Herdius - Next Generation Decentralized Blockchain Financial Infrastructure

Deme Balázs
Founder of Herdius
balazs@herdius.com

Abstract

Blockchain architectures and cryptocurrencies represent a generational innovation in financial system design. However, it remains to be seen how impactful these technologies are going to be. Present day financial blockchain architectures and decentralized exchanges still suffer from several shortcomings such as long transaction confirmation times, suboptimal scalability, a lack of decentralization and liquidity. In this paper, we present an innovative and robust blockchain, powered by an underlying PoS (Proof-of-Stake) algorithm that solves the above issues while still maintaining a cryptographic level of security (Croman et al). The resulting system, Herdius, is what we call a decentralized exchange platform.

Thanks to its innovative, performant architecture, Herdius will be able to meet many of the infrastructural needs of various participants in the token economy ecosystem. We envision that Herdius will become an integral part of the technology stack that powers the next generation of financial applications. Potential use cases range from decentralized financial apps built on top of the Herdius blockchain to financial services using Herdius as the gateway technology to connect their applications to the cryptocurrency market.

1. Introduction

The recent rise and success of Bitcoin (Nakamoto, 2008) and other cryptocurrencies, as well as highly innovative dApps (decentralized applications), has created a new economy worth over 204 billion dollars (Coin Market Cap, 2017). While recent growth has been promising, legacy blockchain infrastructure such as Bitcoin suffers from long confirmation times which result from a lack of scalability. As of now, no mass adopted protocol-level solution to this problem exists. Attempts such as SPV (Nakamoto, 2008) and Lightning Network (Poon and Dryja, 2016) have been great initiatives. However, adoption is slow and these architectures themselves introduce issues of their own. In order to sustain future growth and handle the influx of new users to the growing decentralized economy, better financial infrastructure has to be put in place. It needs to allow for faster and easier trading of tokens and cryptocurrencies – while also maintaining the security and traceability that blockchains offer.

In this paper, we outline our vision and preliminary architecture for what we consider will be a groundbreaking improvement in the cryptocurrency financial sector.

Current real-world blockchain networks suffer from fatal transaction scalability issues with most being limited at a practical rate of 30 transactions per second. We aim to address this issue by introducing a new, secure and highly scalable architecture that can make cryptocurrencies a viable payment option going forward. We believe that solving scalability needs to be approached through the usage of an efficient off-chain architecture which doesn't put unnecessary strain on the

underlying root chains. In order to realize a sidechain that securely keeps track of balances and transactions, we need to make full use of distributed storage, multi-party key generation as well as general distributed network technologies.

Moreover, our approach relies on an innovative and efficient blockchain architecture which we call Blocks-of-Blocks (BoB). The BoB architecture introduces two types of blocks to the Herdius chain: regular *singular blocks* and *stretched blocks*. The latter consist of a base block and multiple child blocks in a merkle tree-like structure (explained in Section 4.4). Finally, by introducing a distributed hallmark storage to the Herdius blockchain, we can significantly decrease the number of blocks that actively need to be confirmed during the validation period. Adding these architectural choices together will render Herdius significantly more performant than current legacy blockchains – without sacrificing security or decentralization.

The Herdius architecture was designed with the goal to build a high-performance, decentralized exchange. But that is far from the only use case we foresee. Herdius will also be able to connect different blockchains in the future. This will be outlined later in the paper when we introduce the transition layer which maintains crucial information within an efficient and public sidechain while relaying data and transactions across and in-between different chains.

Of course, any analysis is incomplete without findings from future alpha chain implementation benchmarks.

2. Overview of the system

When weighing different approaches to solving the scalability problem in blockchains, it has become clear that a new off-chain needs to be introduced which offers superior security and speed compared to the root chains. Herdius can be most easily thought of as an efficient and robust sidechain which contains different distributed elements. All these elements work together closely to create a secure and fast sidechain that can handle a huge transaction number.

At the core of Herdius is a blockchain that is capable of vertically stretching itself, thereby fitting and stacking several blocks on top of each other in the process. This stretching mechanism allows us to introduce parallelization to the validation process and is our solution to make transactions within the Herdius network fast and scalable.

Before blocks are created, a queue mechanism keeps track of all the transactions that took place within the network since the previous block. Based on this information, the constructor mechanism builds up a Merkle tree type block structure which carries so-called *child blocks* on its leaves. These child blocks all point to the *base block* and have a Null pointer at the end, preventing anyone from attaching additional blocks to them. This is necessary in order to prevent forks or parasite chains happening from these blocks. Child blocks are stacked on top of each base block and, therefore, can vertically stretch the blockchain size at block level. A huge number of child blocks can be stacked on top of a single block and each child block contains a cluster of transactions. Transactions in the Herdius chain can represent any transfer of digital assets, regardless of the underlying blockchain or architecture.

To reduce the scope of blockchain verification that needs to be conducted each time a new block is created, we introduce a *hallmark chain*. The hallmark chain acts as a distributed ledger for the Herdius blockchain and it's embedded data. At any single point in time, three blocks are maintained and actively verified on the main chain. Blocks on the main chain are moved to the hallmark chain in the order they have been validated in. Each block pushed to the Hallmark chain becomes a *legacy block*.

Each transaction in the Herdius chain will be validated by subgroups of staked validator nodes. Staked validators must hold and front HER tokens equivalent to a given transaction's value in order to be eligible to verify it. Staked tokens are locked until the block has been confirmed and was added to the hallmark chain. *Validator subgroups* are controlled and monitored by a second class of nodes, called *supervisors*. Supervisors carefully review the correctness of verifications done by the staked validators before the block is approved to join

the Herdius chain. They hold the right to detain a certain amount of staked tokens from validators if any misconduct or fraud has occurred.

The Herdius architecture is also capable of acting as a connection between different chains. By introducing a *transition layer*, users and entities will be able to attach additional data and information to any transaction within the Herdius architecture. The transition layer maintains all information referring to transactions and is accessible to the public at all times. By maintaining all this information in the transition layer, we make it possible for traditional miners or validators of different blockchains to refer to Herdius' transition layer for additional details that have been previously attached to a transaction.

3. Previous Work

In this section, we would like to mention projects and research that had an immense influence on the blockchain space and Herdius itself. Blockchain technology is currently a fast-paced field; new ideas and projects are envisioned almost every week. We believe that, going forward, it's crucial for ideas to mature and evolve as new research and solutions become available. While none of the works mentioned below are close to Herdius on their own, we believe that their design approaches are worth mentioning as they influenced Herdius.

3.1 Lightning Network and Payment Channels

Lightning network was the first huge step for Bitcoin to achieve scalability. At its time, it was the single biggest step forward for the blockchain space. It has propelled many other projects to come out and innovate. The introduction of Bidirectional Payment Channels as well as the network of nodes that make this network possible was a huge step forward in off-chain transaction handling. Perhaps the biggest issues holding back Lightning Network from mass adoption are the technicalities involved when using it as well as the limitations of the architecture itself. (Poon and Dryja, 2016)

3.2. Nxt

Nxt was one of the first projects to implement a Proof-of-Stake consensus mechanism. While the procedure we introduce in the Herdius chain is different from other Proof-of-stake mechanisms, we still employ concepts that correlate stake-size with rewards. What Nxt had in mind is very similar to one future goal of Herdius: to connect digital assets to the real world. The approach we employ at Herdius differs from "coloring" within Nxt, as we instead use the *transition layer* to keep track of coins across different chains (Nxt community, 2014).

