THORChain Whitepaper

THORChain

A lightning fast decentralised exchange protocol

devs@thorchain.org

V0.1 July 2018

Abstract

THORChain is a highly optimised multi-chain using pBFT consensus to achieve sub-second block finality. Tokens are traded on single chains, known as tokenChains with discrete address spaces. Multi-set sharding is proposed to allow byzantine resistant scaling. The native protocol facilitates on-chain trading and order matching at the protocol level, supporting both limit and market orders. Continuous liquidity pools ensure liquidity is always available for any token pair, and double as the source of trustless on-chain price feeds, a cornerstone of the protocol. Fee incentives are designed to continually attract on-chain liquidity. On-chain token generation is built-in for both fixed and variable supply tokens, with the potential for auditable collateralized stablecoins. Two-way pegs with UTXO, account and contract-based cryptocurrencies allow most existing cryptocurrencies to seamlessly move on and off THORChain. Transparent developer incentivisation strategies allow wallet and exchange developers to be funded at the protocol level which will encourage a vibrant developer community. On-chain governance and smart updates with enforced voting and quadratic voting ensures THORChain is an adaptive and iterative platform that grows with its needs without contentious hard forks. THORChain is built to be compatible with the Flash Network; a layer 2 payment channel network.

CRYP2NOVA

Document Set

The following whitepapers should be read in conjunction:

THORChain

A lightning fast decentralised exchange protocol.

Archived copy · cryp2nova.com

THORChain Whitepaper

Bifröst Protocol

Secure and fast cross-chain bridges for THORChain.

Flash Network

A layer 2 Network for instant asset exchange on THORChain.

Yggdrasil Protocol

Dynamic multi-set sharding for THORChain.

Æsir Protocol

A self-amending forkless consensus algorithm for THORChain.

CRYP2NOVA

Overview

Introduction

Centralized and Decentralized Exchanges

THORChain and ASGARDEX

Archived copy · cryp2nova.com

THORChain Whitepaper

THORChain Architecture

Overview

TokenChains

MerkleChain

Consensus

Validator Set

Multiple Validator Sets

On-chain Governance

Overview

Key Aspects

Tokens

Rune Characteristics

CRYP2NOVA

Archived copy · cryp2nova.com

THORChain Whitepaper

Block Rewards and Emission

Fees

TokenChain

Rune Generation

THORChain Bridges

Continuous liquidity pools

Solving Liquidity

CLPs

CLP Transactions

Liquidity Fees

Trustless On-chain Price Feeds

On-chain Trading

Account Types

Transaction Types

CRYP2NOVA

Archived copy · cryp2nova.com

Limit and Market Orders

User Experience

Wallets

Trading

Exchanges

Exchange and Wallet Funding

Cross-token Trading

On-chain Initial Exchange Offerings

Validator Staking

Advanced Transaction Features

THORChain Name Service (TNS)

Account Recovery

Multi-sig Accounts

Transaction Receipts

CRYP2NOVA

Archived copy · cryp2nova.com

Transaction Requests

Recurring Transactions

Escrow Accounts

Automatic Disbursements

Combined Payment Features

Other Features

Collectibles

Messaging Protocol

Smart contracts

Additional On-Chain Commands

StableCoins

Token Baskets & Indexes

Anonymity

FIX 4.4 Protocol

Conclusion

CRYP2NOVA

Archived copy · cryp2nova.com

THORChain Whitepaper

References

Definitions

Item	Definition
ASGARDEX	A Decentralised Exchange built on THORChain
Bifröst Protocol	Bridges built to other networks (two-way-pegs).
CLP	

CRYP2NOVA

Archived copy · cryp2nova.com

Continuous Liquidity Pool

DEX

Decentralised Exchange

ECDSA

Elliptical Curve Digital Signature Algorithm

ERC-20, NEP-5, QRC-20

Ethereum, NEO, QTUM tokens

Flash Network

Layer 2 Payment Channel Network

FIX4.4 Protocol

Archived copy · cryp2nova.com

THORChain Whitepaper

Foreign Information Exchange 4.4 Protocol for Trading Desks

HFT

High Frequency Trading

Mjölnir

Liquidity Nodes.

RNG

Random Number Generation

RUNE

THORChain Native token, Rune token, Rune, RUNE

pBFT

Archived copy · cryp2nova.com

THORChain Whitepaper

Practical Byzantine Fault Tolerance

Runes/Tokens

Tokens on THORChain

TokenChain/RuneChain

Chain for each Token

TPS

Transactions Per Second

UTXO

Unspent Transaction Output

VRF

Archived copy · cryp2nova.com

THORChain Whitepaper

Verifiably Random Function

Introduction

Centralized and Decentralized Exchanges

Exchanges
. One of the most important advents of the cryptocurrency economy is the crypto exchange. While digital assets are useful on their own, exchanges are needed so that creators, consumers, investors, and traders can transfer these assets to one other openly and freely. As digital assets reach market equilibrium prices, all market participants benefit from clearing markets that are otherwise either unavailable or not at true equilibrium due to interference from governments, regulators, or logistical barriers.
Venture capitalists, service providers, employees, founders, and other company stakeholders benefit from freely available liquidity that is not locked up in arcane or encumbering pre-IPO restrictions. This is true for tokens regardless of whether they are asset-backed, service-backed, or infrastructure-backed. In the asset-backed case case, cryptocurrency exchanges allow for easier transactions of securities that are currently more costly in terms of time and money due to reconciliation requirements that are natively on the blockchain and otherwise handled by back-office workers.

Cryptocurrency exchanges allow people increased flexibility in using the cryptocurrency of their choice for each individual transaction they undertake. They can focus on a few key markets that capture their consumption needs. For example, a user who considers data storage a primary consumption need can store much of their wealth in Filecoin or Sia while another who relies heavily on shipping goods can store wealth in Shipchain or VeChain.

As specialized cryptocurrencies gain traction, users can segment their wealth in specific token ecosystems. They can reward these ecosystems with more of their wealth for positive governance and pro-social activity, forcing all token ecosystems to compete with each other to provide the best economy for users. This competition is only feasible with the proliferation of exchanges that enable fast, secure, reliable transfers of wealth from one token to another.

Finally, exchanges allow traders to profit from predicting the future behavior of markets. As global wealth becomes increasingly concentrated in the hands of the few, so too go the resources needed to predict the future of markets and thereby gain additional wealth. Because cryptocurrency markets are new and lightly regulated, large incumbents have less of an overwhelming advantage over lay-traders in forming accurate, profitable hypotheses. Skilled traders can participate more directly in the rewards of insightful analysis without tethering themselves to major institutions, which can disperse market gains.

Decentralized Exchanges
. Of course, the above is only true to the extent that other sources of power are also decentralized. One clear target for decentralization is the exchange itself. Centralized exchanges can hold undue power over users. This allow them to engage in many of the regressive practices that cryptocurrency enthusiasts dislike in the fiat world.

Some hold users funds without explanation or recourse. A decentralized alternative will allow users to maintain self-sovereign rights to their assets at all times. Some centralized exchanges bar users, currencies, or transactions

Archived copy · cryp2nova.com

because of local law in their domicile. A decentralized alternative has no central operator upon which a nation state can impose restrictions. A decentralized alternative with public code can benefit from perpetual public security audits and publicly driven code changes. All things being equal, decentralized exchanges (DEXes) more adequately capture the ethos of cryptocurrencies than centralized ones.
All things are not yet equal, however. Decentralized exchanges need feature and performance parity with centralized exchanges to successfully compete.
Binance
, one of the most popular centralized exchanges, processes over 1 million transactions per second.
Forkdelta
, a popular decentralized exchange, routinely has transactions that take minutes.
Bittrex
, a popular centralized exchange, has a well-designed UI honed by design specialists and a simple-to-use UX.
Barterdex
, a DEX, has an obtuse UI and requires several cumbersome steps in its UX. Centralized cryptocurrency exchanges pride themselves on listing a myriad of different token types. Most modern DEXes operate within a single cryptocurrencys ecosystem (for example, stellar for
Stellardex
 and ether for
Etherdelta
) allowing users to trade only tokens in one protocol.

To avoid some of these constraints, some DEXes take partial steps to toward decentralization but centralize key aspects of the system (like order books and matching engines). This is a nice attempt but these services do not provide censorship resistance and so do not fulfill the promise of decentralization. The market needs a decentralized exchange that adheres to the core tenets of decentralization and still provides a world-class exchange experience that meets the feature and performance standards set by centralized exchanges.

THORChain and ASGARDEX

THORChain
. Thorchain addresses these issues. With an out-of-box performance of 10k TPS and a goal of 1m TPS, Thorchain can power the same traffic as a centralized exchange. Thorchain decentralizes all aspects of a trade, including order matching and order book population, ensuring censorship resistance. Most DEXes have less than $10M in daily trading volume but Thorchain will be seeded with several times that amount in liquidity directly from the Odin Foundation. Moreover, Thorchains continuous liquidity pool mechanism (borrowed from Bancor) provides fees to users who provide further liquidity, incentivizing even more capacity. Thorchains use of cryptocurrency bridges enable it to natively support transactions across cryptocurrency ecosystems. Users can exchange Bitcoin for Ether or an ERC-20 token for an NEP-5 token directly on a Thorchain DEX. Bridges are addressed in more detail separately.
ASGARDEX (the first DEX that will be built on top of THORChain) will have deep liquidity, a publicly available order book, omni-token support for cryptocurrencies compatible with bridges, a world-class UI, and refined UX. ASGARDEX solves the major problems with decentralized exchanges, allowing users to experience the full promise of a secure, transparent, censorship-resistant and fault-tolerant cryptocurrency exchange.

