Omnygram: An Internet Unit of Account for Peer-to-Peer Electronic Cash Systems

Hyungsuk Kang
hyungsuk@omnygram.org
www.omnygram.org

Abstract- Cross-chain projects face significant challenges in terms of security, liquidity, and user experience. This paper proposes a novel approach to address these issues through the implementation of new account abstraction techniques for multiple blockchains. By leveraging account abstraction, the proposed solution aims to enhance asset management, improve security, and provide a personalized user experience within cross-chain environments.

I. Introduction

The blockchain technology has revolutionized financial transactions, but current cross-chain projects encounter several critical challenges related to security, liquidity, and user experience. Existing bridge projects often fail to adequately segregate user assets or consider proper accounting for asset inflows and outflows, leading to substantial losses due to potential hacks. Additionally, liquidity is often siloed, restricting the seamless financial connectivity of new blockchains with the existing ecosystem. Furthermore, the user experience provided by one-to-one bridge solutions is inadequate, as the primary focus of these projects tends to be on establishing connections and acquiring liquidity. Consequently, the quality of the user experience is disregarded, leaving users uncertain about transaction finality and token receipt. Moreover, one-to-one connected bridges lack scalability in naming schemes when connecting additional blockchains, resulting in unnecessary complexity.

Furthermore, the security of the current linearly connected bridge system raises concerns. Many projects prioritize increasing liquidity in two chains rather than implementing robust accounting mechanisms for user deposits and withdrawals across multiple chains. For instance, the Nomad bridge, despite claiming to be a security-first bridge system, experienced a significant hack that could be easily replicated by anyone connecting two blockchains using a single smart contract. A more robust approach would involve accounting for the inflow and outflow of each asset from chains with total supply, as well as diversifying risks through separate smart contracts for individual users.

In summary, the current cross-chain projects face several pressing issues, including liquidity limitations, generic user experiences, and security vulnerabilities. The liquidity is hindered by the joint siloing of duplex connected chains, impeding the integration for new blockchains into the existing system. Moreover, the lack of asset segregation and accounting on a per-user basis exposes these projects to severe losses resulting from attacks on vulnerable points. The user experience provided by one-to-one bridges is suboptimal, with users unable to ascertain when their transactions are finalized due to the mingling of requests from multiple users. To address these challenges, this paper proposes the establishment of a multiplex infrastructure that enables non-linear connections through account abstraction. This approach facilitates the seamless transfer of assets across multiple blockchains using personalized account contracts, eliminating the need for renaming assets whenever new blockchains are connected.

II. Related Works

Bitcoin[1] is the first and most renowned cryptocurrency to the general public. Unlike traditional currencies, Bitcoin is not controlled by any government or financial institution. Instead, it relies on complex mathematical algorithms to control the creation and transfer of bitcoins. There is a limited supply of bitcoins, with only 21 million bitcoins ever to be mined. The maintainer of the Bitcoin network is called miners, and they are incentivized by receiving new bitcoin to the one who finalizes transaction fastest. However, block rewards are decreasing, and Bitcoin does not have sustainable plans on incentivize miners to maintain the network without mining reward. They will have to only rely on transaction fees by end users using Bitcoin as currency. Omnygram network takes the advantage of limited supply and proposes sustainable economics where maintainers of its blockchain network can sustain regardless of decreasing reward with turnover rate and gas price index.

Ethereum[2] is a decentralized, open-source blockchain platform that was created in 2015 by Vitalik Buterin. The platform's cryptocurrency is called Ether (ETH), and it is the second-largest cryptocurrency by market capitalization after Bitcoin. One of Ethereum's primary features is the ability to create and execute smart contracts, which are self-executing agreements between parties that are stored on the blockchain. Smart contracts which are often called as externally owned account(EOA) can be programmed to automatically execute the terms of the agreement once certain conditions are met, which can streamline and automate a wide range of processes. Ethereum Virtual Machine(EVM) enabled this feature, and it is now widely used by many chains due to its high level of security and reliability from iterations of previous trials and errors. Omnygram network primarily applies EVM to make multi chain apps for its widespread use cases and reliability.

Cosmos[3] is a decentralized network of independent blockchains, designed to enable interoperability and scalability across different blockchain ecosystems. It was launched in 2019 and is built on a modular architecture that allows for the creation of customized, application-specific blockchains that can communicate with each other through the Cosmos Hub. They claim that they pioneered on establishing Transmission Control Protocol layer between blockchains, but they did not establish Internet Protocol layer between different blockchains with unified datagram. Omnygram network proposes internet protocol suite for relaying datagrams between blockchain accounts in different system.

Polkadot[4] is one of next-generation blockchain platforms that was created by Gavin Wood, one of the co-founders of Ethereum. Launched in 2020, Polkadot is designed to provide a scalable, interoperable, and secure platform for building decentralized applications (dApps) and blockchain-based services. While it claims to be a multichain platform, it is actually close to a vertical scaling solution of a blockchain and what **Ethereum 2.0**[5] should have been built. Questioning why they just built Ethereum 2.0 in the first place, the system is designed to scale vertically, but it is not considered to be applied in production due to its lack of developer support, horrible user experience and useless technocracy without considering actual adoption within the organization.

Polkadot having excessive arbitrariness on technical implementation without any guideline only causes problems. For instance, one of its parachain called Acala was introduced as a cross-chain decentralized finance (DeFi) hub that issues the aUSD stablecoin based on the Polkadot blockchain. Acala is built on Polkadot's blockchain framework Substrate which claims to be "swiss-army-knife" for blockchain developers, enabling to make application specific blockchains in whatever purpose. However, Implementing their system in Substrate's runtime module led the hack which caused depegging of aUSD by over 99%. An account based approach where the system cares about every individual must be preceded before drawing unimaginable possibilities. Engineering skills are meant to solve people's problems, not for making a new sovereign nation where devs stand above the law hiding on the lack of clarity.

Multi party computation(MPC) protocols with distributed keys such as **Nillion**[6] or **Lit protocol**[7] aim to enhance the security of users' wallets with versatile key generations on different blockchains for interoperability. However, these protocols introduce an element of interference in the user's transaction process. They rely on an off-chain client to not only process messages but also act as an execution layer on behalf of the users. Any failure to process transactions or loss of keys on the provider's side can lead to potential losses for the user.

LayerZero[8] offers a flexible method for exchanging messages between blockchain networks. However, it does not form a network between accounts. It also lacks specific provisions to ensure users' financial sovereignty and security as octets shouldn't be needed for their data model when data is stored in bytes in EVM, thereby leaving room for potential vulnerabilities to arise to its component called Endpoint. While it allows for multiple forms of communication, it still relies on off-chain clients to execute transactions on behalf of users with relayers on its custom proof, which introduces additional risks and uncertainties like Binance bridge. If the project claims to be a messaging protocol, relayers shouldn't involve themselves other than moving messages from one chain to another.

