Analog:

The Decentralized Timegraph

Victor Young[1] Peter Wanjala[2] Florian Franzen[2] Sanchal Ranjan[2]
Eric Wang[2]

Timepaper (Version 4.0)

Abstract

We present *Analog*, a suite of interoperability protocols built on the Substrate SDK. We articulate the vision for Analog, designed to become a one-stop shop for multi-chain and cross-chain solutions, addressing native DApp development needs. At the core of Analog is the *Timechain*—a consensus-agnostic protocol maintained by a dynamic and decentralized validator set—that serves as a core ledger for all activities occurring in the Analog ecosystem. The most appealing feature of the Timechain is the security it offers for DApps that want to operate in multi-chain and cross-chain environments, including Analog Watch, Analog GMP, and Analog Automation, which we will discuss in this paper.

CONTENTS

Abstract
1.0 Introduction & Hypothesis
    About This Timepaper
2.0 The Timechain: Analog Core Protocol
    2.1 Timechain Components
        2.1.1 Threshold Signature Schemes (TSS)
        2.1.2 Consensus Engine
    2.2 Features & Benefits
    2.3 Validator Network
        2.3.1 Timechain Nodes
        2.3.2 Chronicle Nodes
3.0 Analog Tech Stack
    3.1 Analog Watch

[1] The author is a founder member of Analog.
[2] Analog Labs team

3.1.1 Design Overview
3.1.1.1 Hosted Service
3.1.1.2 On-Demand Data Validation Service
3.1.1.3 Watch Tokenomics
3.2 Analog GMP
3.2.1 Architecture
3.2.1.1 On-Chain Components
3.2.1.2 Off-Chain Components
3.2.2 Transaction Lifecycle
3.2.3 Use Cases
3.3 Analog Automation
4.0 Tokenomics
References

1.0 Introduction & Hypothesis

As the use of cryptocurrencies becomes mainstream and mass-market adoption of Decentralized Finance (DeFi) surges, 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 days of the Internet when there was no Transmission Control Protocol/Internet Protocol (TCP/IP)[1].

For users to interact with Web3 products across different domains, there is an inherent need to provide mechanisms for interacting with those chains. Due to a lack of seamless interoperation across different chains, DApp developers must make tough choices on which blockchain platform to build their innovative applications.

Similarly, users must 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 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.

Since the original Analog Timepaper was published in early 2022, dozens of "multi-chain" and "cross-chain" solutions have emerged. While it is evident that there is huge market demand for multi-chain and cross-chain solutions within the Web3 space, none have yet gained widespread adoption. The reasons for this can be summarized as follows:

- The proposed solutions are often specialized and only provide one or two services. For example, Axelar[2] and LayerZero[3] have focused solely on General Message Passing (GMP), while The Graph[4] protocol only provides an indexing solution.
- The proposed solutions are often overcomplicated and badly explained. For example, while Chainlink addresses multi-chain and cross-chain challenges under one roof through its Oracle and CCIP protocol[5], it is not built from ground up.
- In the rapidly evolving Web3 space, the frequency and cost of software vulnerabilities, particularly those related to cross-chain bridges[6], have become a significant concern.

In this paper, we present the first intrinsically secure interoperability platform for Web3 that addresses both multi-chain and cross-chain issues. Analog's primary objective is to streamline DApp development in multi-chain environments by providing pluggable toolkits under one roof, enabling developers to concentrate on their core tasks. Our thesis is that achieving this goal will allow Analog to unlock new possibilities for DApp developers in a multi-chain/cross-chain ecosystem, enabling them to build apps that achieve seamless chain-agnostic interoperability through Analog's Timechain.

1.1 About This Timepaper

This is the fourth iteration of the Timepaper, with the initial version being published in early 2022. This updated version is intended to reflect the actual design and architectural components of Analog in practice, with many changes having been made to the protocol after extensive research and implementation.

This Timepaper does not aim to offer a canonical account of every component within the Analog Network. Instead, it seeks to provide a medium-level overview of the entire protocol, explaining how the various components fit together and the rationale behind the design decisions related to our thesis.

