

SigWorld: a decentralized communication protocol*

SigWorld Lab

May 19, 2023

Abstract

The blockchain ecosystem has experienced tremendous growth ever since the advent of Bitcoin and Ethereum. With the evolution of decentralization, fragmentation has become increasingly prevalent from infrastructural level to application level. As a result, a new challenge has emerged: how can digital identities(DIDs) communicate with each other in a reliable manner within this fragmented digital realm? This challenge extends to all forms of communication, such as DID-to-DID (wallet-to-wallet), dapp-to-dapp, chain-to-chain, wallet-to-dapp, wallet-to-chain, and many more.

In this article, we present a novel decentralized, trustless communication protocol called SigWorld, which solves the inter-DID communication problem by using blockchain for governance, and ensuring full reachability and messaging chronology in a decentralized network. SigWorld is comprised of a communication layer and a governance layer. A blockchain is used for node discovery, incentivization, and node governance. At communication layer, all postal nodes are obligated to construct messages in a hash-based message chain between any DIDs, including those between postal nodes. SigWorld protocol is the first protocol to enable reliable and verifiable message delivery in a decentralized, trustless network. This protocol has great potential for a wide range of applications and scenarios, including decentralized messaging apps, decentralized social networks, IoT communication, financial transactions, online gaming, etc. Its strong consistency guarantees, digital signature-based security, and scalability make it a promising solution for the communication in the digital age.

1 Introduction

Communication is an essential aspect of human interaction. The digital disruption of media and society has altered the way people communicate, and as IoT devices and AI agents are becoming ubiquitous in our life, everyone, everything, is a living digital identity in the hyper-weaved Web. Most digital identities are forced to communicate with each other via someone else's service. This "someone else" is all digital identities have to trust and rely on. However, existing communication platforms are centralized and vulnerable to attacks, censorship, and data breaches. Safeguarding communication rights in the digital era is a complex and challenging task. The advent of blockchain technology offers a real opportunity for change.

The emergence of Bitcoin and Ethereum has demonstrated the viability of sustainable decentralized ecosystems. These systems decentralize banking and financial services by returning ownership of money to users through asymmetric encryption keys. Despite introducing fragmentation problems at both the infrastructural and application levels, which hindered smooth and reliable communication between different parties, decentralized ecosystems have shed a light upon building a trustless and decentralized communication network.

In this article, we propose a decentralized communication protocol called SigWorld, which stands for "Signature-secured World". SigWorld provides end-to-end security, chronological messaging, and full reachability within a decentralized communication network. It leverages an underlying blockchain to incentivize and govern the postal nodes, manage user-owned identities, and provide security for the decentralized network. Figure 1 shows a general overview on how SigWorld works. SigWorld offers a secure, reliable, and scalable solution for a wide range of inter-DID communication scenarios, including messaging apps, IoT devices, decentralized social networks, decentralized financing, and more.

In the following sections, we will present related work in Section 2, describe the core concepts and design of SigWorld in Section 3. Security considerations will be thoroughly discussed in Section 4. In Section 5, we'll explore several application scenarios to demonstrate SigWorld's potential to transform how we communicate and collaborate. Then we conclude with a discussion of future work in Section 6 and conclusions in Section 7.

*This article is based on our patented work, as described in CN113595736B.

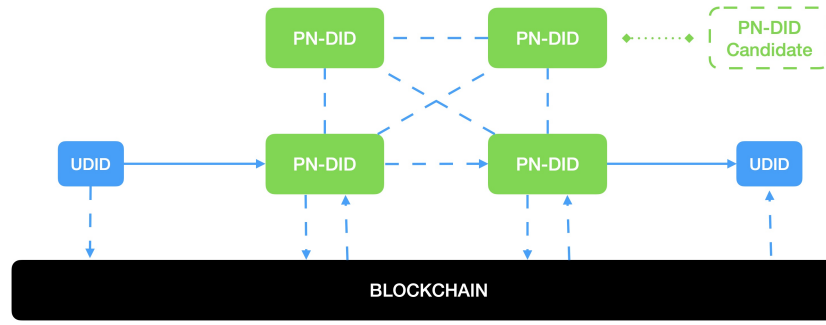


Figure 1: SigWorld Decentralized Network

2 Related Work

The decentralized messaging protocol we propose is designed to provide a secure, reliable, and scalable messaging solution. The protocol is based on a distributed network architecture that enables decentralized messaging without a central server. Our work builds upon several valuable prior research and industrial efforts in this area that have inspired our protocol design. We review related work on similar protocols and products, and justify how SigWorld extends existing approaches. Most of these works adopt peer to peer communication, suitable for small-scale messaging but potentially inadequate for large-scale adoption. An essential feature is yet to emerge that would propel these works into widespread use by the general public.

Matrix *Matrix* [1] is an open source project that publishes the Matrix open standard for secure, decentralized, real-time communication. It aims to create an open platform for communication that is as independent, vibrant, and evolving as the Web itself. Each Matrix server has full sovereignty over its users' data. Each server manages registration and data for its own users. Our work mitigates potential data control risks by storing user registration and preference data onto the underlying blockchain, thereby making servers interchangeable, and granting users full control over their data.

Marlin *Marlin* [2] is an open peer-to-peer communication protocol, that allows users with spare infrastructure to share their computational resources with application developers. Its objective is to build server-less backends for Web3 industry. While serving the underlying network infrastructure of cryptocurrency miners, Marlin's network and token design could be extended to general use cases.

Status *Status* [3] is an open source messaging platform and mobile interface to interact with decentralized applications that run on the Ethereum Network. Status utilizes Ethereum's Waku protocol for clients to initiate peer to peer communication. It also includes a crypto wallet and a growing ecosystem of DApps including marketplaces and exchanges. By leveraging the existing Ethereum ecosystem, Status may lower the learning curve for Web3 users. However, additional improvements may still be needed to simplify the user experience for mainstream adoption.

