Analog: The Decentralized Timegraph

Timepaper (Version 3.0)

January 2024

Executive Summary

Many blockchains have emerged, each providing unique and distinct features that users and decentralized application (DApp) developers find attractive. However, seamless communication across these ecosystems is fragmented and siloed, isolating DApps from the majority of liquidity in the Web3 space that other blockchains provide. To enable DApps to communicate frictionlessly across heterogeneous blockchains, we propose the Analog Network. Analog Network is a suite of software that enables more powerful decentralized applications (DApps) through secure interoperability. The center of Analog is the Timechain, a layer-0 (L0) blockchain network powered by a novel Proof-of-Time (PoT) consensus protocol that facilitates message passing between heterogenous chains. Software is built on top of the Timechain to allow users and DApp developers to extract maximum value from all connected blockchain ecosystems.

CONTENTS

1.0 Introduction

As the use of cryptocurrencies becomes mainstream and mass-market adoption of decentralized finance (DeFi) surge, so does the demand for integrating various decentralized services and products. It is nearly impossible to imagine that a single blockchain would emerge to solve all of society's use cases. A multi-chain future seems unavoidable. However, such a future without interoperability can only be compared to the early Internet days when there was no Transmission Control Protocol/Internet Protocol (TCP/IP).

The current Web3 ecosystem can be paralleled to the early Internet days, with blockchain networks being not interoperable and hindering the mass adoption of decentralized ledger technologies (DLTs). For users to interact with Web3 products across different domains, there is an inherent need to move their crypto assets along with them. Due to a lack of seamless interoperation across different chains, DApp developers are forced to make tough choices on which blockchain platform to build their innovative applications.

Similarly, users have to choose which "*walled garden*" to deploy their assets to maximize utility and capital efficiency. A few proposals and projects have emerged to address this interoperability challenge. However, most of them seem to apply to only specific chains, often standardizing their protocols within their own ecosystems. This interoperation approach requires other blockchains to adopt the standardized framework, often through complicated, restricted, and less secure bridging networks. This timepaper proposes a novel, public permissionless layer-0 (L0) blockchain — also called ***Timechain*** — that actively

and agnostically connects to other blockchain networks to facilitate interoperability.

Moreover, we propose a generic smart contract system that runs on the Timechain — what we are calling ***continuum*** — that users/DApps can use to hold and manipulate assets on different chains directly. The continuum smart contract system is set to open boundless cross-chain applications. Most importantly, we also propose an ***Omnichain SDK*** that incorporates the Timegraph API, ***Analog Watch*** (Query Marketplace), arbitrary smart contract executions, and, later, General Message Passing (GMP) protocol on all the connected chains.

1.1 What Ticks Are and Their Relevance in Cross-chain Communication

At its core, Analog's blockchain — also called the Timechain — is simply a ***state replication machine (SMR)*** that persists validated data (***ticks***) from different connected chains. There is a state which describes the current state of the Timechain and a tick that triggers the state transitions. Given a state ***S*** and a tick ***E***, the Timechain will return a new state ***S'***. Like all blockchains, the tick is bundled in blocks to make the SMR system more efficient.

With remarkable developments in the blockchain space and its mounting prevalence in enhancing efficiency and convenience for users, it is evident that the technology is here to stay. One notable advancement in the sector is the advent of DApps that are censorship resistant. At the core of DApps' operation

is the need for validated data, i.e., ticks, from different sources, which are processed as per users' requests.

Accessing such data in Web3 is essential for an "always-on" world, where DApps are increasingly becoming automated, with users leveraging smart contract capabilities to facilitate conditions for decentralized transactions without intermediaries. Despite data occurring across virtually all these applications, accessing it is prohibitively difficult for DApps, especially when it comes to cross-chain communication.

The inability to share data across different networks has led most Web3 users to associate ***cross-chain communication*** with asset transfers — also called token bridging. However, associating cross-chain communication with token transfers is simply scratching the surface when it comes to the potential that data-driven cross-chain solutions can provide. A tick-driven blockchain ecosystem opens the door for a ***General Message Passing (GMP)*** protocol with boundless use cases, including cross-chain message passing with value/data, smart contract-managed external assets, cross-chain automated market makers(AMMs), multi-chain NFTs, and more.

1.2 Introducing Analog Network

Analog Network is a suite of software tools that enable secure cross-chain communication in Web3. DApp developers can use Analog Network as an application programming interface (API) to pass arbitrary messages between blockchains frictionlessly and securely, enabling them to bring interoperability

functionality to any chain, whether such a blockchain is a layer-1(L1), layer-2(L2), or a rollup.

At a higher level, Analog Network consists of three layers:

- **Decentralized validators**. The Network relies on validators — also called ***time node***s — to deliver cross-chain communication through Nominated Proof-of-Stake (NPoS), which will later be replaced by a novel Proof-of-Time (PoT) protocol. As a Byzantine Fault Tolerant (BFT) protocol, the NPoS algorithm assumes that a supermajority (more than two-thirds) of validators are honest and acting in the best interest of the protocol to extend the state replication machine — also called the Timechain.
- **Decentralized oracle network**. In addition to being a blockchain, Analog Network also observes other connected chains. Therefore, each time node is attached to an off-chain module — also called ***Chronicle Worker*** — and a ***Connector*** that scans other chains (smart contracts) for relevant events (transactions, event logs, or state) at any given time. Whenever a user/application requests blockchain data on a connected chain, any time node on that chain can report the relevant event to the Timechain. This triggers a threshold signature scheme (TSS)-based consensus, where a supermajority (more than two-thirds of active time nodes) needs to sign the transaction for it to be forwarded to the Timechain for further consensus through the NPoS algorithm.
- **Application-level SDKs and API suite**. Sitting on top of the decentralized validators and the decentralized oracle network are software development kits (SDKs) and API — also called the Timegraph API — that DApp builders can use to compose applications across any number of

blockchains. The SDKs and APIs allow DApp builders to go cross-chain without the need for rolling out a complex infrastructure layer or learning new languages for the connected chains.