2.0 The Timechain: Analog Core Protocol

The core aspect of Analog revolves around the ***Timechain***, a sovereign blockchain ecosystem that serves as a ledger for all its activities—essentially acting as an ***Accountability Layer***. Although the Timechain is built on Substrate SDK and leverages some of the platform's features, including forkless upgrades, fast block times, and security-focused development, it remains an independent blockchain.

To interact with external chains, the Timechain uses ***Multi-Party Computation*** (MPC), and in particular, ***Threshold Signature Schemes*** (TSS), to create aggregate keys held by off-chain validators (also called ***Chronicle Nodes***). These Chronicle Nodes manage and control simple contracts/gateways on the supported blockchains by attesting (reading) and executing (writing) on those chains. The Chronicle Nodes are paired with the Accountability Layer (i.e., the Timechain) that processes, tracks, and executes events/instructions.

2.1 Timechain Components

The Timechain has two essential components that allow it to deliver seamless interoperability: **TSS** and ***Consensus Engine***.

2.1.1 Threshold Signature Schemes (TSS)

The blockchains that Analog supports are backed by TSS accounts such that more than two-thirds of Chronicle Nodes must collectively co-authorize any request made to the protocol. The TSS protocol is required for any general-purpose read (view/read) request on external chains and cross-chain smart contract execution calls, a core feature of Analog GMP and Analog Automation.

The TSS protocol is designed with the following principles in mind:

- **Open membership**: While currently designed as a permissioned system, our goal is to create a permissionless network of Chronicle Nodes where anyone can join and leave the network provided they have staked enough ANLOG tokens and their nodes meet the minimum hardware requirements.

- **Economic incentives and slashing**: The TSS protocol requires the Chronicle Nodes to act honestly when attesting to cross-chain requests. This requires them to lock a certain threshold of ANLOG in the relevant shards (see below for more details). At the end of each epoch, each Chronicle Node that participated in the TSS and operations are reimbursed the costs they have incurred in processing the specified operations along with an incentive (i.e., safety/profit factor). However, the protocol also identifies malicious nodes and punishes them via a slashing mechanism. Check out the Tokenomics Paper[7] to learn more.
- **Key rotations**: Besides splitting the keys into multiple shares, the keys are constantly rotated to achieve maximum security.

To enhance scalability, the entire Chronicle Node network is partitioned into ***shards***, with each shard consisting of a number of independent Chronicle Nodes that leverage MPC processes to reach a consensus on the validity of requests made to the network.

To participate in a shard, each Chronicle Node must set up a node[8], which essentially means registering it to the Timechain. The Timechain serves as a record for all the registered shards and the corresponding Chronicle Nodes that should participate in those shards.

2.1.2 Consensus Engine

Timechain is an NPoS-based, public blockchain maintained by a decentralized set of on-chain validators (also called ***Time Nodes***) and ***Chronicle Nodes*** that serve as observers on external chains and reach consensus on those chains through TSS. The Timechain could also be considered a hub, where all protocol events are recorded, executed, and triggered.

The Consensus Engine creates an ***Accountability Layer*** for all transactions across the Analog Network. Some of the protocol events that the Timechain manages include:

- Registration and tracking of validators.
- Registration and management of shards through key rotations.

- Assignment of shards to Chronicle Nodes.
- Initiation of tasks on the destination shards to trigger execution of cross-chain requests.
- Staking and slashing mechanisms.
- Governance mechanisms.

2.2 Features & Benefits

Using the Timechain as an Accountability Layer instead of externally operated Layer1(L1s) or Layer2s (L2s) offers significant benefits to the entire Analog Network ecosystem, such as:

- **Security**: The Timechain is built on the battle-tested NPoS consensus protocol and secured by a set of decentralized Time Nodes that must have a sufficient stake (i.e., self-stake plus that of nominators) to serve as collateral. Any malfeasance behavior can lead to the slashing of this stake, serving as a powerful deterrent against attacks.
- **Scalability**: The Timechain is architected to handle high volumes of requests efficiently through transaction batching and other automated processes that save on gas.
- **Governance**: The Timechain provides governance mechanisms that allow holders to easily undertake protocol upgrades, parameterization adjustments, and other aspects of the Analog ecosystem. This control within the protocol provides a high level of flexibility and upgradeability, ensuring that the network evolves according to Analog Community preferences.