XMTP *XMTP* [4] is an open protocol, network, and standards for secure, private web3 messaging. It allows use cases of sending data between web3 identities, including app to user notification, user to user messaging, and public announcements. While XMTP focuses on Web3 identities and messaging, our work has a broader scope by supporting communication between any digital identity.

Nostr *Nostr* [5] is a simple, open protocol that enables global, decentralized, and censorship-resistant social network. By removing the need for a centralized social media platform, Nostr seeks to give users full control over their data and provide a forum for open expression. However, fully realizing Nostr's vision of a decentralized social network at web-scale, still faces challenges like content recommendation, trust networks, and content monetization, in a decentralized context.

3 SigWorld Protocol

In this section, we will describe the fundamental parts of SigWorld protocol, and how they interact to form a reliable and practically feasible decentralized communication network. SigWorld protocol consists of two main layers, a decentralized communication network, and a underlying decentralized blockchain. Postal nodes

provide the messaging delivery and routing service for all digital identities (DIDs). They form the backbone of the communication network. The underlying blockchain acts as a common registry for identities of both postal nodes and users. Participants also use the blockchain as a fallback channel for messaging in addition to communication via postal nodes, as shown in Figure 2.

Domain Name Upon registration, each postal node is automatically assigned a domain name that resolves to the node's DID. Domain names persist even if the underlying postal node changes, enabling replacement nodes to serve under the same domain. Each user DID is also assigned a domain name under the domain of a postal node.

3.1 DID Registration

As we mentioned, there are two roles in this network, postal nodes (PN-DID) and users (U-DID). In a semi-open postal network, PN-DIDs will be registered with approval from the governance committee. User DIDs must first register under one or more postal nodes before they can send and receive messages within this network. Any two DIDs can securely and privately exchange signed messages in this network.

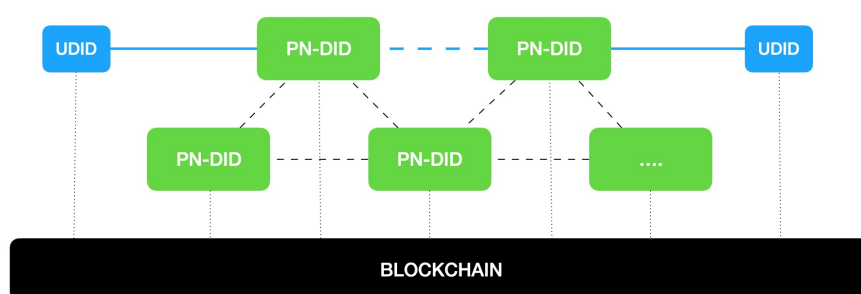


Figure 2: DID Interaction in SigWorld

Registration of PN-DID and U-DID can be defined respectively as follows:

- registerPN: (pnPubKey, pnDomain, keyScheme, pnMeta);
- registerUser: (userPubKey, pnDomain, keyScheme, userMeta);

Note that DIDs are uniquely defined by their public keys, regardless of their IP, geo-location, or any other information that may correlate to their real world identities. Keys, once registered, can be updated independently by DID key owner.

3.2 Postal Node (PN)

Postal nodes are pillars of the decentralized network. They are responsible to provide the following basic services to all users:

1. messaging gateway: through which users and other postal nodes could connect to current postal node;
2. registration service: users may register their DIDs through any postal node. DIDs can communicate with others only after registration via whichever postal nodes they choose;
3. message delivery: postal nodes are in charge of receiving messages from origin DID, and sending messages to target DID;

An incentive mechanism is crucial to ensure that all postal nodes can play by the rules and serve all DIDs equitably. We will extensively discuss the design of this incentive mechanism in Section 3.7.

Postal Node Registration In an open network, any entity can become a postal node, as long as it meets all technical and staking requirements. For a permission-ed network, a candidate node mechanism helps ensure stability by keeping nodes ready to replace the ones that are penalized or removed for malicious behavior.

A governance committee establishes the basic rules for penalizing postal nodes, and provides guidance on how nodes should respond to community needs. The committee also approves the registration of each postal node, for reason of compliance, security, economic sustainability, or operational reliability.

Node Discovery Postal nodes must continuously connect to each other once operational. This requires high availability and discoverability for each postal node. By publishing connection metadata to an underlying blockchain, postal nodes can discover each other. Published metadata may include the transmission protocol, physical network addresses, SLA commitment, and other parameters. For censorship resistance, this metadata could be asymmetrically encrypted using other nodes' public keys.

- publishConnectionMeta: (pnDID, connectionMeta);

As an alternate discovery mechanism, a new node can broadcast signed connection metadata directly to the postal node network.

Replacement Replacement of a postal node happens on two occasions:

1. when a node owner transfers or sells node ownership to others;
2. when a node violates the rules and triggers a forceful removal.

Replacement will be initiated by a node owner, approved by a governance committee, and executed on the blockchain. Upon replacement, ownership of the bound domain name will be transferred to the candidate.

- reassignPN: (pnDID, candidatePubKey);

Postal Node Services Postal nodes can provide a variety of services to enhance the user experience and address issues interoperability and usability.

As postal nodes are identified solely by their signatures in the network, they can provide gateways in many forms to facilitate wide adoption. This allows for diverse devices and applications to connect to the network, regardless of their operating system or platform. This also forms an ideal solution for a range of use cases, such as enabling secure communication between apps, providing access to a corporate intranet, or connecting smart devices in a home automation system.

Postal nodes should collaborate to provide a moderation service that maintains a healthy and sustainable network. This involves monitoring the network status, computing DID reputation scores based on network activities.