3.3 Raiden Network

The Raiden Network is perhaps the most sophisticated and best-executed off-chain scaling solution to transact Ethereum-based ERC20 tokens. It is based on the above-mentioned Lightning Network. While it is a great architecture, it is not a step forward in our opinion. The user experience is far from optimal, because Raiden Network utilizes Hash-locked transfers and payment channels to transmit transactions back to the Ethereum chain. Time-locked contracts and payment networks are sluggish and require both parties to agree and transmit a transaction back to the root chain. Our approach, DIVIWA, is based on splitting private keys into smaller parts, thereby making wallets able to send transactions back to the root chain at all times through a key assembly process (Raiden Network).

3.4 Polkadot

Polkadot's robust architecture makes it potentially the project closest to Herdius in this list. Polkadot is a very elegant and intricately designed system that is intended to connect different blockchains. The main difference between Polkadot and Herdius lies in the respective scope of both systems. Polkadot - while also having general features for inter-blockchain connection - is mainly focused on its Parachain/Relaychain structure. With that, Polkadot aims to facilitate the creation of new, even experimental blockchains without each having to build out its own network. Herdius, meanwhile, is focused on cross-chain transactions and interoperability between established chains (Wood).

4. Herdius architecture

Herdius is composed of the following elements: the Herdius chain, the constructor/queue mechanism, hall mark storage mechanism, blocks of blocks architecture, validator sub-groups, supervisor nodes and the transition layer. These elements work hand in hand to maintain an efficient, distributed network with highest security standards.

4.1 Herdius chain

The Herdius chain is based on a Merkle Tree structure (Merkle, 1988). Each block contains the following information:

- Prevhash: The hash of the previous block
- State reference: The hash reference to the active preliminary state
- cBlock number: The number of child blocks included in the block
- cBlock root: The root of the dBlock trie
- Timestamp: Block creation date

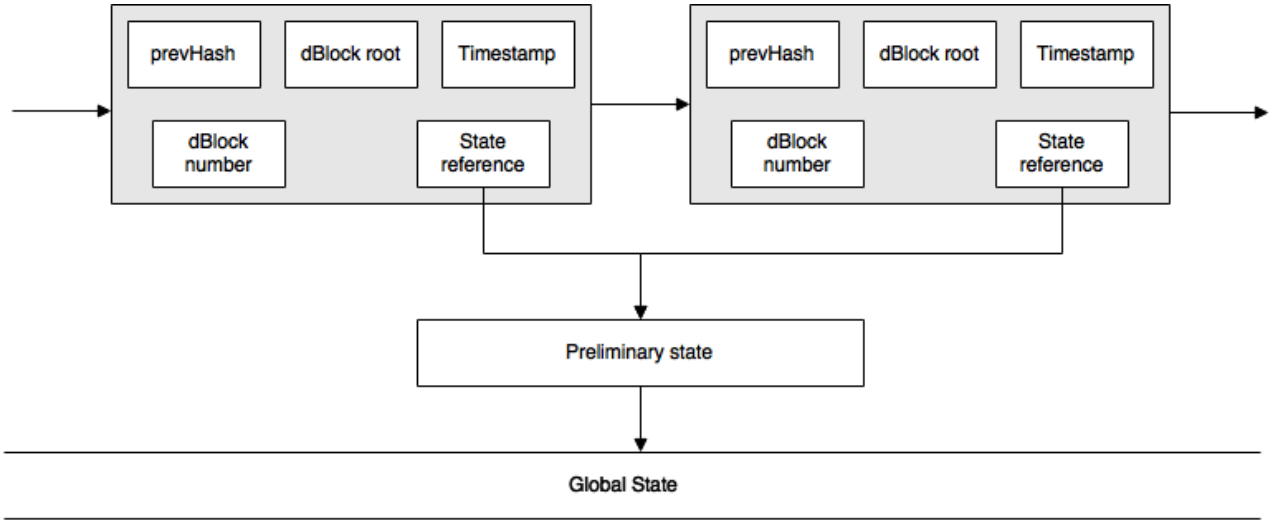

Figure 1: The construction of the chain and interaction with the different states

Blocks on the Herdius chain can be of two types: *singular* or *stretched*. Both block types can be interlinked and are created depending on transaction volume within the chain. In a period of transaction surge, only stretched blocks will be created. During a downturn in transaction volume, only singular blocks will be created. This ensures that no unnecessary computational resources are wasted and that there is an ample pool of validators to choose from at all times. The two block types have the following characteristics.

Singular: Singular blocks are simple blockchain blocks, similar to blocks within the Ethereum architecture (Wood, 2014). When compared to stretched blocks, described below, singular blocks simply have NULL pointers for the dBlock number and dBlock root fields.

Stretched: The stretched block structure is constructed by the Herdius *constructor/queue mechanism*. It builds up and creates the hash tree structure of the block that is being created next with all the child blocks acting as leaves of the base block. Stretched blocks can be scaled and stretched to a huge extent, provided that enough staked validators are available to handle the number of transactions on the network. It very well might be that, in the future, each stretched block acts as a hash table that maps child blocks to the base block and that each of these blocks includes hundreds or thousands of blocks itself.

The underlying consensus mechanism for Herdius is Proof-of-Stake. Therefore, it is important to note that blocks will be filled according to *transaction value*, not transaction number. The higher the value of a transaction, the higher the stake that validators are required to hold to qualify for staking said transaction. Because there is a maximum number of staked validators that can be assigned to an individual block, it is not practical to place any further transactions into the block.

Each individual validator stakes for a share of the total block value. The share is directly proportional to the percentage the validator's bonded HER tokens present of the total transaction value in the block. If, for instance, the transaction value of a given block is 10 BTC and staked validator Alice participates in the block with a total stake worth 1 BTC, then Alice is responsible for 10% of the block confirmation.
(On a system and user level, this doesn't make much of a difference. Since transaction fees will be represented as a minimal percentage of the transaction value, staked validators earn the same share regardless of whether the transaction has its own block or not.)

Moreover, it can be argued that the security of a transaction proportionally increases with the transaction's volume. The higher the value of a transaction, the more tokens need to be bonded by validators. In most circumstances, this will lead to a higher number of validators – which in turn reduces a single node's opportunity to misbehave.