2.3 Validator Network

There are two categories of validators on the Analog Network: Timechain Nodes and Chronicle Nodes.

2.3.1 Timechain Nodes

Timechain Nodes, or simply ***Time Nodes***, secure the Timechain by verifying and committing new blocks. Like all other Substrate-based chains, Time Nodes participate in an NPoS consensus in exchange for block rewards.

The NPoS protocol, a consensus mechanism used in Substrate-based blockchains like the Analog Network, allows holders of the native token (ANLOG in Analog's case) to participate in the network's decision-making processes.

In this protocol, Time Nodes are selected to propose and validate blocks based on their own stake (in ANLOG tokens) plus their nominated stake in the network. Any token holder can nominate their stake to a Time Node candidate. Refer to Tokenomics Paper[7] to learn more.

2.3.2 Chronicle Nodes

Chronicle Nodes provides two primary services:

- **Attestation service**: Chronicle Nodes monitor inbound cross-chain requests from connected chains, validate them using TSS, and then post them on the Timechain. To undertake this role, each Chronicle Node needs to run a Connector instance. A Connector is an entity that listens to inbound requests from other chains, parses them into a unified format, and propagates the message to the Chronicle Nodes.
- **Relaying service**: Chronicle Nodes serve as relayers by picking up the cross-chain request on the Timechain (i.e., GMP payload) and forwarding them to the destination chain for execution.

Multiple chronicles connected to the same external chain form a shard on the Timechain. The Timechain determines the size of the shard. Once a ***shard*** is formed, it can start interacting with external chains.

The security of the Chronicle Node network will ultimately be controlled by a locking/staking mechanism, where Chronicle Nodes effectively loan out ANLOG through their relevant shards to participate in MPC processes. The ***shard stake*** allows Chronicle Nodes to undertake a number of activities, including:

- Funding protocol-level operations/tasks, such as registration and deregistration of shards.
- Funding operations that allow a GMP task to be submitted to the destination chain.

- Funding operations that allow a GMP message to be submitted to the Timechain.

Note that at the launch of the mainnet, Analog will operate Chronicle Node services as a centralized entity. However, as Analog progresses on its roadmap, the goal is to transition to a trust-minimized infrastructure, allowing anyone to join and offer Chronicle Node services to the network. For more details, check out the Tokenomics paper[7].

3.0 Analog Tech Stack

The Analog tech stack consists of the following components, which are all designed to break down the barriers associated with the development of DApps and cross-chain communication in a multi-chain environment, as illustrated in Figure 1:

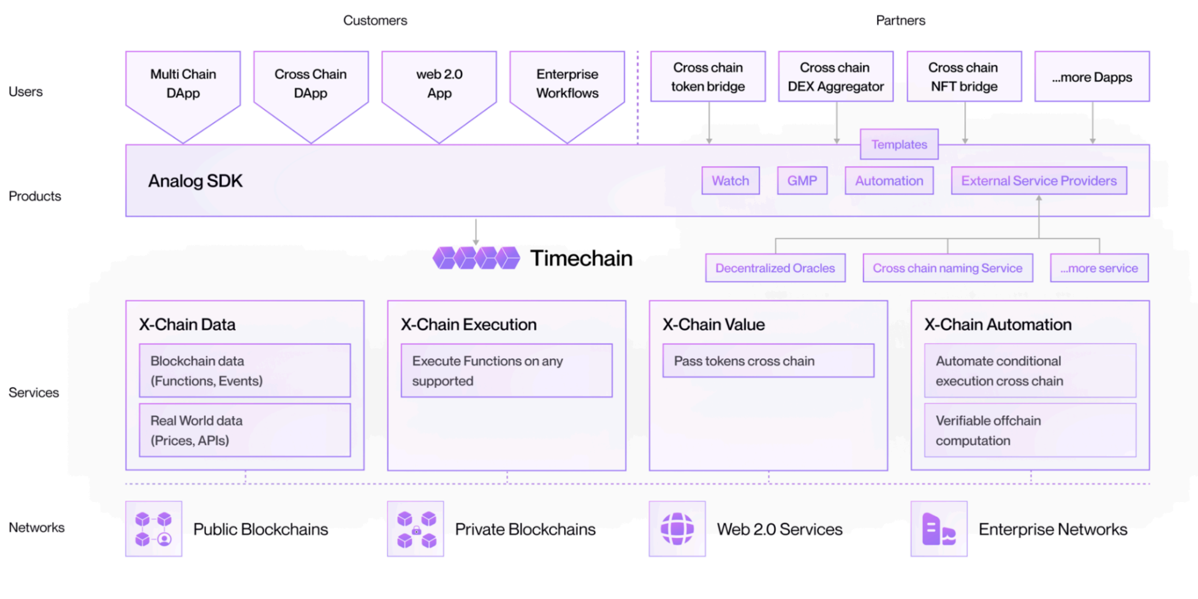

Figure 1: Analog Tech stack

1. **Networks:** Analog's primary goal is to help DApp developers extend their applications beyond a single blockchain, whether those networks are public, private, Web2 services, or enterprise networks. We will first prioritize integrating public blockchains, beginning with ***Ethereum***, ***Astar***, ***Polygon***, and ***BNB***

Chain, before expanding to other networks, including private, Web2 services, and enterprise networks.

2. **Services**: Analog aims to support multiple chains while offering integrated services under one roof. This contrasts with other solutions, such as [2], [3], [5], that specialize in one or two services or require developers to integrate multiple services. Some of the services the network will support include: ***Cross-chain Data***, ***Cross-chain Execution***, ***Cross-chain Value Transfer***, and ***Cross-chain Automation***.
3. **Timechain**: For more details, see "The Timechain: Analog Core Protocol."
4. **Products**: Each product we support will include easy-to-use and pluggable toolkits (i.e., SDKs) that run on the Timechain. Currently, two products are running on the Timechain: ***Analog Watch***, which provides developers with seamless access to blockchain data, and ***Analog GMP***, which enables cross-chain smart contract execution calls.
5. **Unified API**: A single GraphQL API that provides a turnkey developer experience, allowing builders to query smart contracts from any supported chain in a seamless and scalable manner.

In the subsequent sections, we present three examples of Timechain-enabled platforms and explain them in more detail just to illustrate the versatility of the Timechain and how it can enable a host of new services.