3.3 Message Structure

Inspired by the chained blocks of transactions, message structure in SigWorld is defined in a similar manner. A message between two DIDs is defined as follows:

- from: DID of message sender;
- to: DID of message receiver;
- content: content of the message;
- timestamp: the time when this message is sent;
- sequence: sequence number of current message in the whole message chain;
- prevhash: hash digest of previous message, not needed for the very first message.

Alongside the message, digital signatures are appended to ensure authenticity of each message, which is defined as $signature = sign(privKey, content || from || timestamp || to || sequence)$. All parties involved in message sending and delivery append their signatures, to form a transparent and verifiable delivery. Signatures appended to each message are:

- signature0: signature of message sender;
- signature1: signature of origin PN;
- signature2: signature of destination PN;
- signature3: signature of message receiver.

signature3 is optional, as a follow-up reply message from receiver will function as a signed confirmation of all previous messages.

By including prevhash in the message structure, messages between any two DIDs form a chain of hashed records, ensuring integrity and authenticity of messages. As depicted in Figure 3, this is the same structure for

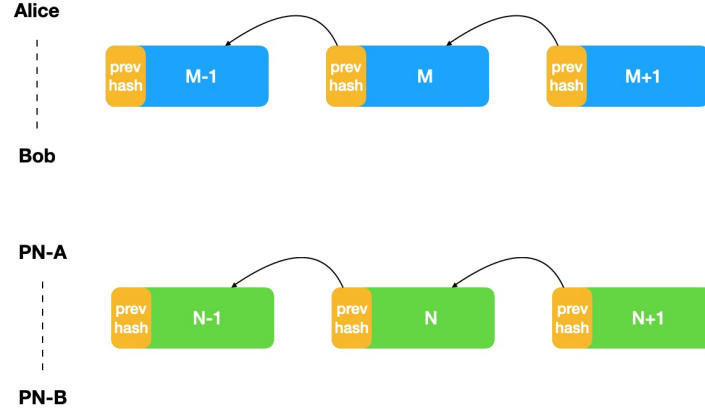


Figure 3: Message Chain between DIDs

messages between PNs. Enforcing this rule forms chronological messaging. This lays a foundation for a full reachable network. A detailed discussion on reachability will be continued in 3.9.

Chronological messaging is indexed by wrapping a *sequence* number in the message structure. As shown in Figure 3, $M-1$, M , $M+1$ are sequence numbers in 3 consecutive messages. The uniqueness and continuance of *sequence* number and the message hash, helps avoid possible disorders and collisions in the asynchronous messaging context. Collisions may still exist, but this can be resolved at client side of message senders. Although linearizability may incur limitation and increased complexity in terms of multi-session support for DIDs, this feature could unlock surprising potential in order-sensitive scenarios, such as financial transactions and online gaming. We will demonstrate this in Section 5.

3.4 Message Flow

Given user Alice("sw.x.a") registered on PN X("sw.x"), and user Bob("sw.y.b") registered on PN Y("sw.y"), when user Alice sends k^{th} message to Bob at time t :

1. define a standard message structure as $M = \{from, to, sequence, content, timestamp, prevhash, signatures[]\}$;
2. Alice signs message $m_A = \{"sw.x.a", "sw.y.b", k, m, t, H_{k-1}\}$, $sig0 = sign(m_A, sk_A)$, where H_{k-1} stands for hash of $(k-1)^{th}$ message;
3. Alice sends $M_A = \{m_A, [sig0]\}$ to PN X;
4. PN X signs on M_A : $sig1 = sign(m_A, sk_X)$, $M_{AX} = \{m_A, [sig0, sig1]\}$;
5. PN X sends M_{AX} to PN Y, and Alice gets $sig1$ from X;
6. PN Y signs on M_{AX} : $sig2 = sign(m_A, sk_Y)$, $M_{AXY} = \{m_A, [sig0, sig1, sig2]\}$;
7. PN Y sends M_{AXY} to PN X;
8. Bob fetches M_{AXY} from PN Y, Alice gets $sig2$ from X.

In this process, *signatures* serve as a proof for message delivery. In an open and incentivized network, *signatures* are also receipts for postage fees, which will be explained in 3.7.

Message Routing When Alice cannot reach PN X directly due to network issues, the routing mechanism would help Alice send her message via some other PN Q("sw.q") that Alice could connect to.

1. define a routing message structure as $RM = \{from, to, sequence, content, timestamp, prevhash, signatures[], routingScheme, routingSignatures[]\}$;
2. Alice signs message $m_A = \{"sw.x.a", "sw.y.b", k, m, t, H_{k-1}\}$;
3. Alice sends $RM_A = \{m_A, [sig0], "sw.q", [rsig0]\}$ to PN Q, where $rsig0 = sign(m_A, sig0, "sw.q")$;
4. PN Q signs on RM_A : $rsig = sign(m_A, sk_Q)$, $M_{QA} = \{m_A, [sig0], "sw.q", [rsig0, rsig]\}$;
5. PN Q sends RM_{QA} to PN X;
6. PN X verifies RM_{QA} , signs on m_A : $M_{AX} = \{m_A, [sig0, sig1]\}$ and marks routing scheme for M_{AX} such that X can route follow-up receipts back to Alice;
7. PN X sends M_{AX} to PN Y and PN Q, and Alice gets $sig1$ from Q;

