

RedStone

WHITE PAPER

CRYP2NOVA

Archived copy · cryp2nova.com

Contents

INTRODUCTION **3**
- The oracle problem 3
- Current solutions to the Oracle Problem 3
- Our vision 5
  - Affordable Storage 5
  - On demand fetching 5
  - Flexible Data Streams 6
  - Data Integrity 7
  - Cross-Chain Oracle Solutions 8
- Possible applications 8

SYSTEM ARCHITECTURE **10**
- Top level design 10
- Data fetching 13
- Data management 14Data delivery 19
  - Data packing (off-chain data encoding) 19
  - Data unpacking (on-chain data verification) 20
- Data integrity 21
  - Dispute process 22
- Node management 25
  - Node deployment 25
  - Nodes monitoring 26

BUSINESS MODEL AND TOKENOMICS **29**
- Flow of Funds 29
- Tokenomics 29
  - OBLIGATIONS OF TOKEN HOLDERS 30
  - BENEFITS FOR TOKEN HOLDERS 30
  - TOKENS DISTRIBUTION 31

ROADMAP 31

CRYP2NOVA

Archived copy · cryp2nova.com

INTRODUCTION

The oracle problem

Blockchain networks do not communicate with the outside world directly as they, by their very nature, have been designed to act independently of intermediary involvement and in an isolated manner. Blockchains form a consensus to execute agreements or commands using data that is stored on their ledger, all of which is validated using nodes across distributed networks. Allowing data from external sources to take part into the consensus process will not only delay the process but also make it non-deterministic, as it is impossible to guarantee uptime and responsiveness of third-party APIs.

Blockchain nodes must be kept in an isolated sandbox and as such cannot directly access traditional services or generate data in-house. Without this data, they cannot execute contracts such as deciding results of insurance policy payouts, determining financial settlements or knowing when to release or settle payments — in other words a specific blockchain is helpless in decisions that are dependent on data from outside of its own infrastructure. All nodes must also operate in the same, predictable and deterministic environment.

CRYP2NOVA

Current solutions to the Oracle Problem

During the early days of blockchain, there was a distinct lack of standards and market leaders. This “wild west” landscape saw almost every protocol create its own proprietary oracle solution. However, building an oracle is not a trivial task, and home-made oracles turned out to be vulnerable which resulted in many hacks. Some of these hacks were far from low profile, with millions of dollars being stolen or lost due to poor practice or human error.

Archived copy · cryp2nova.com

As exploits continued, protocol users started migrating towards professional solutions and common standards started to emerge. The most popular approach came in the form of a “two phase process”. In this model a contract submits a request for data to an Oracle Service first and an Oracle Service sends back a response with data. This simple and flexible solution was pioneered by Oraclize (now Provable) and Chainlink as the Request-Response Pattern. However, the main disadvantage of this approach is that the contract cannot access data immediately as it requires two separate transactions; this constrains usability as the client needs to wait until the data comes to a contract to see the result of an action. An even bigger problem is that fetching data is not atomic — meaning not in a single transaction. As a result, an Oracle has to synchronise multiple contracts, which is complex, slow and ultimately kills interoperability.

Currently, in an attempt to address the aforementioned issues, the most popular approach taken by current blockchains is to persist all data directly on-chain so that the information is available in the context of a single transaction. Protocols have also formed syndicates around the most popular oracles using common standardised configuration.

However, the high maintenance costs of this approach mean that storing data directly on-chain is extremely expensive. Ethereum in particular was never designed to be a storage network. As Oracles cannot “pause” the feeding data, they have to absorb the transaction costs of a network they are using. Ethereum by its design favours low-latency computation and heavily penalises any input/output operations such as data storage. These costs force protocols to pull together funding which means that they agree to use the same configuration and cannot tailor data to their needs. With such high operating costs, it is currently only possible to support a small number of tokens, meaning less popular tokens are excluded from oracle support.

Another market trend that creates a challenge for Oracles is the proliferation of alternative EVM-compatible chains targeting the DeFi market, such as Celo, Polygon, BNB, Avalanche, Solana, Fantom, Klaytn, Aurora, Optimism, Arbiitrum,

Archived copy · cryp2nova.com

Evmos or zkSync. All of these chains require reliable feeds. However, with the current Oracles design, data has to be pushed on each of them independently resulting in redundant costs and additional delays.

Despite facing scalability and maintenance cost obstacles the Oracle market cap is valued at around $14 billion as of January 2022 (according to Coingecko). With the expanding DeFi and crypto industry this number is only going to rise in the near future and so the need for diverse and reliable data is sure to grow.

Our vision

RedStone looks to mitigate the presented issues faced by blockchain projects by offering flexible and affordable oracle solutions. We have designed technical innovations that are expected to lower operating costs which lead to better scalability and lower latency.

Affordable Storage

RedStone leverages a new generation of blockchain storage thanks to Arweave, which is designed to keep large amounts of data at a fraction of the cost of chains like Ethereum. As of December 2021, storing 1GB on Arweave would cost $35, while on Ethereum: 1.7M$. Low operating costs allow us to process more data with a higher update frequency.

