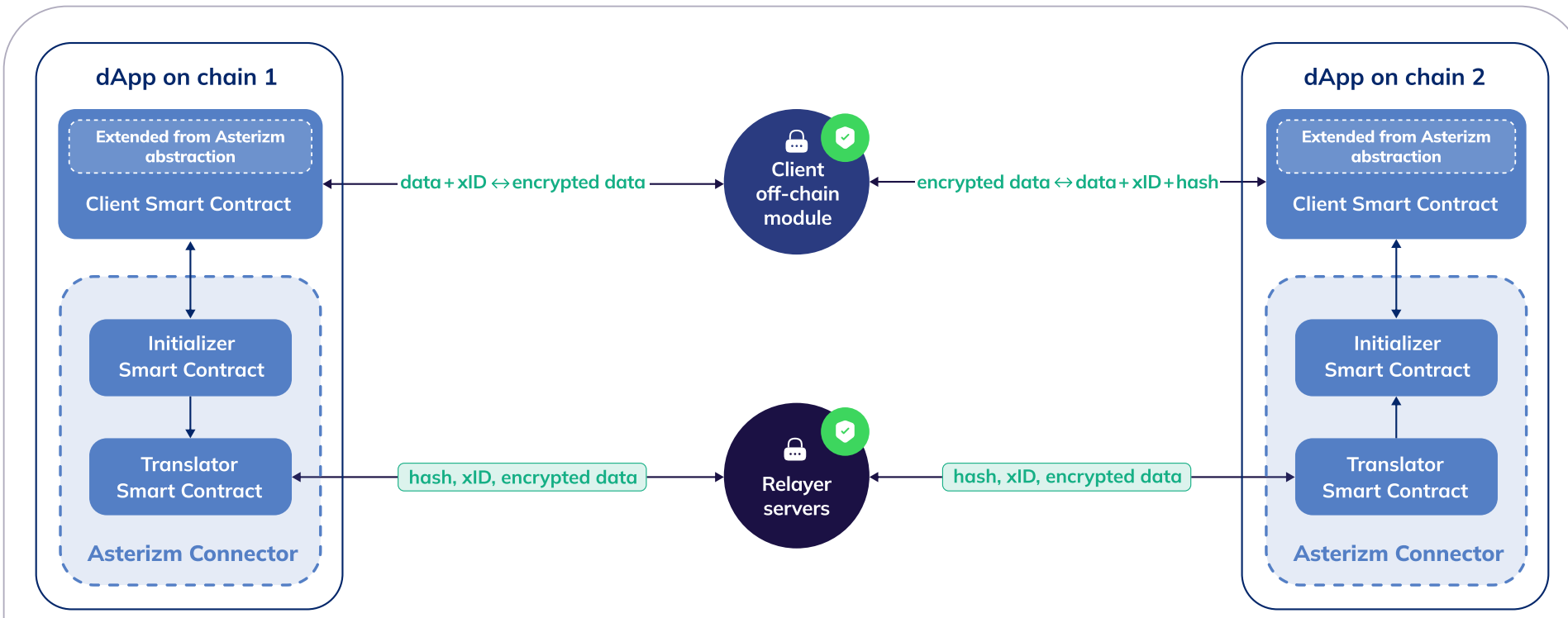

Figure 7

4. Economics of the protocol

As we mentioned above, every interaction with the network requires gas fee payment, therefore, as already clarified above, the Asterizm protocol is also designed to minimize these costs by transferring a number of operations to a trustless off-chain environment.

However, there are the following transaction costs:

- **The source network fees:**
  - When the developer of cross-chain dApp deploys a Client smart contract to send and receive cross-chain messages;
  - When Initializer and Translator smart contracts communicate with each other.
- **The destination network fees:**
  - When the developer of cross-chain dApp deploys a Client smart contract to send and receive cross-chain messages;
  - When Initializer and Translator smart contracts communicate with each other;
  - When the Client smart contract executes a message (instructions) on the destination chain.

For the convenience of developers, Asterizm provides the ability to top up client smart contracts with gas tokens on all supported networks with just 1 transaction from the client's preferred network. This can be done using the innovative Gas Charger solution from the Chainspot project with the support of the Asterizm team.

In order to popularize the Asterizm protocol at the initial stages no fees will be charged over those necessary to perform a cross-chain transaction.

In order to promote the Asterizm protocol, there will be no extra charges at the initial stages - the users will only cover the amount required to perform a cross-chain transaction.

At the same time, taking into account the design features of the solution, further monetization of the protocol is possible in at least two ways:

- **Fixed service** fee for the use of the Asterizm infrastructure, which will be factored into the total amount of all fees required to perform cross-chain transactions (charged by the Translator smart contract).
- **Charging fee** in the form of a certain percentage of the total amount of gas tokens required to perform cross-chain transactions. Paid on the Translator smart contract.

For enterprise customers, there is an option to pay periodically for a certain period. This can be especially relevant in the case of an enterprise private blockchain, where there are no publicly traded tokens and it is necessary to deploy a custom version of the Asterizm Connector into the private network.