THORChain Architecture

Overview

Archived copy · cryp2nova.com

THORChain Whitepaper

THORChain is a multi-blockchain protocol comprised of side chains TokenChains and the master MerkleChain. The MerkleChain tracks the entire state of the network, storing the latest hash of the Merkle Tree to prevent double-spending and syncing the state of network. All side chains in the protocol are first-class citizens with no delineation in security, technology or preference in the Validator Set.

TokenChains have a genesis account that describe the characteristics of the token, and solely track transactions for that token. Each TokenChain maintains a discrete address space allowing deconfliction of transaction mempools.

TokenChains are account-based in a similar manner to Ethereum (as opposed to UTXO-based where unspent transactions are tracked such as Bitcoin). A nonce tracks the latest state of each account, invalidating all previous account states.

The native token of the ecosystem is Rune and is created at the genesis of the protocol, with all Rune transactions tracked on the first side chain,

T0

. The Rune is unique as it is the settlement currency of the ecosystem and is stored in all on-chain liquidity pools.

The genesis block of every sidechain has a continuous liquidity pool (CLP) that defines a price ratio for Rune and that token. CLPs are explained in more detail later in this paper. For now, consider CLPs as providing a counterparty to anyone wishing to make a Rune-denominated trade. If two users trade between two non-Rune tokens, one of the tokens is automatically converted to Rune before the transaction is complete.

Figure 1: Each token is tracked on a discrete TokenChain. Genesis accounts track Token information and hold on-chain liquidity.

TokenChains

Each TokenChain has a discrete address space.

CRYP2NOVA

Figure 2: The THORChain Address Space; here for Rune (T0).

The prefix deconflicts the address space with all other protocols. The TokenIndex deconflicts the token from other on-chain tokens, where Rune is

T0

, the first on-chain token is

T1

etc. The Separator separates the TokenIndex (which may be variable length) and the public address, derived from a SHA256 hash of a private key.

Alice, who has a public address for her Rune,

T0xAlice

, will also have a matched address on every other TokenChain that she has tokens for. Her single private key unlocks all addresses. If Alice is sent a new token; her new address is automatically created.

Archived copy · cryp2nova.com

T0xAlice -> Rune address for Alice

T1xAlice -> T1 address for Alice

T5xAlice -> T5 address for Alice

TnxAlice -> Tn address for Alice

Deconflicting address spaces in multiple chains have the following advantages:

Discrete transactions for each token, with discrete mem-pools.

Each token has a clear and distinct lineage.

Each TokenChain inherits a genesis account, which contains and performs cornerstone functionality for the network.

Tokenchains can represent tokens internal to Thorchain or external tokens (like Bitcoin or Ethereum). These tokens can be traded within Thorchain but represent actual external tokens stored within a Thorchain bridge.

The UX for trading
BTC
on ASGARDEX involves user selecting a bridge, transferring the
BTC
to a wallet controlled by that bridge, and specifying in the data field for that
BTC
transaction which Thorchain wallet they want the corresponding Thorchain proxy bitcoin
tBTC
stored in. Bitcoin sent to a bridge without a corresponding
tBTC
wallet address should be sent back automatically. Once this happens, on-chain trading can occur. To withdraw
BTC
, users specify a bridge and a
BTC
wallet and then initiate a
tBTC
transfer from their wallet to the bridge wallet. The bridge then moves the
BTC
to the corresponding BTC wallet. Likewise,

Archived copy · cryp2nova.com

tBTC
sent without the right data fields and formatting will be returned.

MerkleChain

The MerkleChain tracks and syncs the network state by hashing and merkelizing the latest transactions for each TokenChain in the network. The following is the overview of this mechanism:
The Validator Set collects Merkle Roots from each tokenchain, assembled into a Merkle Tree, and hashed as a single Merkle Root for block n for each TokenChain.

The Token Merkle Roots are then collected into another Merkle Tree and a final Master Merkle Root produced. This is then inserted into the MerkleChain block
n + 1
by the Validator Set.
The Validator Set at the same time then inserts the same Master Merkle Root into every tokenchain at block
n + 1
.
The MerkleChain also stores network wide information such as the TokenIndex counter that specifies and deconflicts TokenChains on the network.

Consensus

THORChain implements Tendermint, a byzantine fault tolerant state machine replication algorithm. Tendermint has the necessary maturity, performance and security to serve THORChains requirements. Tendermint is most useful in that it supports instant finality, which is an indispensible property for a blockchain. Tendermint achieves this by only committing transactions that have already reached super-majority consensus, rather than waiting for consensus to be achieved after a transaction is committed (such as Bitcoin and Ethereum V1). As such, Tendermint can support sub-second block times and process up to 10,000 TPS.
Tendermint performance will be sufficient for THORChain initially, but modifications will be required to support the end state of achieving performance parity with centralised exchanges at over 1 million TPS. To achieve this performance, THORChain will need to research and implement various implementations of sharding.

Validator Set

Tendermint requires full-nodes as Validators or block producers, and each Validator must have a weight on the network. The weight is determined by their staking, so Tendermint can support Proof-of-Stake out of the box. The first implementation of THORChain has a single Validator Set drawn from all available Validators through an auction; the
100
highest staked Validators form the Validator Set. Validators propose blocks, agree and commit them. To join the Validator Set, a Validator must stake higher than the lowest Validator, and by this process ensures that the Validators with the greatest economic investment secure the network. Staking pools are possible to allow wider participation, and anyone can delegate their stake to a chosen Validator. Validators are paid from the block reward, and are paid evenly regardless of stake held. Proxy stakers are paid pro-rata to the total stake of a Validator; and so by design and self-interest; should choose the Validator with the lowest current stake, but in the top
100

Archived copy · cryp2nova.com

. This encourages proxy stakers to distribute their stake where they have the most earning potential, and thus consequently will ensure a very flat and competitive group of 100, where the minority can easily influence Validator participation in the Validator Set.

Figure 3: Validator Set

An in-protocol penalty is required to discourage validators from misbehaving on the network. This is heavily researched in both Casper and Cosmos. Misbehaviour may be double-signing blocks, being off-line or lack of participation in on-chain voting. The amount that a validator is slashed depends on the severity of the incident and will likely be finalised with on-chain voting prior to mainnet. If a Validator is slashed to a level below the highest waiting Validator, the dishonest Validator is evicted and the waiting Validator automatically joins the Validator Set. As a result of the flat group, the heavy level of competition and the likelihood that dishonest participation will cause both economic loss to stakes and eviction from the Validator Set, an attacking Validator is heavily discouraged.
Network security is a consideration for THORChain. Historically on-chain voting has extremely low turn-out, as shown by EOS, CarbonVote and Steemit voting events. Cosmos is attempting to solve this by increasing the rate of inflation of the staked token to increase the incentives to bond tokens to validator stakes (directly or via pools). Inflation pivots between a floor of
7%
and a ceiling of
20%
to drive holders to bond tokens at a desired bond state of
2/3
of the entire circulating supply. With bonded, the network is optimally secure and resistant to cartels and plutocracy. A downside to this that the staked token loses ideal currency properties, and as such Cosmos have introduced a dual-token system, with a secondary token being used to pay for transaction fees. It is not known if Casper V2 will implement such a mechanism and if staking for collators will cause Ether to lose ideal currency properties.
THORChain will adopt a single token for simplicity but research the real-world economics of other Proof-of-Stake chains until mainnet. If a secondary token is implemented, it can be done using THORChain on-chain governance.

CRYP2NOVA

Multiple Validator Sets

THORChains primary performance metric of TPS is most dependent on the bottlenecks of the single Validator Set. Adding more validators to a Validator Set will not improve performance, instead it will decrease it. There is an opportunity given the design of THORChain to investigate a Multiple Validator Set architecture, whereby transactions are sharded to each tokenChain and maintained by separate Validator Sets. By processing the transactions on each tokenChain as separate, the network can perform at a potentially unbounded level.
Sharding is possible in THORChain as separate mempools for each TokenChain are maintained by unique address spaces. This allows pending transactions to easily be segregated. The GasLimit is capped for THORChain to allow the network to track saturation and signal for splitting Validator Sets. If the GasLimit approaches upper limits (above
90%
for
100
blocks), then a separate Validator Set will form to meet the scaling demands. In this case, the next 100 highest staked Validators will be selected to form another Validator Set. The primary Validator Set (Validators

Archived copy · cryp2nova.com

1-100
) will evict all TokenChains bar the TokenChain with the highest gas demands and the MerkleChain from their mempool. The secondary Validator Set (101-200) will then maintain all other TokenChains. Further, if the secondary Validator Set approaches its GasLimit, it will also then evict all but the highest TokenChain, and nominate the Tertiary TokenChain, and so on. The MerkleChain stores and tracks multiple Validator Sets. In this way a hierarchy of Validator Sets are established, each tracking their immediate child.
At any stage if a Validator Sets GasLimit reduces below
10%
saturation for longer than
100
blocks then the set is instantly dissolved into its parent set. If a parent set signals to dissolve before its child, then the Master VS re-shuffles all child VSs to re-establish an unbroken hierarchy and prevent disruption. In this way THORChain can scale so that it can always cater for the demands of the network.