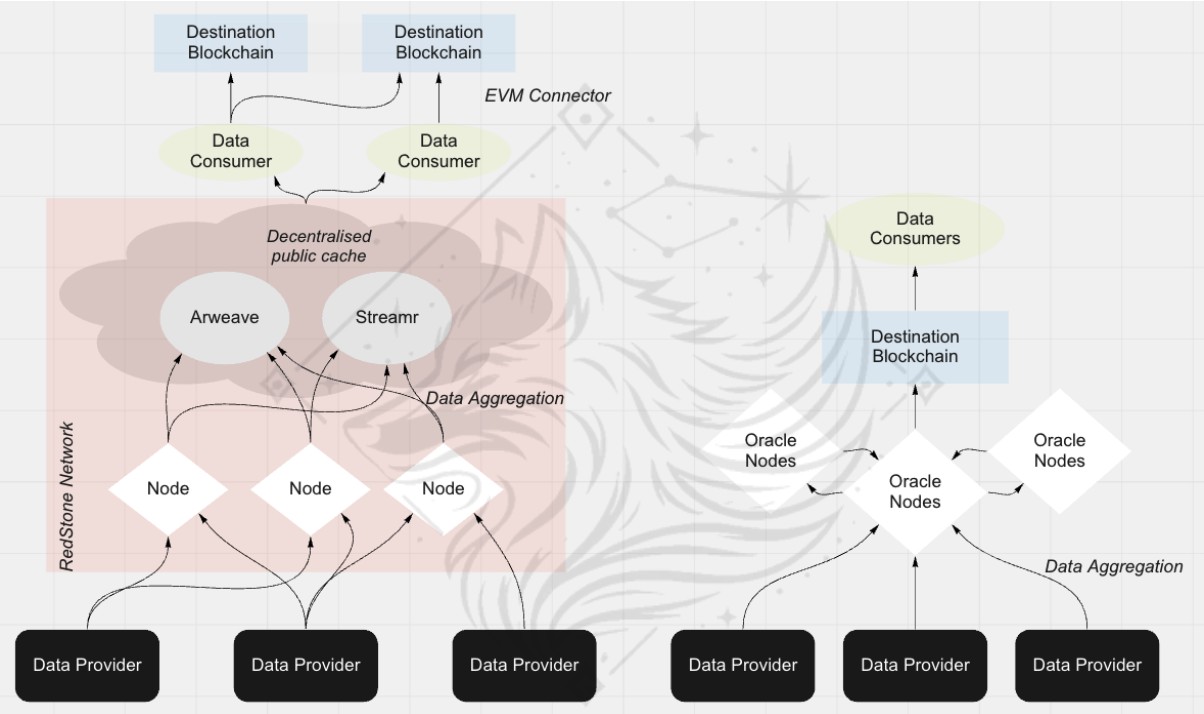
On demand fetching

At RedStone we derive our innovation from an unique method of delivering feeds to various chains. We do not simply push information to a blockchain every couple of minutes as the majority of Oracles do. Instead, our idea is to keep data on a multi-chain intermediary layer which is built on top of the data distribution chains such as Streamr and permanently persisted on Arweave. Afterwards, the data is available to other blockchain projects, which could fetch it on demand. In other

Archived copy · cryp2nova.com

words, the blockchain projects cover the transaction costs only when the data is utilised, making the solution extremely cost-efficient.

The signature ensuring integrity is attached to the transaction which requires the data, similar to a meta transaction. The chart below shows simplified architecture of RedStone data ecosystem (left) and standard Oracle architecture for reference (right).



Flexible Data Streams

The capacity to process more data creates an opportunity for multiple data providers to enter the blockchain ecosystem. Each of them may apply different aggregation rules offering services tailored to the needs of different DeFi protocols. A concrete example of this is when a lending pool requires time-averaged data not to liquidate users for a price flash-movement. On the other hand, a synthetic DEX or a margin-trading protocol will have totally different needs, wanting to receive the most up to date information. The RedStone network can satisfy both needs by configuring separate data feeds manifests.

RedStone can source various types of data — we're not limited to price data for assets only. Currently, there are underserved areas in the Web3 space, such as

Archived copy · cryp2nova.com

data related to NFT, gaming, insurance, sports statistics, green bonds or even credit scoring. Our focus is to fill these niches with customisable and secure data feeds. Technically, it is already possible to add other sources to our Oracles.

The process of adding new data-feeds to the platform starts with reviewing suggestions of use cases from developers and the community to decide what type of data would be the most useful and could enable innovative protocols. Once there is enough demand for the data a dedicated adapter is implemented and deployed to the network.

At the same time, becoming a data provider is simple and fully automated due to its execution via RedStone smart contracts. Consequently, the whole process is transparent and truly decentralised.

Data Integrity

Allowing multiple data providers generates a need to curate them and select the most reliable ones. Therefore, data providers join the network by locking RedStone tokens as collateral. The RedStone tokens are fully refundable as long as the data providers will keep operating and providing high quality data.

RedStone also keeps a full record of the provider's activity that will be permanently available on the Arweave chain so anyone could challenge a data feed and receive a part of that collateral in case the data is misrepresented.

The logic that governs challenge processing is powered by the Dispute Resolution Protocol (internally called ARgue). The protocol is based on Schelling point consensus, which is described in detail in one of the next chapters.

Cross-Chain Oracle Solutions

RedStone's data is cryptographically signed by providers and could be verified on any chain that supports basic cryptographic primitives. As a result, we abstract storage from usage; although the data at RedStone is hosted on Public Cache

Archived copy · cryp2nova.com

powered by Streamr, among others, and backed on the Arweave chain, it could be used with any other blockchain.

When it comes to blockchains, our first choices are EVM (Ethereum Virtual Machine) compatible chains like Ethereum, Celo, Polygon and Avalanche. As of December 2021, over 75% of the whole Total Value Locked (TVL) is based on EVM chains (according to DeFi Lama). Our team has invested a lot of resources into maximising the efficiency of data transmission and has reached a point where importing and verifying a single data feed could be cheaper than reading from an external contract's storage (as it's done by the most popular other oracle providers).

In the future we plan to offer solutions across all other, non-EVM, blockchains as part of our deliverables — becoming fully chain agnostic Oracles of the new generation.

Possible applications

Lowering the operating cost enables data feeds for less popular tokens which allows them to be used as collateral by various lending and derivatives protocols. It's surprising to realise that out of about ~20,000 of tokens aggregated by popular explorers, less than 100 is fully supported by on-chain oracle providers.

Being EVM-compatible makes the solution available to less popular chains which support the Ethereum execution model. Currently, the traction and trading volumes makes it less attractive for market incumbents to provide services there. However, our flexible & modular infrastructure makes it possible to serve these environments.

Currently, the data feeds mostly target financial protocols which are willing to pay the highest fees. However, solutions like environmental monitoring protocols or insurance services by definition require access to larger amounts of less structured

Archived copy · cryp2nova.com

and more diversified data. We may expect more demand from that side when the traction will follow the market share observed in the real world economies.

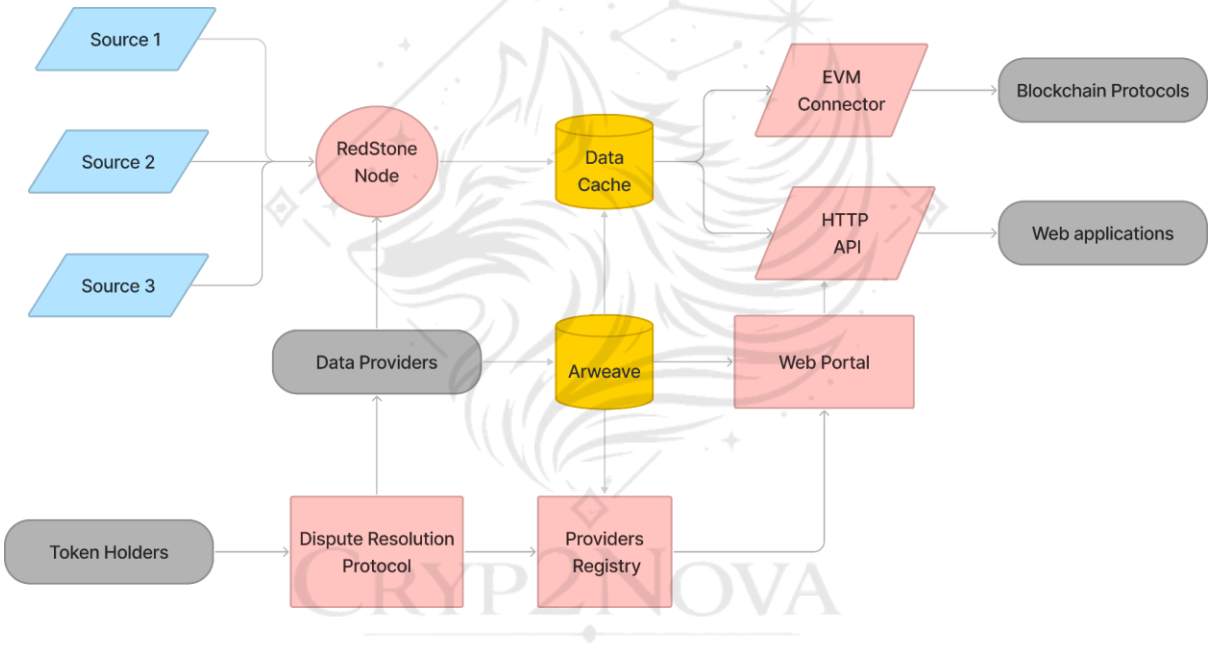
Another interesting use case is the provision of the NFT-related data. It's much harder to accurately price this type of asset due to the subjective nature of criteria applied to art-related content. Current pricing models try to aggregate multiple data sources related to social sentiment, user's ranking and view counts. Due to the ubiquitous nature of information related to NFT's our flexible and customisable data feeds seem to be a perfect match to tackle this problem.