1.3 Value Proposition

Blockchain innovations are still in their infancy, and breakneck competition is essential for the ecosystem's growth. However, today's blockchain ecosystem is subject to the same pressures that the early Internet had in the 1980s and 1990s, and balkanization risks are imminent.

At Analog, we believe platform builders, DApp developers, and users can only realize the true potential of blockchain if an interoperability framework exists to connect multiple networks. We also believe that data is the lifeblood of many DApps that want to unlock liquidity across multiple chains. This is why we are committed to building an omnichain interoperability infrastructure that:

- Enables seamless flow of liquidity across multiple heterogeneous blockchains.
- Allows DApp developers to leverage and extend the composability of decentralized finance (DeFi) applications across multiple blockchain networks.
- Facilitates liquidity migration and developer efforts towards emerging blockchain ecosystems and solutions.
- Spurs mass adoption of Web3 products and services.

2.0 Analog Network Architecture

Analog Network has five essential components that allow it to deliver seamless interoperability:

2.1 Threshold Signature Schemes (TSS)

The protocol relies on TSS where a group of Chronicle Workers — affiliated routines implemented within time nodes and operating in off-chain mode — needs to reach a consensus on the validity of fetched ticks. The Chronicle Workers undertake this functionality by fetching finalized transactions from Connectors and jointly participating in key generation (KEYGEN) and key signing (KEYSIGN) processes, enabling them to attest to the validity of the fetched transactions. For any fetched ticks to be relayed to the Timechain network, a supermajority of Chronicle Workers (more than two-thirds of Chronicle Workers) must append their partial private signatures to the fetched transaction.

The KEYGEN and KEYSIGN processes are performed via multi-party computation (MPC) processes that reveal no secret of any participating Chronicle Worker. Because the decentralized chronicle worker network can hold a single public key (***group TSS key***) and address, Analog Network is able to support smart contracts and manage vaults or liquidity pools on virtually any connected chain.

This is because the TSS keys are stored on the Timechain and used to verify the signed transactions. To ensure economic safety, each time node running a chronicle worker is required to stake a certain amount of $ANLOG tokens in the

network. The staked $ANLOG tokens are subject to forfeiture upon failure or malfeasance by the Chronicle Worker. In the blockchain context, this mechanism is often known as ***slashing***.

To enhance scalability, the entire Chronicle Worker network is partitioned into ***shards***, with each shard consisting of a number of independent Chronicle Workers that leverage distributed TSS processes to reach a consensus on the validity of fetched transactions.

2.2. Consensus Engine

Analog Network is a NPoS-based, public blockchain that will, in the future, migrate its consensus engine to the PoT protocol. The Network is supported by a decentralized set of time nodes (validators) that also serve as observers on external chains and reach consensus on those chains through TSS.

Once the TSS consensus is reached, the payload is propagated to the time node network where it undergoes validation to be added to the Timechain. Each time node can vote on block proposals with voting power proportional to the amount of staked $ANLOG tokens.

Time nodes need to be online (active) at all times, ready to participate in the consensus process. Time nodes are incentivized for each block they add to the Timechain. The locked tokens are subject to slashing in case the time node fails or deviates from the protocol.

2.3 Continuum Smart Contracts

Gateway smart contracts provide connectivity between the Analog network and external chains. Time nodes monitor the gateway API for incoming cross-chain requests and then come to a multi-party cryptographic consensus on the validity of those requests.

In case a supermajority of time nodes attests to the validity of the cross-chain request, the transaction is passed to Analog Network, where it undergoes further consensus via the NPoS algorithm. Put simply, gateway smart contracts enable the GMP functionality, where the DApp states and logic are spread across all the Analog-connected chains. Here, the Timechain only acts as a verifier and relayer network.

Besides the gateway smart contracts, Analog will also deploy its own native smart contract system (***Continuum***) in the future. Continuum smart contracts are native programs on the Timechain. Unlike the gateway smart contracts, where the logic and the state of the DApps live within the external chains, continuum smart contracts will exclusively reside on the Timechain.

Here, the Timechain serves as a single platform with a unified interface that interacts with external networks. We believe this approach simplifies DApp development (minimal efforts required to accommodate new chains). This approach is also flexible. For example, DApp developers will no longer be constrained by chain peculiarities and message passing.

2.4 Gateway Smart Contracts

The gateway smart contracts connect Analog Network with external chains. Each chain onboarded to Analog Network has an associated smart contract deployed to that network and managed by a set of Chronicle Workers through a single public key or TSS key. The associated private key pair is generated via a multi-party cryptographic scheme that divides the key into multiple shares, assigning the key share to each of the participating Chronicle Workers.

The platform uses the deployed smart contract to pass messages from the connected chain to Analog Network. The gateway smart contract can only execute actions on an external chain if the number of Chronicle Workers connected to that chain attains a set threshold (more than two-thirds need to sign the transaction).

The primary purpose of gateway smart contracts is to enable the ***generic message passing (GMP)*** functionality on the Network. DApp developers can use GMP protocol to:

- Bring cross-chain network effects and composability to their blockchains of choice, whether such chains are L1 or L2. For example, they can call a smart contract on chain Z from chain Y.
- Augment fungible tokens and non-fungible tokens (NFTs) with message-passing functionality. For example, they can call a smart contract on chain Z from chain Y and attach some tokens.
- Tap into users from other connected blockchain ecosystems.

2.5 Timegraph SDK and Developer Tools

Accessing data from blockchain ecosystems at scale can be tedious and challenging. Common approaches that DApp developers often use, like public blockchain nodes, are limiting, expensive, and frustrating. Analog Network enables users/DApp developers to access data from any connected blockchain via a simple and intuitive interface — also called Timegraph API — that is delivered through ***Timegraph SDK***. Besides the Timegraph SDK, Analog also provides ***Timechain SDK*** and ***Connector SDK*** to enable time node operators to maintain the network.