III. Decentralized Internet Protocol

This section describes the motivation of building internet of blockchains over cross-chain projects. There is an existing off-chain client called bridge which refers to an external wallet with smart contract to execute transactions of behalf of users on blockchain justifying to connect blockchains all together instead of accounts inside them. Current cross-chain bridges operate under the assumption that all blockchains can be interconnected individually. However, issues arise when attempting to connect a new chain to an asset for which a connection has already been established. However, this implementation must stop for the future.

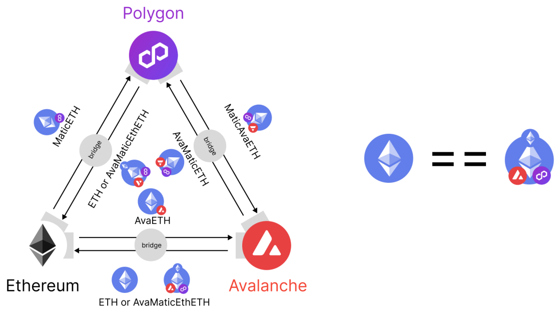


Figure 1. Semantic Satiation from Cross Chain Bridges

Semantic satiation is a phenomenon in which a word or concept loses its meaning or impact due to excessive repetition. In the context of blockchain interoperability, this phenomenon can occur when the idea of connecting Ethereum (ETH) across multiple chains is reiterated excessively. As depicted in Figure 1, this repetitiveness can diminish the significance of the concept and impede the development of effective solutions.

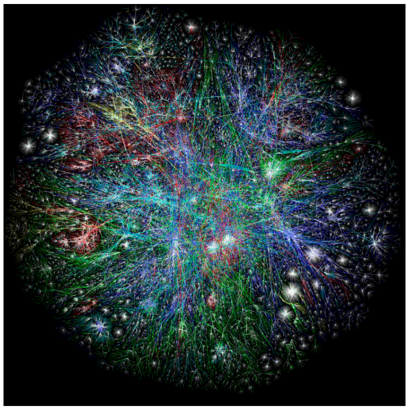


Figure 2. Map of Internet

Syntax satiation can also be observed in programming codes, particularly when enabling cross-chain transfers of Ethereum (ETH). There is at least 2 addresses of ERC20 bridged tokens in a network from multiple cross-chain bridge projects. This process introduces technical complexities arising from variations in smart contract addresses, diverse data models, and distinct transaction formats across chains and bridges. Such differences present a challenge for developers and users seeking seamless interoperability. Figure 2 illustrates the potential future of blockchain connectivity as more blockchains emerge. The resulting chaos would be unimaginable, overwhelming anyone attempting to establish connections due to the sheer volume of data and infrastructure intricacies that would need to be unraveled.

A. Decentralized Internet Protocol(DIP)

To address the problem of semantic satiation and improve overall digital asset connectivity across chains, a decentralized internet protocol where it considers each blockchain network as a subnet of blockchain accounts and unifies data model for communication is proposed. The difference would be that blockchains cannot send packet to each other being a set of network instead of a computer, so a relayer is needed to check finalized messages to be sent on each network. This paper defines packet like data stored in blockchains as ***Stacket(/ ˈstækɪt/)***. Blockchains also use bytes to define data, so the *stacket* has layouts in bytes instead of octets.

In the decentralized internet protocol,

1. A ***stacket* must** include a unique identifier of the destination blockchain.
2. A ***stacket* must** have encoded format which destination chain can decode and execute in its virtual machine.
3. A ***stacket* must** the block which has been submitted at the source blockchain state and the number of confirmation in order to be relayed when the data is finalized without fork.
4. A ***stacket* must** be able to be processed in local transactions.
5. A ***stacket* must not** be only executed by other fully on behalf of an individual.

Field	Description	Size (Bytes)
chainId	Represents the unique identifier of the destination chain.	2
confirmations	Specifies the number of confirmations on the source chain needed for the stacket to relay.	2
transport	Represents serialized transmission data for a stacket in a byte array with a flag in front. k is the total size of the data to represent valid transmission.	$1{+}k$
submittedAt	Indicates the block height when the packet was submitted in the source chain state.	32
payload	Represents an array of n ATTP stackets to be sent.	$\sum_{i=1}^{m}(\theta(M_j)+\sum_{j=1}^{n}\theta(\delta_{ij})+20)$

Table 1. DIP Stacket Fields and Descriptions

Table 1 specifies 5 fields to satisfy all the conditions. Function θ represents the function which returns the byte data size of its argument. Based on an argument's data type, Function θ is computed as follows:

$$\theta(x)=\begin{cases}\text{size}(x) & x\in\ \text{a primitive type}\\ \text{len}(x) & x\in\ \text{an array or tuple}\\ \sum\text{size}(p) & x\in\ \text{a struct or an object with properties } p\end{cases}\tag{1}$$

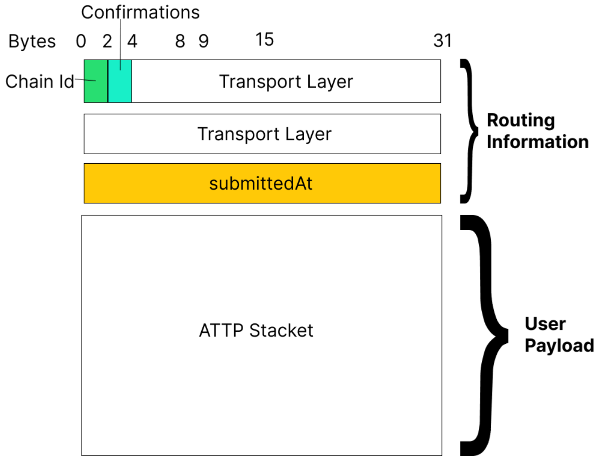
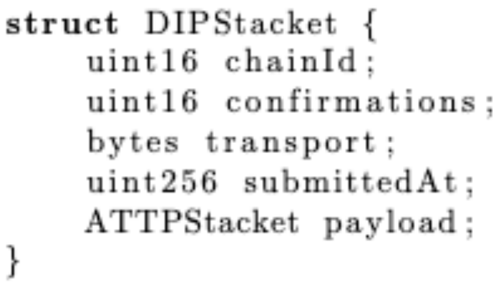


Figure 3. DIP stacket datagram

In Figure 3, DIP stacket has minimum 36 byte layout with header for routing information. DIP stacket encompasses ATTP packet as payload in its type on code implementation as below:



```
struct DIPStacket {
    uint16 chainId;
    uint16 confirmations;
    bytes transport;
    uint256 submittedAt;
    ATTPStacket payload;
}
```

Figure 4. DIP Stacket Implementation

C. Transport layer

The Transport layer is responsible for facilitating communication between accounts in different blockchains. Within this layer, there are two cases of transmission protocols: one for app-to-app communication and another for app-to-account communication.

i) App-to-app communication

In app-to-app communication, denoted as Case 1 (Transaction Communication Protocol), the Data link layer verifies if the sender's address, from which the stacket is stored, is accurately recorded. The Relayer then runs the receptor within the app to execute the received message. To ensure the stacket originates from the app address of the source chain, the app needs to verify the 'from' address. The Transaction Communication Protocol (TCP) utilizes ABI encoded bytes. DIP stacket layout with TCP protocol is illustrated in Figure 5.