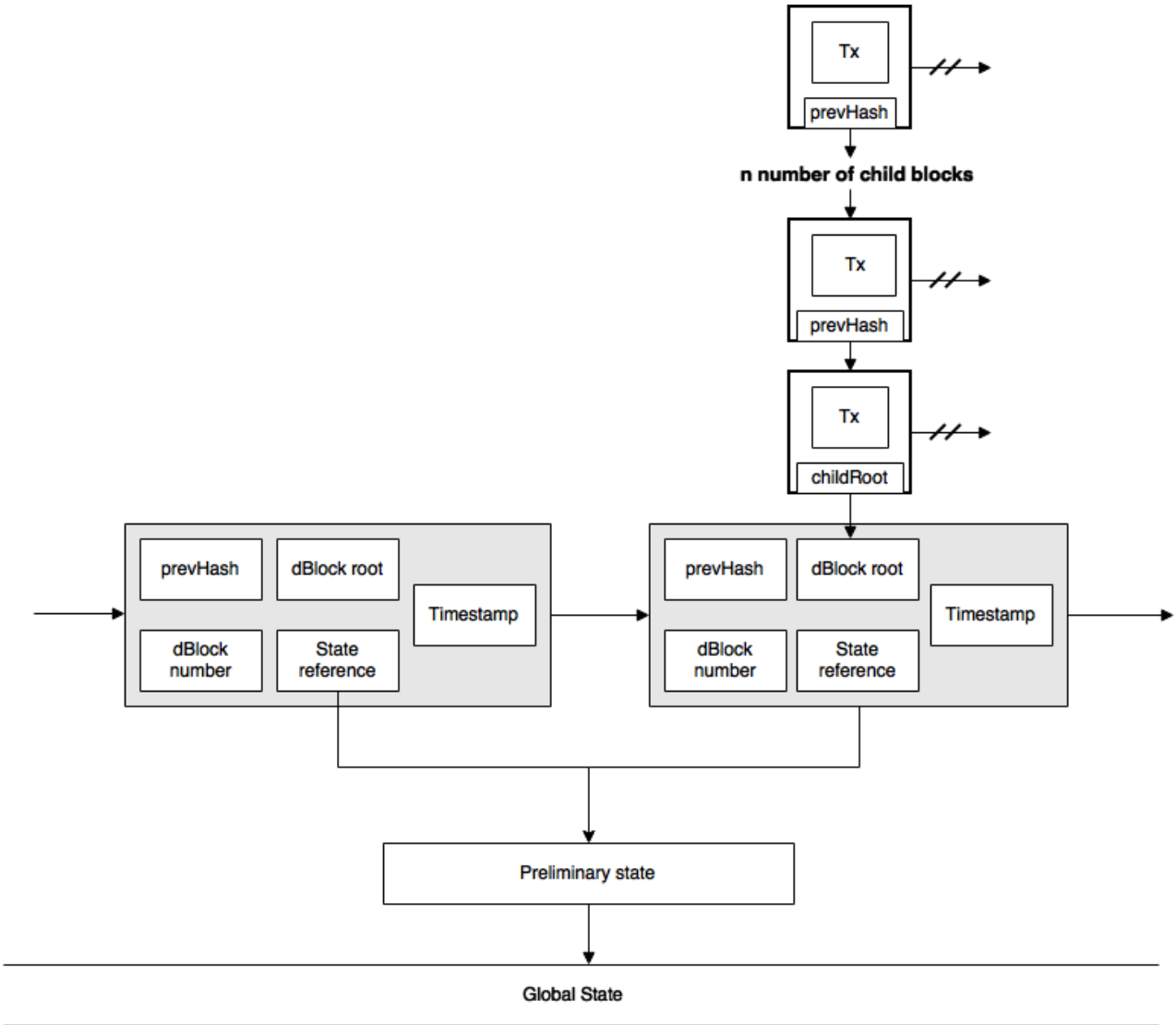

Figure 2: Connection of Singular blocks with Stretched blocks

4.1.2 Block Time & Stake Lock-in

Block time within the Herdius chain will initially be set at 5 minutes for *root blocks* (singular blocks or base blocks of a stretched block structure). For child blocks, the confirmation time will be three minutes. This time difference is necessary in order to allow for re-verification: *Supervisors* will have two minutes to inspect child blocks for traces of misconduct

from validators and, if necessary, to invalidate those transactions. If validators create invalid transactions, supervisors are responsible for reporting it.

The chances of fraud and misconduct are further minimized by locking-in the funds of staked validators until an additional three blocks are created, validated and added to the network. Thereby, supervisors have time to review transactions within a block until it becomes a *legacy block* within the *hallmark chain*. This gives the supervisors additional time to carefully review each transaction within a block, detect frauds, trace them back to the responsible validator, and take the necessary corrective action (even though it's after the conclusion of block validation).

The definitive configurations will be tested, measured and adjusted based on performance benchmarks before the final release of the Herdius chain.

4.1.3 Entity Highways

High block times are a serious area of concern for most legacy blockchains and one of the factors that hamper a wide acceptance and adoption of cryptocurrencies. Herdius has been designed to solve this problem. in order to create a fast way to transfer cryptocurrencies,
Herdius introduces the concept of *entity highways*. This instrument allows us to transfer the infrastructural requirement of transaction confirmation onto entities.

Instead of solely relying on public nodes for confirmation, any entity on the Herdius network can opt to do the required confirmation of incoming transactions and then broadcast this transaction to the network (including its own, unique co-signature). Once the entity receives the signed transaction and does the verification on their own part, it can release the product at once and transmit the transaction to the network at an instant. Thus, entity highways make transactions between users and entities fast and seamless.

A simple example to illustrate entity highways in practice: Alice wants to buy a product for 1 BTC from Bob's store which runs on the Herdius chain. At the checkout stage, Alice is requested to send 1 BTC to Bob's listed address. Alice initiates the payment and signs it with her key to validate the transaction. Bob verifies the authenticity of her signature and makes sure that she owns the transferred funds, just like a public node would (though the entity would collect only a nominal fee or no fee at all.) After completing this level of verification, Bob co-signs the transaction and transfers it to the network for approval from public validators. The validators cross-check the entity's signature with the entity register and verify Alice's signature and balance before adding this transaction to the latest block, thereby amending the Herdius chain.

Anyone can become an entity and receive a special key pair to sign transactions. Entities are stored in a distributed database decoupled from the Herdius chain. The distributed database is connected to the global state to provide for simple verification. It's not efficient nor a good system design choice to litter the chain and state with unnecessary data. Running an entity, however, is computationally expensive and every entity is expected to be able to do fast verification of transactions on its own.

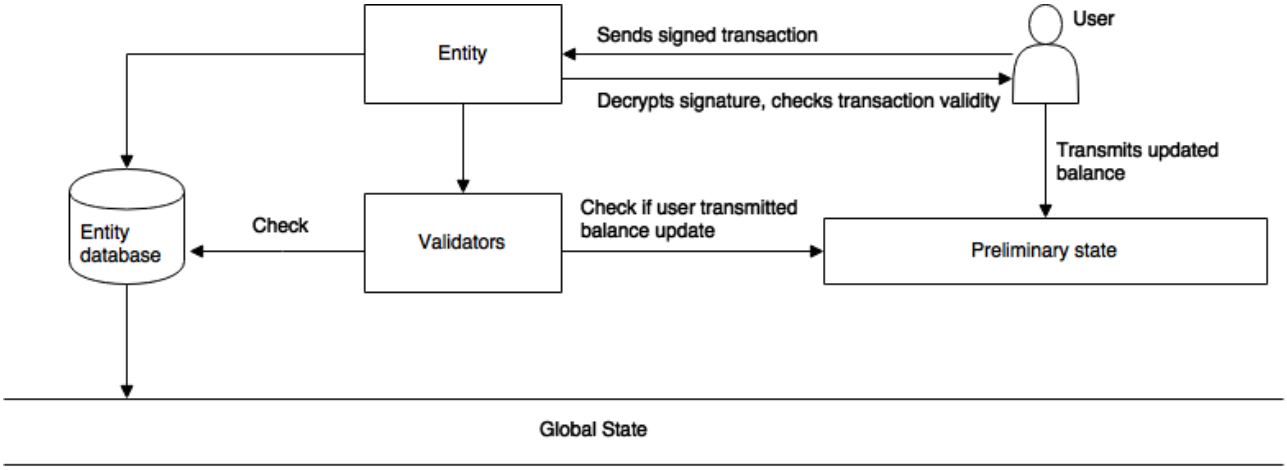

Figure 3: Entity Highway process

4.1.4 Payment Channels

Entity highways will be implemented at protocol level. To that end, they will utilize payment channels as previously worked out in the Lightning Network and similar architectures. Once a payment channel has been opened between a user and an entity, entities will only have to verify the balance of the user before transmitting a transaction as they are already aware of the user's identity from previous transactions. This cuts down on transaction time and makes the whole system work faster.

zkSNARKs are a great mechanism to keep the privacy of the transaction intact while still maintaining the same level of cryptographic security as public key cryptography. We believe that zkSNARKs are the optimal route to take when implementing entity highways. We want to preserve user privacy while still providing the network with proof that the transaction is valid without revealing the context of the transaction. While zkSNARKs are computationally efficient, we will have to test if they introduce additional risk and attack vectors before making a final decision upon implementation (Bowe et al).