3.0 Analog Interoperability Stack

At an abstract level, the interoperability stack is a four-layered solution as shown below:

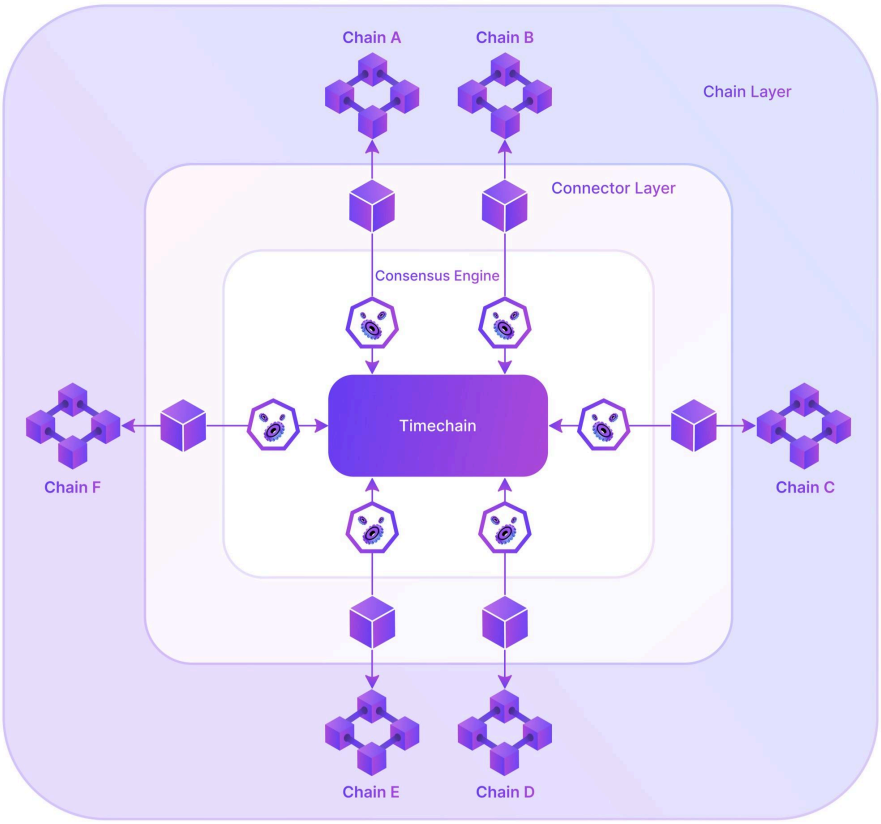

Figure 1: Interoperability stack

Analog Network has four primary layers:

3.1 Chain layer

This layer consists of multiple sovereign blockchains that the Analog onboards on its network. As a chain agnostic platform, the Analog network will onboard EVM-based, Cosmos-based, and Polkadot chains via a simple SDK that any validator (i.e., time node) can download and install.

3.2 Connector layer

This is the core layer within the network that facilitates monitoring and message passing in cross-chain transactions. The connectors that are hosted by time node operators are the primary actors within this layer that ensure trustless interoperability between multiple chains.

Applications running on the Analog Network can leverage Chronicle Workers, which interacts with connectors, to write custom logic on any connected chain. The Chronicle Workers on the external chain can trigger events in response to the specified state changes on the source chain.

3.3 Consensus Engine

Analog Network is essentially a public blockchain that will initially run a NPoS consensus engine and later the PoT consensus protocol. As such, its consensus engine consists of distributed time nodes that are incentivized to act responsibly while producing blocks and maintaining the state replication machine (SRM).

The following processes explain how the NPoS consensus engine works in Analog Network at a higher level:

1. A time node called the ***time elector/proposer*** is selected to submit a new block to the Timechain.
2. Other time nodes — also selected within the given active set — vote on whether to accept or reject the proposed block. If the block is rejected, a new time elector is chosen, and the process starts again.

3. If the block is accepted, the block gets signed and is appended to the Timechain.
4. The network distributes the staking rewards and a portion of gas fees to time electors and nominators. Time electors get rewarded extra for their participation in the consensus process.

3.4 Timechain

This is the heart of the Analog Network that stores the metadata associated with fetched and confirmed ticks emanating from multiple chains. During block generation, time nodes pre-process the fetched and confirmed ticks into a prescribed format based on the transaction type. Pre-processing involves separating the key fields from the fetched ticks that would form the metadata while storing other fields in off-chain storage.

The Timechain has the following primary responsibilities:

- It serves as a ***public and auditable ledger*** for users to interact with the time node network.
- It creates a robust infrastructure for building other DApps, including the ***Analog Watch,*** that would otherwise be impossible to build in a chainless environment.
- It serves as a ***redundant security mechanism*** (defense-in-depth security strategy). For example, an attacker that successfully compromises the TSS-based consensus needs to compromise more than two-thirds of time nodes on the Timechain network to successfully attack a GMP-based transaction on the destination chain.

- It provides ***governance infrastructure***, where users propose and vote on crucial decisions, such as staking and slashing on the network.
- It provides an infrastructure for running ***Continuum smart contracts***.

3.0 Generic Message Passing (GMP) Protocol

The core of the platform's interoperability feature is a generic message passing (GMP) protocol. The GMP protocol sits atop the permissionless network of time nodes which provides routing and validation services. The GMP functionality fundamentally changes how DApps in a multi-chain ecosystem are built and used.

DApp developers can leverage the GMP protocol to build cross-chain applications with coherent application logic, shared states, and efficient liquidity utilization. With GMP protocol, Analog Network users will enjoy the benefits of diverse blockchain ecosystems with the simplicity of a single-transaction user experience (UX) without the need for complex manual interactions inherent in the current Web3 space.