3.1 Analog Watch

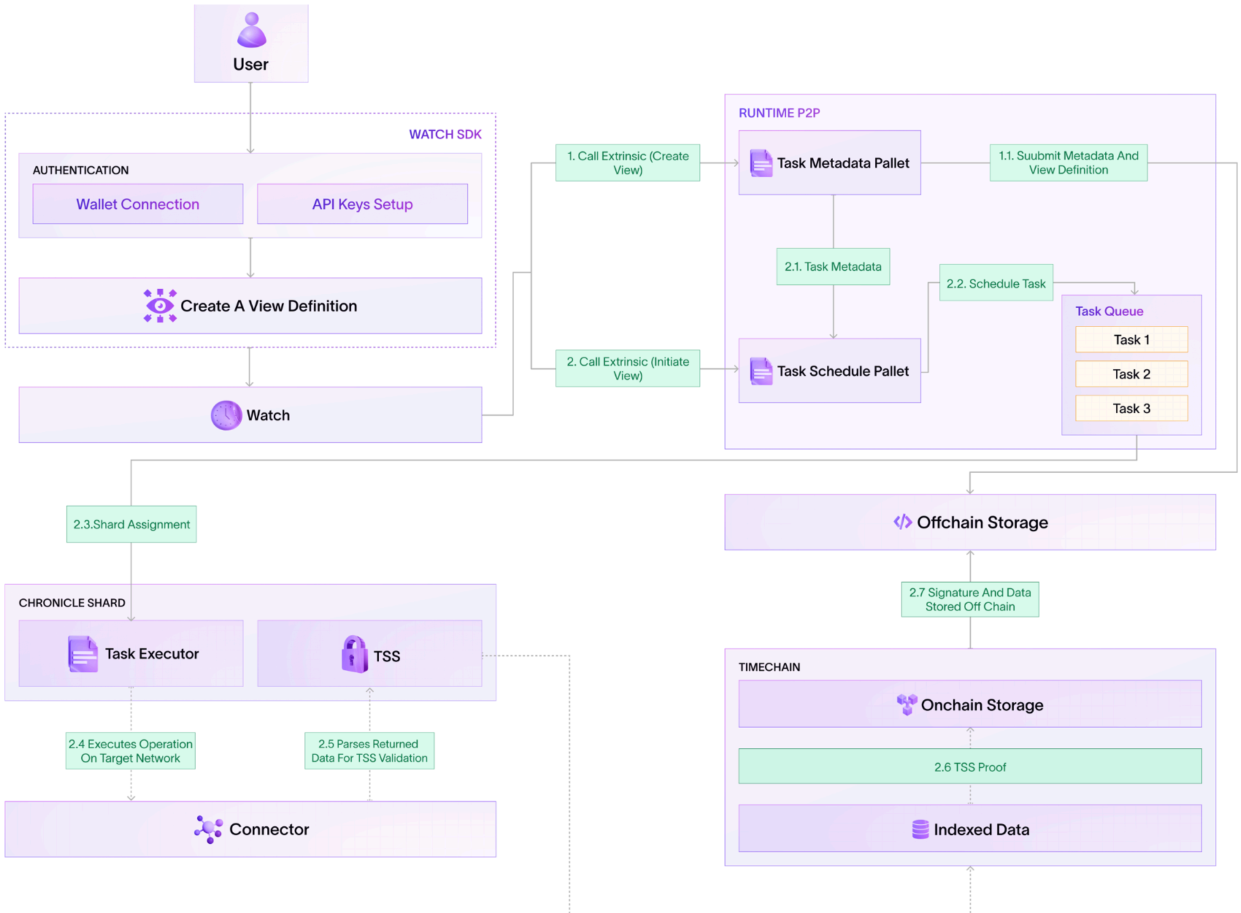

Figure 2: Analog Watch

Analog Watch is a Web3 data provider solution that aggregates and indexes data from various blockchain networks, including Ethereum, Astar, and Polygon. This curated data is then made available to DApp developers for use in various projects. It facilitates operations between L1 and L2 blockchains and DApps, allowing developers to focus on their core use cases without the need to invest in costly and tedious custom backend systems for Web3 data processing.

Currently, anyone can use Analog Watch to extract and query data from supported networks quickly and at zero cost, albeit through a centralized approach. In the future, Analog will enable this same scalable and performant solution but in a trust-minimized and verifiable manner.

3.1.1 Design Overview

The protocol is designed to accommodate two deployment options: the ***Hosted Service***, currently in use, and an ***On-Demand Data Validation Service*** that will become available in the future.

3.1.1.1 Hosted Service

The Hosted Service is a centralized model that allows anyone to deploy and query Views without running their own infrastructure. It offers a user-friendly UI for listing smart contracts and deploying Views[9] (via the ***Watch Portal***), allowing anyone to explore them on the network.

Currently, the Watch Portal does not allow direct data querying from the protocol. Users must utilize the ***Watch SDK***[10] or the ***GraphQL IDE*** to query data directly from Analog Watch via a unified GraphQL endpoint.

In the current model, Analog Labs serves as a Data Provider for Analog Watch, ensuring that data specified by the View Definition is ingested block-by-block and served to users. The View Definition is saved off-chain and is updated by the protocol each time a new piece of data is updated as specified by the number of Cycles. For the View data to be queryable and earn revenues for the relevant Data Designers, they need to be sponsored by Data Collectors.

The sponsorship metadata is saved on the Timechain and updated by Analog Watch each time a user queries them. The sponsorship metadata describes the following details:

- Data Collector (s)who has sponsored the View.
- The number of Cycles the View has been sponsored.
- etc.

3.1.1.2 On-Demand Data Validation Service

Although the Hosted Service provides convenience, it is just the beginning. The On-Demand Data Validation Service, a trust-minimized approach to data access, is on the horizon. It will provide cryptographic guarantees for the validity of the deployed artifacts (e.g., smart contracts and Views), opening up new possibilities.

It will also allow anyone to validate specific queries by submitting requests to the Timechain. All the query responses are signed via Threshold Signature Schemes (TSS) by the Chronicle Nodes (Data Providers)/shard that executed the query, serving as a commitment to the query response.