4.2 Constructor/Queue Mechanism

The *constructor/queue mechanism* is the highest-level consensus mechanism within the Herdius system. It guides the creation of new blocks and assigns validator nodes to blocks.

The queue mechanism counts and keeps track of all bonded validators available in the network. Moreover, it tracks all unconfirmed transactions that will be included in the next block and forwards them to the constructor.

Based on this information, the constructor will define the structure of the next block – either stretched or singular – and divide validators into subgroups using a randomization process. Each subgroup will be responsible for confirming the batch of transactions included in the block assigned to them. The block design choice will depend on the number of available validators and the total value of pending transactions. It is better to disperse transactions into as many blocks as possible when enough validators are present: this increases security and reduces the confirmation time of individual blocks.

To avoid congestion within this mechanism, the Herdius chain will have a cutoff period. Cutoff period refers to the period after which the queue mechanism will not count transactions that are initiated in the last 30 seconds of the previous block being confirmed. The constructor/queue mechanism closely mimics the working of a Certificate Authority that governs the structure of the blockchain.

Eventually, it will be performed by public nodes and, hence, represent a distributed feature that is run by the network. The first, non-public version of the Herdius alpha chain will feature a central authority that is maintained by Herdius. But once the blockchain is fully ready, it's crucial to have this consensus mechanism running on public nodes through a Byzantine

fault tolerant (BFT) system which we call “The Table” (further explained in Section 4.2.1).

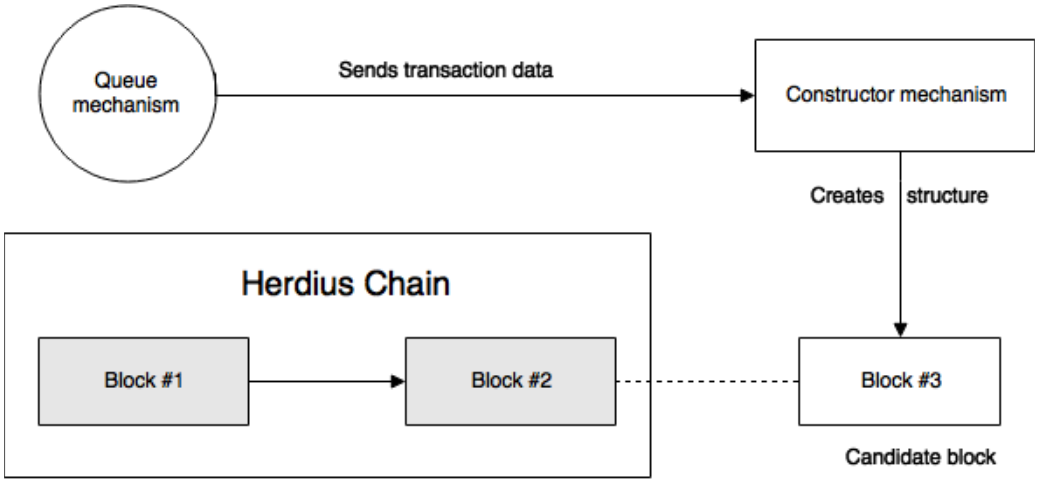

Figure 4: The operation of the Queue and Constructor Mechanism

4.2.1 “The Table” BFT Consensus Mechanism

“The Table” is our first vision of how the consensus mechanism within the constructor and queue mechanism would look like when running through public nodes. Because the control of that mechanism is critical for the network, it is important that only the most trust-worthy, proven participants of Herdius run this part of the system. Thus, it will be made up of nodes that earned the trust of the system (e.g. by employing a voting mechanism and/or relying on nodes with a significant stake). Also, we will likely create a waitlist which ensures that at least 100 or more participants are involved in decision-making at all times. This waitlist would be nomination-based, so every user could nominate nodes to be included in the mechanism. Including a higher number of nodes in the process additionally lowers the risk of nodes conspiring in order to create validator subgroups that favor them.

Our preferred Byzantine fault tolerant protocol will be closely related to HoneyBadgerBFT (Miller et al), an asynchronous agreement protocol that allows every node to nominate a certain path (=block structure) to follow, whereupon other nodes stake the decision based on their own calculation. In practice, all participants will track transactions happening on the network. Based on this, each node can propose a block structure to be followed. Others will stake different outcomes and the proposal with the highest placed stake wins and gets chosen as the eventual block structure.

Latency is an important factor when being part of this mechanism. Participants who miss transactions on the network will have bad predictions on the block design outcome and so will not receive the support of other participants. Thus, the overall decision is robust to occurrences of latency.

Once an agreement has been reached on the block structure, the division of validators into subgroups follows. To that end, The Table appoints the validator subgroups, instructs them about the terms of the agreed block structure and then apportions the responsibility among the validators.

4.3 Hallmark storage

Blockchains are a reliable way to store critical data. Yet, a big constraint of today’s legacy blockchains is the size they eventually reach. To ensure that the Herdius chain is capable of handling thousands of transactions, we have to make certain trade-offs. The biggest trade-off we intend to make is to reduce the size of the blockchain that validators actively have to confirm at each validation event. As a result, we introduce *hallmark storage* on the Herdius chain. The purpose of hallmark storage is to act as a distributed and decentralized storage network for storing the entire Herdius chain from to the genesis block onwards.

To that end, the Hallmark storage stores all the legacy blocks transmitted and signed by validators on the Herdius chain. A block

becomes a legacy block once the Supervisors have finished their job and all the transactions within the block are confirmed. Hallmark storage is also a solution to the block size problem as it reduces work load across the network.