3.1 GMP Architecture

The architecture for the GMP protocol has several noteworthy components, as illustrated below:

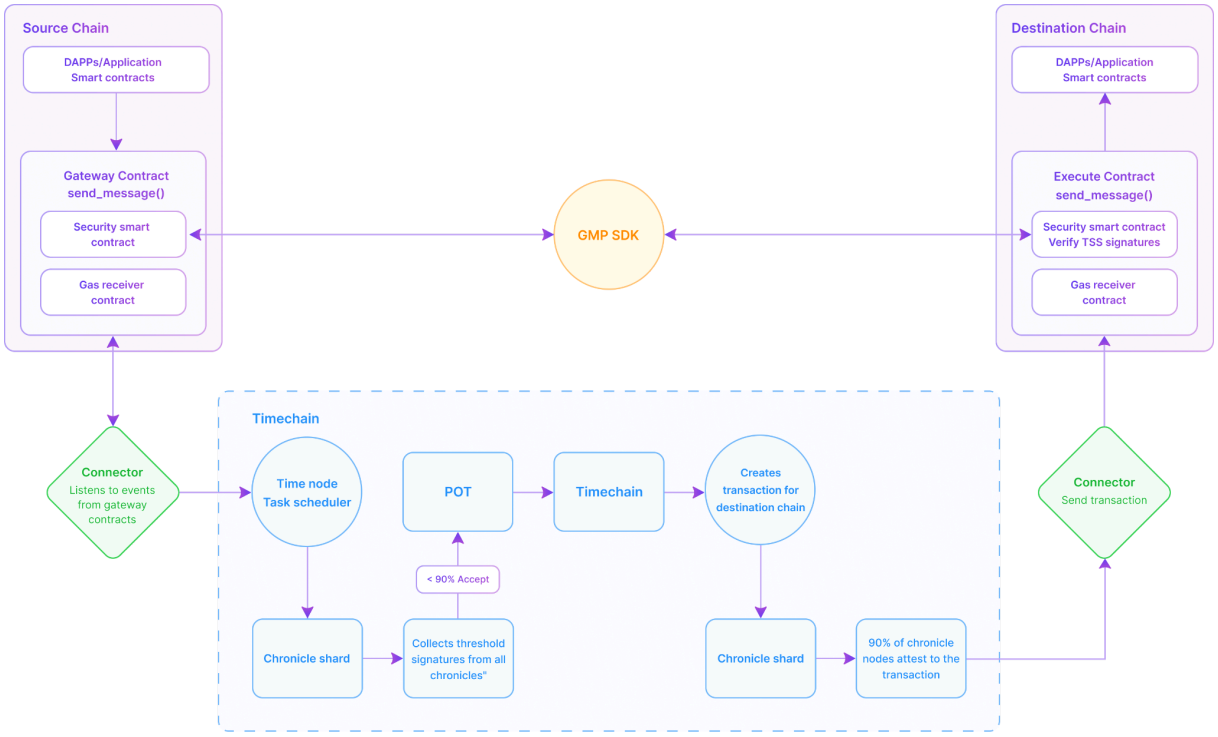

Figure 2: High-level illustration of GMP protocol

At a high level, the GMP protocol consists of the following components, deployed in both on-chain and off-chain environments:

3.1.1 On-Chain Components

There are four primary on-chain components:

- **Gateway smart contracts**. These are the contracts the time nodes (through Chronicle Workers) observe and which fundamentally facilitate cross-chain communication. The network deploys one (1) smart contract for each connected chain to be managed by time nodes.
- **Gas Receiver Services (GRS) smart contract [TBD]**. This is a contract deployed on the source chain to manage the payment of transaction

fees. It enables the user/application to pay transactions as a single bundle in the source chain's denominated native token.

- **Security smart contract**. These are smart contracts that verify that cross-chain messages being delivered to the destination chain were actually signed by the requisite number of chronicle workers (as specified by the shard) on the source chain. DApp developers can optionally override the source chain's default shard (i.e., a shard consisting of 3 chronicle workers) by specifying the shard according to the needs of their applications.
- **Time nodes (validators)**. These are on-chain nodes that vote on proposed blocks to the Timechain, with voting power being proportional to the staked $ANLOG tokens. A time node operator can also delegate its voting power to another time node. They also relay these messages across the network. In this regard, these nodes serve as byzantine fault tolerant (BFT) notaries that attest to the validity of blocks added to the Timechain.

3.1.2 Off-Chain Components

There are two primary off-chain components:

- **Time nodes (blockchain oracles)**. Each time node has two routines implemented in off-chain mode:
    - ***Chronicle Worker***. This is a module that participates in TSS-based consensus (i.e., KEYGEN and KEYSIGN processes) to reach consensus on the state changes occurring in external chains. Chronicle workers can use this module to update the various states

on the connected chains (i.e., execute cross-chain transactions on the target/destination chains).

    - ***Connectors***. To connect to these chains, each time node runs its own ***connector(s)*** — also implemented as an off-chain routine. The chronicle workers fetch the validated transactions/data from the connectors and generate a signed payload that gets transmitted to the time node for verification.

- **Sentinels**. These are special nodes on Analog Network that are not part of the time node network. Any Sentinel on the network can observe the gateway smart contract and connector signatures to detect fraudulent activities. If any fraudulent activity is detected on the gateway, any Sentinel can submit evidence (i.e., fraud proofs) to the Timechain for verification. If a supermajority of nodes attests to the fraud, the time node whose connector submitted the transaction is punished through slashing mechanisms. The presence of one or more sentinels on the network serves to deter malfeasance behavior.

3.2 How It Works

To operationalize this functionality, the developer will need to leverage two sets of gateway smart contracts: one deployed on the source chain (chain X) and the other on the destination chain (chain Y). Since Analog Network handles this functionality in GMP protocol, the developer does not need to implement this feature. With Analog-managed gateway smart contracts on chains X and Y, developers can start building cross-chain applications that serve users on any connected chain with a simple integration.

To understand how the GMP protocol works, let us consider an application on chain X that wants to send an arbitrary message to the destination chain Y.