Archived copy · cryp2nova.com

SYSTEM ARCHITECTURE

Top level design

The RedStone ecosystem enables secure and reliable flow of data from external sources (such as off-chain APIs) to consumers (blockchain protocols or web integrators). The data feeds are curated, filtered, aggregated, permanently persisted and distributed. The process is facilitated by various components and realised by distributed actors incentivized to perform their duties by smart contract rewards and punishments. The below chart shows a simplified flow of data and actors that participate in the RedStone ecosystem.



Data sources (blue)

Provide any kind of data or information that could be useful for end users

- Off-chain sources - websites, platforms and services providing data via publicly available http endpoints
- On-chain sources - blockchain contracts and protocols that could be queried directly

Archived copy · cryp2nova.com

- Private sources - password protected resources that are available only to authorised nodes

Users (grey)

- Data providers - need to register by locking RedStone tokens as collateral and may fetch, process and serve data afterwards
- RedStone Token holders - protect data integrity by participating in disputes around data quality
- Blockchain protocols - consume data coming from the EVM-connector by validating the origin and integrity on-chain
- Web applications - may fetch and display data through integration via http API, can easily browse and render historical data

RedStone modules (red)

- RedStone node - fetches data from external sources via api, aggregates and attest information with a cryptographic key and broadcast the results
- EVM connector - it provides the data to on-chain protocols. Initially, it targets Ethereum compatible chains but it could be easily extended to support additional environments. It verifies signed data on-chain.
- HTTP api - a web based endpoint that serves information via REST api. It can power websites, node-js scripts and social media bots. There is also a dedicated Javascript wrapper (library) that will abstract request formatting and error handling to make the service easier to integrate
- Providers registry - an Arweave smart contract (based on SmartWeave protocol) that manages the provider's manifest (SLA of data feed) and allows the checking of historical performance (showing any data inconsistency and downtime)
- Dispute resolution protocol - a set of contracts that allow the challenge of any existing pricing feeds and manage the dispute process which enable token holders to vote for the verdicts
- Web portal - a web application that acts as an interface to browse data, check providers' statistics and see historical feeds with an option to raise a dispute and participate in the voting process

Archived copy · cryp2nova.com

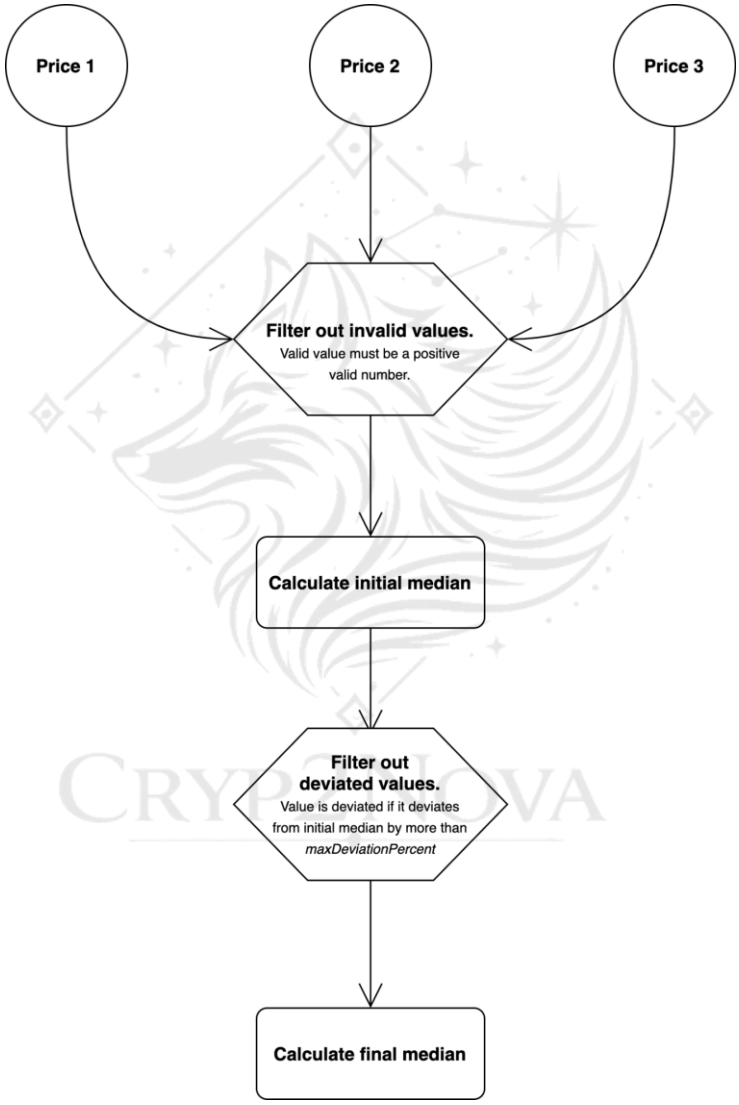
Data delivery layer (yellow)

- Broadcasting layer - a module that temporarily keeps data and makes it available to interested parties. Multiple services might implement it on various infrastructure to increase availability. It should achieve minimum latency and scale to handle large volumes of requests. The broadcasted data feeds are further persisted in the storage module (Arweave).
- Storage (Arweave) - a permanent storage that persist all the feeds and make it available for queries, which is helpful for efficient dispute resolution process

Archived copy · cryp2nova.com

Data fetching

The RedStone ecosystem could be connected to any number of external data sources regardless if they are natively present on a chain or expose web-based API. The data is periodically queried, filtered, and aggregated as shown in the following diagram:



The data is fetched according to a contract that is agreed upon between a data provider and data consumers. The contract specifies the list of trusted sources, the probing interval and the data aggregation algorithm. The full list of parameters is shown in the table below:

Archived copy · cryp2nova.com

interval	required	Number	Data fetching interval in milliseconds
priceAggregator	required	String	Aggregator id. It specifies the strategy for data aggregation (e.g. median).
defaultSource	optional	String[]	Array of fetcher names that will be used by default for tokens that have no specified sources
sourceTimeout	required	Number	Default timeout in milliseconds for sources
maxPriceDeviationPercent	required	Number	Default maximum price deviation percent for tokens. It may also be set for each token separately
tokens	required	Object	Object with tokens in the following format: { "TOKEN_SYMBOL": { "source": ["source-name-1", "source-name-2", ...], "maxPriceDeviationPercent": 25 }, ... }. Note that source and maxPriceDeviationPercent params per token are optional. This is also a correct tokens configuration: { "TOKEN_SYMBOL_1": {}, "TOKEN_SYMBOL_2": {} }

CRYP2NOVA

Archived copy · cryp2nova.com

Data management

One of the core values of RedStone protocol is maintaining the full transparency and accountability around the data delivery process. This includes the ability for every user to access current definitions of data providers in a way that cannot be blocked or censored by any party. The users should also be able to monitor all the changes in the configuration and be assured that modifications are introduced in a transparent and predictable manner.