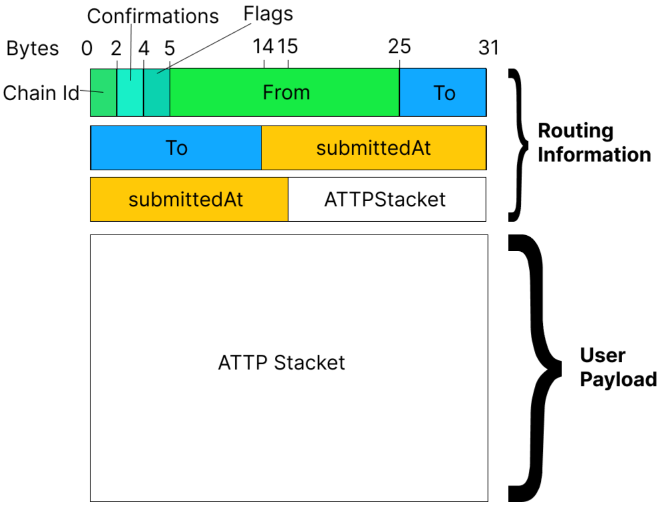


Figure 5. TCP Stacket Datagram

ii) App-to-account communication

In app-to-account communication, the process is for bundling multiple transactions from multiple apps together and delivering them to user. This User Delivery Protocol (UDP) allows for a streamlined approach where a relayer takes on the task of stacking ATTP packets within the alt-mempool of account abstraction, facilitating their execution. ATTP, which stands for AccountText Transaction Protocol, comprises ABI encoded bytes, forming a standardized format for seamless communication and interaction within the system. DIP stacked layout with UDP protocol is illustrated in Figure 6.

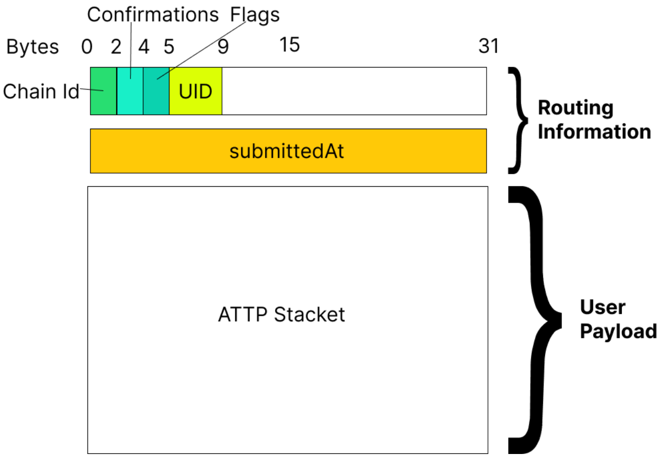


Figure 6. UDP Stacket Datagram

iii) Other communication methods

If there is i DIP packets to send with ATTP packets with j accounts to call methods to call with k arguments for each, the total stored data size for sending DIP packets G in an EVM blockchain is computed as follows:

$$G = \sum_{i=1}^{l} \left(\sum_{j=1}^{m} \left(\theta(M_{ij}) + \sum_{k=1}^{n} \theta(\delta_{ijk}) + 20 \right) + 41 \right) \tag{2}$$

Using (1), (2) the gas cost to send packets can be estimated. There are certain rate where each EVM networks charge per bytes of data. For instance, Ethereum yellow paper states that it costs 20000 gas for allocating 32 bytes size of words. If ε is constant gas cost for storing 1 byte of data, ρ is the gas price in Gwei, The total allocated gas cost T is computed as follows:

$$T = \rho * \epsilon * G \tag{3}$$

To optimize T on (3), G *can* be adjusted by the protocol to have minimum value. A separate smart contract can be built to provide stateless function to encode ATTP stackets into smaller data with compression or zero knowledge proofs. Relayer then will be able to either decompress data in its client or verify and send the transaction call data to other blockchain account.

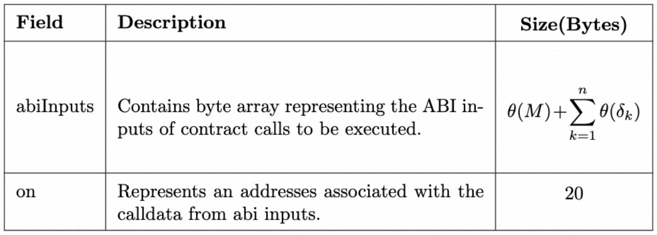
The default stacket flags within the system serve the purpose of distinguishing between different types of bitwise communication. Specifically, the flag "1" is assigned to indicate app-to-app communication, while the flag "2" is designated for app-to-account communication. In the event that new encoding and decoding methods for user payload are introduced, it is essential to provide two distinct cases, each accompanied by its respective flag. To ensure clarity, app-to-app communication should be flagged with an odd number, while app-to-account communication should be flagged with an even number. This systematic approach aids in effectively categorizing and identifying the nature of communication within the system.

Multichain Account Communicator smart contract is the contract to encode or decode data where G can be optimized. The contract acts as transport layer of TCP/IP protocol. It provides end-to-end connection between application layer and below. The difference is that while tcp focuses on transmission, the transport layer of DIP protocol focuses on optimizing data storage to transport data between blockchains.

B. AccountText Transaction Protocol(ATTP)

The AccountText Transaction Protocol (ATTP) draws inspiration from the widely used hypertext transfer protocol

(HTTP) and serves as the application layer within the TCP/IP protocol framework.



Field	Description	Size(Bytes)
abiInputs	Contains byte array representing the ABI inputs of contract calls to be executed.	$\theta(M)+\sum_{k=1}^{n}\theta(\delta_k)$
on	Represents an addresses associated with the calldata from abi inputs.	20

Table 2. ATTP Stacket Fields and Descriptions

Table 2 shows the fields and size of each payload.