Seq	Event	Action
1	>90% Saturation	Validator Set (VS1) start signalling to split

Archived copy · cryp2nova.com

2

100 Blocks

3

101st Block

A new Validator Set (VS2) commissioned

4

<10% Saturation on both Chains

VS1 and VS2 signal to merge

5

100 Blocks

Archived copy · cryp2nova.com

6

101st Block

VS2 decommissioned and VS1 returns to process both chains.

Table: Order for splitting and merging

The Master VS, using the functions of the MerkleChain and processing subservient tokenChain MerkleRoots, will continue to ensure that the network is synced and interoperable. Subservient tokenChains will be less secure than the Master VSs MerkleChain and RuneChain as the Validators who service their mempools will have less stake.

Figure 3: Segregated Validator Sets

There are a number of unanswered questions around network security, performance and the characteristics of having multiple Validator Sets that will require further research and development, however the fundamentals of how it could be achieved are outlined here. Additional research to our plan for implementing multiple validator sets via sharding is in a separate paper.

On-chain Governance

Overview

On-chain governance on THORChain is known as Validator Signalling as it is a continual process performed by validators, and that votes are more aptly referred to as signals. Validator Signalling is covered extensively in the Æsir-Protocol Whitepaper
. An overview is provided here.

Any block producer can propose a change in the core software and consensus rules structured as a data packet:

{description, newCode, diffPatch}

All other validators vote to accept the change and if it reaches supermajority consensus the updated code can be

Archived copy · cryp2nova.com

immediately brought in to operation. The proposing and agreeing validators run the compiled core software on standby, so that once approved, the core software is live to produce the very next block.
The types of updates that can be rolled out are essentially unlimited and could be:

Consensus rules such as supermajority thresholds, or voting rules (relating to on-chain governance itself)

Protocol architecture such as a change to consensus algorithms, integration of sharding, change to the blockchain structure or signature schemes.

Native on-chain commands as discussed, whereby additional trading rules can be integrated at the protocol level.

Changes to the token structure such as supply or inflation.

Changes to state, such as amending exploited or unused accounts.

Key Aspects

Economically Enforced Participation
. Voter participation is enforced by in-protocol slashing rules. Not voting on a proposed update or poll will result in a Validators stake being slashed and redistributed to other Validators who do vote. There is a grace period of n blocks allowing Validators time to poll the community (or their staking pool) and take up a position before casting a vote.

Empowered Minorities
. Quadratic voting is implemented inside a validators staking pool. Each member that stakes with a Validator can cast votes quadratically proportional to their stake, where each vote is a single whole unit, and increases with the number of subsequent votes.
1 vote
costs
1 token
,
2 votes
cost
4 tokens (2 * 2)
,
3 votes
cost
9 tokens (3 * 3)
etc. This appropriately removes the biases that large holders can have on voting, preventing a plutocracy. It also

Archived copy · cryp2nova.com

enables and empowers the minority to know their vote is meaningful and represented. A Validator can submit votes on behalf of non-voting stakers in their pool but the vote is quadratically weighted as above. Thus non-voting stakers are empowered to easily swing their Validators final vote as their individual vote has more comparative weight than the Validator who representatively voted for them using their tokens.

Flexibility
. Any staker can change their vote at any time to signal differently. Delegators who switch staking pools to swing vote a different Validator are bound by bonding periods and can effectively only switch once.

A carefully designed on-chain governance solution (which itself can update) will see THORChain become a forkless self-amending ledger and increase the likelihood of persistence well into the future.

Tokens

Rune Characteristics

The Rune is the token of the ecosystem and resides on Address Space
T0
. Initial supply will be an arbitrary but finite number. The following are the uses of Rune:

Validators are required to stake Rune to be part of the Validator Sets. Once staked the Rune are bonded for a period of time to prevent nothing-at-stake attacks.

All network transaction fees (gas) are paid in Rune. Fees may be transaction fees, trading fees, bridge fees and liquidity fees, imposed by the different elements of the ecosystem.

Liquidity is always backed by Rune in the Continuous Liquidity Pools; so the Rune functions as ecosystem settlement currency.

The Flash Network requires Rune as liquidity to join Liquidity Hubs and fees are paid in Rune.

Block rewards for Layer 1 Validator Sets and Layer 2 Liquidity Nodes are paid in Rune.

Block Rewards and Emission

Archived copy · cryp2nova.com

THORChain Whitepaper

Rune ticker
RUNE
 is a predictable inflationary currency that aims to create a price-competitive ecosystem, driving transaction fees to zero. Inflation is designed to reward those that stake to either secure the network or add either Layer 1 or Layer 2 liquidity. Noting the current inflation rates of Ethereum, Bitcoin, EOS and Cosmos, an acceptable starting inflation of
2-5%
 is proposed.

Block Rewards are issued with
50%
 to Validator Sets and
50%
 to the Layer 2 Liquidity Nodes. Each Validator in the Validator Set will receive:

Liquidity Nodes will receive rewards in accordance with the Layer 2 Incentivisation Plan which rewards for liquidity and reliability.

If Layer 2 is not launched alongside the mainnet, then Layer 2 rewards may be hard-coded to pay back to the Foundation to prevent a stalemate whereby validators refuse to launch the Layer 2 Incentivisation Plan for the incorrect perception that they are being diluted or penalised.

Fees

THORChain retains the concept of gas fees and a gas limit for each TokenChains blocks. The GasLimit is designed to solicit Validator Set splitting when required to scale. Gas fees are paid for each transaction type; of which there are several distinct types. Each transaction type is hard-coded in the protocol so gas estimates can be made.
Layer 1 Liquidity Fees are collected when utilising Continuous Liquidity Pools (CLPs), and are a function of liquidity in the CLP. The higher the slips incurred in CLPs, the more fees are charged, thereby attracting more users to offer liquidity to the pool. Adding liquidity depth reduces slip and thus reduces the fees collected. Liquidity makers can withdraw their liquidity at any stage, including their collected fees. Self-interested liquidity makers will likely bootstrap new CLPs, collect fees, and then withdraw just their initial stake to continue to earn fees in the CLP indefinitely.
Layer 2 Trading Fees are paid to Liquidity Nodes for providing liquidity to Liquidity Hubs. The fee for using each Hub is the average of each fee nominated by each Node as they join. There may be multiple Hubs for each Rune Pair, thereby encouraging a price competitive network, where traders will use the Hub with the lowest average fees. Nodes are free to join any Hub they wish, or create their own. All Liquidity Nodes are paid from the Block Reward, but are weighted to their pro-rata liquidity and time online:

By paying trading fees and block rewards to Liquidity Hubs, nodes are incentivised to be liquid, online, reliable and there is no limit to the number of nodes; thus the network becomes quickly decentralised with many channels. This is outlined in more detail in the Flash Network paper.

Archived copy · cryp2nova.com

TokenChain

Each TokenChain is created in a special genesis transaction
GenTX
on the primary Rune chain
T0
. Once created, the TokenChain is initiated in the next block with a Genesis Account
GenAcc
. The
GenACC
is both the on-chain specification for the characteristics of the token, as well as the account that hosts the tokens Continuous Liquidity Pool (defined later).
GenTXs
require a fee to be paid in Rune; which is an effective anti-sybil measure to spamming the network with new tokens.

Figure 5: The Genesis Account for Token1

The
GenAcc
stores the following information about the Token which can be publicly queried and displayed on order books, wallets and exchanges:

Ticker

TKN1

Ticker to display

Archived copy · cryp2nova.com

Name

Token1

Name to display

Supply

100,000,000

Total Supply (100m)

Decimals

18

Decimals

Reserve

1.0

A parameter to specify the fractional reserve of the CLP (default is 1.0).

Archived copy · cryp2nova.com

Owner

Self

If self; the details above are immutable as there is no private key for the genesis account.

Table: Fixed Supply Token

If the Account Owner is SELF then the token details are immutable and cannot be changed, as is expected in most digital assets. The GenAcc doubles as an easily identifiable continuous liquidity pool (CLP) for that token and with protocol-level security. The CLP can be interacted with by sending transactions to it from any other wallet. In this case the transaction will execute the liquidity transactions discussed further.
If there is an external Account Owner specified, then that Account Owner can change any characteristic of the token, including Supply by proxy. They cannot change the TokenIndex which is permanent and identifies the TokenChain. These tokens are known as variable supply tokens as the account owner is permitted to change the supply of the token. This is necessary for tokenised assets, security tokens and even pegged tokens, whereby the account owner can be a single owner or multi-sig owner that is permitted to change the token characteristics in line with external non-THORChain logic.
Account owners can even hand over control of the GenAcc to other accounts, including accounts without a private key; effectively turning the token into a fixed supply immutable token.

Ticker

TKN1

Ticker to display

Archived copy · cryp2nova.com

THORChain Whitepaper

Name

Token1

Name to display

Supply

100,000,000

Total Supply (100m)

Decimals

18

Decimals

Reserve

1.0

Archived copy · cryp2nova.com

A parameter to specify the fractional reserve of the CLP (default is 1.0).

Owner

T1xa1b2c3d4e5f6

Owner can be another address which will allow mutable token details, necessary for Asset/Security/Pegged Tokens.

Table: Variable Supply Token

A minimum
GenTX
initiation fee can be set
(100 * GasPrice)
to prevent token spam, but stopping runaway token initiation cost.