Here is how the On-Demand Validation Service would work from a technical standpoint:

1. First, users list the smart contract(s) and the relevant functions that are of interest to particular use cases.
2. A Data Designer uses the listed contract(s) to specify the View Definition, describing how data from supported chains needs to be indexed and queried. When specifying the View Definition, Data Designers can select the Hosted Service or the On-Demand Data Validation Service.
3. The View is deployed on the Analog Watch. Any Data Collector who deems the View useful can sponsor it to make it queryable for a certain number of cycles. Alternatively, a Data Designer who has deployed the view can sponsor it. The sponsorship metadata is stored on the Timechain.
4. If the Data Designer opts for the On-Demand Data Validation Service:
    - The View metadata is stored on the Timechain.
    - The Timechain schedules a task—also called a *Timegraph task*—on the Timechain, and the source shard (i.e., Chronicle Nodes assigned to the relevant blockchain) is randomly assigned to handle that task.
    - Upon successfully scheduling the Timegraph task, the source shard continuously scans for new blocks on the defined blockchains. Whenever a new block is discovered, each Chronicle Node on the source shard appends its signature to the fetched data.
    - If a supermajority of Chronicle Nodes within the shard attests to the validity of the fetched data, an aggregated signature (i.e., TSS Data Hash) is generated and stored on the Timechain, and the View metadata is updated.

- Meanwhile, the actual data is stored in an off-chain Postgres database.
- At this point, anyone can query the database via the Watch SDK or submit a query for on-chain validation, provided the View remains active (i.e., funded).

3.1.1.3 Watch Tokenomics

Check out [11] for more details.

3.2 Analog GMP

General Message Passing (GMP) is a general interoperability notion in Web3 that a DApp or smart contract on one chain, say Ethereum, should be able to invoke an action on another chain, say Astar. The invoking DApp or smart contract sends a message, which can be an asset or function call, to another DApp or smart contract and relies on these artifacts and their supporting infrastructure to run some code on another chain.

Analog GMP is a core interoperability feature that allows messages to flow freely and securely across supported blockchains. The GMP concept is encapsulated in the phrase "***cross-chain smart contract execution calls***[12]" and is what Analog envisions for the future of Web3. The ultimate goal of Analog GMP is to standardize DApp development in a multi-chain environment, the same way TCP/IP standardized Web2 applications.

DApp developers can leverage Analog GMP to build cross-chain DApps with coherent application logic, shared states, and efficient liquidity utilization. With GMP protocol, 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.

Let us consider a simple Cross-chain Domain Name Service as an example application that would use Analog GMP.

Decentralized Name Service (DNS) platforms like ENS are gaining popularity in Web3 because they enable the translation of human-readable names into wallet addresses. Ideally, in an ideal Web3 ecosystem, DNS platforms would not be confined to a single blockchain. This is not possible in the current Web3 ecosystem because of the siloed nature of blockchains.

Integrating Analog GMP would enable DNS platforms to interact with any blockchain, provided it is supported by Analog Network. This would provide users with a unified identity across all blockchains, eliminating the need for multiple domain names on different chains. The GMP protocol would notify each instance of the naming service on supported chains whenever new domains are registered using a GMP protocol. It would also require a method to perform "lookups" against a global name registry across all supported blockchains.