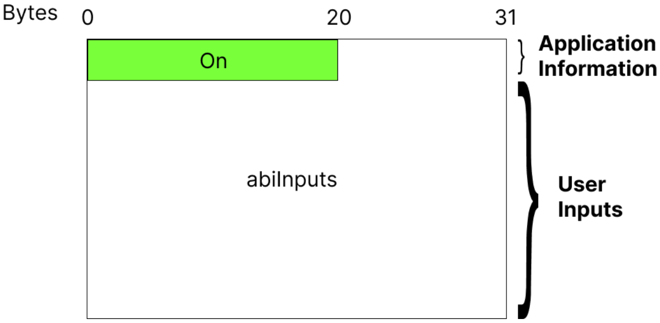
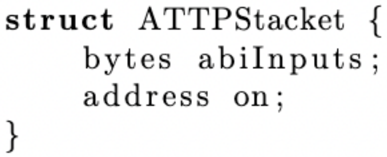


Figure 7. ATTP Stacket Datagram

In Figure 7, ATTP stacket has a header for application information and user inputs to interact with. The data type in programming code is implemented as below:



```
struct ATTPStacket {
    bytes abiInputs;
    address on;
}
```

Figure 8. ATTP Stacket Implementation

D. Link Layer

Network interfaces of DIP protocol is formed with stacket queue contract which stores DIP stacket and relayer to poll. Similar to event queues in distributed system, it coordinates the order of DIP stackets to be delivered on each source chain. Only registered app which uses DIP protocol verified by Omnygram community can store DIP stackets in the queue.

E. Physical layer

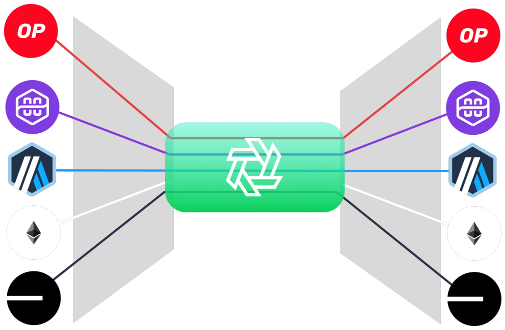
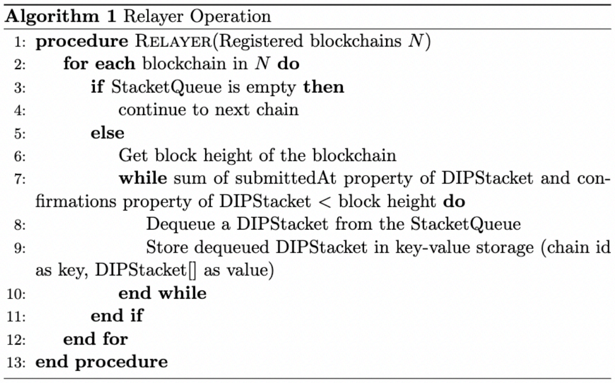


Figure 6. Concept Map of Relayer as Multiplexer

Relayer takes part of the physical layer of DIP stack same as physical layer of TCP/IP. It relays DIP packets stored in stacket queue contracts in each blockchains and send them to destination. Relayer relies on the trustless setup assuming that each blockchain state is finalized and immutable after a consecutive number of blocks are stored in the network regardless of its consensus algorithm and the trusted setup by ensuring that no single entity has control over the initial cryptographic keys or parameters.



```
Algorithm 1 Relayer Operation
procedure RELAYER(Registered blockchains $N$)
    for each blockchain in $N$ do
        if StacketQueue is empty then
            continue to next chain
        else
            Get block height of the blockchain
            while sum of submittedAt property of DIPStacket and confirmations property of DIPStacket $<$ block height do
                Dequeue a DIPStacket from the StacketQueue
                Store dequeued DIPStacket in key-value storage (chain id as key, DIPStacket[] as value)
            end while
        end if
    end for
end procedure
```

Trustless setup in relayer can be displayed as Algorithm 1 to check confirmations of block produced by the blockchain.
On the other hand, trusted setup relays stackets with multiple trusted clients depending less on block confirmations believing that the ones running clients have enough funds to cover the loss by miscommunications.

F. Account Abstraction

Account abstraction[9] in DIP protocol stands above all layers and send/receive stackets to other abstractions across blockchains. Account abstraction is the concept to have an abstraction of a wallet for better user experience. The abstraction can enable smart contract accounts to execute transaction representing its owner as sender in EVM.

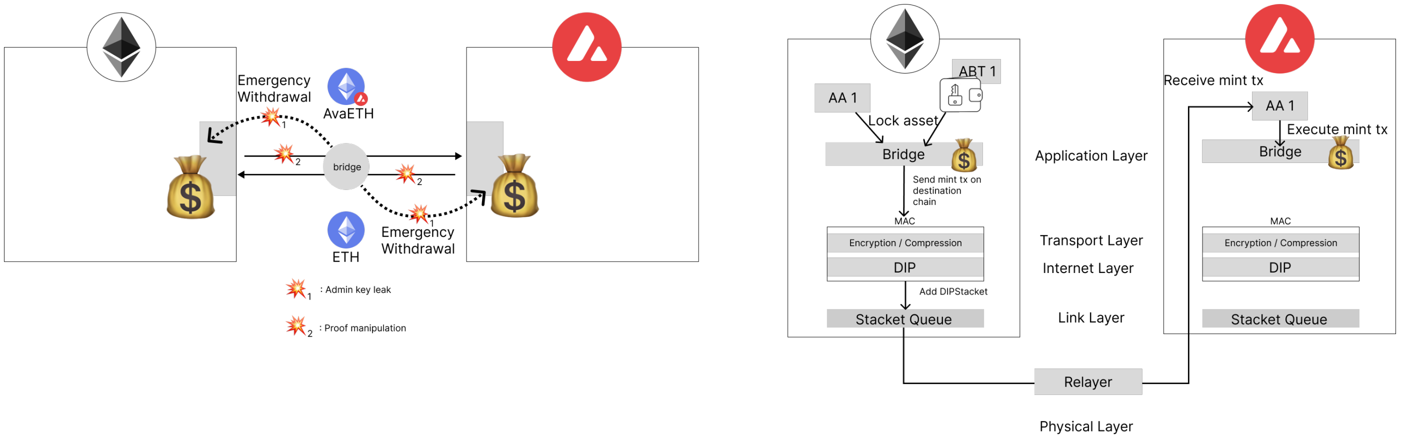


Figure 7. Attack Vector Comparison between Existing Bridge Architecture and Bridge on Top of Decentralized Internet Protocol

The difference between account abstraction in DIP and **ERC4337**[10] is that its alt-mempool consists of DIP stacket. To execute contract calls in received stackers, account abstraction holder bounded with account bound token executes the stacket in the mempool. Algorithm 2 specifies the process how DIP stacket is processed.

```
Algorithm 2 ProcessDIPStacket
recv := DIPStacket[]
function PROCESSDIPSTACKET(index)
    CTTPStacket[] cttpArray := recv[index].payload
    for i in cttpArray do
        if address(i.on).call(i.abiInput) then
        else
            Revert Error(”CTTP tx error”)
        end if
    end for
    Revert ”Item not found”
end function
```

Another significant difference lies in the implementation of non-fungible tokens (NFTs) for verifying ownership of the account abstraction. Each NFT is assigned a distinct identifier and is associated with the account abstraction. These NFTs, known as account bound tokens (ABTs), are managed by the ABT registry contract within the DIP protocol. This contract oversees the management of user ‘account abstractions. In the Omnygram network, this mechanism ensures that a user retains a consistent identifier across all interconnected blockchain chains.