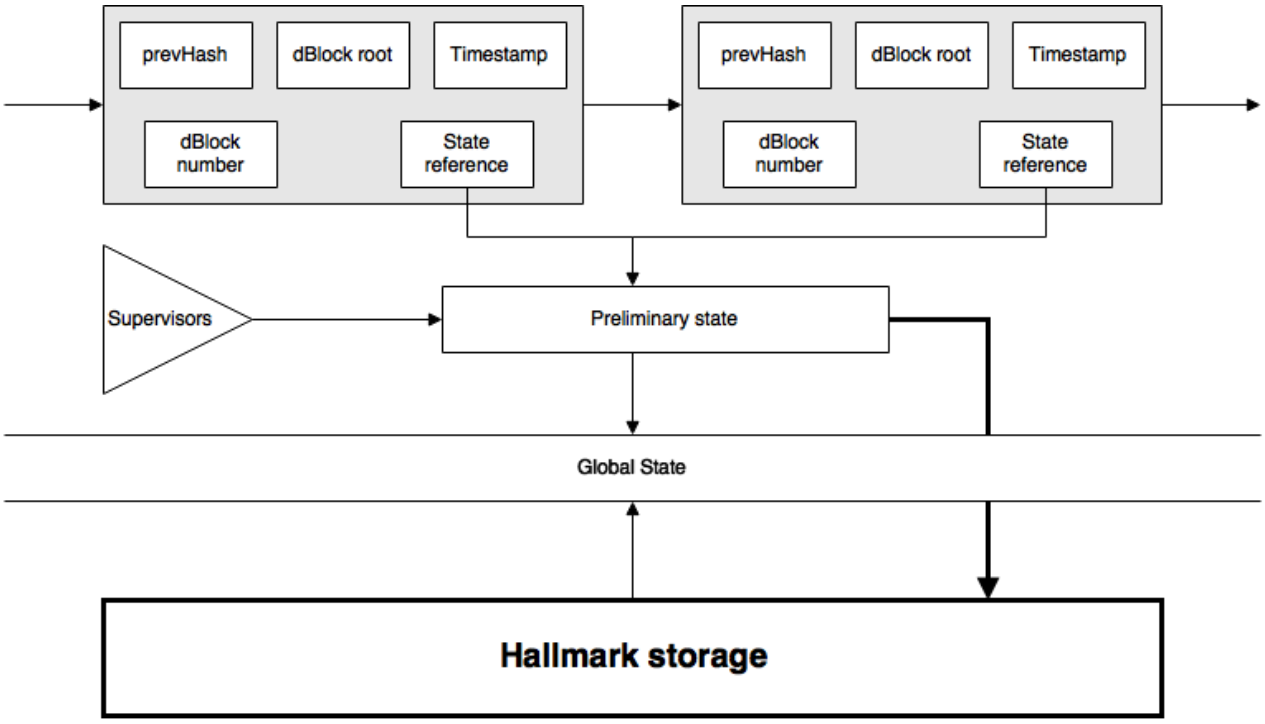

Figure 5: The Hallmark storage

The global state attached to each Herdius block is what carries vital account balance and general information forward within the system. This, however, can be traced back all the way to the genesis block by going backwards through the hallmark storage and re-checking the status of the state at every single block. Every node is incentivized to download, update and maintain the latest copy of the Herdius chain and the Hallmark storage. Similar to Filecoin (Protocol Labs, 2017) and other blockchain-based storage networks, all parties who maintain the Hallmark storage will be incentivized through receiving a small share of new tokens that are generated from each block that is about to be pushed to hallmark storage. As the blockchain grows, the rewards of hosting the chain increase. We believe that once Herdius can maintain thousands of transactions, hallmark storage hosting will become almost as profitable as being a validator within the network itself.

One of the most obvious advantages of hallmark storage is security. By having a distributed and constantly updated chain that is maintained by hundreds of nodes, it becomes way less likely that an attacker can create an efficient Sybil attack by creating a parasite chain. An attacker would have to convince all the hallmark storage nodes as well as validator nodes on the Herdius chain that his version of the blockchain is the correct one. When dealing with a robust and sizeable chain, this becomes almost impossible, especially when considering the rate of new additions to the blockchain every 5 minutes.

Speed is the second key factor for building the Herdius chain using hallmark storage. Unlike other blockchain architectures that use proof-of-work, Herdius already reduces computational wastage by relying on proof-of-stake. On top of that, by using hallmark storage, only 3 blocks have to be verified by validators at every single point in time. This equals a big reduction in unnecessary over-verification - which becomes increasingly expensive as the chain grows. Mature and popular proof-of-work based legacy systems clearly suffer from that today.

Finally, scalability also benefits from the introduction of a separately maintained hallmark storage. Block size becomes practically irrelevant: a higher block size does not put any additional strain on the blockchain

because the majority of blocks don't have to be actively verified at each validation event.

4.4 Child Blocks

Fundamentally, *child blocks* contain the same information as base blocks, except for the NULL pointer at the end of them, signaling that no additional blocks can be attached to that certain block. Child blocks are placed on top of each other, with the blocks in the bottom containing transactions with higher value and fee amounts attached to them. These blocks are prioritized over the others as more validators are needed to confirm the contents of the block. Each child block contains a nextHash field referring to the next block.

By stacking multiple blocks on blocks, we achieve parallelization at the individual block level. As all blocks are separated from one another, the transactions within them can be verified in parallel by unrelated groups of staked validators. Since each of these validator groups performs unrelated confirmation processes at the block level, validation happens without having to rely on other blocks being validated or not. Technically, the only limit on the parallelization of blocks is the number of validators present. The decision of how many blocks will be created ties back into the *constructor/queue mechanism* which is responsible for creating the block structure and assigning validators to each block.

The parallelization of block validation introduces a dangerous attack vector against the network. Double spending attacks are a regular issue within blockchains and the introduction of parallelized validation in Herdius' case introduces new risks. Our solution to this problem relies on the direct connection of the global state underpinning the blockchain to the transactions that are happening inside the network. To that end, we are introducing the concept of *preliminary state*. Essentially, the preliminary state tracks transactions and holds timestamped balances of the period between two block creation events. At every block creation event, the preliminary state is reset and starts at the official state according to the most current block.

An illustration: When sending a transaction, Alice first generates a random string that she encrypts using the preliminary state's public key. The transaction first goes to the preliminary state that decrypts the transaction and updates Alice's balance in the preliminary state itself. This process is followed by a sign-off by the state. The validators then receive the transaction with the respective signature of the preliminary state which signals the specific user's balance has been checked and updated by the preliminary state. So even if Alice tried to create three transactions in a very short timeframe, each worth her entire balance – which could end up in three child blocks that are being independently validated in parallel – only the first of these transactions would be validated as the preliminary state would not sign off the latter ones. Preliminary state is maintained by Supervisor nodes.

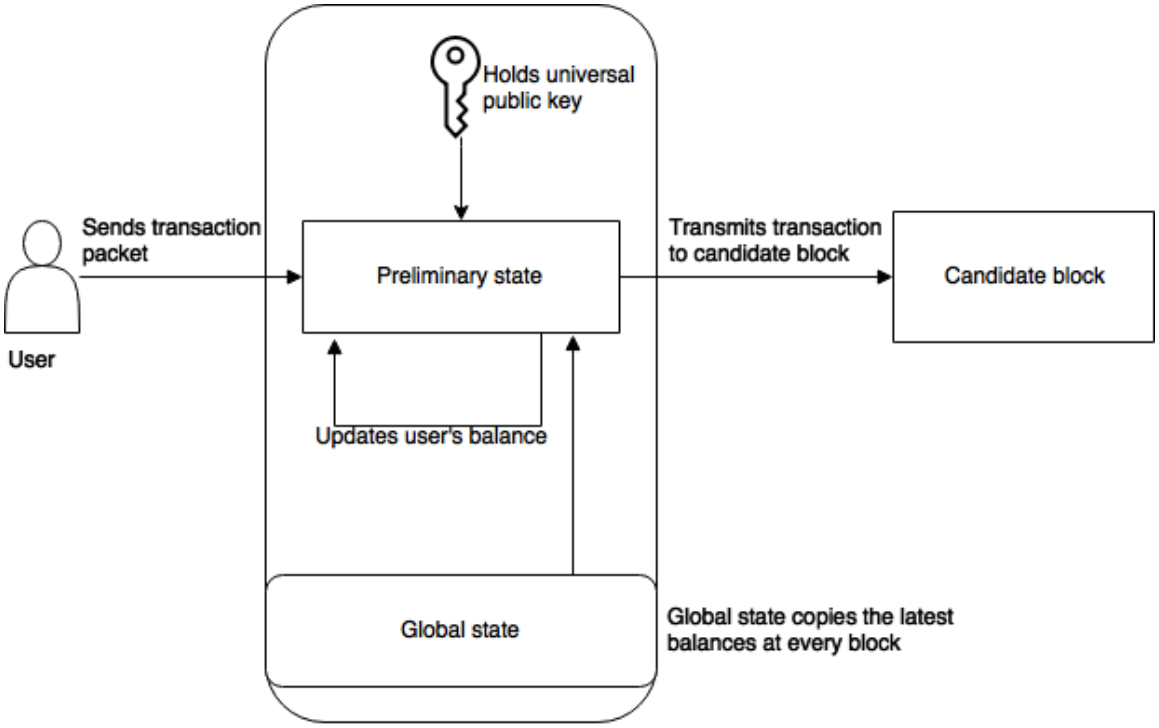

Figure 6: Prevention of Double-spending attacks through the Preliminary state

4.5 Validator subgroups

Validator subgroups are the “miners” within the Herdius architecture. They make sure that transactions are legitimate and that no foul play or double spending has occurred. Nodes that want to participate in the next block as a validator must

1) commit to performing the necessary work and
2) put up HER tokens as a locked-in bond which they are prepared to stake in the process.