3.2.1 Architecture

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

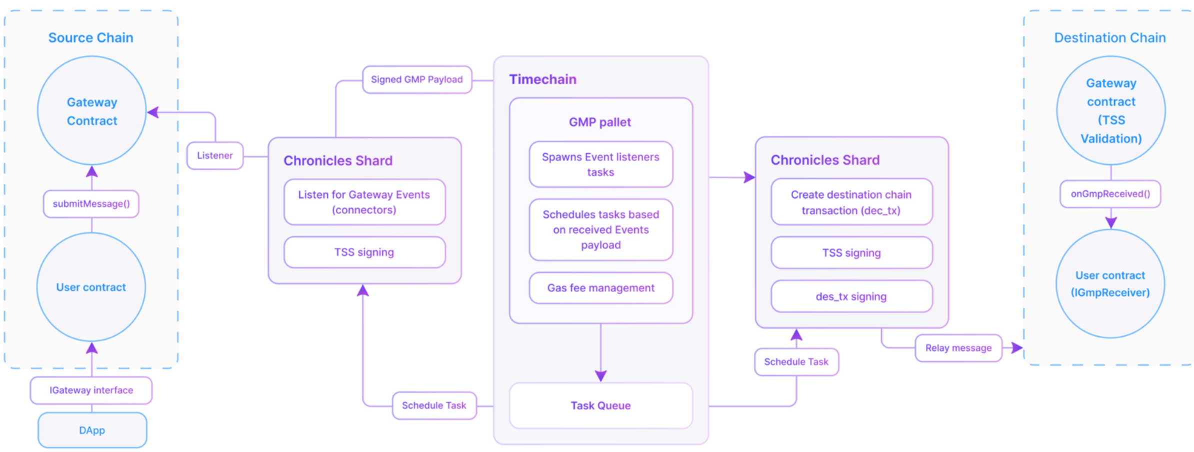

Figure 3: High-level illustration of GMP protocol

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

3.2.1.1 On-Chain Components

There are three primary on-chain components:

1. **Gateway Smart Contracts (GSCs)**: These are the contracts the Chronicle Nodes observe and which fundamentally facilitate cross-chain message passing.

Analog Network supports one GSC on each blockchain, and this is the contract that the source and destination ***shards*** need to observe.

2. **Application Smart Contracts (ASCs):** These are smart contracts that developers deploy on the source and destination chains. An ASC is essentially the cross-chain DApp or an existing ecosystem protocol.
3. **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. Any token holder can also nominate/delegate their stake to any existing Time Node operator and share the staking rewards. See the Tokenomics Paper[7] for more details.

3.2.1.2 Off-Chain Components

There are three primary off-chain components:

1. **Chronicle Nodes**: These nodes observe external chains for certain state changes, transactions, or events associated with a registered smart contract address and report them to the Timechain. To interact with external chains, each Chronicle Node runs a ***Connector***. ***Connectors*** are entity instances that listen to incoming requests from other blockchains, parse them into a unified format, and send them to the Chronicle Nodes. The Chronicle Nodes then verify the validity of these requests using MPC and forward them to the Timechain. To be part of the Analog Network ecosystem, all Chronicle Nodes must operate a Connector instance and lock a sufficient number of ANLOG tokens. For more details, check out the Tokenomics Paper[7].
2. **Shards**: Sharding is a process that Analog uses to parallelize the Chronicle Nodes network into partitions—called ***shards***—to achieve multi-chain operability. Analog GMP uses the sharding mechanism model to promote scalability among its Chronicle Node networks. In the context of Analog GMP, each cross-chain request is monitored by two shard committees: one on the source chain that attests to the transaction's finality and reports the outcome to the Timechain, and the other one on the destination chain that executes the

request on that network. Each of these shards needs to run a GSC, whose public key is registered and tracked by the Timechain.

3. **Threshold Signature Schemes**: See "Threshold Signature Schemes (TSS)" for more details.

3.2.2 Transaction Lifecycle

To understand how Analog GMP protocol would work, let us consider an ASC on chain X would send an arbitrary message to another ASC on the destination chain Y.

Chain X:

1. A user on chain X prepares the message payload.
2. An ASC on chain X uses the X's GSC interface to call the submitMessage() function on that Gateway contract.
3. The GSC processes the transaction and emits an event (i.e., GmpCreated event).
4. The source shard assigned to manage the GSC on chain X listens to and validates the GmpCreated event.

Timechain:

5. The Timechain executes the transaction and creates a GMP message for the destination shard to process.

Chain Y

6. The destination shard randomly selects a Signer Node[3]. The Signer sends and executes the transaction by calling the execute() function, and the GSC emits the GmpExecuted event.
7. The GSC calls the onGmpReceived function on the ASC/DApp to complete the cross-chain execution call.
8. Timechain waits for the transaction to get finalized, marking it as successful.

---

[3] An Analog Signer is any Chronicle Node selected—via a randomized algorithm— within a shard to sign and execute Timechain-validated transactions on the destination chain's GSC.

3.2.3 Use Cases

In this section, we outline the use cases of Analog GMP. We anticipate a growing demand for the cross-chain smart contract execution functionality offered by Analog GMP. This functionality will complement and enhance both existing and new DApps in the cross-chain ecosystem by providing a secure and reliable mechanism for cross-chain composability.