IV. Design

This section provides a clear explanation of the design of the DIP protocol in comparison to other cross-chain bridge projects. It also explores the various use cases of the protocol. The primary emphasis of the design is on bridges for comparison, as they are widely utilized for achieving interoperability between different blockchains. Additionally, the section introduces a novel design for applications other than bridges in the context of cross-chain use cases.

A. Trustless Validation with On-Chain Native Proof

Validations on cross-chain bridges rely on wallet and smart contracts with shared user funds to verify and execute transaction on behalf of users. Executing transaction on behalf of users related with their digital assets always caused massive security hack.

Nomad[11] bridge lost 150 million dollar, because the smart contract with shared funds was giving asset users' deposit in the destination chain with the amount of 100 times more from than the actual deposited amount from source blockchain. Additionally, the hack was able to replicate, and hundreds of users drained the fund in the destination chain. Ronin bridge lost 650 million dollars with wallet managing bridge hacked from email, stealing private keys from the bridge developer. The hacker used it to approve admin action to withdraw shared users' funds which was meant only to be used in emergency. Binance bridge hack lost 570 million dollar from validating proof in their relayer connected with wallet. The hacker manipulated the proof from only one smart contract's business logic to validate, making it to deliver users' funds to the hacker. To stop the hacks from bridges, the bridge must validate flow of funds only on chain without relying on off chain clients such as relayers. Relayers must not store user's funds and only relay verified messages from the smart contract without administrative methods to move users' shared funds.

Figure 7 provides a comparison between attack vectors in existing bridges and the DIP (Decentralized Interoperability Protocol) bridge. The DIP protocol introduces a distinct approach to validation, where tasks are carried out on the dapp contract without requiring the generation of proofs. Instead, network validators verify the validation process through block confirmation. Proofs are already generated and verified during block finalizations, leveraging the existing maintainers of each blockchain network. Notably, the admin keys of the relayer are exclusively utilized for moving stackets and have no authority over users' assets. In the event of a potential hack of the admin key, the worst-case outcome would involve the relayer losing a client for relaying purposes, without any adverse effects on users' funds. This effectively mitigates off-chain risks arising from operational security vulnerabilities. Omnygram governance assumes the responsibility of managing the stacket queue, thereby eliminating the risk associated with admin keys. As a result, the bridge app can focus solely on verifying its own business logic, minimizing potential vulnerabilities even further.

The process of the bridge interacting with DIP consists of five steps:

Step 1: The Omnygram bridge app verifies whether the user has deposited and locked their assets in the contract. If so, it sends a remote transaction to mint the asset on the destination chain. The bridge app generates ABI input and sends it to the multichain account communicator contract to encode the data with a specific type.

Step 2: The MAC contract is called to encode the data using a particular encryption or compression algorithm. The MAC assigns the corresponding transaction type flag to the DIPStacket and sends it to the stacket queue contract.

Step 3: The stacket queue contract verifies that the source of the DIPStacket is registered and stores it in its queue. The stacket remains in the queue until the relayer detects and dequeues it.

Step 4: The relayer decodes the encoded data in each ATTPStacket and constructs transactions to be executed in the account abstraction. The relayer retrieves the account abstraction contract address from the ABT registry contract and sends the decoded ATTPStackets to the abstraction contract.

Step 5: The user with an ABT holding wallet calls the account abstraction contract to execute the ATTPStacket, resulting in the minting of assets to the wallet.

B. *Sovereign Apps*

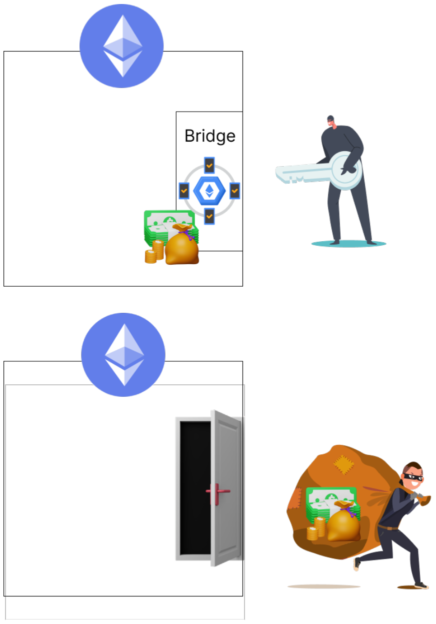


Figure 9. Decentralized App hacking

A smart contract as an account has two kinds of data storages, one for storing business logic to operate funds, and the other for storing cryptocurrency balances. Current decentralized apps have business logic and account balances together, and hackers use its business logic to move funds from the smart contract to themselves. Decentralized app can be compared to a pile of money stored in a house with one door to be breached as shown in Figure 9.

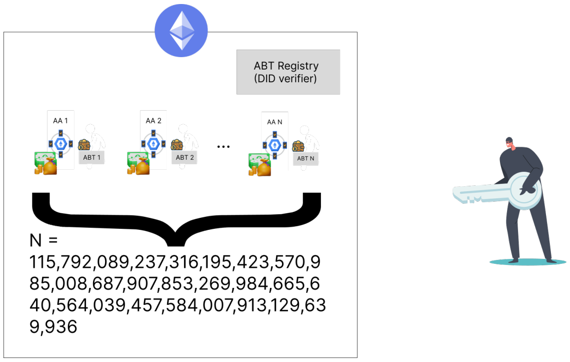


Figure 10. Sovereign App Hacking

Sovereign apps separate business logic in accounts per its user with customized smart contract accounts and limits access to the shared funds based on each user's deposit. This makes hackers turn from grand thief of whole users' fund into pickpocket for one. As blockchains discourage hackers to manipulate its state by dealing with enormous amount of blocks, Having unimaginable number of accounts makes computationally impractical to hack, and operation security team can buy enough time to investigate. Being an account abstraction, the DIP integrated app accounts can interoperate across blockchains without facing bridge hacks with enormous amount of loss of fund. Sovereign app can be compared to passbooks stored in an apartment where there are numerous doors to be breached with customized keys for each resident as shown in Figure 10.

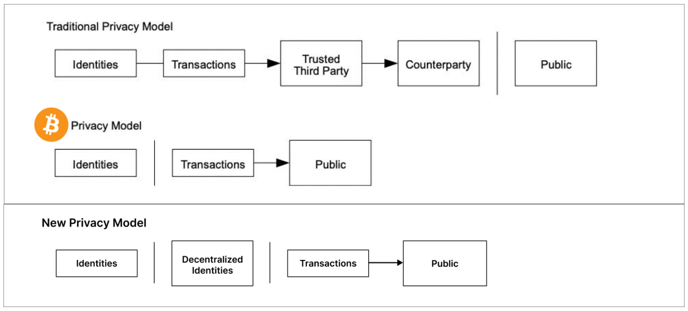


Figure 11. New Privacy Model

Current design only shows an application for cross-chain asset transfer, but having an app account enables to add more business logics to track users activity and add another identity layer. Having a record of user activity in an app account can build credits which can be used for not limited to loans based on its credit score, membership benefits, gamification, social proof. Manifold opportunities are available without revealing who the user is. The app only knows there is a user with certain id. Figure 11 shows new privacy layer from Bitcoin by Omnygram network.