Rune Generation

THORChain allows a unique mechanism of generating both the full supply of a token, as well as its on-chain CLP at the same time. The special GenTX which creates a new TokenChain requires a non-zero amount of Rune to be transacted. This non-zero amount pays for the GenTX transaction fee, as well as funding the CLP for the first time by trapping the Rune inside the GenACC alongside the newly minted tokens; this sets the initial price of the token.
The token owner will subsequently transact in a specific amount of Rune to emit the full token amount to their custody. The following will happen in the case that 100 Rune was the GenTX, and 1mn tokens were created (decimals not included):

Genesis Emission

Archived copy · cryp2nova.com

TX in (Rune)

Rune Locked

Tokens Locked

Price (RUNE)

Tokens Emitted

GenTX

(100 Rune)

100

1,000,000

0.0001

Subsequent CLP Transaction

TX in (Rune)

Archived copy · cryp2nova.com

Rune Locked	90
Tokens Locked	190
Price (RUNE)	100,000
Tokens Emitted	0.001
	900,000
	99
	199
	10,000

Archived copy · cryp2nova.com

0.01

990,000

99.9

199.9

1,000

0.1

999,000

99.99

199.99

100

1.0999,900

99.999

Archived copy · cryp2nova.com

199.999

10

10

999,990

99.9999

199.9999

1

100

999,999

Table: Creating and emitting a new token. There are two transactions; (1) the GenTx and (2) the subsequent CLPTX that emits a specified amount of tokens. They can be performed in one request.

The locked Rune and tokens that remain in the GenAcc can never be fully emitted, thus the pricing of the tokens are set from the first block. If initial pricing or liquidity is unsatisfactory, special one-way Liquidity Transactions can be performed to correct pricing and add liquidity.
It is imagined that the developer community will build software that allows creating tokens to be performed with an easy to understand user experience. The following is an example of how the creation of the token can be done in one client-side order:

Archived copy · cryp2nova.com

Parameter

Field

Notes

Token Parameter

{Ticker: TKN1, Name: Token, Supply: 100m, Decimals: 18, Owner: SELF}

Create a data packet to attach to GenTX.

CLP Parameter

{Reserve: 1.0, InitialPrice: 0.001, InitialLiquidity: 1000 Rune, InitialEmission: 99%}

Set and calculate the initial price, liquidity and emission of the token in the CLPTX.

Table: Parameters to create a new token

THORChain Bridges

THORChain will have native cross-compatibility with UTXO, Account and Contract-based cryptocurrencies. The heart of this is are Trustless Two-Way-Pegs (2WP) known as the Layer 1 Bridges, and multi-signature accounts. The core

Archived copy · cryp2nova.com

Validator Set (
100
 staked Validators) are signatories to
67/100
 multi-sig accounts to external chains, and a
2 / 3
 signature threshold on THORChain. This allows Bitcoin, Ethereum (and their forks) as well as ERC-20 tokens to be seamlessly moved onto the THORChain ecosystem and off again. The external coin is moved into a multi-sig account, signed by the Validator Set. After observed finality, the Validator Set create a signature threshold to mint new tokens in THORChain via CLPs. BLS signature thresholds are used to prevent issues during Validator Set re-orgs, as only a threshold of signatures is required to be achieved. The reverse occurs to move the coin off the ecosystem. BLS signature threshold theory has been extensively researched by DFinity. The benefits of signature thresholds is that it only requires a threshold of signatures from a super-set, and not specifically a certain sub-set from that super-set. This translates to flexibility in who can be part of a sub-set, and tolerates validators leaving and re-entering the Validator Set, which could be a frequent occurrence in a healthy and competitive environment of validators.
Cryptonote Coins such as Monero and Loki do not support
m of n
 multi-signature, but can support
n of n
 or
n-1 of n
. It is possible to support these coins, however there is a risk in a Validator Set re-org that more than one of the signatories is replaced. If the evicted Validators do not stay online, then the Cryptonote coins that require their signature will remain locked indefinitely on THORChain. There is no risk of theft, and indeed the user can trade from
tXMR
 to
tBTC
 and exit on the Bitcoin bridge safely, putting this inconvenience on a low significance. A mitigation plan to this is to re-shuffle validators immediately if just
1
 validator is lost. The BiFrost paper contains additional research on Thorchains bridge protocol research.

Continuous liquidity pools

Solving Liquidity

Decentralised exchanges are notorious for having low or zero liquidity to the point where they can be unusable for low liquid pairs. This can also be the case for low-tier centralised exchanges.
Solving liquidity has been a large focus for Bancor Network and arguably a successful endeavour. Bancor introduced the concept of continuous liquidity by using smart tokens and connectors built on smart contracts deployed on Ethereum. Tokens and Ether are held in smart contracts in such a way that they become bonded and are priced according to the ratio at which they are held. Users can send either tokens or ether to these smart contracts and the other asset is emitted according to a slip-factored price, which takes into account the liquidity depth of the pool. As a result, liquidity is always available for these tokens.
THORChain integrates two on-chain liquidity strategies; an adaption of Bancors continuous liquidity strategy, the CLP, and on-chain order-book liquidity multiplication adapted from the Komodo blockchain, discussed later.

CRYP2NOVA

Archived copy · cryp2nova.com

CLPs

The CLP is arguably one of the most important features of THORChain. By building in on-chain liquidity the ecosystem receives the following benefits:

Provides always-on trustless liquidity to all tokens in the ecosystem.

Improves the user experience for low-liquidity tokens.

Functions as source of trustless on-chain price feeds to power all aspects of the ecosystem, including advanced trading types.

Generates arbitrage opportunities, further increasing token liquidity.

Allows users to trade tokens at trustless prices, without relying on centralised third-parties.

Provides trustless price-anchors to the Layer 2 Flash Network, allowing instant trading at the layer 2 level.

CRYP2NOVA

Figure 6: The CLP is the genesis account.

CLP Transactions

The
GenAcc
is the CLP for each TokenChain, holding both Rune and the full supply of the created tokens as locked liquidity. The account owner can not move the tokens; they can only interact with the
GenAcc
on-chain command. There are six valid transaction types:

Rune In
. Anyone can send Rune to the
GenAcc

Archived copy · cryp2nova.com

. The
GenAcc
will emit
TKN1
(minus a fee), sent to the senders
TKN1
address

TKN1 In
. Anyone can send the specific token to the
GenAcc
. The
GenAcc
will emit Rune (minus a fee), sent to the senders Rune address.

Figure 7: CLP Transactions

Rune LiquidityTx
. Anyone can add Rune to the locked liquidity in the
GenAcc
, and no
TKN1
will be emitted. This is a special transaction that registers the senders address as well as the balance sent in order to track and pay them pro-rata liquidity fees.

TKN1 LiquidityTx
. Anyone can add TKN1 to the locked liquidity in the GenAcc, and no Rune will be emitted. Again, the senders address and balance will be registered for payment of liquidity fees.

Figure 8: Adding liquidity to a CLP

Rune LiquidityWithdrawTx
. Anyone that added liquidity to the CLP is permitted to withdraw up to the maximum of their initial liquidity and earned fees.

TKN1 LiquidityWithdrawTx
. Again, anyone can withdraw their staked Token1 liquidity up to the maximum.

Archived copy · cryp2nova.com

Figure 9: Rune FeeWithdrawTx. Anyone that added liquidity to the CLP is permitted to withdraw their earned fees.

TKN1 FeeWithdrawTx
. Again, anyone can withdraw their earned fees.

Importantly, the Locked Liquidity will only emit tokens at the internal price; inferred by the locked
RUNE:TKN1
ratio. If
10 RUNE
is locked alongside
1000 TKN1
; then the price of
1 TKN1
is
0.01 RUNE
, and tokens will be emitted at this internal price. The emission rate factors in slip which is the price of the token after the emission occurs, governed by the following equation:

Reserve Ratio can be adjusted to be lower than
100%
to prevent large slip rates and is akin to fractional reserves.

Liquidity Fees

The Liquidity Fee is a function of slip; the rate at which the price will change in a transaction that emits Rune or Tokens. Importantly, there will always be a slip, as every transaction (no matter how small) will change the ratio between locked Rune and Tokens, so the subsequent change in ratio defines a new price. To couple fees with slip and thereby encourage more self-interested users to add liquidity in CLPs that need it most (high volume and low liquidity CLPs), fees are paid out as:

Liquidity Depth

Archived copy · cryp2nova.com

Transaction

Slip

Liquidity Fee

1m Rune

100 Rune

0.01%

0.001 Rune

100,000 Rune

1000 Rune

1%

1 Rune

1000 Rune

Archived copy · cryp2nova.com

500 Rune

50%

25 Rune

Table: Example Liquidity Fees

Trustless On-chain Price Feeds

On-chain Arbitrage
. Any interaction with the GenAcc will emit the opposite pair at a slip-factored internal price. The slip that is caused by a large emission may move the GenAcc pricing away from fair market price. In this case, self-interested arbitrageurs will immediately correct the price by performing a reverse transaction.

Figure 10: On-chain Arbitrage

Price Oracle
. Token pricing in the GenAcc should always the fair market price, else self-interested arbitrageurs would have taken action. After large price movements there may be a delay, but large price movements cause large Liquidity Fees which self-corrects by attracting more liquidity. Thus the THORChain network can employ token pricing trustlessly throughout the ecosystem based on GenAcc pricing.
Trustless token pricing could be employed by future versions of the protocol in the following ways:

To facilitate advanced order types such as margin trading and stop loss/take profits.