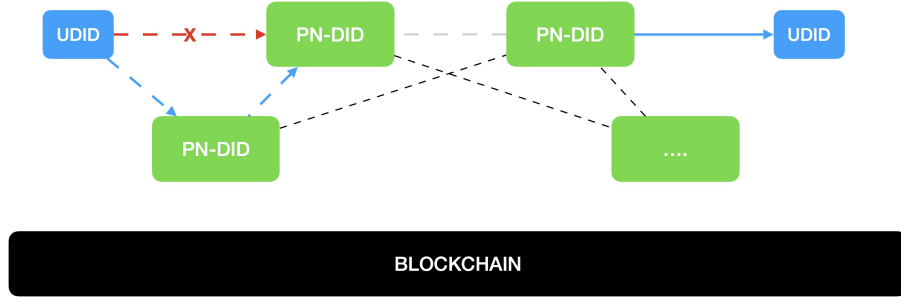


Figure 4: Message Routing

8. PN Y signs on M_{AX} , and sends M_{AXY} to PN X;
9. Bob fetches M_{AXY} from PN Y, Alice gets $sig2$ from Q.

On-chain Messaging When Alice needs a strictly guaranteed message delivery to Bob, or Alice cannot directly send message or send through routing, she can utilize a underlying blockchain as a fallback communication channel, to communicate with others.

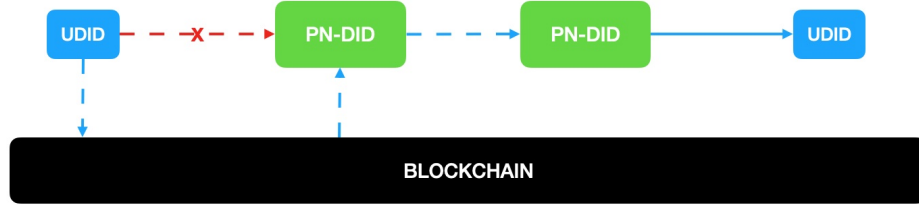


Figure 5: Fallback Mechanism: On-chain Messaging

1. Alice signs message $m_A = \{ "sw.x.a", "sw.y.b", k, m, t, H_{k-1} \}$ and puts $M_A = \{ m_A, [sig0] \}$ onto blockchain **B**;
2. PN X listens for message M_A , signs on M_A , and send M_{AX} to PN Y;
3. PN X sends M_{AX} to PN Y, and puts M_{AX} onto **B** such that Alice can get $sig1$ from **B**;
4. PN Y signs on M_{AX} , and sends M_{AXY} to PN X;
5. Bob fetches M_{AXY} from PN Y, PN X puts M_{AXY} to **B** such that Alice can get $sig2$.

Note that this fallback mechanism also applies to communication between postal nodes.

3.5 On-chain Commitment

Each postal node periodically commits a message delivery proof onto blockchain **B**. DIDs can get their chain of proofs related to their latest message tree from registration PN. Such commitments serve as checkpoints for service rule validation.

User DIDs can request a forceful a on-chain commitment when sending messages. The commitment is a public notarization.

3.6 End-to-end Encryption

End-to-end security for all DIDs' communications are ensured by implementing end to end encryption protocols between DID clients and between postal nodes.

Adapted X3DH In an asynchronous chronological messaging context, an adapted version of the X3DH protocol provides enhanced security and privacy protection for users. By leveraging the chronological message sequence, the modified X3DH protocol in SigWorld simplifies key ratcheting for both sending and receiving sessions. This makes key storage and cryptographic operations less complex and error-prone.

3.7 Token Design

SigWorld achieves security, high availability and reachability through token staking, incentive distribution and penalties. Postal nodes are responsible for message verification and delivery. They deposit crypto tokens onto blockchain **B** to be eligible for running postal node softwares. In return, they are paid postage gas for successful message delivery and routing. In practice, tokens deposited could be any crypto mined through PoS consensus, such as ETH.

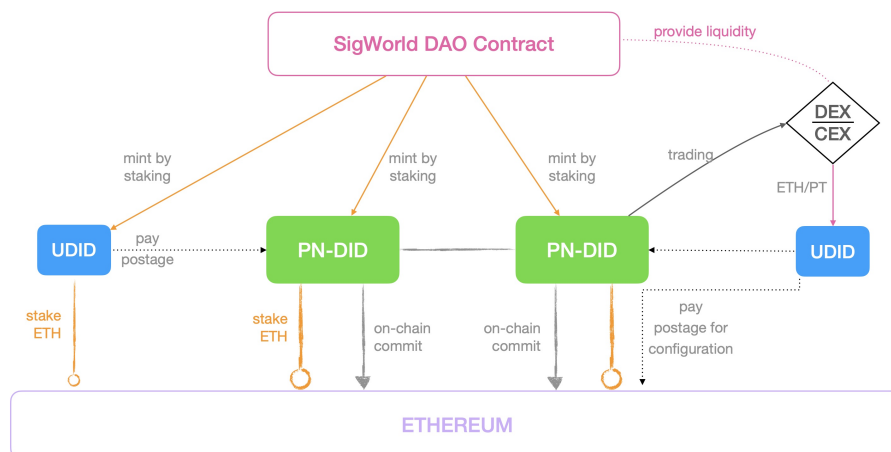


Figure 6: Token Design of SigWorld based upon Ethereum

Postage In SigWorld, postage token keeps the network secure and available. It exists to prevent spamming and malicious bandwidth occupation. Postage token also acts as a measurement for postal nodes contribution in the network. Postage token are acquired by staking cryptos and purchasing from marketplace. If implemented on Ethereum, it could be a ERC-20 token.

Postage is charged when any DID need to send messages to other DIDs. The basic amount of fees charged is determined by the message size. In addition, user DIDs can charge extra postage token or set token gate on message receiving. This configuration allows users to filter out bothering messages from strangers, or as a claim about necessary fees to pay for advertisers. Such configurations are maintained on the underlying blockchain and executed by postal nodes on message delivery.

Staking & Rewards There is a minimum staking requirement for any postal node to join the network. It is meant to help ensure the security and stability of the network.