Processing large amounts of data, such as configuration of a complex data fetching mechanism, is a problematic task for many popular smart contract frameworks. These platforms focus on increasing the throughput and treat data as a side-product that slows down processing. Data is digested, pruned and pushed out to side chains. Nonetheless, the bloated global state is still the biggest challenge of modern blockchains.

We decided to implement providers registry on top of Arweave blockchain and keep the content and data access rules there.

SmartWeave Protocol

SmartWeave is a data-centred protocol built on top of the Arweave chain that allows the implementation of smart contracts that could govern access to data. The key difference between SmartWeave and other similar solutions is that the contracts can be evaluated on-demand and "lazily".

The SmartWeave protocol consists of a few key concepts:

1. Contract creation
2. Contract interaction
3. Lazy-evaluation of the contract state

Archived copy · cryp2nova.com

Contract Interactions

Contract interactions allow modifying of the contract state. Each interaction MUST be deployed as a separate Arweave transaction:

Transaction field	Value
tag['App-Name']	SmartWeaveAction
tag['App-Version']	the current version of the SDK, e.g. version from package.json
tag['Contract']	Transaction id of the contract
tag['Input']	JSON stringified representation of the input

Lazy Evaluation

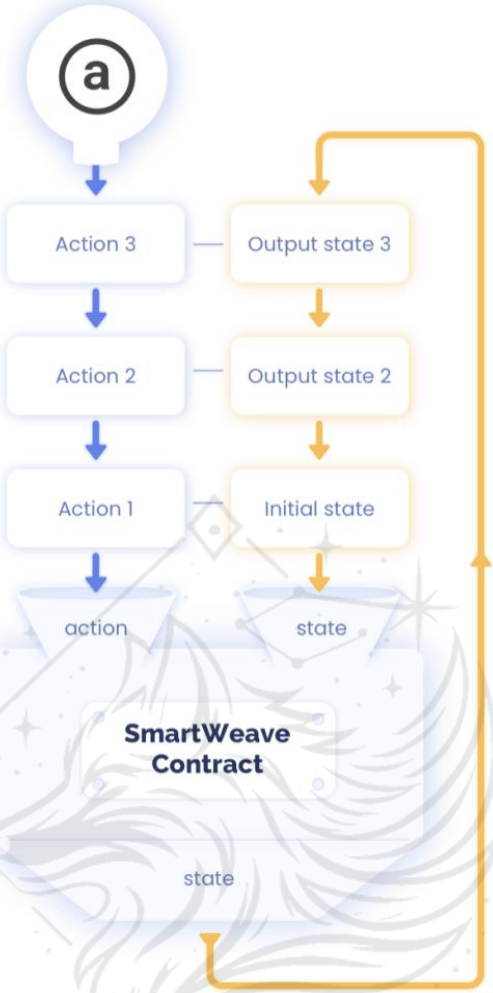
In its current form, the SmartWeave protocol assumes that smart contract code is a Javascript function, defined as:

contractFunction = (state, action) => state;

This function takes a state and an action as input arguments - and produces a new state as a result. In functional programming terms, it acts as a *fold* function - the state of the contract is derived from:

1. An initial state
2. A contract function
3. An ordered list of actions

Archived copy · cryp2nova.com

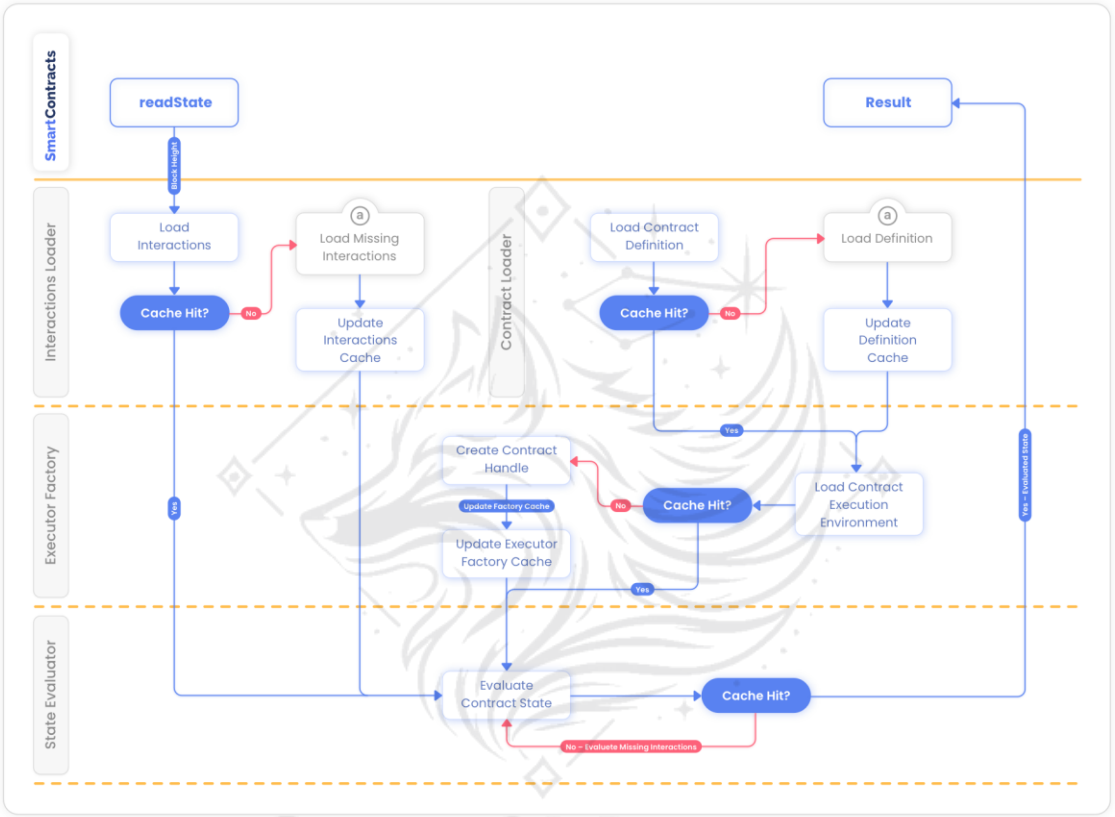


CRYP2NOVA

Archived copy · cryp2nova.com

Contract State evaluation

In order to perform contract state evaluation (at a given block height), The Smartweave SDK performs certain operations. The diagram above and the description assume the most basic client version with the computation executed in memory.