The concept of achieving self-sovereignty over self custody in smart contract accounts aligns with the ethos of Bitcoin. In 2019, Jimmy Song, a Bitcoin educator, emphasized that decentralization is not a spectrum and that users should have control over their assets without reliance on a central entity. Unfortunately, neglecting this advice had negative consequences, as evidenced by the case of Terra. Terra's Anchor Protocol, created by a central entity called Terraform Labs, attracted capital through deceptive practices, including the use of bot accounts to manipulate the total value locked in the app. This behavior resembled the fraudulent actions of Bernie Madoff's hedge fund, which falsely inflated its profit growth. Despite warning signs that Terra was centralized, many individuals invested in its Ponzi scam without considering the importance of maintaining personal control within the decentralized app (dapp). Ultimately, the founder of Terra, Do Kwon, took advantage of the situation by selling $LUNA in the market, leveraging the liquidity provided by speculators. Instances like Terra highlight the need to prevent such centralized dapps and encourage the development of new growth strategies that prioritize self-sovereignty. It is crucial to ensure that self-sovereignty remains a fundamental principle, regardless of the circumstances.

The potential for gamification and the implementation of self-sovereign accounts for both users and bots opens up new possibilities for sustainable growth strategies. Rather than relying on Ponzi-like economics, this approach encourages fair competition among users on the blockchain, fostering personal growth. Increased competition not only benefits individual users but also generates more revenue for app developers. With higher rewards at their disposal, developers can effectively attract a larger user base, creating a positive cycle of growth.

Standard Protocol[12] plays a crucial role in implementing sovereign apps by fully utilizing the capabilities of the Omnygram network. This collection of apps provides a robust infrastructure within the multichain ecosystem. One key component is Somad, which acts as a bridge for securely locking and minting assets. It enables seamless transactions across different blockchains while implementing stringent accounting measures to enhance security. Somad account serves as a passport between blockchains, facilitating smooth asset movement. Additionally, Safex operates as a fully decentralized order book exchange, ensuring transparency and security in trading activities through personal trader accounts. Another notable feature is SAFU, an overcollateralized stablecoin that goes beyond being solely tied to the US dollar. It leverages self-custodial smart contract accounts to effectively manage risks, distribute collaterals in the market to enable its use as a medium of exchange, and maintain alignment with the value of various fiat currencies. Together, these elements contribute to the establishment of a comprehensive and resilient financial ecosystem in Omnygram network.

V. Economics

A. *$OMNY*

$OMNY(/ˈɑːmniː/) is the native currency of the Omnygram network, which powers the network's multichain self-sovereign account abstractions and sovereign apps. There are two ways to acquire $OMNY: purchasing it from existing holders or participating in the validation of blocks on the Omnygram network. This currency is typically used by subscribing to membership plans to maintain a Omnygram account and access the internet of blockchain created by the Omnygram network. Holders of $OMNY have the option to stake their tokens with network validators, granting them voting rights within the Omnygram economy.

B. *Dynamics*

The Omnygram network serves as a blockchain platform that facilitates remote transactions across multiple blockchains using the cryptocurrency $OMNY. To become a central hub, the network needs to exist for a longer duration compared to others. However, several obstacles hinder its progress. Some self-proclaimed investors are expressing discontent over their losses and advocating unnecessary regulations, which pose a threat to the development of an unbiased, connected, and transparent blockchain-based economy. It is crucial to prioritize rewarding actual users over arbitrageurs.

Many cryptocurrencies experience significant value depreciation, often resulting in a 99% loss. Since cryptocurrencies are not classified as securities, venture capitalists who sell their holdings at a 20% price reduction should not be held accountable. Cryptocurrencies, including Bitcoin, was not originally designed for profit generation. Instead, people should view them as currencies to be used rather than investments for monetary gain. The potential for extraordinary returns, such as 100x gains, stems from the intrinsic value of financial freedom and self-sovereignty offered by an open global financial system.

However, the true value of cryptocurrencies is being undermined by unsustainable economic practices, including projects built on fake foundations or toxic trades initiated by early adopters. Whales who became overnight billionaires due to Bitcoin or venture capitalists who lack understanding of stocks and solely prioritize displaying remarkable returns on liquidity providers often engage in mass cryptocurrency sell-offs, irrespective of market growth. This harmful behavior is prevalent across other cryptocurrencies and is difficult to counteract, especially since cryptocurrency is not classified as a security. This rampant practice has resulted in significant losses for individuals and has prompted regulators to overprotect the cryptocurrency industry.

Cryptocurrencies are often not classified as securities and may face regulatory challenges that limit their protection. In

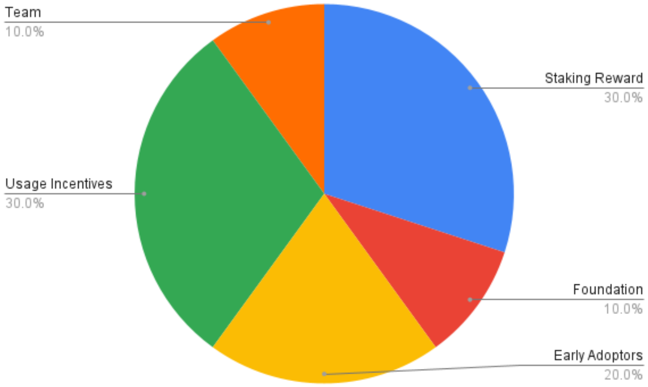


Figure 12. $OMNY Initial Supply distribution

light of this, it is worth considering an approach that ensures sustainability by establishing a constant demand and continuous use case, rather than solely relying on selling activities. The team behind $OMNY views it as both a commodity and a currency, functioning as a network currency within the ecosystem. To preserve the functionality and value of the blockchain, two metrics are implemented, serving as important factors in maintaining the long-term viability of the $OMNY network.

The first metric is the *turnover rate*. $OMNY should function as a widely circulated currency to build trust. The turnover rate represents the percentage of tokens used in continuous demand, such as through membership subscriptions, compared to the total token supply per month. A high turnover rate indicates a tendency for the $OMNY token's value to increase. If the turnover rate becomes excessively high and affects gas prices, scaling the decentralized app or increasing the $OMNY supply may be necessary to attract more users into the $OMNY economy. Conversely, a low turnover rate suggests a tendency for the $OMNY token's value to decrease. In such cases, proposals such as burning $OMNY supply or reducing rewards can be considered to retain users in the $OMNY economy.