Based on the staked amounts, the queue mechanism places validators into subgroups and assigns blocks to them. All transactions within Herdius require at least two separate validators to place stakes equivalent to the transaction amount. In the case that a transaction requires a significant portion of the available validator pool to participate, each validator can attest to only a certain share of the total transaction for which she is responsible for.

The actual validation process progresses as follows: the nodes

1. check block validity,
2. verify the correctness of the block,
3. test the correctness of hash values pointing to the previous block,
4. for each transaction, they check the balance of the sender address by looking it up in the current and preliminary state.

Since validators are risking their bonded stake in the process, they are directly incentivized to perform their work thoroughly and make sure that each transaction is valid.

The state directly pulls balances from the newest available block that has been confirmed. That means that each amendment to the blockchain directly impacts and updates balances of different users. Each validator subgroup involves its own decision-making process in the way that each validator validates transaction on its own and then transmits the results to the other members of the validator group. Other validators can dispute the verification of others, but a general agreement has to be reached before transmitting the block to the chain as final. It is the Supervisors’ responsibility to then check the submitted work of the validator subgroups once again.

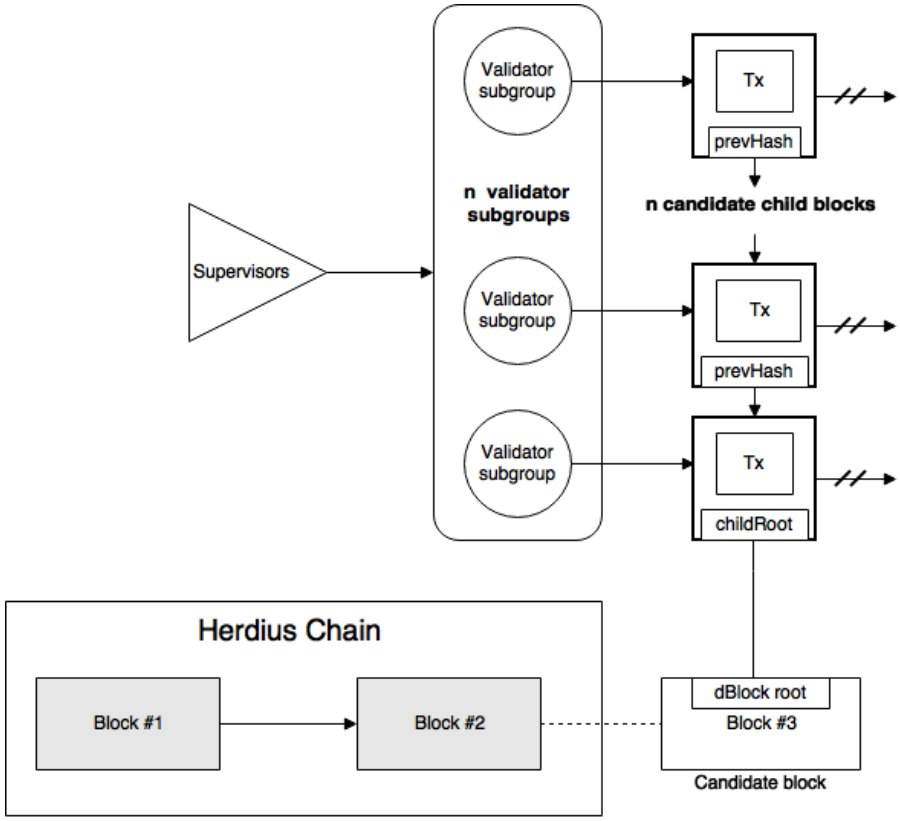

Figure 7: Block validation through Validator subgroups

4.5.1 Stake time-lock

Once a validator commits to performing validation on a block, the stake she puts up as collateral for the work will be locked-in three blocks in the future. The validator subgroups' data will be linked to the previous block, meaning that the network will reject any kind of outgoing transaction from that certain address unless the following is true:

Outgoing transaction value + staked amount ≤ the address's total balance.

The placed stakes are freed once three future blocks have been confirmed. This gives the Supervisors the time required to spot and catch misbehavior among validators. There won't be a staking limit present, so validators can commit all of their stake to verify transactions. Stake age, however, will play a role in the reward amount a validator receives. Rewards will be proportional to the staked amount as well as the number of blocks that the validator has successfully verified.

As far as stake age goes, there will be a multiplier lambda that offers a small, but not insignificant reward based on how many blocks the stake has been locked-in for. Stake age will have an upper cap which we currently plan to place at 60 blocks. Hence, the multiplier proportionally increases with each block that has been confirmed and then reverts back to 1 as soon as block 60 has been reached. This does not mean that each validator has to re-stake at every 60 blocks, but merely that stake age loses its significance. The reasoning behind designing the system this way is to always keep a fresh pool of validators and make sure that each validator re-commits and is actually available to do the confirmation work.

4.6 Supervisors

Supervisors play a crucial role in the Herdius chain as they are the final layer of security before a block is fully confirmed. The supervisors' role within the system is to catch and punish misbehavior by the staked validators. It is important that supervisors are fully independent of validators and that there is no way that the two groups can communicate or identify each other. This, however, means that any discrepancy detected has to be relayed to the staked validators as a group instead of individually.

For example, when a supervisor detects misbehavior, she is able to communicate this to other supervisors within the group. Once at least 2 additional supervisors confirm that fraud or misconduct occurred indeed, the whole group relays this information to the staked validators as well as to the highest-level consensus mechanism within the system - the constructor/queue mechanism.

Punishments are handed down to validators by the constructor/queue mechanism. Based on the severity of the offense, anything from partial to full stake revocation is possible. Validators found guilty are also marked as previously convicted when taking part in future validation sessions as well as barred from participating in validation for a prolonged period.

Supervisors, however, have their own set of punishments for raising false flags. Supervisors that are repeatedly raising false flags are warned and ousted at once should a false flag occur in the future. Supervisors who detect cheating are entitled to the full stake amount that is taken away from the cheating validator and this amount is added to their balance included in the next block.

Supervisors are also responsible for maintaining the preliminary state (see 4.4) between two consecutive block creation events. For that effort, they receive a share of newly generated HER from each block.

4.7 Transition layer

While the *transition layer* does not play a crucial role in the efficient operation of the Herdius Chain, connecting and interoperating different blockchains has been a long-discussed topic. The Herdius transition layer can be best thought of as a bridge that connects different chains while maintaining an accurate sidechain which contains relating information to each individual transaction.

When connecting different chains, problems usually arise from incompatibility in the underlying consensus mechanisms involved. It would be naive to believe and expect miners outside of the Herdius chain to maintain and keep track of incoming data outside of their own architecture. Thus, we propose a simple labeling mechanism where each individual asset is labeled so that receivers know that additional data belongs to the tokens. For instance, a Bitcoin transaction that is handled on the Herdius chain could include an invoice number – something that isn't possible natively in the bitcoin protocol – and always be retrieved from the transition layer.

This additional data can be claimed from the transition layer by proving ownership of the involved tokens. The simplest form of such proof will be the sending of a signed message from the wallet address containing the tokens to the transition layer. As a next step, the transition layer forwards the query to validators within the transition layer who verify the request's authenticity.

The transition layer, however, is more than just a simple bridge that connects the different chains. It can also act as a link between transactions and any kind of data, be it order numbers, text, or other sensitive data. In the process, it becomes possible to connect assets on the Herdius chain with distributed ledgers, private databases, private blockchains as well as any other data sources.