CRYP2NOVA

1. Users who are interacting with the contract call the “readState” method.
2. Interactions Loader and Contract Definition Loader modules are then called in parallel - to load all the data required for state evaluation. Both Interactions Loader and Contract Definition Loader firstly check their corresponding cache to see whether data is already loaded - and only load from Arweave what is required..

Archived copy · cryp2nova.com

3. With interactions and contract definition loaded - Executor Factory creates a handle to the SmartWeave contract main function (or loads it from its own cache)
4. With all the interactions and a contract handle - the State Evaluator evaluates the state from the lastly cached value - and returns the result to User.

Data delivery

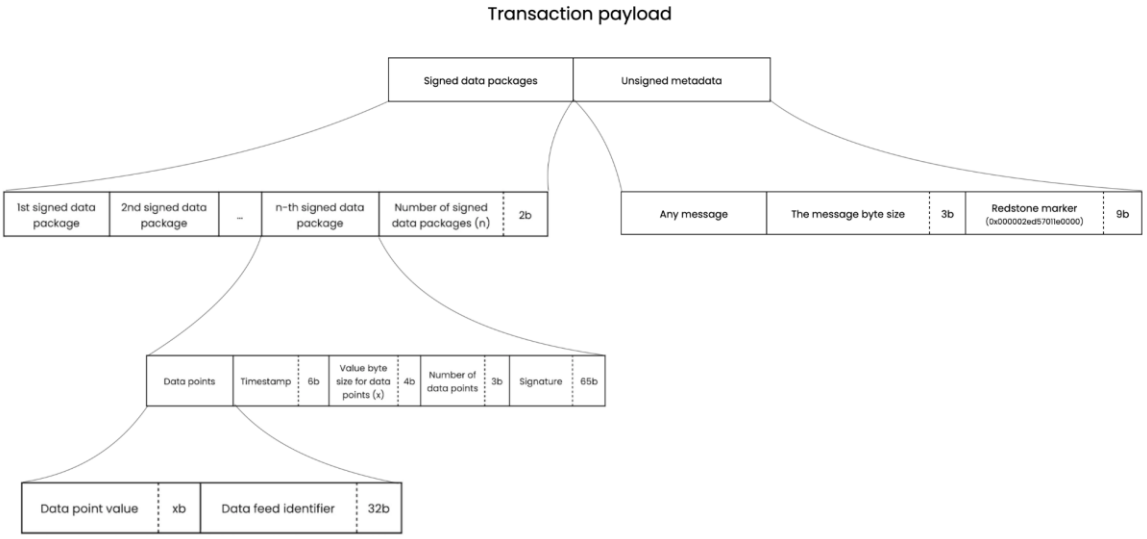
RedStone EVM-connector implements an alternative design of providing data to smart contracts. Instead of constantly hosting data on the EVM storage, the information is brought on-chain only when needed (on-demand fetching). Until that moment, the data remains available in the broadcasting layer and on the Arweave network where data providers are incentivised to keep information accurate and up to date. Data is transferred to EVM via a mechanism based on a meta-transaction pattern and the information integrity is verified on-chain through signature checking.

At a top level, transferring data to an EVM environment requires packing an extra payload to a user's transaction and processing the message on-chain.

Data packing (off-chain data encoding)

Data is packed into a message according to the following structure:

Archived copy · cryp2nova.com



The package is appended to the original transaction message, signed and submitted to the network.

The data is propagated by a network of light-weight nodes called broadcasters whose only responsibility is to pass along information. Moreover, the information is transmitted to a fully decentralised pub/sub network called Streamr, which consists of thousands of independent nodes.

As a backup the information can always be fetched directly from the Arweave blockchain where it is permanently stored.

All the steps are executed automatically by a Javascript library called ContractWrapper which makes the process transparent to the end-user and easy to integrate. The library is implemented on top of the ethers-js smart contract proxy, which is the most popular library to interact with the smart contracts environment.

Archived copy · cryp2nova.com

Data unpacking (on-chain data verification)

To ensure the maximum level of security and reduce the possibility of corrupting the data (man-in-the-middle attack) we authenticate the information directly on the target chain. The process consists of the steps listed below:

1. The appended data package is extracted from the msg.data
2. The data signature is verified by checking if the signer is one of the approved providers
3. The timestamp is also verified checking if the information is not obsolete
4. The value that matches a given symbol is extracted from the data package

As the process is executed in the on-chain environment, we optimised the execution using a low-level assembly code to reduce gas consumption to the absolute minimum.

To increase reliability the on-chain aggregation could process data from multiple providers reducing the risk of relying on a single node which could be corrupted.

Data integrity

Decentralised data feeds have many advantages over centralised competitors. They are more secure, lack a central point of failure and are more censorship-resilient because of diversified governance. However, it's not trivial to manage data integrity and accuracy without a central point of control.

RedStone's proprietary Dispute Resolution Protocol, internally called ARgue, enables high data integrity preserving the benefits of keeping a fully decentralised and diverse set of data providers. Users are incentivised to report problems with data quality by opening a dispute which is resolved by a panel of token holders who vote by locking their RedStone tokens. The consensus is based on the majority rule, where voters are rewarded for voting alongside others and penalised for being outliers.

Archived copy · cryp2nova.com

The majority rule is the most common strategy for reaching coordination in a decentralised environment. It was researched as a part of game theory studies in the 60's, and the current implementations are based on the work of Thomas Schelling and his idea of the focal point described in the book "Strategy and Conflict". One of the adaptations of this idea was drafted by Vitalik Buterin in 2014 as a Schelling Coin concept. It was also proposed as an architecture for the Ethereum Price Feeds: "*For financial contracts for difference, it may actually be possible to decentralize the data feed via a protocol called SchellingCoin*" (from the Ethereum whitepaper). However, the network congestion, high gas price and the extreme storage cost of the Ethereum Virtual Machine rendered the solution impractical at the current level of technology.

The progress in scaling smart contract execution efficiency makes it possible to implement the dispute resolution logic on-chain. The challenging part is setting the system parameters to keep the user friction as low as possible and reduce the voting requirement while maintaining the high quality of decisions. Hopefully, the recent simulations done by the RedStone team prove that even a small percentage of expert jurors could produce high-quality results when the incentives are properly set up (see the simulations of TCR-based decision model presented during EdCon Paris 2019).