The second metric is the *Gas Price Index(GPI)*. To maintain the internet of trust, $OMNY should serve as a commodity. One important aspect is the usage of $OMNY as a commodity within the network. Effective gas cost management is essential for sustainable growth. With a Gas Price Index of 1, $OMNY as a utility token for network services remains unaffected. The Gas Price Index is calculated as 65000 * (150 GWei) * ($OMNY/USD price) equals 1 USD.

If the Gas Price Index is greater than 1, $OMNY experiences deflation, indicating a higher value. However, this high gas cost impedes the growth of dapps and the overall economy. To address this, $OMNY may consider providing more $OMNY rewards to users, incentivizing greater usage and rewarding network validators.

If the Gas Price Index is less than 1, it indicates that $OMNY is experiencing inflation, resulting in a decrease in its value. This inflationary environment allows for the expansion of dapps and the economy, creating space for growth. More $OMNY tokens are needed to facilitate trading with other currencies. However, it is important to avoid prolonged inflation, as it can lead to a situation called stagflation, where both $OMNY and its economy depreciate. To address this, adjustments are made by decreasing $OMNY rewards and network validator rewards. Additionally, there is an encouragement for leveraging $OMNY to stimulate economic activity and counteract the negative effects of stagflation.

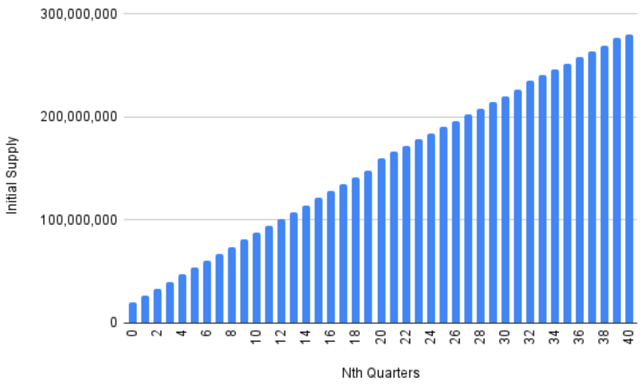


Figure 13. Estimated Initial Supply Dilution

By closely monitoring these two metrics, turnover rate and Gas Price Index, the Omnygram network aims to maintain a sustainable and balanced ecosystem. A high turnover rate ensures widespread circulation and trust in $OMNY, while an optimized Gas Price Index promotes effective gas cost management for network services. The goal is to create an interconnected and resilient blockchain economy that rewards active participants and supports continuous growth.

It is imperative to overcome obstacles posed by discontented investors and unnecessary regulations. By prioritizing actual users, valuing long-term use cases, and carefully managing economic dynamics, Omnygram strives to establish itself as a hub for blockchain connectivity, fostering an unbiased, connected, and transparent economy. Through these efforts, the network aims to enhance the value and functionality of $OMNY while driving the adoption and advancement of decentralized technologies.

C. *Initial Supply*

The early distribution and market dynamics of $OMNY, the native cryptocurrency of the Omnygram network, are pivotal factors in determining its value and liquidity. These early distribution parameters significantly impact the economic metrics of $OMNY.

Figure 12 shows the early distribution of $OMNY. Liquidity mining and Airdrop portions may vary based on the gas price index and turnover rate determined from circulating supply. Figure 13 shows the initial supply vesting schedule of early adopters and the foundation. It can have an impact on the value of $OMNY and the ecosystem's overall liquidity. The vesting schedule determines when and how much of the

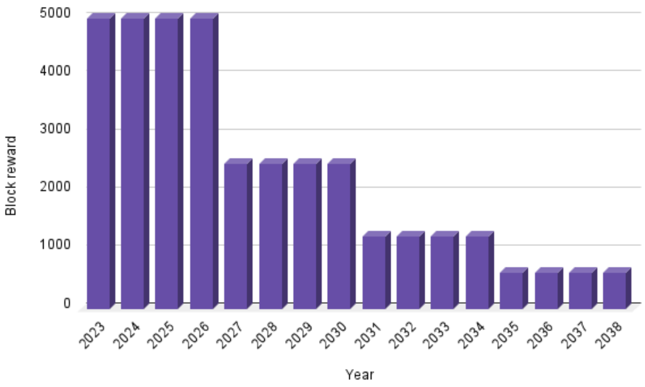


Figure 14. Block rewards of $OMNY

tokens are released into circulation, which affects the supply and demand dynamics of the market.

If the initial vesting schedule allows for a large number of tokens to be released into the market quickly, this could lead to an oversupply of $OMNY, resulting in a decrease in its value. On the other hand, if the vesting schedule is too restrictive and releases too few tokens, it could result in a lack of liquidity and difficulty in trading $OMNY.

In addition, the initial vesting schedule can also affect the distribution of tokens among early adopters and the foundation. If the vesting schedule favors the foundation, this could lead to a concentration of tokens in the hands of a few entities, potentially leading to centralization issues. Conversely, if the vesting schedule is more favorable to early adopters, this could help to distribute the tokens more evenly, creating a more decentralized ecosystem, but there are more possibilities that early adopters sell off.

The availability of airdrops and liquidity provision can also impact the value of $OMNY. Airdrops, which involve distributing tokens to users for free, can increase demand and create a sense of community within the ecosystem. However, if airdrops are too frequent or too large, this could lead to oversupply and decrease the value of $OMNY. The worst case would be users grifting airdrops just to sell for profit.

Liquidity provision, which involves providing access to liquidity pools for trading $OMNY, can increase the availability of $OMNY and make it easier for users to trade. However, if liquidity provision is too low, this could result in difficulty in trading and reduce the overall value of $OMNY.

Therefore, it will be crucial to understand how each emission from initial supply affect turnover rate and gas price index in Omnygram network.

D. *Inflation*

Bitcoin, the pioneering cryptocurrency, was designed to have a maximum supply of 21 million coins. As of May 2023, approximately 19 million bitcoins have already been mined, with the remaining 2 million bitcoins set to be gradually released over the next century through mining rewards. Once the maximum supply is reached, no new bitcoins will be created, and people will acquire bitcoins solely through purchasing them from existing holders. This limited supply of bitcoins contributes to its value storage mechanism. Similarly, the Omnygram network introduces new $OMNY as rewards to network validators who successfully finalize blocks based on the **Tendermint**[13] consensus algorithm. The initial supply of Omnygram is 1 billion, with a maximum supply of 2.1 billion $OMNY. Figures 14 and 15 illustrate the approximate time frames for block rewards and inflation rates, assuming a 6-second finalization time. Once the maximum supply is attained, no further $OMNY will be minted, and the network will rely on transaction fees and protocol revenues to reward network validators. The challenge for the Omnygram network lies in maintaining a stable status, where a sustainable turnover rate can cover block rewards based on the gas price index. If successful, $OMNY will be recognized as a valuable commodity and currency within the internet of trust.