Several theoretical ways exist by which such a transition layer could be implemented. The critical passage from one chain to another could be most easily done by collaborating with other chains and setting up a whole new miner/validator segment within that blockchain that is solely responsible for handling incoming transaction coming from the Herdius chain. This would be the easiest solution at first sight, but this approach would introduce problems of its own, not to mention that it's highly unlikely that, at this stage, other chains will go out of their way to implement something similar. The main problem that arises is the interlinking of different data points across different chains and the backlinking of the transaction to the respective data stored in the Herdius chain.

4.7.1 Augmented data exchange protocol

The *augmented data exchange protocol* is the solution we foresee for the near future as something that can solve the problem of chain interlinkability. This data exchange protocol can co-exist next to the Herdius chain without interference as it is more of an add-on that introduces additional functionality to the root

chain. The augmented data exchange protocol will be especially useful for new financial applications as well as already existing decentralized applications that need a flexible backend infrastructure. For instance, a new e-commerce shop will be able to hash and store all order data in a distributed ledger and link it to a specific transaction through the Herdius chain. Decentralized insurance companies could track user credit scores across the chain or issue loans based on spending habits while storing this data in a hashed distributed ledger.

User data can be stored in vaults that preserve everything inside them in a hashed format. These vaults can be multi-signatory, meaning the data within can only be accessed by third-party applications if the user gives permission and any new addition has to be done through the whole process.

The whole purpose of the protocol is to create a thriving, data-rich ecosystem where a user's chain identity can be used to access all the different services. It is also a protocol that still maintains privacy, since every sensitive info is hashed and only services the user opts-in to can receive it.

5. Decentralized Exchange

The key use case that Herdius has been optimized for is the truly decentralized exchange of cryptocurrency and cryptographic assets. Unlike other approaches like Bitcoin SPV, Lightning Network (and similar payment channel solutions) or protocol-level solutions like Airswap (Oved and Mosites, 2017), our proposed decentralized exchange architecture solves the problem of underlying chain limitations.

It is important to note, though, that the proposed decentralized exchange platform is more than a traditional trading service. Off-chain transactions can be settled, and new services and infrastructure can be built on top of the system.

5.1 Peer-to-peer trading

The core of Herdius' decentralized exchange platform is its peer-to-peer trading functionality. Instead of solely relying on a centralized order book for matching orders across the network, Herdius enables peer-to-peer trading by transmitting and propagating orders through nodes within the network.

Once a matching trade order has been found, the order gets instantly fulfilled and the account balances are updated on the Herdius chain. Instead of paying fees per each transaction, there will be a percentage transaction fee imposed for the whole order. That way, if the first matching order has not fulfilled the full order amount, the rest of the order can be propagated further across the network.

An upper bound will be set on how many times a transaction is routed between the different nodes. As every hop in-between nodes introduces further time and latency problems, it is best to set a limit at which point the order is transmitted directly to the central orderbook system within the Herdius chain.

5.1.2 Order routing

Orders are routed using a gossip protocol (Demers et al, 1987) where each wallet receives and forward-propagates orders that come in. At every single point where a match is found, the order is executed, and the new order is routed on with the outstanding amount that has to be traded. Orders are routed per hop, outside of the Herdius chain, so orders don't stress the chain at all. As mentioned above, there is a hop-limit which shows up as a counter for trade orders. When an order is transmitted to a new node, the counter is increased and once the counter hits 10, the order is then submitted to the central order book.

5.2 The order book

The order book at the core of the decentralized exchange differs a lot from traditional decentralized exchange approaches. The Herdius order book should be imagined like a central hub that matches orders from all the makers as they come in. Thus, it is a collection of order information and has the purpose of order settlement and peer matching rather than anything else. The order book is most similar to an order-matching-party that instructs users about which other users to query regarding a specific order.

Matching up different nodes as well as their trade orders will be done using a communication system that provides bi-directional sender and receiver anonymity as well as outside unobservability. As low-latency is key, our proposed peer-order-matching mechanism will be based on the Loopix anonymity system (Danezis), a mix network-based architecture that uses Poisson mixing which is capable of independently delaying messages to further prevent third party linkability.
The order book also involves an anti-flood mechanism that prevents users from flooding the book with inexecutable orders. To tackle this problem, the book employs an oracle that collects prices through an API from trusted sources and only allows orders with a certain variance from those prices to be executed through the order book. In the early alpha chain stage, Herdius will maintain the orderbook on its own. However, for the full release it is planned to turn it into an entirely decentralized and distributed mechanism.

5.2.1 Linking other order books to Herdius

One current drawback of decentralized exchanges is the lack of liquidity available within them. In our view, the Herdius decentralized exchange can succeed only when it democratizes order books and provides significant improvement in liquidity to all ecosystem participants when compared to traditional exchanges.

We want the Herdius decentralized exchange to be used as backbone order book infrastructure that other exchanges can link their order books to – so Herdius will become a liquidity provider in the process. By creating dedicated APIs for order books, we want to allow other order books to link to the Herdius chain and use it as a reliable and fast solution for order fulfillment and settlement. This way, we hope to create scalable backend network for the cryptocurrency exchange space.

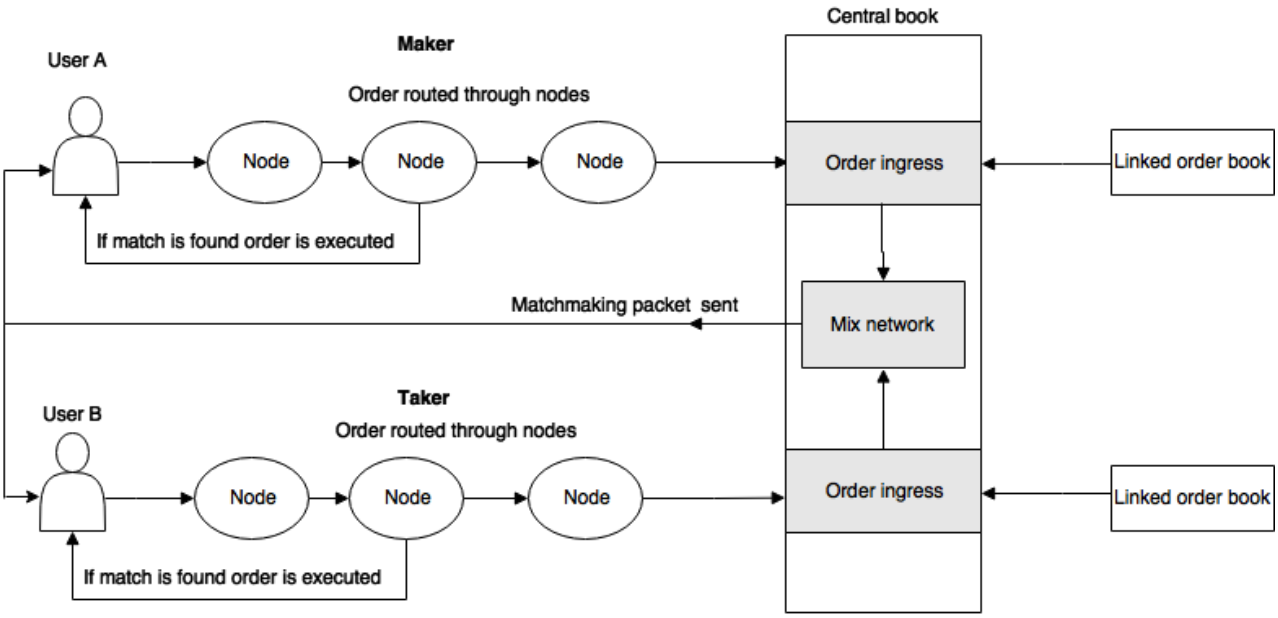

Figure 8: The Herdius Central book order matching

5.3 Virtual Wallet Network

Our decentralized exchange makes use of the fact that wallets are the underlying storage mechanism of each blockchain containing cryptocurrencies or tokens. Instead of going to the root chain for every single transaction, we do lateral transactions by making use of the Herdius chain. The virtual wallet network plays a vital part in this and the goal of the to-be-implemented system is to be just as secure, if not more secure than traditional offline wallets.