CRYP2NOVA

Dispute process

The dispute is a multi-stage process that involves two main counter-parties: a challenger of a certain data entry, who initiates the dispute, a provider, who originated the challenged data entry and a jury panel that makes the final decision. The actors are incentivised to participate with potential token rewards. All of the stages of the process are described below:

1. Pre dispute / Provision of a RedStone Node

A data provider shall deposit collateral as a signal to the users that there is a strong incentive to maintain the quality of data. The higher the collateral means a higher

Archived copy · cryp2nova.com

potential reward for the challenger, but it also translates into a higher fee to open a dispute.

2. Opening dispute

Anyone could open a dispute providing he can lock an amount of tokens that is proportional to the provider's deposit. There is also a minimum cap introduced to avoid spamming. The challenger deposit is locked until the dispute is settled.

3. Voting

Jurors vote by locking their tokens either to support or reject the dispute. The maximum number of tokens that could be used is limited by an additional parameter called "voting capacity". Voting lasts for an adjustable pre-defined period, which is typically a week.

4. Verdict

If a required quorum (defined as a percentage of all eligible voters) is reached before the deadline, the voting is stopped. The choice with the highest amount of token locked is called a verdict. A party that is not satisfied with the verdict may submit an appeal during the appeal window (typically 3 days). If there is no appeal the verdict is considered final, and the settlement is executed.

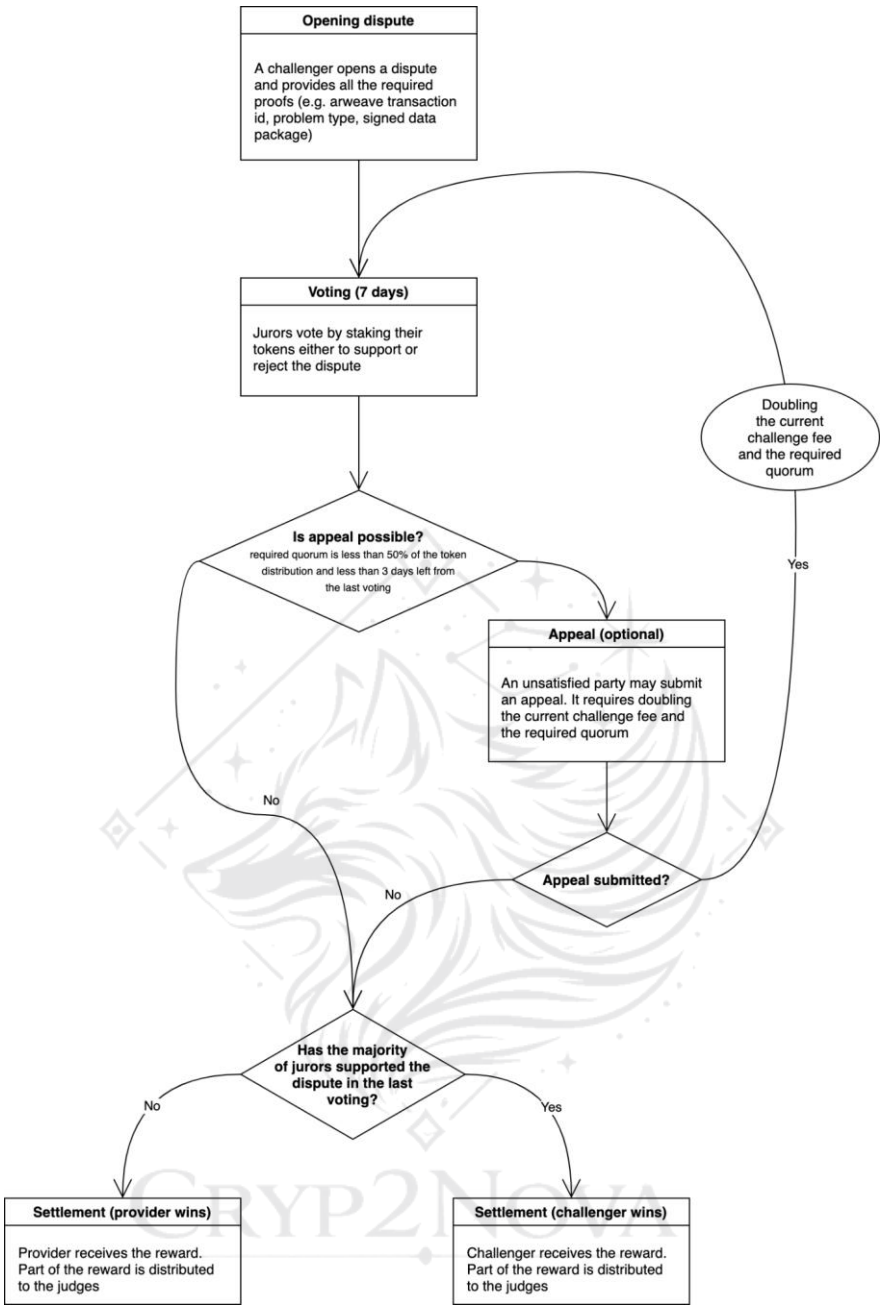
5. Appeal

An appeal submission requires doubling the current challenge fee. It will restart the voting and double the required quorum. If the quorum is already above 50% it is impossible to submit another appeal and the current verdict is considered final.

6. Settlement

After the verdict is final the winning party receives a reward which is taken either from the challenger's or provider's deposit. Part of this reward is distributed to the judges. There is also an internal redistribution of tokens among the judges. Those supporting the majority choice get part of the tokens locked on the losing side. The jurors who were on the winning side also increased their voting capacity. The results are hosted on the Arweave blockchain and could be used as a base for reputation score for providers, watchers and jurors.

Archived copy · cryp2nova.com



Archived copy · cryp2nova.com

Node management

Data providers may effectively start offering their service by running a node. A node is an application that automatically runs in the background and is responsible for data fetching, processing and broadcasting.

There are two sub-types of Data Providers. Since operating RedStone Nodes requires high levels of technical knowledge, mainly companies specialized in server administration will be acting as operators of the RedStone Nodes (Node Operator(s)). The second type of Data Providers can review and verify the specifics of an SLA configuration and data aggregation methodology but lacks the technical skill to administrate RedStone Nodes manually (Node Curator(s)). This group may include data experts such as analysts, scientists or data brokers. Node Curators are able to provide Collateral to existing RedStone Nodes of Node Operators. Both Node Operators and Node Curators will need to register on the RedStone Network by providing a certain minimum amount of RedStone Token. By operating the RedStone Nodes, data is being fetched based on the terms as set out in the SLA. During that process, RedStone Tokens are locked and remain blocked on the Data Providers' own wallets. Each Node Operator needs to publish an SLA where the scope of data, the source of information and the frequency of updates are defined (=manifest) as well as terms of services such as price and warranties. Both Node Operators and Node Curators are responsible for any legal and regulatory implications its SLA may have. Node Operators are also responsible for data cleaning, verification and data aggregation according to the specified methodology (e.g. median value or time-weighted average).

Node deployment

For the convenience of node operators the application is packaged as a docker container and could be easily deployed either on local infrastructure or on any of the most popular cloud computing platforms. RedStone maintains both the code of a node application and the necessary documentation including the step-by-step guideline of the deployment process.