Staked cryptos can be re-staked to a PoS ecosystem **B**, and a large portion of the staking earnings will be shared to postal nodes. Another part of rewards for postal nodes, comes from shares of governance DAO's income. These incentives for postal nodes will be distributed both in forms of postage tokens and cryptos. Reward distribution is executed periodically, in accordance to postal nodes' on-chain commitments, SLA metrics and sender retention.

Note that normal user DIDs can also stake cryptos to mint postage tokens, enjoying the same minting rate for postal nodes.

Penalties Postal nodes will be slashed for the following misbehavior:

- missing on-chain commitment: PN failed to make on-chain commitments;
- unavailability: PN service unavailable;
- message sequence conflict: multiple conflicting signatures of same message sequence from PN.

There will be a forceful removal of malicious PN from the network, when its staking is penalized to a certain portion of shares.

Burning On redeeming rewards, postal nodes submit a fair amount of postage tokens in proportion to its rewards, many of which will be burnt. Also, for each message delivery, a portion of the charged postage tokens are burnt as network fees.

3.8 Governance

Targeting SigWorld as a decentralized communication infrastructure and as public goods, we envision the SigWorld network to be a community-managed ecosystem, aligning core values of SigWorld to the ideals of Ethereum, Uniswap and more.

In this perspective, SigWorld will grow to be a community-led and community-managed protocol. A governance token SWGT will be issued to help community members on protocol governance, and help foster a healthy ecosystem growth.

SWGT holders have ownership of:

- SigWorld governance;
- community treasury;
- the SigWorld.eth ens domain;

SWGT holders are responsible for:

- making governance rules for postal node;
- fee charging rules;
- minting rate and liquidity of postage token;
- find optimal balance between ETH and postage token rewards needed to sustain a healthy network;
- income distribution and expenditure management;
- voting on improvement proposals.

SWGT holders should ensure that governance decisions are made in compliance with applicable laws and regulations. The community is encouraged to consult knowledgeable legal and regulatory professionals before implementing any specific proposal.

Initial governance committee will be the founder team of SigWorld. As more postal node and user participant in the SigWorld network, SWGT shall be distributed accordingly.

3.9 Reachability

By adapting SigWorld token economy to that of staking pools, we have:

1. for any message sent from u_i to p_i , p_i has an acceptance probability of a and rejection/loss probability of $1 - a$;
2. for any node p_i in the network, it has staked P_i tokens;
3. each user u_i stakes U_i tokens;
4. all staked tokens will be offered rewards of e_0 APR from staking pools, where e_0 is a variable rate for different chains, but will fluctuate around a fixed number; for any node p_i , the APR will be e_1 , where $e_1 > e_0$ and for APR e_2 of any user u_i , $e_2 < e_0$;
5. with each message accepted and delivered by p_i , it will be rewarded with g , where g represents the minimal share of the network's income I ;
6. for each message rejected, or accepted but not delivered by p_i , it will be slashed by sP_i . For any node p_i , when a rejection of a message happens, the message sender u_j or the source node p_j will put the message on a blockchain **B** as a proof (as shown in Figure 5) that a certain message is being sent to p_i and waiting for p_i 's ack signature. In this scenario, p_i must send ack to blockchain **B**, and if p_i failed to submit the ack before deadline, u_j or p_j can submit a slashing request for p_i , and a large portion of the fine shall go to u_j or p_j . In this case, p_i will have to refill its stake to the number of T_i to continue its delivery service;

So, given N_i incoming messages for node p_i , with C_{p_i} as operational cost, the expected profit of p_i would be

$$E_{p_i} = P_i \cdot e_1 + \frac{aN_i}{\sum g} \cdot I - (1 - a)N_i \cdot sP_i - C_{p_i}$$

where $I > 0$ and (suppose that staking pools are from Ethereum):

$$I = \lfloor \frac{\sum U_i + \sum P_i}{32} \rfloor \cdot 32 \cdot (1 + e_0) - (1 + e_2) \cdot \sum U_i - (1 + e_1) \cdot \sum P_i$$

Obviously, to maximize E_{p_i} , node p_i should ensure that $a = 100\%$. However, there still might be a good chance that a node could provide poor delivery service and walk away with considerable profits. Let's take Rocket Pool's reward model (see Appendix ?? for more details) as an example.

Suppose there are 200 postal nodes and 1,000,000 users; each node stakes 16 ETH; each user stakes 0.01 ETH on average, and each of them could send 500 messages a day; if all users are equally distributed among all nodes, that is, $N_i = 91,250,000$. Set $e_0 = 5.29\%$, $e_1 = 6\%$, $e_2 = 4.45\%$, with a minimal slash of $s = 1\%$, then:

$$\begin{cases} I = 1.0529 \times 412 \times 32 - 1.0445 \times 10000 - 1.06 \times 3200 = 44.4336 \\ E_{p_i} = 0.96 + a \cdot 14600000.222168 - 14600000 \end{cases} \quad (1)$$

In this case, for any p_i to earn profit, it shall process messages with an acceptance rate $a \geq 99.999992\%$ to ensure a positive E_{p_i} , meaning that out of the 91,250,000 messages, p_i could not afford rejection of more than 7 messages to avoid any asset loss. Comparing to $a = 100\%$ for perfect all time full reachability, this is a very promising result. When raising the slash rate s a little higher, to 5%, a reasonable acceptance rate for p_i would be $a \geq 99.9999984\%$, as a result, at most 1 message can be rejected for p_i to earn any profit.

Serious caution should be taken to avoid possible damages an evil node p_e could do to the network. If we use Ethereum as blockchain **B**, with slash rate $s = 5\%$, the max number of messages p_e could reject would be 14, resulted from $\{16(1 - 0.05)^n < 8 | n \in \mathbf{Z}^+\} \Rightarrow n > 13$ (remaining stake less than half of initial staking will face a forceful removal). Each rejected message would cost p_e about 0.5714 ETH $\approx$ \$1088 (ETH priced at 1904.4 USD/ETH on Apr. 21th, 2023).