Used across THORDApps to allow advanced dApp features around token prices.

Used to transmit fair market prices to the Layer 2 Payment Network, facilitating instant trades across supported assets.

Archived copy · cryp2nova.com

Price Manipulation
. The cost of price manipulation is high in THORChain, due to extant economics. The CLPs that are mostly likely to attract manipulation are the most used pairs, however the act of manipulation immediately exposes the manipulator to on-chain arbitrage and high liquidity fees. Sustained manipulation in a CLP will increase volumes across that CLP, increasing the amount of liquidity staked by self-interested liquidity stakers, thereby reducing the effect of manipulation in the long run.

On-chain Trading

Account Types

THORChain integrates on-chain trading at the protocol level, fulfilling all aspects of the desired trade experience. THORChain makes use of account-based architecture which offers flexibility over the UTXO model. There are three account types in the ecosystem, created with the appropriate native on-chain command:

Wallet Account
. Wallet Accounts have the following characteristic:

- Store a balance that can only be transferred by the Account Owner.
- Store a nonce.

Balances are updated with a unique nonce, and old balances are invalidated if a later nonce is published.

CRYP2NOVA

Figure 11: The THORChain Wallet for Rune (T0). Alices public address is [aaa...aaa]

Trading Account
. Trading Accounts have the following characteristic:

- Store a balance that can only be transferred by:
  - The Account Owner
  - A valid incoming trade, and only after performing the Trade Execution.
- Store a Price; that generates a Trade Execution, such that:

Archived copy · cryp2nova.com

- Store an Expiry time in blocks.
- Store a Host Fee; such that user-facing clients can add an optional fee.
- Store a Host Address to send Host Fees to.
- Store a nonce.

Figure 12: The THORChain Trading Account for Rune (T0). Alices public address for the pair (T1) [aaa....aaa] is automatically inserted, but can be specified.

CLP Account
. A CLP is a special account that resides on genesis account of each tokenChain:

- Store an array of stakers (address and commitment amount) to collect fees for.
- Store balances of Rune and Token.
- Store fees for Rune and Token.
- Store Token Data.
- Store CLP Data.
- Store a nonce.

Figure 13: The THORChain CLP Account for TKN1 (T1).

Transaction Types

Alongside Accounts, THORChain adds unique Transaction types that cover transactions and trades. These are executed as native on-chain commands that perform math on each trade and update account nonces to save state.

TX
. Send a Balance to a recipients wallet. If sent to a CLP, corresponding tokens are emitted to back to the senders token wallet.

- Transfer Rune or TKN.

If the recipient does not have the required wallet, it is created using the same publicAddress.

Archived copy · cryp2nova.com

Figure 14: Alice paying Bob on a single chain.

Figure 15: Alice paying Bob across chains. Alice can pay to T0x(bob) or T1x(bob) or T2x(bob); but balance only updated at T1x(bob).

Figure 16: Alice interacting with TKN1 CLP with T0. She receives TKN1.

LiqTX
. Send a balance to a CLP to add liquidity. Senders address is added to collect earned fees.

- Transfer Rune or TKN.

If the incorrect TKN is sent, it will instead be sent to the matching token CLP. This may not be desired, but easily recovered.

Figure 17: Alice adding liquidity to CLP1.

LiqWdTX
. Withdraw liquidity from a CLP.

- Call Withdraw on the CLP.

If signatures match with Stakers Array, all liquidity is emitted to senders address, including earned fees.

Figure 18: Alice withdrawing her liquidity + fees.

FeeWdTX

Archived copy · cryp2nova.com

. Withdraw fee from a CLP.

- Call WithdrawFees on the Account.

If signatures match, all fees emitted to senders address.

Figure 19: Alice withdrawing her fees.

GenTX
. The Genesis Transaction is used to create a new Token and TokenChain.

- Transfer RUNE.
- Set Ticker and Name.
- Set Supply. Leave 0 for variable supply.
- Set Decimals (18 default)
- Set Reserve Ratio.
- Set Account Owner, default is Self.

Figure: Alice creating Token1 TokenChain.

CRYP2NOVA

TradeTx
. The Account Transaction is used to create a Trade Account.

- Transfer Rune or TKN.
- Set Price. In Rune or TKN.
- Set Expiry (in blocks).
- Set destination account (default is Alices token address).
- Set Host Fee. Optional.
- Set Host Account. Optional.

Figure: Alice setting a T1 Sell Order for
T0

Archived copy · cryp2nova.com

WalletTx
. The Account Transaction is used to return a Trade Account to a Wallet Account.

- Transfer Rune or TKN.

Figure: Alice cancelling her trade.

TradeSolicitTx
. The Trade Transaction is used to solicit a Trade.

- Transfer Rune or TKN.
- Set Price. Desired Price to fill at.
- Set Limit. Furthest deviation from Price allowed. Leave blank for Market Order.
- Set Expiry (in blocks). Used if the Order is not immediately filled, and becomes a WalletTX.
- Set Host Fee. Optional.
- Set Host Account. Optional.

Figure: Bob buying Alices T1 Sell Order. The transaction is sent across chains from
TO
to
T1

TradeExecuteTx
. The successful outcome of TradeSolicitTx.

- Transfer TKN to Buying Party Destination Account.
- Refund spare Rune back to Buying Party Account.
- Transfer Rune to Selling Party.
- Update nonces.

Figure: Bob buys Alices T1 Sell Order, is refunded the extra. Alice gets the Rune.

Archived copy · cryp2nova.com

Limit and Market Orders

Refund
. As transaction fees are non-deterministic and may not be known ahead of time, the final balance sent to an order may create dust or excess if the trade amount exceeds the target order. Thus the concept of a Refund is introduced. This Refund becomes a useful mechanism to support other trade types such as Market Orders and Advanced Trading by relaying instead of refunding.
Relaying Refunds from one order to another may be seen as inefficient as it generates additional consequential transactions that rely on the previous being filled, however the benefits are twofold. The first is the trader can fire and forget their trade, knowing that it will be executed as intended prior to the expiry specified. The second is that the trade remains trustless as it does not involve client software splitting up a traders orders on behalf of them which can be vulnerable to interception attacks. Ultimately it is a better user experience as the trade can be performed in one user-facing action instead of multiple.

Limit Order
. A trader wishes to spend their entire balance on open orders within a price limit. They specify their starting price as well as a limit, and collect all open orders inside that range, before the order expires. The mechanism of this is to relay the Refund from one order to another until all is fulfilled (or not).

Figure: Bob issues a limit order.

Market Order
. A trader may just wish to clear their trade at any market price available. This is done by relaying the Refund from one order to another until all is fulfilled (or not).

CRYP2NOVA

Figure: Bob issues a market order.

User Experience

Wallets

The following is the user experience of interacting with THORChain-based wallets and exchanges.

Instantiating Wallets
. Users create a new wallet by generating some form of off-line entropy that is used to create the private/public key

Archived copy · cryp2nova.com

pairs. These key pairs can be represented and secured by seed-words (BIP39), key-stores or even memorable brain wallets. An innovation proposed by FairLayer is to use [email + password] as the source of entropy as it is both unique, memorable and portable. A single private key controls all of a users token addresses (one for each tokenChain). Wallets can support Hierarchical Deterministic key generation which allow users to generate n number of private/public key pairs from the same entropy.

Maintaining Assets
. All of a users assets can be viewed inside a single wallet interface. Asset details
(name, ticker)
can be trustlessly queried from the
GenAcc
whilst prices for assets can be queried from CLPs. Bridge accounts can also be queried from GenAccounts for tokenised assets (tokenised Bitcoin), so users can easily send external assets across the ecosystem.

Trading

Trading a CLP
. Users can swap assets from inside their wallets by simply sending to CLPs for each assets. Pricing is shown ahead of time, as well as any expected liquidity fees. The advantage of trading a CLP instead of a public order book is that the swap can be performed immediately, with transparent fees and trustless pricing.

Executing a Trade
. Users can import their key pairs or entropy to a decentralised exchange built on THORChain. They have immediate access to all of their assets for public trading.

Creating & Updating an Order
. Users can seamlessly move Rune or any other token into a trade account and create a public Buy or Sell order (depending on the side). The Order can be updated at any time. If performed in a Wallet or on an Exchange front-end, the developer may insert an optional fee for development revenue. This fee is public and encourages price competition.

Large Orders
. Traders and developers have two options for completing large orders. Traders can split up their assets manually and pick off individual orders, sending a small amount of asset to each discrete order to complete trades. Developers can also build client-side software to split up orders for traders and send these trades to multiple orders but this is not preferred.

Exchanges

Traders will have access to exchange interfaces that are familiar in experience and features to current exchanges. All exchanges are accessing the same order book and assets, so they all share liquidity. The ASGARDEX exchange is one such exchange to be launched for mainnet but will be open-source and able to be forked easily. This will drive developers to continually innovate instead of building closed-source moats.

Archived copy · cryp2nova.com

Viewing Trades
. Once orders are served, a block explorer can index trading accounts and display on a public order book. Trade sides are always paired to
RUNE
; but
TKN:TKN
markets are also available if specified. In this case the pair is ordered respective of the underlying TokenIndex. The following would be displayed on a client-side exchange, and can be displayed on any client side interface:

Liquidity Depth.