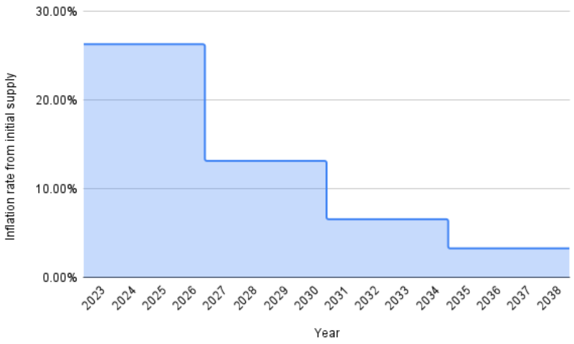


Figure 15. Inflation rate of $OMNY

VI. Future works

In light of the findings presented in this new system of interoperability, several promising avenues for future research in the field of renewable energy can be identified. Firstly, further investigation is warranted to explore the extension of Stacket to be compatible with non-EVM chains or smart contracts. Stackets can be encoded into byte arrays, enabling multiplex communication with a customized decoder in the relayer client. The use of encoded byte arrays in the ATTP Stacket could be a breakthrough for compatibility. Ordinals inscriptions could also leverage Stacket to move assets from Bitcoin to other chains. Additionally, the economics of $OMNY as a commodity currency represents a crucial area for future exploration. Research should focus on optimizing the emission and operation of currency liquidity to enhance its value stability, enabling a more sustainable environment where true users benefit over short-term speculators. By addressing these future research directions, we can accelerate the transition from a fragmented, chaotic cross-chain ecosystem to a sustainable and reliable financial system for everyone with three phases.

Phase 1: A multichain liquidity app. The Omnygram network is expected to undergo three phases of growth. In Phase 1, utilizing the DIP protocol, it will initially apply for a bridge on the omni liquidity of an asset. The use of the Standard protocol's fully on-chain orderbook decentralized exchange Safex, along with its first app-to-app communication implementation bridge Somad, will ensure that all bridged Omny assets are valued the same as native assets, providing arbitrage opportunities across the entire EVM ecosystem. A multichain transaction scanner and dashboard will be utilized to compare bridges based on cross-transfer time and gas costs, providing valuable research insights. The DIP protocol's reliance on block finalization for proving validity is expected to outperform other solutions in terms of time and gas costs.

Phase 2: Composable multichain transaction. As Omnygram progresses, it will establish its own mainnet to accommodate sovereign apps and advanced account abstraction methods. The launch of the Omnygram app will allow users to maintain their cross-chain account-bound tokens and assets across multiple chains. Users will be able to compose batches of multichain transactions and send them to account abstractions. Account abstractions will execute these transactions and provide notifications. The combination of Omny liquidity, money market, and future market will further enhance the growth of the network.

Phase 3: Omni layer of multichain. Omnygram, with its unified liquidity and the ability to compose multichain transactions, will introduce a new paradigm for dApp development. Sovereign apps leveraging multichain transactions will emerge, utilizing a build-to-earn mechanism based on the accounts sovereign apps generate. The multichain transaction capability will accommodate more users, allowing app developers to horizontally scale their web3 systems. For those seeking to develop multichain apps for mass adoption, Omnygram will be the preferred solution.

VII. Conclusion

In conclusion, the existing cross-chain projects encounter significant challenges in terms of liquidity, user experience, and security. These projects often prioritize segregation and accounting of assets, which can lead to liquidity losses and vulnerability to hacking incidents. Despite attempts to create bridge protocols, the user experience remains subpar, leaving users uncertain about the finalization of their transactions. To solve these issues, Omnygram network's proposed solution advocates for a multichain with decentralized internet protoocl and account abstraction. This approach offers distinct advantages such as financial sovereignty, transparency, and sustainability. By employing smart contract accounts bounded with NFT and low-level compatible internet protocol between blockchains, the multichain approach facilitates the seamless movement of assets across multiple blockchains, while the introduction of multichain dApps and a multichain account ID simplifies asset management for users. In summary, the proposed multichain approach with account abstraction presents a more viable solution to overcome the challenges faced by current any other cross-chain projects.

References

[1] Nakamoto, S. (2008). Bitcoin: A Peer-to-Peer Electronic Cash System. Retrieved from https://bitcoin.org/bitcoin.pdf

[2] Wood, G., et al. (2014) Ethereum: A Secure Decentralized Generalised Transaction Ledger. Ethereum Project Yellow Paper, 151, 1-32.

[3] Kwon, J., & Buchman, E. (2016). *Cosmos: A Network of Distributed Ledgers* (Whitepaper). Retrieved from https://v1.cosmos.network/resources/whitepaper

[4] Wood, G. (n.d.). Polkadot: Vision for a Heterogeneous Multi-chain Framework, Draft 1. Retrieved from https://polkadot.network/whitepaper/

[5] Ethereum Foundation. (2019). Ethereum 2.0: Serenity. Retrieved from https://ethereum.org/eth2/

[6] M. de Vega, A. Masanto, R. Leslie, A. Yeoh, A. Page, and T. Litre, "Nillion: A Secure Processing Layer for Web3," 2022.

[7] Lit Protocol Contributors. (n.d.). Lit Protocol GitBook. Retrieved from https://developer.litprotocol.com/

[8] Zarick, R., Pellegrino, B., & Banister, C. (2021). LayerZero: Trustless Omnichain Interoperability Protocol.

[9] Ethereum. (2023). Account Abstraction. Retrieved from https://ethereum.org/en/roadmap/account-abstraction/

[10] Vitalik Buterin (@vbuterin), Yoav Weiss (@yoavw), Kristof Gazso (@kristofgazso), Namra Patel (@namrapatel), Dror Tirosh (@drortirosh), Shahaf Nacson (@shahafn), Tjaden Hess (@tjade273), "ERC-4337: Account Abstraction Using Alt Mempool [DRAFT]," *Ethereum Improvement Proposals*, no. 4337, September 2021. [Online serial]. Available: https://eips.ethereum.org/EIPS/eip-4337.

[11] Nomad. 2022. Nomad Docs. Retrieved from https://docs.nomad.xyz/

[12] Kang, H. (2023). Whitepaper 2.0. Retrieved from https://github.com/standardweb3/Whitepaper/blob/main/whitepaper_en.pdf

[13] Buchman, E., & Kwong, K. (2014). Tendermint: Consensus without Mining. Retrieved from https://tendermint.com/static/docs/tendermint.pdf

[14] WebAssembly. (2016). Retrieved from http://webassembly.org/

[15] ECMA International. (2017). ECMAScript® 2017 Language Specification: JSON (2nd ed.) [Standard]. Retrieved from https://www.ecma-international.org/publications/files/ECMA-ST/ECMA-404.pdf

[16] Ordinals Handbook. (2023). Retrieved from https://docs.ordinals.com/