5. Use cases and prospects

Cross-chain operations are a highly demanded internal function for any popular protocol and will remain so. This is also a crucial aspect for classic online businesses that integrate blockchain into their business processes. Some of them consider network reliability and robustness truly important. Still, for others, speed and gas prices are even more important, so the companies need to split processes across multiple chains suitable for each business task. Still, all those networks should interact and sync with each other

Trends are constantly changing, but interoperability between networks and free movement of liquidity and data will always be in demand among the audience of any projects from the currently popular industries of the crypto- and real markets.

To date, the most in-demand potential use cases for the technology can be summarized as follows:

- **CBDC & Enterprise business logic implemented on-chain**
- **Uniting liquidity and users in GameFi and Metaverses**
- **Cross-chain liquidity farming aggregators and lending platforms**

CBDC & Enterprise business logic implemented on-chain

The Asterizm protocol provides a confidential communication channel between different chains, including private networks. It is perfect for building truly trustless and secure cross-chain dApps while saving money on cross-chain transactions and avoiding the risks of data leaks.

The encryption of the transmitted data allows companies to remain legally compliant in terms of data protection and corporate law.

For example, company A is an insurance company with its private blockchain, which they use to run part of the company's business logic and to store client data. Company B is a Neobank that has also implemented blockchain in its business processes. In the event of an incident, Company A needs to send a cross-chain message to Company B containing information about the incident and instructions to release funds to the affected client (possibly even in the form of the CBDC of the client’s country of citizenship).

This is one of many great cases of using Asterizm to create a cross-chain application with a confidential channel for transferring data and assets between both public and private networks.

Asterizm allows the implementation of any cross-chain business logic implemented by companies in public or private chains. This will boost the development of the industry and mass adoption of blockchain and digital assets.

GameFi & Metaverse

Last year's GameFi boom with Axie Infinity set off a chain reaction of new Web3 game development and also attracted the attention of major players in the classic gaming market, who are now actively exploring Web3 and incorporating blockchain and DeFi mechanics into their new games.

The fragmentation of the blockchain market has made it profitable for developers to create different app modules and games on different networks in order to attract a larger audience and use the best perks and features of each blockchain.

Asterizm is designed to seamlessly and securely link business logic across networks, as well as eliminate legal risks when private blockchains are involved.

Cross-chain farming and lending

Asterizm will allow lending protocols to implement cross-chain deposit and credit logic across networks, providing users with an intuitive interface and significant savings in time and money on fees. In addition, farming aggregators will get access to liquidity and farming protocols in different networks to leverage the best rates and APY.

Similar to cross-chain yield aggregators, lending protocols work only within a single network, even if other networks have copies of the protocol. The problems remain the same:

- **Accessing liquidity on other networks**
- **Synchronized liquidity and data management tools.**

Today, for a user to take advantage of the difference in loan or deposit staking rates in the two networks, they need to use third-party solutions in the form of blockchain bridges to move assets, which increases the costs and complicates the process of interacting with the protocol.

Asterizm will allow lending protocols to implement cross-chain deposit and credit logic across networks, providing users with an intuitive interface and significant savings in time and money on fees.

Summary

Three of the most sought-after examples represent just a small fraction of the many possibilities that Asterizm offers.

With Asterizm protocol, companies don’t have to spend time and money developing their own cross-chain communication solutions or integrating third-party solutions that require trust in their system in terms of data privacy or centralized entity.

We anticipate new innovative applications at the intersection of Web2 and Web3 developed by companies that leverage trustless, cost-effective, confidential, and fast cross-chain messaging powered by Asterizm Protocol.

6. References

[1] - Survey of Crosschain Communications Protocols Peter Robinson ConsenSys Software R&D https://arxiv.org/pdf/2004.09494.pdf

[2] - In search of interoperability: An Overview of the Cross-Chain Market. Crypto.com Research Manager Kevin Wang; Crypto.com Research Analyst Donovan Choy. https://assets.ctfassets.net/hfgyig42jimx/6muXpjWea1oCM0Wwovi7Vp/b3cdcd739f0def5af74fd541ac4af7e5/211029_Cryptodotcom_An_Overview_of_the_Cross-Chain_Market.pdf

[3] - Navigating Arbitrary Messaging Bridges: A Comparison Framework, Arjun Chand. https://blog.li.fi/navigating-arbitrary-messaging-bridges-a-comparison-framework-8720f302e2aa

[4] - Golden gate – trustless-bridging ethereum (evm) blockchains – part 1: Basics. https://loredanacirstea.medium.com/golden-gate-trustless-bridging-ethereum-evmblockchains-part-1-basics-d016300ea0dd

[5] - SPAIN, M., FOLEY, S., AND GRAMOLI, V. The Impact of Ethereum Throughput and Fees on Transaction Latency During ICOs. In International Conference on Blockchain Economics, Security and Protocols (Tokenomics 2019) (Dagstuhl, Germany, 2020), V. Danos, M. Herlihy, M. Potop-Butucaru, J. Prat, and S. Tucci-Piergiovanni, Eds., vol. 71 of OpenAccess Series in Informatics (OASIcs), Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, pp. 9:1–9:15.