Candlestick and Volume charts.

Previous Closed Trades.

CLP Accounts

Token Information

Bridge Accounts

CRYP2NOVA

Figure: Reading and displaying the trades.

Trading
. Traders can access all features of the protocol to perform trades on their assets and leave at any time. Private keys are held on client side and signatures may be queried from hardware wallets.

Exchange and Wallet Funding

Exchanges and Wallets have client-side access to insert a transparent Hosting Fee and Account to each AccountTX. As each trade is made (maker, taker, CLP trades) the fee is immediately paid to the host account.

Archived copy · cryp2nova.com

Figure: Exchange inserting their maker fee.

This will support the development of excellent user-experiences in exchanges and wallets that use the THORChain protocol. The pricing mechanism only gathers fees from trades originating from each Wallet and Exchange, so it will naturally support apps that are great experiences. The fee mechanism is also price competitive and public, ensuring that users get the best fees.

Cross-token Trading

For Users
. The Rune is the settlement currency of THORChain and is paired to all tokens in their CLPs, (no token can exist on THORChain without a CLP being created). Users who wish to trade from one token to another such as
T1
 to
T2
 do not have a CLP available that matches
T1:T2
. Instead they can perform two transactions that settle with Rune;
T1:Rune:T2
. Which uses the
T1:Rune
 and
T2:Rune CLP
. In this case the emitted Rune is immediately re-routed to the next CLP. The user can be assured that both prices are trustless and represent fair market price. Fees can also be transparent and known ahead of time, and the entire trade will take
2
 blocks to complete.

CRYP2NOVA

Figure: Routing via two CLPs.

For Traders
. With the mechanism described above, all tokens now have liquidity with each other. This may be sufficient for the needs of traders, but advanced traders may prefer direct order-book markets from one token to another, such as for tBTC:tETH
. This is up for users to create by simply specifying a pair that is not in Rune when creating orders. The THORChain order-book explorer will collect orders and pair them appropriately on any DEX built for THORChain.

Archived copy · cryp2nova.com

Figure: A T1:T2 sell order that is not paired to Rune
T0
.

On-chain Initial Exchange Offerings

THORChain has all the mechanisms to support native on-chain IEOs. The benefits to this instead of a contract-based token is that the token standard is immutable, auditable and no back doors can be built. Additionally, the token would gain instant liquidity with an immediate market. Lastly, users can participate directly from their wallets which minimise interception attacks. The following would be the process for any team to create their own crypto-asset:

The TokenChain is minted with token details.

The project team then emit their token to their own account: liquidity is created and pricing is set.

The project then create a number of large public Sell Orders (may be with different price points bonuses.

Contributors then send Rune to each Sell Order. Projects may also choose to pair to non-Rune tokens (like
tBTC
or
tETH
).

CRYP2NOVA

Contributors receive the new token in their wallet immediately.

This will herald a new generation of liquid, auditable and trustless cryptoassets.

Validator Staking

A key part of the protocol is that Validators stake or bond Rune to be part of the 100 in the Validator Set and agree to be bound by slashing rules. The economic cost of being slashed defines the security of the protocol. With 67% requirement for consensus, the entire protocols security can be objectively observed to be:

Archived copy · cryp2nova.com

Thus THORChain attempts to create the right mechanisms to encourage Rune holders to stake as much of the circulating supply. The following are aspects of this:

A proposed
5%
Annual Inflation encourages Rune holders to reduce the cost of inflation and stake as Validators.

An auction-based entry to the Validator Set encourages Rune holders to outbid each other in order to enter.

A bonding period of
14
days to lock tokens once accepted as a Validator. This prevents long-range nothing-at-stake attacks.

An unbonding period of
14
days to prevent a rogue Validator attempting a quick getaway with no repercussions after malicious activity.

A method to allow delegation of stake to a Validator without the cost of running validator infrastructure. This empowers minority Rune holders to participate in the security of the network.

The mechanisms and UX to staking is simple in THORChain making use of the
T0
Genesis Account. Ordinarily GenAccounts are CLPs, but in the special case for
T0
, there is no CLP required as the currency pair is Rune itself.
T0
GenAccount stores the token information for Rune, such as current circulating supply, decimals, as well as the name and ticker. Instead of Liquidity Stakers staking to earn Liquidity Fees, Validators Stake in
T0
to earn block rewards. The transaction types for this use the same base logic as CLP staking transactions.

CRYP2NOVA

Ticker

RUNE

Ticker to display

Archived copy · cryp2nova.com

THORChain Whitepaper

Name

Rune

Name to display

Supply

100,050,000

Total Supply after 1 year at 5%
inflation.

Decimals

18

Decimals

Reserve

1.0

A parameter to specify the fractional reserve of the CLP (default is 1.0).

CRYP2NOVA

Archived copy · cryp2nova.com

Owner

Self

Owned by the Protocol

Table: Rune Information. Total Supply updated every block by the Validator Set.

The first
100
Validators Auction their way into the Validator Set by staking into
T0
. The Top
100
are assigned to be Validators, assessed every block. Staking is simply a transaction to
T0
, where the incoming address is saved in the Staking Array, alongside balance.

Figure: A prospective Validator stakes.

The Top
100
Validators in the Staking Array Stakers are Validators that secure the protocol, and this is tracked in the Validators array. If a new validator stakes more than the
100th
Validator, the old Validator is removed from the Validators array and the new one is added. This can be observed by anyone and is updated every block. A bondPeriod specifies the minimum amount of blocks before a Validator can withdraw stake (
14
days). Block Rewards accumulate in
T0
. When a block reward is issued, the tokenData field is updated to increase total Rune supply commensurately. Staked Validators can withdraw their split share of Block Rewards at anytime (
1/100th
of the accumulated rewards). Reward entitlements are tracked in the Validators field.

Archived copy · cryp2nova.com

Delegating tokens to an existing Validator (or Staker who wishes to become a Validator) is performed by the Rune holder simply performing a transaction that specifies an address that is already in the Staking Array. Not specifying a Staking Address, or an address that is invalid will result in them contending to be a Validator instead of a Delegator:

Figure: A rune holder delegates to a Staker.

The Delegators address is nested against a Staker, storing balance and their address. The Staker has no authority to spend a Delegators balance, but it counts towards their total balance. If Delegators add enough balance to a Staker to become a Validator, the Staker enters at the next block. Delegators are entitled to withdraw their share of rewards at any time, pro-rata against what has been accumulated towards their Validators address. Delegators are bound to the bonding period and unbonding period of their parent Validator.
Withdrawing Stake and Rewards is via the same logic that allows Liquidity Stakers to withdraw Liquidity and Fees in CLPs, covered in
§ 6.2
. Voting and Validator Signalling is covered in the Validator Signalling Whitepaper.

Advanced Transaction Features

These features require more research and is not in the primary development pathway

THORChain Name Service (TNS)

Users who wish to assign short human-readable names to their accounts to improve the user experience around payments, such as
@satoshi
, can use the TNS. Short names are unique, stored on-chain and indexed in the MerkleChain.

The user names their account by performing a naming transaction.

Their chosen name alice is checked against the MerkleChain for uniqueness and stored.

A single name can be used for all of their asset accounts, but can be specified, such as
alice.rune
,
alice.btc

Archived copy · cryp2nova.com

,
alice.0
by setting either the index or the ticker as a suffix.

Sending RUNE to
alice.btc
will forward it to
alice.rune
regardless.

Figure: Alices Rune Account is alice.rune

THORChain has the opportunity to rethink how a name service can be operated such that it achieves both effective allocative and investment efficiency. Names will end up being redistributed to parties who can derive the most value from it, and each name can become an investment to the owner. The principle behind the TNS is a hybrid between Harbinger taxes and staking auctions to prevent name-squatting.
If a squatter owns a name that another user wishes to purchase, the buyer can simply stake Rune with the squatters account, known as Name Staking in a special transaction. If the squatter does not hold more than the buyer in their own account, the buyer can purchase the name trustlessly after m blocks. If the squatter does not wish to sell, or cannot afford to hold more in their account, they begin paying a fee to anyone who Name Stakes in their account. The squatter can sell at any time to the highest bidder by simply withdrawing the highest bidders stake and the name is trustlessly swapped. A very keen buyer must either be patient to acquire the name, or stake a higher amount to increase the fees the squatter has to pay. There can be multiple Name Squatters in an account.

CRYP2NOVA

Account Owner

Name Staker

Fee

Archived copy · cryp2nova.com

10 Rune

20 Rune

20 *
every
m blocks

12 Rune

12 *
every
m blocks

5 Rune

5 *
every
m blocks

Archived copy · cryp2nova.com

1 Rune

1 *
every
m blocks

Total:

38 Rune

0.1 Rune
every
m blocks

Table: If out-staked, the Owner pays a Fee based on the difference between their stake and the highest bidder. They can sell the name any time and if they dont pay the fees the name is open to be acquired.

Figure: Name staking to acquire a desired name.

CRYP2NOVA

A benefit to this approach is that squatters are always eventually coerced out of the name they are squatting, but at an appropriate price. If the squatter is in fact a disused account, then eventually the account will be emptied via fees and the name can be acquired.

Account Recovery

Account Permissioning was first described in the
EOS whitepaper
. While more granular controls over an account can be helpful, the key aspect is account recovery. Account recovery processes are important for mainstream adoption.