Chain X (source chain):

1. A user/application on chain X calls the $send\_message$ ( ) on the gateway smart contract to initiate the message transfer with the following parameters:
    - destination chainID
    - address
    - payload
    - amount of $ANLOG tokens
    - Contract message (memo) for the destination transaction

    Note: The user/application can also specify the number of Chronicle Workers on chains X and Y that needs to sign the message by calling the ***security smart contract [TBD],*** which defines the shard to be used. However, if the user/application does not specify the shard, then the smart contract uses the default shard, consisting of three Chronicle Workers. Once the call has been initiated, users can track the call by viewing the transaction on the Analog Explorer or through the GMP SDK.
2. User/application prepays the transaction fees to cover the gas fees on the source, Timechain, and the destination chain.
3. The Gas Receiver Smart Contract on chain X converts the paid gas from the source chain's native token into $ANLOG tokens (to cover fees on the Timechain) and into the destination chain's native token (to cover fees on the destination chain).

4. Any Chronicle Worker on chain X can route this message to chain X's gateway smart contract.

Analog Gateway (chain X)

5. Chronicle Workers operating at the gateway on chain X receive the message from their respective connectors and authenticate them. Afterward, each Chronicle Worker forwards the authenticated message to a Collector Worker for TSS-based consensus to be performed. A Collector Worker is a special Chronicle Worker that is elected through NPoS algorithm in each shard for every epoch to coordinate TSS-based processes (KEYGEN and KEYSIGN activities).
6. If a supermajority of the Chronicle Workers appends their signatures to signs the transaction, the Collector Worker propagates the signed payload to the Timechain network.

Timechain Network

7. Time nodes validate the transaction and append its metadata, i.e., an on-chain hash consisting of the event hash, block ID, timestamp, group TSS key, and chain IDs to the Timechain.
8. The rest of the transaction details are stored in off-chain storage.

Chain Y (Destination chain)

9. The connectors on chain Y prepares a signed command authorizing $execute\_message$ ( ) on the destination chain.

4.0 Nodes

This section describes various nodes that run Analog Network.

4.1 Time Nodes

Analog Network maintains a set of validators — also called ***time nodes*** — that participate in the network at two levels:

- Listening to events from external chains, generating proposals for those events, and submitting signed payloads to the Analog Network for further validation processes. ***Connectors*** and ***Chronicle Workers*** handle this role. A connector is a special routine that allows the time node to observe external state changes and report them to Analog Network. In contrast, a Chronicle Worker is an off-chain module implemented within the Timechain SDK that enables the time node to participate in multi-party computation processes.
- Validating proposed blocks to the Timechain. As an NPoS-powered blockchain, Analog Network relies on the time nodes — operating with economic incentives — to maintain its ledger. For example, to participate as either a block proposer or confirmer, the time node needs to stake a minimum of $ANLOG tokens [TBD] in the network. However, token holders that do not intend to run time nodes can also delegate their $ANLOG tokens to existing time node operators and participate in the validation process. The more $ANLOG tokens staked in the network, the more secure the network becomes. Elected validators who successfully sign the blocks receive block rewards denominated in $ANLOG tokens as compensation

for their services. Similarly, nominators (token holders who delegate their $ANLOG tokens to validators) also earn a percentage of the block rewards.

To participate as a time node on the network, nodes must fulfill the following conditions:

Hardware specifications:

1. CPU
    - CPU specifications: x86_64 (Intel, AMD) processor with at least 8 physical cores
    - CPU features: CMPXCHG16B, POPCNT, SSE4.1, SSE4.2, AVX
2. RAM
    - 16 GB DDR4
3. Storage
    - Storage hardware: SSD
    - Storage capacity: xxx TB

Software specifications:

The time node operator needs to run two modules:

1. **Timechain SDK**: This is an SDK that implements the Timechain components, including block authorship, staking, rewards/slashing, connector registration, and vesting schedules, among others.
2. **Connector SDK**: This is an SDK that implements the connectors, allowing the time nodes to fetch finalized transactions from external chains.

Networking:

The internet bandwidth should be at least 300Mbps. In addition, the uptime for the time node should be greater than 99% for maximum profitability.

Besides time nodes, the network will also incorporate two other categories of nodes: Sentinels and storage nodes.

4.2 Sentinels

Sentinels are full nodes operating as off-chain entities on the Analog Network. They can submit fraud proofs that show that a particular fetched block violates the protocol rules. For example, if an external chain is under a 51% attack, any chronicle worker on that chain can submit fraud proof to the Timechain to forestall the fetched event data from being used as a basis for cross-chain communication.

Besides submitting fraud proofs from external chains under attack, Sentinels can monitor chronicle workers and report any dishonest behavior. For example, they can report colluding connectors on the network, allowing the protocol to trigger slashing mechanisms for the affected nodes.

4.3 Storage Nodes

Like Sentinels, storage nodes will also be full nodes that store the entire Timechain data from the genesis block to the current block in off-chain mode. Our tokenomics design has incorporated a storage fund that the protocol will redistribute from past event data transactions to future time nodes. Users who transact on the Timechain will be required to pay the entire transaction costs, consisting of computational and storage fees upfront.

5.0 Proof-of-Time Primitives

PoT is a completely decentralized consensus algorithm where any node (user) can join and propose/confirm blocks without being hindered by excessive levels of hardware or money. It works by selecting validators based on their ranking scores and a *minimum stake of 1 $ANLOG token*. A ranking score is a numerical weighting measure that the algorithm assigns to each validator based on its historical experience (the accuracy with which the node validates data) and tenure of participation.

On the other hand, a stake serves as collateral to deter misbehavior while validating blocks on the network. At a high level, the PoT consensus protocol is a two-step process involving ***soft*** and ***hard voting***. During the soft voting stage, a selected time elector (validator) proposes a block to be included on the Timechain. This process triggers the hard voting phase, where a committee of 1,000 time nodes votes to determine whether the transaction is valid or not. If more than two-thirds of the 1,000 time nodes attest to the transactions, the block gets appended to the Timechain.

5.1 Verifiable Delay Function

The PoT protocol uses a verifiable delay function (VDF) as a mechanism for deterring simulation attacks. The network uses VDF as a random beacon to select new block proposers (time electors) at regular intervals, i.e., time slots with the probability of becoming a time elector biased by the time node's ranking score and stake.