Archived copy · cryp2nova.com

In order to operate the node, a data provider needs to lock a certain amount of RedStone tokens (as collateral) to get access to the infrastructure. There is also a need to register the service in the network so the potential users will be able to discover it and monitor the performance. The registration requires the specification of a few obligatory parameters:

id	unique id of a given data provider
name	human-readable name of the data provider
description	description of the provided data
url	url to the data provider's website
urlImg	url to an img with provider's logo

The registration process is fully decentralised and managed by the smart-contract layer. We list all of the provider's registry features below:

CRYP2NOVA

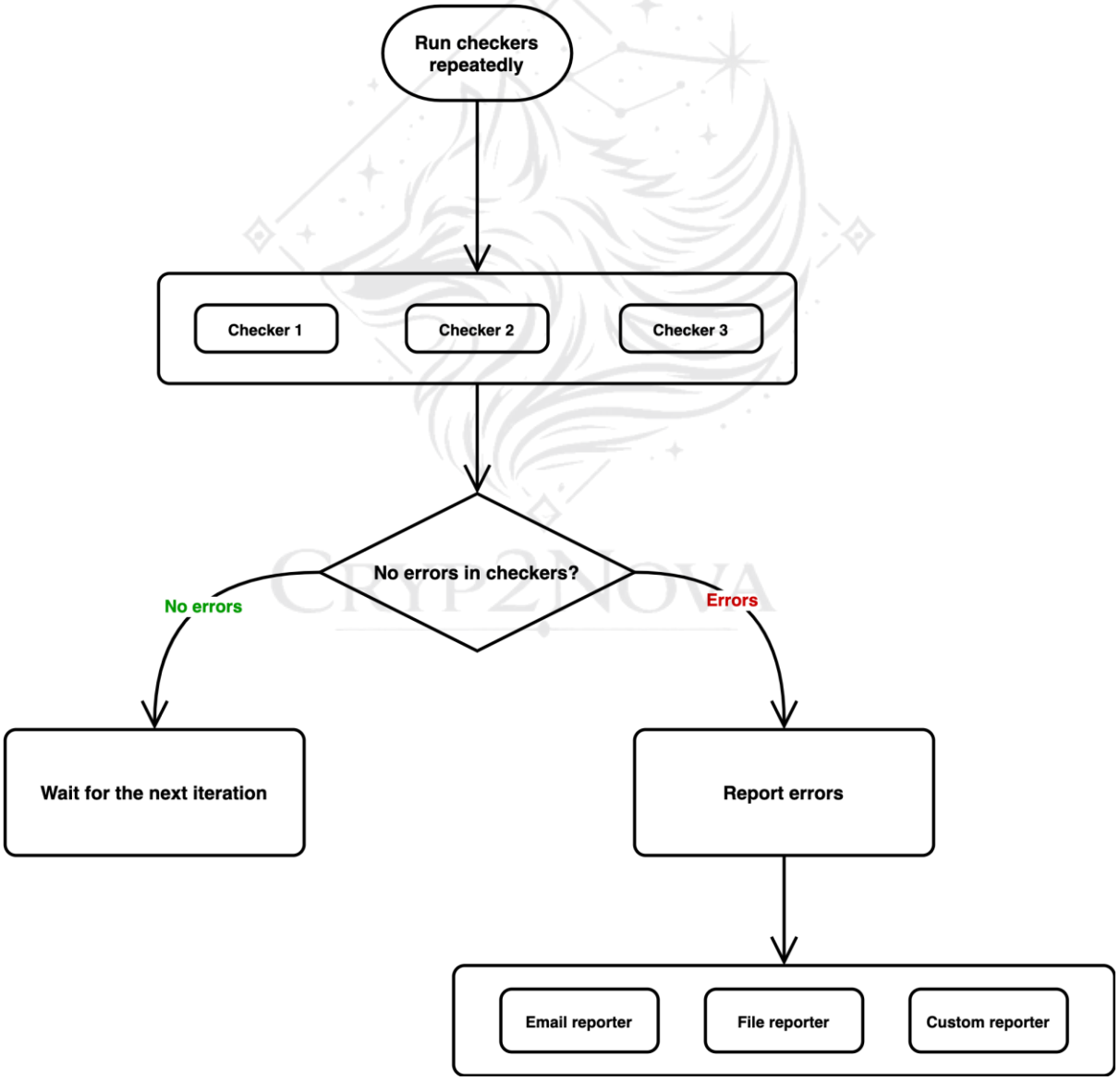
1. Tracks profile data of all the currently available RedStone data providers. Keeps current and historical manifests data for all the registered nodes.
2. Allows updating of a given provider's manifest.
3. Allows updating of a given provider's meta-data.
4. Allows adding and removing providers.
5. Allows configuring provider's admin accounts (i.e. accounts that are allowed to change provider's manifest and meta-data).

Archived copy · cryp2nova.com

Nodes monitoring

It is of the utmost importance for oracles to provide high quality data without interruption. Therefore, the node operators need to monitor their nodes and immediately fix any potential problems.

That is why we have implemented a special app dedicated to node monitoring. This app is executed in a loop and automatically verifies if a selected node regularly publishes new data to the RedStone cache layer and the Arweave blockchain. In each loop iteration multiple checkers are executed. And if any of these checkers fail then the error is reported.



Archived copy · cryp2nova.com

The architecture is based on a Checker - a module responsible for data integrity checking. It can check the number of saved data points during the last 5 minutes or the timestamp of the latest saved data point.

Implemented checkers:

- ArPriceReturnedRedstoneRapid
- ArPriceReturned
- ArweaveTimestampDelay
- HistoricalPricesReturned
- StockPricesReturnedRedstoneStocks
- TimestampIsCloseToNowRedstoneRapid
- TimestampIsCloseToNow

Reporter - a module responsible for error/warning reporting. For example, it can notify a node operator via email, SMS or Discord. It can also save a notification to a text file. Currently we send email notifications to our developer team and save logs in AWS Cloudwatch.

CRYP2NOVA

Archived copy · cryp2nova.com

BUSINESS MODEL AND TOKENOMICS

Flow of Funds

RedStone as an association doesn't collect any fees and commissions and neither does it mediate in the flow of funds between node operators and their customers. Instead, the funds necessary for the ongoing maintenance and potential further development of network infrastructure come from selling the RedStone tokens. In return, entities which purchased the tokens have got access to the infrastructure and could develop their business there by charging data access fees. The fees are denominated in the stable coins or the most liquid tokens, but not in Redstone token. The RedStone tokens will be sold based on the capital needs to reach the planned milestones and ensure proper support for the users.

This strategic approach to financial success contributes towards the value of the RedStone network and the growth of any ecosystem partners. The more people involved in the ecosystem as node operators, data consumers, judges and other participants, the bigger the value of the whole system.

The majority of funds collected in the funding rounds will be spent on product development while the remaining budget will be used for market & community growth plus standard administration and operating costs.