A user specifies two other types of accounts on their primary account:
Recovery

Archived copy · cryp2nova.com

and
Guardian
.
Recovery
accounts are accounts that can move all funds and TNS name to their account, but are always deactivated. A
Guardian
account activates
Recovery
accounts. The recovery process is as follows, relying on primarily out-of-band communication and existing social or trust networks:

A user creates an account.

The user nominates a
Guardian
account (or multiple). The
Guardian
is notified and signs the request.

The user nominates a
Recovery
account (or multiple).
Recovery
accounts are notified and sign the request. They are deactivated.

The user loses access to their private key.

The user contacts their
Guardian
to activate one of their
Recovery
accounts.

The user can then move their funds from their lost account to their activated recovery account, signed from their
Recovery
address.

This mechanism is safe as it requires both their
Guardian

CRYP2NOVA

Archived copy · cryp2nova.com

and
Recovery
accounts being compromised.
Guardian
accounts can do nothing but activate pre-selected
Recovery
accounts. Additionally,
Recovery
accounts are specified before the fact, so funds can only move to an address specified by the original owner.

Figure: Guardian and Recovery addresses are stored on-chain.

Figure: A Guardian activates a Recovery Address. The Recovery Address withdraws the balance.

Multi-sig Accounts

Multi-signature accounts are supported at the protocol level as a key part of managing assets. The account owner simply adds external accounts in special transactions. Once added all future transactions require all parties to sign. The signature is aggregated in the account and the last party performs the transaction. The signature is cleared once the transaction is performed. The default is
n of n
but this can easily be changed at any time by the parties.

CRYP2NOVA

Figure: A THORChain multi-sig.

Transaction Receipts

Sending payments on a public blockchain is problematic for the point of privacy; but invaluable for recording an immutable history of transactions. For public adoption, transactions need to be tracked and reconciled and these are often done through third party software. THORChain has an opportunity to re-think how transactions can be tracked and reconciled for users on-chain with no third party software, but still done in the private fashion using the assymetric key cryptography.

Each transaction is sent with a hash of a transaction description:

Archived copy · cryp2nova.com

type

Tx

struct
 {
  balance
int64

//
  Balance to transfer from Sender

  token
int64

//
  TokenIdentifier

  to_address
string

//
  Address of the Receiver

  data_hash
string

//
  Optional. hash.SenderPrivKey(description) or

//
  hash.ReceiverPubKey(hash.SenderPrivKey(description))

}

The description is hashed and stored on-chain. There are two options:

Only the
Sender
 private keys are used to hash the description. In this case only the
Sender
 can decrypt.

Both the

Archived copy · cryp2nova.com

Sender
 private keys as well as the
Receiver
 public keys are used to hash. In this case both (and only) the
Sender
 and
Receiver
 can decrypt and read the message, yet the hash is stored publicly on-chain.

Considerations
. There are a number of considerations that need to be factored into this functionality and its implementation:

Transaction descriptions will add an extra 256 bits to each transaction; which can add up and increase block sizes. As transaction descriptions are not
critical
 information and present no risk to assets if a collision is detected, smaller SHA digests or other hashing functions can be considered such as SHA-224 (224 bits) SHA-1 (160 bits) or MD5 (142 bits). A user who wants full privacy will not use on-chain descriptions anyway.

For full two-way decryption, the
Sender
 must know the
Receiver
 Public Key. As THORChain addresses are trimmed hashes of the Public Key, it is not possible to extract the PubKey from the address, so the PubKey must be shared using off-chain channels, or by using the THORChain Messaging Protocol. Exposing a PubKey
may
 increase the probability of an address being compromised in future by advanced key breaking algorithms. Users should be communicated the risks so they do not expose the PubKeys on high-value addresses.

With this lightweight functionality built on-chain any wallet for THORChain can decrypt and show transaction information universally. Third-party services can use the
data_hash
 field to insert payment references allowing automated reconciliation and payment tagging. With the correct cryptography, THORChain transactions are private yet useable.

Transaction Requests

Archived copy · cryp2nova.com

Using similiar functionality to
Transaction Receipts
, payments can be requested on-chain from the
Sender
. With the correct implementation the
Sender
 will be notified and can simply pay the request. The
Requester
 may or may not be the final
Receiver
.

Each transaction is sent with a hash of a transaction description:

type

TxRequest

struct
 {
  balance
int64

//
  Balance Requested from Sender

  token
int64

//
  TokenIdentifier

  to_address
string

//
  Address of the Receiver

  data_hash
string

//
  Optional. hash.RequesterPrivKey(description) or

//
  hash.SenderPubKey(hash.RequesterPrivKey(description))

CRYP2NOVA

Archived copy · cryp2nova.com

}

Once the
TxRequest
is processed on-chain the
Sender
is notified by an Events Service from supported wallets, or simply shows up in their wallet. The
Sender
can then simply pay the request. This functionality has the same considerations as
Transaction Receipts
.

Recurring Transactions

By advancing the functionality of
Transaction Requests
, psuedo recurring payments can be built on-chain.
Requesters
send thru batches transaction requests (such as 12 requests for a 12 month subscription), which are pre-approved by the
Sender
. Once approved, the
Requester
can execute the payment with the pre-signed authorisation. Start and expiry block times are inserted to prevent the transaction being executed before or after pre-authorised time window.

type

TxRequest

struct
{
balance
int64

//
Balance Requested from Sender

token
int64

//
TokenIdentifier

to_address
string

Archived copy · cryp2nova.com

//
  Address of the Receiver

  auth_hash
string

//
  Initially empty. The Sender inserts their signature here once they approve.

  start_blocks
int64

//
  The earliest the TX can be made.

  expiry_blocks
int64

//
  The latest the TX can be made.

  data_hash
string

//
  Optional. hash.RequesterPrivKey(description) or

//
  hash.SenderPubKey(hash.RequesterPrivKey(description))

}

Once the
TxRequest
 is processed on-chain the
Sender
 is notified by an Events Service from supported wallets, or simply shows up in their wallet. The
Sender
 simply authorises the payments one-by-one, but they can't be executed by the
Requester
 until the window of time between
start_blocks
 and
expiry_blocks
. This functionality has the same considerations as Transaction Receipts. The Sender can remove their
auth_hash
 to cancel the payment.

Archived copy · cryp2nova.com

Escrow Accounts

Escrows are a vital part of a payment protocol. Alice pays into an escrow, and nominates Bob to receive, and Charlie to escrow. Alice can only
release
the payment to Bob. Bob can only
refund
the payment to Alice. Charlie can
refund
or
release
, or do a partial
refund
. All transactions can be done in range
0% : 100%
allowing partial refunds or releases.

The implementation is trivial. Alice makes an
EscrowCreate
transaction and observes the newly created escrow address. She can include an optional transaction message.

type
txEscrowCreate
struct
{
balance
int64

//
Balance to transfer from Sender

token
int64

//
TokenIdentifier

to_address
string

//
Address of the Receiver

agent_address
string

Archived copy · cryp2nova.com

```
//
  Address of the Escrow Agent. Set Null for Protocol-level escrow

  data_hash
string

//
  Optional. hash.SenderPrivKey(description) or


//
  hash.ReceiverPubKey(hash.SenderPrivKey(description))

}
```

To release, Alice makes an
EscrowRelease
 transaction, which checks her address signature and releases the balance (fully or partially) to Bob.

```
type

EscrowRelease

struct
 {
  escrow_address
string

//
  Address of the Escrow

  amount
int64

//
  Amount to release

}
```

To refund, Bob makes an
EscrowRefund
 transaction, which checks his address signature and refunds the balance (fully or partially) to Alice.

```
type
```

CRYP2NOVA

Archived copy · cryp2nova.com

EscrowRelease

struct
 {
  escrow_address
string

//
  Address of the Escrow.

  amount
int64

//
  Amount to refund

}

To refund or release, Charle makes an
EscrowRelease
 or
EscrowRefund
 transaction.

Automatic Disbursements

Trustlessly disseminating to a number of accounts can be easily done through THORChain's Automatic Disbursement feature. Alice creates a Disbursement Account and adds third-party addresses to the account. Any tokens she sends to this account will now disburse to the addresses listed, in specified proportions. Alice can also specify how to treat different tokens on disbersement.

type
 txCreateDisbursement
struct
 {

DisburseRecipient
       array
//
 Array of [addr, rate]

}

type

Archived copy · cryp2nova.com

DisburseRecipient
 array {
  to_address
string

//
  Address of the Receiver

  rate
int64

//
  Rate to send to Receiver

  token
int64

//
  Token type to disburse. Null to send all

}

Combined Payment Features

Any one of THORChain's Payment features can be combined to create powerful payment facilities, such as:

A bounty escrow being created, with the protocol being the escrow agent. The recipient address is a list of community contributors who are paid from the bounty.

Payroll being disbursed to staff from a recurring payment from an escrow account.

A transaction request to an escrow account that is signed by a multi-signature account.

Account recovery being affected by a multi-signature account.

A THORChain name being bought and owned by a multi-signature account.

Through the

Archived copy · cryp2nova.com

Æsir Protocol
more payment features can be easily added as uses for them arise.

Other Features

Collectibles

The ERC-721 standard of collectibles for Ethereum was the dawn of a new era of digital assets that can be collected and traded in marketplaces. A collectible, also known as a non-fungible token token, is a token that (typically) is not divided below 1 and each token has meta-data that can uniquely add value to the individual token. Each token is give an index and owners move individual tokens around. Some of the metadata that can be attached are names, descriptions and even resource links.