The PoT consensus algorithm runs VDFs for periods called slots, which are periodically computed after one epoch. An epoch is the time duration for which the network randomly determines which nodes propose the blocks (time electors) and which ones confirm the blocks (time nodes) in each time slot. Each epoch has 7,200 time slots.

During each epoch, the network releases a random seed that each node uses to compute VDF based on its ranking score and the fixed stake. Whenever a node finds proof that it qualifies to propose or confirm blocks, it broadcasts it and undertakes its responsibility during its allocated slot, and broadcasts the result alongside VDF proofs. This way, the network prevents malicious nodes from faking data when proposing or confirming blocks.

5.2 Participation Keys

Analog provides two distinct sets of keys:

- **Spending keys**. These are keys that enable a user to prove ownership of $ANLOG tokens. Users can leverage their spending keys to stake the tokens on the network and participate in decentralized governance.
- **Participation keys**. These are specialized keys located on a single node that allows an online user to participate in the consensus process.

The primary goal of separating spending keys from participation keys is to ensure that users' spending keys do not get exposed when their accounts are validating data or participating in the consensus process. Any node that wants to validate data or participate in consensus on the Analog network can generate a participation key for a particular account.

However, such a node can only validate data or participate in consensus if its private key authorizes the transaction, registering the account to go online with a specific participation key. This way, Analog Network keeps the spending keys in cold storage. For enhanced security, Analog Network automatically renews individual participation keys after every 40,000 rounds of data validation and consensus.

5.3 Ranking Score

Any time node can request to validate the submitted data or confirm new blocks. However, for the network to grant permission, such a time node must have a higher ranking score, which is simply a numerical value that the algorithm assigns to each time node. The network uses three parameters to arrive at a ranking score:

1. **Node's tenure on the Timechain**. This indicates how other nodes in the network perceive the time node in terms of its overall contributions to the Timechain. The network uses crypto-economic concepts to assign this value, such as rewards and penalties the node has generated on the network.
2. **Node's historical validation accuracy**. This indicates the number of times the time node has validated accurately. The more the correctly validated data, the higher the value for the node's historical validation accuracy.
3. **Average weighted value of ranking score for the vicinity nodes**. Time nodes do not carry the same weights when it comes to the ranking

score. For example, if time node A validates data and other nodes, say B, C, and D, find this data to be useful, then a special relationship forms between A, B, C, and D. In other words, the ranking score associated with A increases on the network. Suppose another time node, say X, validates data and only one other node, say Y, finds that data to be valuable. The ranking score associated with X will also increase. However, A will have a higher weight than X because it has more relationships. Therefore, when computing the final ranking score for each time node, the network must factor in the connections between the given node and the vicinity nodes.

5.4 PoT Workflow

Consensus via PoT is a two-stage process: block proposal and block confirmation.

5.4.1 Block Proposal

Time electors are a category of nodes that submit block proposals. The network rotates the time electors at regular intervals, known as slots. Each time elector can only validate data during its allotted slot. The soft vote phase starts with a collector submitting the signed TSS payload to the network. During this time, any time node that receives the broadcasted data forwards such data to the time elector.

The time elector collates the posted data, verifies the publisher's signature, and generates the VDF proofs. It then gossips the hashed transaction alongside a VDF proof to the rest of the time nodes in the network.

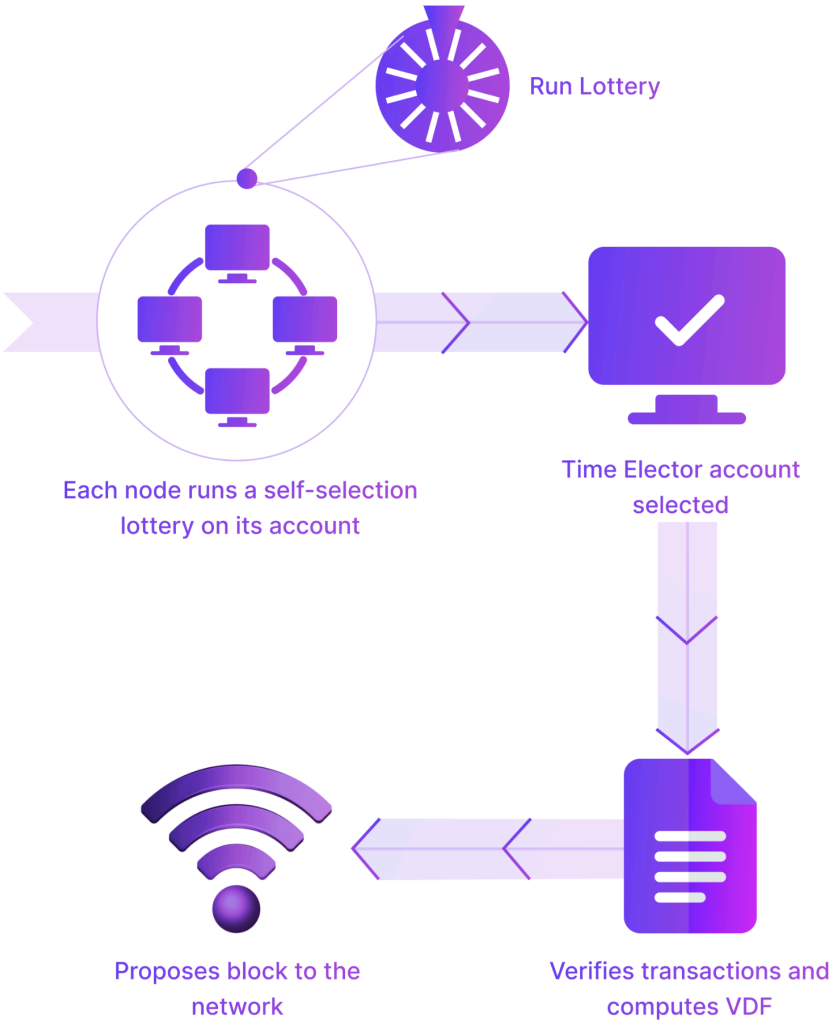

Figure 3: Block proposal process

5.4.2 Block Confirmation

At this stage, the network generates a new set of 1,000 time nodes via a VDF-based lottery process to participate in the consensus process based on their staked coins and ranking scores. Every time node again loops through its accounts to determine if it has been selected to participate in the consensus process.