Tokenomics

Tokens are proven to be a very useful tool for increasing coordination in distributed systems and aligning the incentives of various actors. RedStone token facilitates a data-sharing ecosystem incentivising participants to produce, publish and validate data both continuously and diligently. Below we describe two crucial utility use cases for the token.

Archived copy · cryp2nova.com

OBLIGATIONS OF TOKEN HOLDERS

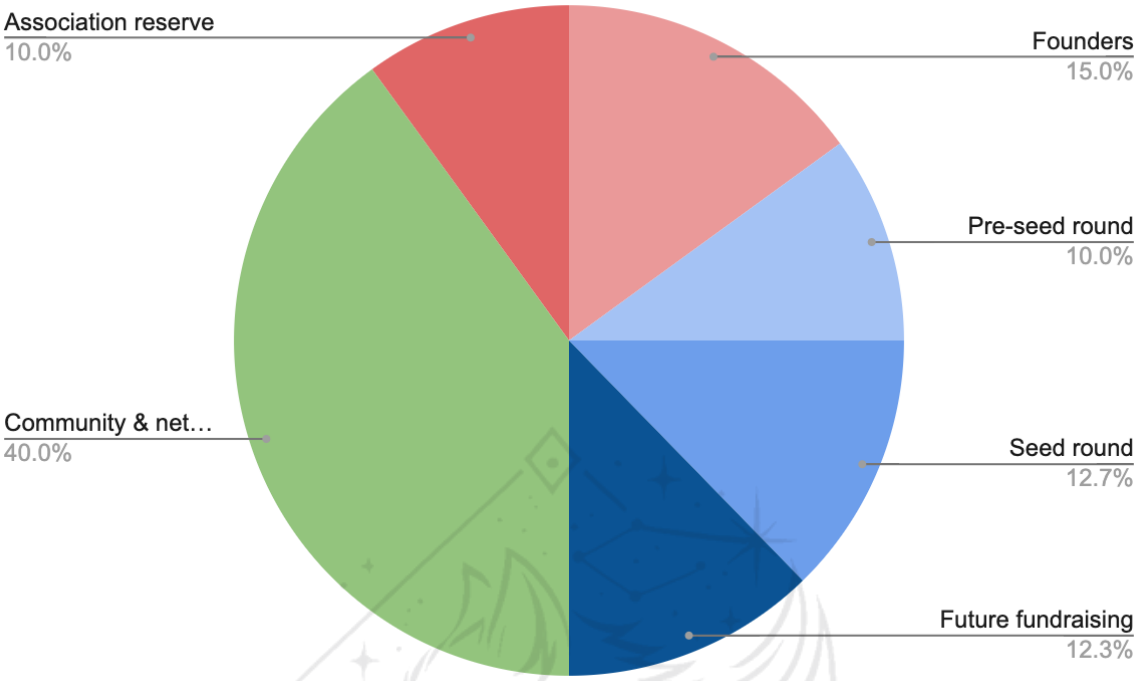
Every data provider (node operator) needs to publish a manifest (acting as a Service Level Agreement) describing the scope of data being served, the source of information, and the frequency of updates. The provider needs to deposit RedStone tokens as collateral in order to commit to the quality of service. To assure users that any claim resulting from a dispute process could be fully covered, providers need to put their collateral in a contract that will lock the tokens for a period of time. The Redstone Association won't keep any custody of the tokens during that period, as the funds will be managed by a trustless smart contract that will automatically return tokens if the terms of the SLA contracts are met. If a provider breaches the terms of service there will be a penalty applied which is also denominated in tokens. The collateral also plays an important signalling role allowing users to select the most reliable provider who offers the largest commitment to provide accurate data.

BENEFITS FOR TOKEN HOLDERS

Any data provider that fulfils the need to publish a manifest and deposit collateral is eligible to collect fees from data feeds users. The fees can be expressed as a fixed rate subscription or charged separately per every access. The protocol doesn't impose any restriction on the price model and means of payment agreed between data providers and users. There is also no commission or other administrative fee collected by the protocol itself. The fees could be expressed in the most popular and liquid blockchain tokens like ETH, AVAX, CELO, MATIC, etc.. or stable coins like USDC, USDT or DAI, but not in the RedStone token itself.

Archived copy · cryp2nova.com

TOKENS DISTRIBUTION



Part of the tokens reserved for the "Community and Network growth" will be distributed to early node operators in form of grants and stipends. This will encourage them to provide the service before there is enough demand to make the operation profitable. This activity is going to be temporary and applied only at the network bootstrapping level. Redstone Association will decide such additional distribution of RedStone Token on a case-by-case basis at its full discretion. There is no intention to directly reward Data Providers with RedStone Tokens as the grants will be awarded per data feed type. This is a common approach to solve network bootstrapping challenges (colloquially called "chicken and egg dilemma") where lack of supply discourages users from building up the demand side of the market. The other part of this allocation will incentivise various actions such as writing tutorials, improving documentation, implementing integrations with new products and engaging in marketing activities.

The "Association Reserve" groups all the potential funding targets that are dependent on the unforeseen challenges for the project and market conditions. It may involve funding development of crucial upgrades, making strategic

Archived copy · cryp2nova.com

partnerships or investing into any form of activity that will bring the most benefit to the network.

ROADMAP

Timeline	Scope
Q1 / 2022	1. Stabilising the data-processing layer, Moving from alpha to beta by increasing the test code coverage, automating regression tests and implementing a set of proof-of-concept contracts 2. Preparing a thorough documentation accompanied by a set of tutorials and guides 3. Performing audits of data delivery contracts 4. Production deployment of nodes with a diligent monitoring tools
Q2 / 2022	1. Research and early implementation of decentralised data aggregation logic across multiple data nodes 2. A web interface to define data sources and simple transformation enabling custom fetchers 3. Implementation and testing of smart-contract transaction gateway and sequencer service
Q3 / 2022	1. Implementation of the on-chain verification of data from multiple providers 2. Preparations for the testnet and token lockups infrastructure 3. Implementing a set of tools for efficient node deployment and monitoring
Q4 / 2022	1. Researching trustless data validation technologies such

Archived copy · cryp2nova.com

	as TLS notary and secure enclaves based environments 2. Implementation of dispute resolution protocol 3. Deployment of the test network
Q1 / 2023	1. Monitoring the test network activity with telemetry and implementing necessary fixes and upgrades 2. Implementing tools for active monitoring of on-chain activities collecting statistics about data transmission and usage
Q2 / 2023	1. Expanding oracles to most popular non-evm chains 2. Launch of the mainnet with production-grade oracle services
Q3 / 2023	1. Researching privacy-preserving methods for data aggregation and tools for integrations with access protected APIs 2. Researching and designing more efficient way of off-chain data aggregation 3. Extensive support for other project with custom data-fetchers and on-chain adapters
Q4 / 2023	1. Implementing of more efficient off-chain aggregation logic based on lightweight consensus model and threshold signatures 2. Improving tools and documentation for integrating with Oracles and creating custom data feeds

Archived copy · cryp2nova.com