CRYP2NOVA

{

"
name
"
:
"
ODIN
"
,

"
description
"
:
"
Odin, ruler of the Æsir, father of Thor.
"
,

"
image
"
:
"
https://ipfs.infura.io/ipfs/QmPS7SbxcQoKAcFmTBxqm1c43sWztnFaPrCfgGDy3FhskN
"
,
}

Collectibles are different to tokens in that collectible trades happen much more sparsely and they normally have far few owners. Additionally, collectibles do not require continuous liquidity as individual items cannot be divided. As such it

Archived copy · cryp2nova.com

would be prudent to track all collectibles on a single discrete chain with address space
TCx<address>
. This will reduce blockchain bloat and allow the
collectibleChain
to maintain separate features to
tokenChain
.

On the creation of a collectible; the collectible is indexed inside the
TCx0
account and the initial details are set:

```
type
 collectible
struct
 {
  name
string

//
 String of the Collectible Collection "Æsir Gods"

  symbol
string

//
 Symbol of the Collectible "ÆGODS"

  supply
int64

//
  Number of different Collectibles in the Collection

  tokenURI   uint256
//
  A distinct Uniform Resource Identifier (URI) for a given Collectible

  owner
T0xBob

//
  Allows the owner (Bob) to details in future, such as name or URIs.

}
```

Once created all collectibles are emitted to the creator's address, allowing them to transfer further. The simple transfer of

Archived copy · cryp2nova.com

a collectible is:

Transfer
(address _from, address _to, uint256 indexed _tokenId);

All collectible and collection details can be read from
TCx0
account at any time. Further features such as minting, burning, pausing or blacklisting can be included in further interations of the standard.

Collectibles can also be bundled together and transfered in very large quantities. The modified transfer function includes an array to detail a batch transfer:

batchTransferFrom
(address _from, address _to, uint256[] _ids, uint256[] _values);

Messaging Protocol

Sending transaction receipts using assymetric key cryptography across THORChain allows private and end-end encrypted data transfer. This functionality allows THORChain to experiment with fully private and peer-peer on-chain messaging. Senders can send messages to the Receiver's address.

type
msg
struct
{
from_address
string

//
Address of the Sender, such as @alice or T0xaaa...aaa

to_address
string

//
Address of the Receiver, such as @bob or T0xbbb...bbb

msg_hash
string

//
hash.ReceiverPubKey(hash.SenderPrivKey(description))

Archived copy · cryp2nova.com

}

Smart contracts

THORChain may support smart contracts to enhance the ecosystem, but this is not in the primary development pathway. It will most likely integrated by on-chain governance. Any potential smart contract that benefits the protocol as a whole should be added as a native on-chain command instead. THORChain is not intended to be built to be a dApp platform; rather it is focussed on digital assets and the trading of those assets.

Additional On-Chain Commands

THORChain employs on-chain commands which are lightweight methods hard-coded into the protocol, and called by transactions. Compared with smart contracts, these native on-chain commands do not need Virtual Machine support and are part of the underlying protocol. Whilst smart contracts are generalist and need to be compiled to bytecode, these function as advanced transaction outputs/inputs. Each time a one is executed, the state of an account is updated in accordance with the script.
On-chain commands are built into the protocol as standardised methods; however once a standard is agreed upon, it is added into the protocol layer through on-chain governance. The following on-chain commands are tabled to be built into THORChain:

Security Tokens

Stablecoins

Collectibles (ERC-721)

Identity (ERC-725, 735)

Recurring Payments

Account Permissioning

Escrows

CRYP2NOVA

Archived copy · cryp2nova.com

THORChain Whitepaper

StableCoins

THORChains macro vision is to create a highly liquid payment network built on a completely trustless structure. An imperative feature for mainstream adoption are stablecoins that are pegged to existing fiat currencies such as USD, YEN, EURO and AUD. The first way to achieve this is to simply tokenise existing stablecoins, while the second is to support the development of stable coins on the platform. The first will naturally be easy if the bridges in
§ 8
are built.
THORChain has all the required features to create StableCoins with auditable supply and collateral using a Variable Supply token and its CLP and full Validator Set participation. The following would be the process:

A tokenChain
tUSD
is created by the Validator Set. A
Rune:tUSD
CLP is created that is variable supply, requiring
2 / 3
signatures from Validators, or by a on-chain smart contract.

At each round, Validators (on smart contract) propose a price
$
that the Rune is valued at. They can infer this by any means possible, most likely by reading APIs off a conglomerate of external exchanges of Rune. Validators can also nominate Rune pricing by watching Bitcoin pricing off external exchanges, and converting to Rune price by the
Rune:tBTC
price feed.

CRYP2NOVA

At each round
n + 1
the median of nominated prices is stored in the block. Outliers may invoke slashing rules to penalise poor price nomination.

As Rune price fluctuates,
tUSD
supply in the CLP is negatively changed by the Validators. As an example, Rune price decreases by
2%
, so
tUSD
Supply increases by
4%
. This causes
tUSD

Archived copy · cryp2nova.com

to become inflated and cheap.

Self-interested arbitrageurs will then send in Rune to buy cheap
tUSD
, until
tUSD
returns to
1:1
backed in Assets.

Additionally, the liquidity fee will increase the staked collateral in the CLP to match volume and reduce slip.

Figure: Collateralized on-chain StableCoins.

This process can be repeated for any other fiat currency and is very simple. Once an external price of Rune is set (in the fiat currency), then the protocol deliberately creates arbitrage opportunities by influencing money supply to attract third parties to act in a self-interested manner to restore any price imbalance. By requiring full Validator Set participation in the price nomination the price of the StableCoin can be relied upon with the same assurance as the entire protocol itself. The pricing/supply feedback loop can be modified to reduce damping and price sensitivity.

Token Baskets & Indexes

A THORChain can go further than a single token being represented in a CLP. With a combination of CLP scripts and trustless price feeds, token baskets and indexes can be created and represented by a single asset. This allows users and traders to carry a single token and know that the tokens price trustlessly represents other assets.

A new tokenChain
TKNIndex
can be created that represents
TKN1
,
TKN2
, and
TKN3
, where the
TokenData
and
CLPData
for
TKNIndex

Archived copy · cryp2nova.com

is linked to the CLPs of the indexed assets. The
TKNIndex
CLP is bonded to include liquidity in all other CLPs so the price of
TKNIndex
is thus representative of the linked assets.

Figure: Token Baskets

Conclusion

THORChain is a lightning fast decentralised exchange with protocol-level trading features to achieve feature parity with the best centralised exchanges of the day; all with full self-sovereign asset management. THORChain solves the fundamental problems of existing decentralised exchanges with a fast on-chain trading experience, on-chain continuous liquidity and correct incentivisation economics for exchange and wallet developers.
THORChain is built for a new class of decentralised exchanges and transactional networks and will rapidly increase the useability of transacting with crypto-currency assets.

References

Anon. (n.d.). Brainwallet. [Online]. Available at:
https://en.bitcoin.it/wiki/Brainwallet
.

Ethereum. (n.d.). Reduce ETH issuance before proof-of-stake · Issue #186 · ethereum/EIPsGitHub. [Online]. Available at:
https://github.com/ethereum/EIPs/issues/186
.

Interchain Foundation. (2017). Consensus Compare: Casper vs. Tendermint Cosmos BlogCosmos Blog, Cosmos Blog. [Online]. Available at:
https://blog.cosmos.network/consensus-compare-casper-vs-tendermint-6df154ad56ae
.

King, R. (2018). Binance Review - Complete Overview of Binance ExchangeBitDegree Tutorials, BitDegree Tutorials. [Online]. Available at:
https://www.bitdegree.org/tutorials/binance-review/
.

Archived copy · cryp2nova.com

Anon. (n.d.). POA Network. [Online]. Available at: https://poa.network/
.

Posner, E. (2014). Quadratic votingERIC POSNER. [Online]. Available at: http://ericposner.com/quadratic-voting/
.

Unchained, C. (2018). Cosmos Validator Economics-Bridging the Economic System of Old into the New Age of BlockchainsCosmos Blog, Cosmos Blog. [Online]. Available at: https://blog.cosmos.network/economics-of-proof-of-stake-bridging-the-economic-system-of-old-into-the-new-age-of-blockchains-3f17824e91db
.

Unchained, C. (2018). Proposed Cosmos Fee Token - Codename Photon Tendermint MediumMedium, Augmenting Humanity. [Online]. Available at: https://medium.com/tendermint/proposed-cosmos-fee-token-codename-photon-e0927daf5c4c
.

Unchained, C. (2018). The Technicals of Interoperability-Introducing the Ethereum Peg ZoneCosmos Blog, Cosmos Blog. [Online]. Available at: https://blog.cosmos.network/the-internet-of-blockchains-how-cosmos-does-interoperability-starting-with-the-ethereum-peg-zone-8744d4d2bc3f
.

Anon. (n.d.). What is Tendermint?¶. [Online]. Available at: https://tendermint.readthedocs.io/en/master/introduction.html
.

bitfly.at. (n.d.). Miner Block Gas Limit VotingEvolution of the total Ether supply. [Online]. Available at: https://www.etherchain.org/tools/gasLimitVoting
.

CRYP2NOVA

Archived copy · cryp2nova.com