Note that our protocol does not allow p_e to selectively reject messages from/to targeted users (see Figure 4 and 5). For example, when p_e rejects a certain message m_r from a user u_i , u_i could send m_r via other nodes p_o , and p_e will have to accept m_r routed via p_o . If p_e rejects this message m_r from p_o , p_e will have to reject all follow-up messages from p_o ; this applies to all other nodes as well. So an end-game for p_e in this case, is that p_e will be an isolated node and will soon get kicked out of the network, losing a fair amount of its staked tokens. Besides, p_o and u_i could fall back to the on-chain messaging solution, which will cost p_e much more to deliver m_r , or p_e will get slashed. Overall, it would be reasonable for any node p_i to go with an acceptance rate of 100%, making every DID in the network fully reachable.

4 Security and Usability Considerations

SigWorld aims to provide privacy-protected, secure and reliable decentralized communication services to all DIDs. SigWorld is a free, open, secure, trust-less communication solution, achieving an experience comparable to centralized competitors. In this section, we analyze potential attack vectors and counter-measures. We also discuss how governance and interoperability impact security and the user experience in SigWorld.

4.1 Security

Communication within an anonymous and decentralized network poses great risks and challenges. In addition to the following analysis, independent third-party security audits on the protocol design and software implementation are imperative.

Sybil Attack In a decentralized network based on SigWorld protocol, anonymous parties may assume fake their identities as postal nodes or regular users, to manipulate the network.

Through measures such as sufficiently high staking requirements, severe slashing penalties, sufficient incentives, signature validation and hash-chained messages, SigWorld is able to mitigate most sybil attack vectors, such as faking identities of postal nodes or regular users, tampering with messages, colluding attacks. However, stringent precautions should be taken in software and hardware implementations to avoid vulnerabilities. Postal nodes are also responsible for protecting their keys from theft. Constant vigilance is still needed to identify and address evolving threats.

Eclipse Attack Since postal nodes are identified by their public keys rather than IP address or physical locations, a node facing an eclipse attack can easily transfer its service to another network region. Postal nodes can take several measures to mitigate such attacks, including maintaining backup connectivity and attempting to communicate through gossip protocols.

Replay Attack Messages in the network are susceptible to replay attacks, where attackers intercept messages and resend them in an attempt to spam the network.

To mitigate replay attacks, all messages in SigWorld are cryptographically signed and linked together using hash chaining with sequence numbers and timestamps. These features enable any SigWorld client to easily detect forged or relayed messages. As messages in the network can be intercepted, attackers may resend them and try to spam the network.

MitM Attack An attacker could hijack the routing of packets between two nodes, inspecting and modifying messages in transit.

However, since all messages in SigWorld are encrypted end-to-end between the sender and recipient, an MitM attacker would be unable to inspect or modify message contents, even if they could intercept transmitted messages. A successful MitM attack would have to trick a DID client into using an impersonated public key belonging to the intended recipient. Such impersonation attacks could be mitigated by requiring clients to verify keys through independent channels, rather than just locally.

Routing Attack A malicious postal node could attempt to launch a routing attack by selectively dropping, modifying, or delaying messages that are routed through it.

While technically feasible, there are economic disincentives that discourage such attacks. First, participants have multiple routing paths for messages, so avoiding the malicious node is possible. Second, the malicious node would have to spend more resources to launch attacks without being detected, offsetting any gains. Third, postal nodes earn income by providing message delivery services, so disrupting that service undermines their own revenue model. Finally, end-to-end encryption prevents postal nodes from viewing message contents, limiting the effectiveness of possible attacks.

DoS Attack Postal nodes are operated by entities that have staked a large amount of funds and have an economic incentive to run their nodes reliably. They thus likely have sufficient resources to mitigate basic DoS attacks. With rate limiting, gateway redundancy, and DDoS protection, postal nodes are able to detect, mitigate and isolate attacks as much as possible.

When a user DID serves digital content or services, it may be susceptible to DoS attacks targeting those services. Unlike traditional networks where DoS attacks can originate from the application layer or network layer, user DIDs in SigWorld are relatively immune due to several factors. First, most services provided by DIDs are session-less and asynchronous. Second, all messages to a user DID flow through the postal node it is registered on. These features make user DIDs less vulnerable to coordinated volumetric attacks from multiple DIDs. In the event of a large volume of messages due to an attack, service providers have the freedom to customize their message-receiving rules. In such cases, postal node services can help mitigate the attack.

Stake Reuse and Cartel Formation There is a risk that postal nodes may attempt to reuse a single stake to operate multiple nodes, or that cartel-like groups may control a large portion of nodes in SigWorld.

Though SigWorld forbids stake reuse and sharing of identities across postal nodes, some nodes could potentially be operated by cartels. Postal nodes held by cartels pose a threat to users. To mitigate this risk, the initial set of postal nodes could be distributed among a diverse set of reputable and independent entities such as universities, organizations and institutions. A portion of nodes could undergo periodic turnover through a fair voting process by user DIDs. This would help limit concentration of control among cartels.

Spamming The measures taken to mitigate DoS attacks in SigWorld remain relevant for countering spamming attacks. However, users may still be exposed to risks like phishing, disinformation, and malware distribution. To further address spamming issues, postal nodes could provide reputation services to collect user reports. With the widespread adoption of AI technologies, client-side AI models could also be integrated into local clients to help mitigate spam messages.

Privacy Attack Although SigWorld employs strong end-to-end encryption, attackers can still monitor messaging traffic and analyze metadata to reveal user relationships and activities.