If selected, the time node determines whether the time elector is indeed a valid proposer that the network selected to propose a new block. Each time node then checks for VDF proofs, double-spending, overspending, and other problems with the proposed block. If the proposed block is valid, the time node accepts it.

When all the 1,000 time nodes have voted to accept or reject the proposed block, the network triggers an end to the confirmation round, triggering the tallying of the votes function that works as follows:

- Each time node broadcasts its vote outcome alongside a VDF output to the other nodes in the consensus committee via a gossip protocol in a P2P manner.
- When each time node receives the vote result, it computes its own VDF to determine if the time node is indeed a valid member of the consensus committee.
- If the received vote output emanates from a valid member of the consensus committee, it increments its vote tally by one. This process

continues until each time node has tallied all the votes from the other consensus committee members.

- If more than two-thirds of the time nodes vote to accept the proposed block, the block gets appended to the Timechain, and all the nodes in Analog Network get notified via a gossip protocol about the new blockchain status. This concludes the block confirmation process and triggers a new round of consensus.

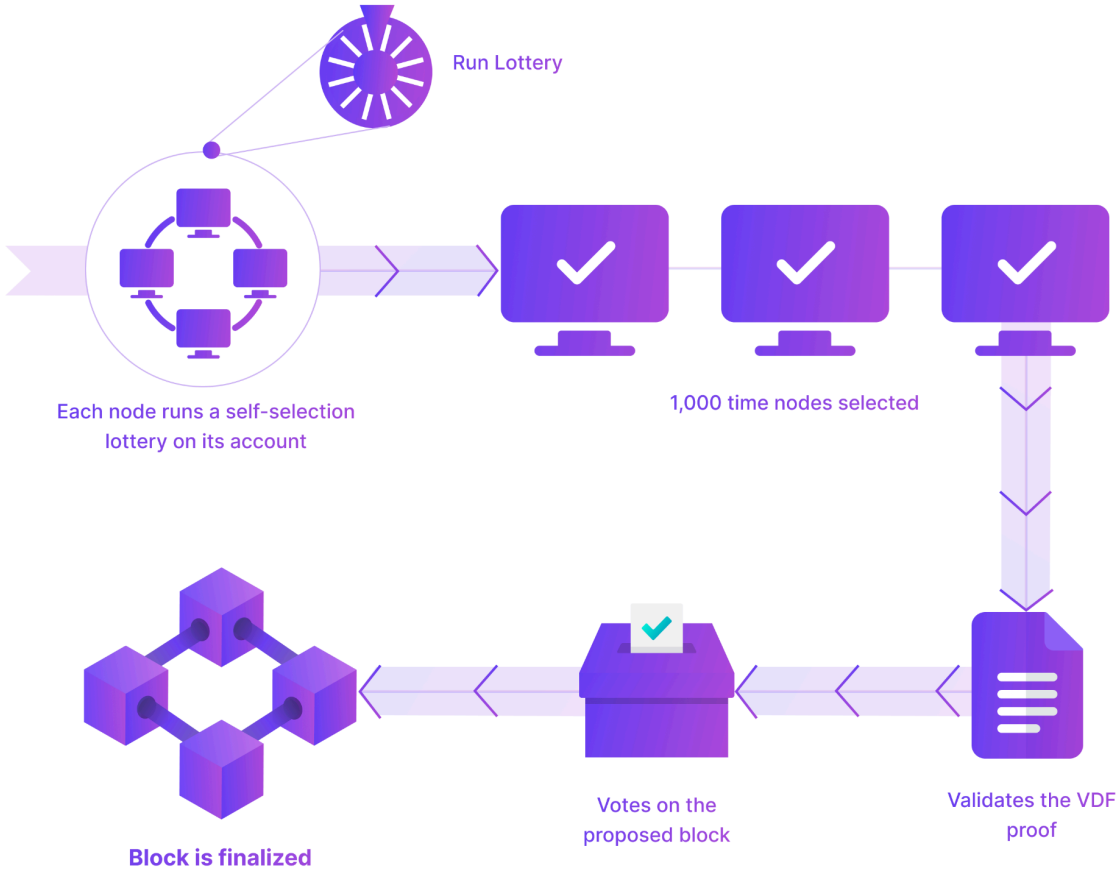

Figure 4: Block confirmation process

6.0 Security Architecture

According to the Fischer-Lynch-Paterson (FLP) impossibility result, a deterministic asynchronous decentralized consensus protocol can have at most two out of three properties. The first is safety, where outputs must be valid and identical for all nodes. The second is liveness, where nodes that do not fail should always generate a valid result. The third is fault tolerance, where the system should survive even if one or more nodes fail.

This section explores PoT's safety and liveness properties, the circumstances under which they can be compromised, and how the protocol prevents them from occurring. Ultimately, we will also explore various practical complexities adversaries can use to attack the platform.

6.1 Safety Guarantees

Unlike classical consensus protocols that usually leverage the longest chain principle, such as proof of work, the PoT mechanism adopts a probabilistic safety mechanism. Practically, this probabilistic safety guarantee is as strong as conventional safety assurances. In the PoT case, there are two stages in each consensus round: *soft vote* and *hard vote*.

During the soft vote, there is only one vote (from the time elector). However, this single vote is accompanied by a hard vote where the process must collect more than two-thirds of the votes for the consensus to take place. Since more than two-thirds of 1,000 time nodes are required for an agreement to occur, two different blocks cannot be generated simultaneously.

This ensures that:

1. The network can never generate two or more blocks at the same height.
2. The network can never process a transaction from a block at height (H+1) before executing all the transactions from the current height (H).
3. The network can never append the cross-chain data to the Timechain unless it has been signed by Chronicle Workers and time nodes in a soft vote and hard vote.