We present five examples of Analog GMP-enabled use cases and how the protocol can realize them: (1) ***Cross-chain Domain Name Service***, a form of providing unified identity across all blockchains that eliminates the need for multiple domain names on different chains; (2) ***Cross-chain NFTs***, a platform that allows users to mint NFTs on a single chain and propagate/leverage these assets on any supported chain; (3) ***Cross-chain DEXs***, DEXs that provide a more efficient liquidity routing via seamless message passing between the supported chains; (4) ***Cross-chain Lending/ Borrowing***, platforms that allow users to seamlessly utilize their assets on one chain as collateral for another protocol on a supported blockchain without performing multiple manual steps; and (5) ***Cross-chain Liquidation Protocols***, that is, platforms that can serve as central routing and aggregation layer in the Web3 ecosystem.

For more details about these use cases, see [13].

3.3 Analog Automation

Check out the Analog Automation position paper [14] for more details.

4.0 Tokenomics

See [7] for more details about Analog Tokenomics.

References

[1] M. Popa, “Drawing Parallels between TCP/IP and the Blockchain,” Blockchain Works. Accessed: Jul. 22, 2024. [Online]. Available: https://blockchain.works-hub.com/learn/Drawing-Parralels-between-TCP-IP-and-the-Blockchain

[2] Axelar, “Axelar:Connecting Applications with Blockchain Ecosystems,” p. 15, 2021.
[3] LayerZero Labs, “LayerZero.” 2024. [Online]. Available: https://layerzero.network/publications/LayerZero_Whitepaper_V2.1.0.pdf
[4] “The Graph Protocol,” The Graph. Accessed: Jul. 23, 2024. [Online]. Available: https://thegraph.com/
[5] “Chainlink 2.0 and the future of Decentralized Oracle Networks | Chainlink.” Accessed: Jul. 22, 2024. [Online]. Available: https://chain.link/whitepaper
[6] “A Timeline and History of Recent Cross-Chain Bridge Attacks | Analog Insights.” Accessed: Jul. 22, 2024. [Online]. Available: https://www.analog.one/blog/a-timeline-and-history-of-recent-cross-chain-bridge-attacks
[7] “Timechain Tokenomics Documentation.docx.” Accessed: Jul. 23, 2024. [Online]. Available: https://docs.google.com/document/u/1/d/e/2PACX-1vRCpMd7VmDxqGYXYZfj-1RyHdWRU6DfBf1tTad4vlpmmvp4Coavwh9mmT42Te665A/pub
[8] “Running a Timechain Node | Documentation.” Accessed: Jul. 23, 2024. [Online]. Available: https://docs.analog.one/documentation/node-operators/running-a-timechain-node
[9] “Architecture | Documentation| What Is a View?” Accessed: Jul. 23, 2024. [Online]. Available: https://docs.analog.one/documentation/developers/analog-watch/overview/architecture
[10] “Quickstart | Documentation.” Accessed: Jul. 23, 2024. [Online]. Available: https://docs.analog.one/documentation/developers/analog-watch/quickstart
[11] “Analog Watch Tokenomics .docx.” Accessed: Jul. 23, 2024. [Online]. Available: https://docs.google.com/document/u/1/d/e/2PACX-1vR6vgYFvmS3kDDmwjuJMG6hcy9g58sUi90nY6vC8sbztaF0pSrVWyURmRClbtqnvA/pub
[12] “Introduction To Cross-Chain Smart Contract Execution Calls | Analog Insights.” Accessed: Jul. 23, 2024. [Online]. Available: https://www.analog.one/blog/introduction-to-cross-chain-smart-contract-execution-calls
[13] Analog Labs, “Analog GMP Use Cases,” Notion. Accessed: Jul. 23, 2024. [Online]. Available: https://teamanalog.notion.site/GMP-Use-Cases-455934a1b1324d8bb8d377305636959a
[14] “Analog Automation: Position Paper.” Accessed: Sep. 04, 2024. [Online]. Available: https://docs.google.com/document/u/1/d/1A7dewi_4VZ4S2PyrpVfoM9TtKxOAEf9XxEacDw3X9Bk/pub