SigWorld is designed to discourage unnecessary message routing, avoiding the dispersal of metadata across the network. Aside from their public domain names and public key addresses, no more additional metadata is required when sending messages. Though privacy attacks cannot be completely prevented, SigWorld offers greater privacy guarantees than general blockchain ecosystems by limiting metadata exposure through minimal routing and reduced information requirements for messages.

4.2 Usability

Privacy and Censorship Resistance In most networks, messages are susceptible to surveillance, censorship, and analysis. However, through the use of the adapted X3DH protocol, messages in SigWorld are strongly resistant to censorship. The responsibility for ensuring key security ultimately lies with users and postal nodes themselves, representing the weakest point in the system.

Message Cost Users expect messaging services to be cheap and reliable, and often expect social messaging to be free. However, in SigWorld, user DIDs must pay postage tokens to send messages, which conflicts with expectations of free messaging services.

Several operational measures could mitigate this barrier. First, postal nodes could provide users with free initial token allowances to reduce friction for new users. Instead of paying per message, postal nodes could offer subscription plans where tokens are paid periodically. When implemented on Proof of Stake consensus blockchains, users could stake native tokens to mint postage tokens while also earning stake rewards. Postal nodes could allow staked users to vouch for others and share their subscription balances, encouraging cooperation between users and fostering network effects that benefit both users and postal nodes.

Compliance While anonymity can enable privacy, it should not enable criminal abuse. However, tackling criminal abuse at the protocol level is challenging. SigWorld provides cryptographically verifiable electronic records that can serve as evidence when reporting criminal activities, enabling law enforcement investigations while preserving users' privacy and anonymity.

However, public postal nodes and users may still face challenges complying with lawful government requests for information. In practice, postal nodes may need to find a balanced approach, to embrace core privacy and anonymity principles while implementing responsible policies.

5 Case Study

5.1 Decentralized Super App

Centralized platforms capture most of network values created by users, as shown in Figure 7. SigWorld revolutionizes the communication paradigm of traditional web applications and digital services.

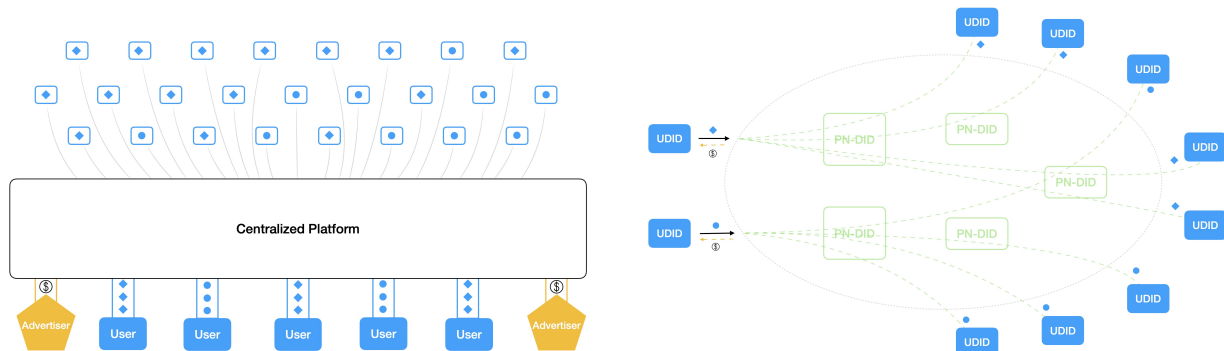


Figure 7: Centralized Platform compared to Decentralized Communication Paradigm in SigWorld

By allowing user DIDs to connect safely and reliably to each other, regardless of geolocations, IP addresses, any DID can serve content from anywhere and from any device, without relying on any third party intermediaries or monolithic platforms to publish and serve the content. This fosters greater innovation as any developer can build new apps and serve its digital content and services right from developer's laptop.

Users will have full right and freedom to customize their experience by combining and integrating different services provided by DIDs however they want. Users have full control to their data and content, and full control over how they want to interact with different features. With the advent of AI, AI can monitor users' behaviors locally, and optimize for each user for a personalized, context-aware experience.

5.2 Decentralized Network Gaming

Network games nowadays are based on a centralized network. Centralized, closed network gaming systems are known for manipulation risks, hacking and cheating problems. Game servers are basically blackboxes

to players. The game can be easily manipulated by developers, admins, or other parties. Once hacked or breached, centralized gaming systems players' data would be at risk, and the fairness of the game cannot be guaranteed. Also, due to lack of transparency, players have to face the risk of account being locked out of the system, game assets lost, etc.

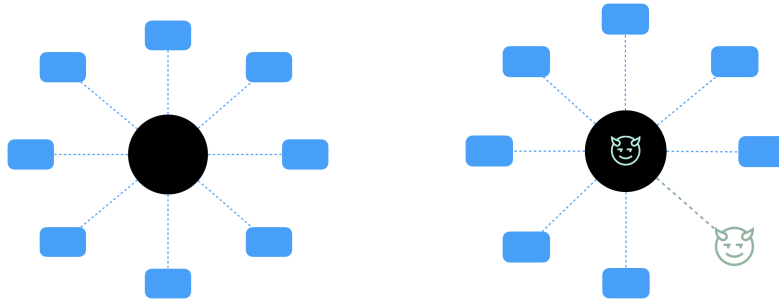


Figure 8: Security and Fairness of Centralized Network Games are not Guaranteed

A decentralized game protocol would largely mitigate these issues. Thanks to message security, reachability and chronology, SigWorld makes decentralized network gaming possible. All game data will be owned by players, gaming procedure and player interactions flow through SigWorld network, making every piece of game data secure and verifiable. With transparent gaming mechanism, decentralized gaming protocol brings verifiable fairness to all players. In turn-based decentralized games, SigWorld helps guarantee fairness and security in most extreme adversarial environment, as shown in Figure 9.