6.1.1 Safety Analysis

First, let us consider safety property number one. The PoT consensus is a two-stage process involving a *soft* and *hard vote*. A single vote from the time elector must be accompanied by a supermajority (two-thirds of 1,000 time nodes) for the decentralized network to commit any block to the Timechain. In practice, no two blocks can be committed to the Timechain simultaneously.

Next, let us consider safety property number two. The PoT algorithm has an inbuilt $isitValid()$ function that determines whether a block is valid. This function guarantees that even if adversarial nodes exceed one-third, no invalid block gets appended to the Timechain. In other words, for a block at height $(H + 1)$to be considered valid, PoT requires it to embed the current state of the block (H).

Finally, let us consider the safety property (3). From (1) and (2), it follows that unless more than two-thirds of the nodes are adversarial (which is highly unlikely due to inbuilt incentive mechanisms), the network can never append the cross-chain data to the Timechain unless it has been signed by Chronicle Workers and time nodes in a soft vote and hard vote.

6.1.1.1 Safety Incentive Mechanisms

The PoT protocol ensures safety guarantees under the "*less than a third*" adversarial conditions. However, due to safety property (2) outlined earlier, it is vital to know why the "*less than a third*" adversarial condition holds.

The protocol requires both Chronicle Workers and time nodes to lock a fixed number of tokens (to be decided through a decentralized governance structure). The locked tokens are registered with a smart contract. After the onboarding process, each tesseract and time node must wait until the start of the next epoch before participating in the cross-chain transfer process.

If any Chronicle Worker or time node decides to opt-out of the interoperability process, it has to pause until the start of the next epoch to unlock its tokens. If a group of Chronicle Workers decides to collude in a TSS process and fetches erroneous data that time nodes do not approve, their staked tokens are automatically slashed.

6.2 Liveness Guarantees

The liveness property of the protocol is guaranteed so long as "*greater than or equal to a third*" of the time nodes are ever offline. No epoch should ever comprise more than a third of time nodes that cannot communicate with the rest of the decentralized nodes. With regard to liveness, PoT ensures that:

1. Time nodes will eventually append a block to the Timechain at every height.
2. Time nodes will eventually process every transaction for any appended block.

6.2.1 Liveness Analysis

Let us start the analysis by considering the liveness property (1). As is the case with safety property (1), this property assumes that any block committed to the Timechain must undergo two stages: soft vote and hard vote. If more than two-thirds of the time nodes are online, the data will eventually be appended to the Timechain.

Now let us examine the liveness property (2). Again, as is the case with safety property (2), the nodes can always generate the block so long as more than two-thirds of them are online. This essentially means that executing the transaction will progress under the "*less than a third time nodes*" condition.

6.2.1.1 Liveness Incentives

The liveness property requires the time nodes to contribute non-zero cost resources ($ANLOG tokens) to participate in the consensus process. Similarly, this would also require an incentivization mechanism that exceeds non-zero costs to reward honest block proposers and confirmers.

To this end, time nodes that want to participate in the validation consensus process must lock a fixed amount of $ANLOG tokens in a smart contract to be selected. Before an interoperable execution takes place, consensus must occur on the network. Time electors and time nodes that successfully validate and confirm blocks to the Timechain receive $ANLOG tokens as rewards and have their ranking scores automatically increased.

6.3 Attack Vectors

In the subsequent section, we look at two key attack vectors that are likely to be perpetuated on the network: Sybil and bribery attacks.

6.3.1 Sybil Attacks

A Sybil attack is a security threat in Blockchains where a user attempts to take over the network by generating multiple accounts and nodes. Sybil attackers may out-vote the honest nodes in a decentralized network if they generate enough fake identities. Typically, such attackers can decline to collate or broadcast the transactions, effectively blocking other nodes from undertaking their functions.

In Analog, a Sybil attacker must control more than a third of the time nodes in a particular slot to corrupt the communication. This is highly unlikely because an attacker will need to lock more $ANLOG tokens in their wallet to launch the attack. If honest nodes in the network detect this kind of attack, the attacker would be severely punished via a slashing mechanism.

6.3.2 Bribery Attacks

This is an attack where an adversary attempts to corrupt the network by coordinating with more than a third of the nodes through bribery. If the attacker succeeds in bribing more than a third of the nodes, they can corrupt the network.

In a PoT-enabled chain such as the Analog network, the attacker has to pay at least a third of the time nodes. This is not profitable, provided the sum value of all

staked $ANLOG tokens for all the malicious nodes is greater than three times the value of tokens in the source chain. Analog enforces this constraint through a dynamic decentralized governance mechanism that adjusts the minting, burning, and transaction fees.

7.0 Analog Network Economics Overview

We have conceived Analog Network's tokenomics system as a healthy, long-term, self-sustaining platform with participant incentives aligned towards achieving complete decentralization and security. The contributions of publishers, subscribers, and time nodes — the main participants in the Analog ecosystem — are discussed below.

7.1 Token Specifications

Analog Network is powered by $ANLOG tokens. The network uses the $ANLOG token to incentivize positive actions by nodes and subscribers while penalizing deceptive entities on the platform.

Table 1 summarizes the token specifications:

Table 1: $ANLOG specifications

Feature	Specification
Token name	ANLOG
Token Ticker	$ANLOG
Total token supply	90,579,710 $ANLOG
Mintable?	Yes
Burnable	Yes
Pre-minted	Yes

Legal Disclaimer

$ANLOG Token is designed to be a utility token that functions as the unit of payment and settlement between participants who interact within the Analog ecosystem. $ANLOG Token does not in any way represent any shareholding, participation, right, title, or interest in the Governing body, the Issuer, its affiliates, or any other company, enterprise, or undertaking, nor will $ANLOG Token entitle token holders to any promise of fees, dividends, revenue, profits or investment returns, and are not intended to constitute securities in the United States or any relevant jurisdiction. The ownership of $ANLOG Token carries no express or implied rights other than that which may be afforded by Analog and/or any other third parties who may use such Tokens.