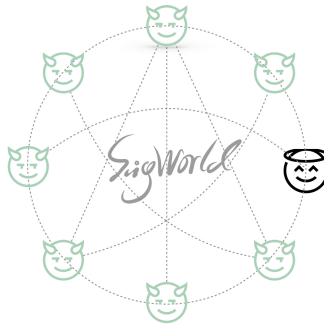


Figure 9: SigWorld-based Decentralized Network Board Gaming Guarantees Fairness in Extreme Adversarial Environment

5.3 Reliable Messaging Relay in Cross-chain Communication

LayerZero Protocol [6] is an omni-chain interoperability protocol designed for lightweight message passing across chains. LayerZero relies on two parties to transfer messages between on-chain endpoints: the oracle and the relayer, as depicted in Figure 10. When a user application sends a message from chain A to chain B, the message is routed through the endpoint on chain A. The endpoint then notifies the UA specified Oracle and Relayer of the message and its destination chain. The Oracle forwards the block header to the endpoint on chain B and the Relayer then submits the transaction proof. The proof is validated on the destination chain and the message is forwarded to the destination address.

The trustlessness of LayerZero lies in the independence of the Oracle and the Relayer. In practice, LayerZero leverages *Chainlink DON* [7] framework as the Oracle, and the team recommends applications to use the LayerZero Relayer or build their own Relayer service. However, allowing user applications to set their own Oracle and Relayer is a security risk, as proven by Krzysztof Urbański from L2BEAT[8]. If the Relayer goes offline or submit invalid transaction proofs as shown in [8], it could prevent valid cross-chain transactions from being confirmed. The Relayer exposes a single point of failure.

Postal nodes in SigWorld network are perfect relayers to help solve the SPoF problem of LayerZero. LayerZero interacts with smart contracts living on multiple chains. In SigWorld, a smart contract can be seen as a special "DID" that could receive messages. Though smart contracts cannot be registered on any PN, these contracts can be set as receivers, as transactional messages posted to contracts are publicly verifiable through state change and events. By setting a message subscription configuration in SigWorld, Alice, also a LayerZero

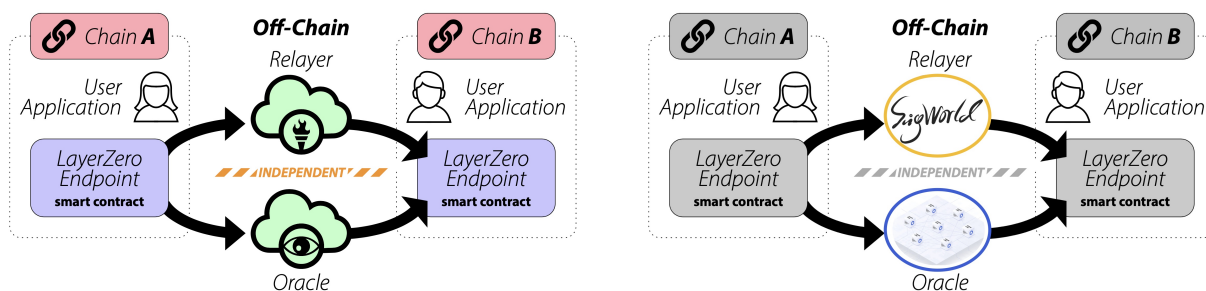


Figure 10: SigWorld as a Decentralized Relay Network in LayerZero

user, makes a postal node listen to specific **Packet** event emitted by *UltraLightNodeV2* on source chain. The PN will see this event message as a message from Alice and deliver the transaction proof to target chain's contract.

6 Future Work

Future work for technical issues includes:

- optimizing the parameters of the token model, such as postage minting rate, reward distribution;
- enabling DID communication between smart contract wallets;
- developing tools, libraries, and integrations to simplify development for ecosystem participants.

7 Conclusion

This article outlined the network design, messaging protocol, and security considerations of SigWorld, a decentralized inter-DID communication protocol. We discussed how SigWorld addresses threats like spam, DoS attacks, and privacy risks through its cryptographic design and economic incentives. Several case studies demonstrated innovations that SigWorld enables, including decentralized super apps, marketplaces, and gaming experiences.

SigWorld's key strengths lie in its end-to-end security, high reachability across postal nodes, and verifiable message chronology. These features overcome traditional challenges of scale, privacy, and trust in decentralized networks. Taken together, they have the potential to radically transform how individuals, businesses, devices, and applications exchange value and communicate at a global scale.

Further research is needed to evaluate SigWorld's real-world performance, economic viability at scale, and ability to navigate complex policy issues like user compliance. Nevertheless, SigWorld represents a promising step toward a decentralized communication infrastructure that preserves user privacy while fostering innovation through open and interoperable standards. With continued development and ecosystem growth, it has the potential to reshape how we perceive digital identity, data ownership, and free communication on the Internet.

References

- [1] *Matrix*. URL: <https://matrix.org>.
- [2] *Marlin. Design and analysis of a decentralized relay network*. URL: <https://www.marlin.org/whitepaper>.
- [3] *Status*. URL: <https://status.im>.
- [4] *XMTP*. URL: <https://xmtp.org>.
- [5] *Nostr*. URL: <http://nostr.com>.
- [6] *LayerZero Protocol*. URL: <https://layerzero.network>.
- [7] *Chainlink DON*. URL: <https://chain.link/whitepaper>.
- [8] Krzysztof Urbański. *Circumventing Layer Zero. Why Isolated Security is No Security* | L2BEAT. URL: <https://medium.com/l2beat/circumventing-layer-zero-5e9f652a5d3e>.