Once digital assets have been deposited to the Herdius wallet, the transaction capabilities are endless. Users can freely and securely transact as well as trade within the Herdius decentralized exchange network. Users can

also access their traded and received tokens instantly within their wallet without the need to withdraw them. Any already existing wallet can participate in this new network since the source code will be open-sourced from the start.

5.4 Keys within the system

Transactions aren't transmitted to the exchanged asset's underlying root chain when performing transactions within the Herdius decentralized exchange network. If, for instance, BTC is moved between Herdius users, this transaction won't be transmitted to the bitcoin blockchain. Instead, ownership of assets is maintained on the Herdius chain. Therefore, it's crucial to consider how to handle the private keys of the underlying assets.
The hard thing to solve here is that all wallets participating in the network have to be online and able to transmit transactions back to the root chain at all times, regardless of the original owner of the wallet being online or not. In order to solve this problem, we utilize Schnorr signatures (Savu, 2012) in a multi-signature setup to combine several signatures into a single one. Schnorr signatures are our preferred choice because they allow to aggregate multiple signatures into a single one while still keeping the final signature low in size. In the Herdius chain, all wallets will have their own private-public key pair which they use to sign transactions within the network. Each signature represents digital assets moved around within the Herdius chain.

By utilizing Schnorr signatures, it is possible for any user at any time to use the signature to transmit coins and currencies she owns back to the root chain. The Herdius chain, however, penalizes such outgoing transactions in the sense that it becomes more expensive to leave the network and transmit a transaction to the root chain than it is to transact within the chain itself.

At wallet creation, private keys are split across the network through a mix network that prevents traceability. Utilizing a mix network makes sure that no other node in the network other than the original wallet owner and split key holder know of each other. This way, other nodes in the network are incapable of performing a takeover attack where the compromise of individual nodes would lead to the theft of tokens. By utilizing Schnorr signatures in a multi-signatory threshold-optimal signatory scheme (Genaro et al), we make sure to mitigate the risk of node comptonization. An outgoing transaction from the Herdius chain towards the root chain requires an intricate key assembly. During this assembly, individual key holders would come together in the above-mentioned threshold signatory scheme and assemble the key which will then be able to move assets outside the Herdius chain.

6. Open-source architecture

The Herdius core technology and protocols will be fully open-sourced and, thus, be available to the public at large. We strongly believe that open-sourcing key distributed infrastructure is the best way forward for the blockchain ecosystem to flourish. Also, we think it makes a lot of business sense: By providing accessible and testable technology, we will have a big advantage when it comes to driving adoption of the Herdius network. Once the infrastructure layer of the Herdius system is running, the Herdius legal entity will be a regular participant in the ecosystem and use the Herdius architecture to build commercial applications on top of it.

We would also like to share our experiences and research moving forward. This space is only at its beginning, and only with joint efforts can we all make it move forward!

7. System Parameters and Community Involvement

Careful readers of this paper certainly realized that the specifics of several system parameters haven't been defined precisely thus far. The following parameters fall into this category:

- Transaction fee height
- Supervisor node incentive height
- Hallmark storage node incentive height
- The Table node incentive height
- Block reward algorithm parameters

These parameters' values are going to be critical to build a healthy, thriving ecosystem. The right structure and balance influences everything from performance to attractiveness

to ecosystem participants on all levels. Therefore, we refrained from publishing (and thereby setting) precise values so far. The Herdius system is relatively complex. While our initial analysis gives us a good idea about adequate ranges for all these parameters, we strongly feel that in order to make a solid decision, it will take detailed modeling of the system and feedback from test environments.

Therefore, we decided to take a different approach than simply defining the parameters outright. Instead, we will build models and run tests with different configurations, publish the results and give HER token holders a say in the final decision.

This comes with an additional benefit: In comparison to traditional software projects, public blockchain systems are heavily reliant on the respective ecosystem participants and their support of critical design decisions. Once the system is live and running, it gets increasingly more complicated to make changes to the system's core while maintaining an active, happy community. Therefore, any critical design decision needs the utmost buy-in.

By opening the design choice up to Herdius ecosystem participants - and not opting to set the parameters as a central decision maker ex ante - we hope to set a solid baseline and precedent for the way we envision the Herdius community to operate.

References

Bowe S., Gabizon A. and Green M. D. (na). A multi-party protocol for contructing the public parameters of the Pinocchio zk-SNARK. Retrieved October 20, 2017, from https://github.com/zcash/mpc/blob/master/whitepaper.pdf

Coin Market Cap. (2017). Cryptocurrency Market Capitalizations. Retrieved November 13, 2017, from https://coinmarketcap.com

Croman K., Decker C., Eyal I.,Gencer A. E.,Juels A., Kosba A., Miller A., Saxena P., Shi E., Sirer E. G., Song D., and Wattenhofer D. (n.d.). On Scaling Decentralized Blockchains. Retrieved October 20, 2017, from http://www.comp.nus.edu.sg/~prateeks/papers/Bitcoin-scaling.pdf

Danezis G., Piotrowska A., Hayes J., Elahi T. and Meiser S. (na). The Loopix Anonymity System. Retrieved October 20, 2017, from http://www0.cs.ucl.ac.uk/staff/G.Danezis/papers/loopix.pdf

Demers A., Greene D., Hauser C., Irish W., Larson J., Shenker S., Sturgis H., Swinehart D. and Terry D. (1987 August). PODC '87 Proceedings of the sixth annual ACM Symposium on Principles of distributed computing. Canada: Vancouver.

Gennaro R., Goldfeder S. and Narayanan A. (na).Threshold-optimal DSA/ECDSA signatures and an application to Bitcoin wallet security. Retrieved October 20, 2017, from https://eprint.iacr.org/2016/013.pdf

Merkle, R. C. (1988). Advances in Cryptology — CRYPTO '87. Germany: Springer.

Miller A., Xia Y., Croman K., Shi E. and Song D. (na). The Honey Badger of BFT Protocols. Retrieved October 20, 2017, from https://eprint.iacr.org/2016/199.pdf

Nakamoto S. (2008, November). Bitcoin: A Peer-to-Peer Electronic Cash System. Retrieved October 20, 2017, from https://bitcoin.org/bitcoin.pdf

Nxt community (2014, July). Nxt Whitepaper. Retrieved October 20, 2017, from https://www.dropbox.com/s/cbuwrorf672c0yy/NxtWhitepaper_v122_rev4.pdf

Oved M. and Mosites D. (2017 June 21). Swap: A Peer-to-Peer Protocol for Trading Ethereum Tokens. Retrieved October 20, 2017, from https://swap.tech/whitepaper/

Poon J., Dryja T. (2016, January). The Bitcoin Lightning Network: Scalable Off-Chain Instant Payments. Retrieved October 20, 2017, from https://lightning.network/lightning-network-paper.pdf

Protocol Labs (2017 July 24). Filecoin: A Decentralized Storage Network. Retrieved October 20, 2017, from https://coinlist.co/assets/index/filecoin_index/Filecoin-Whitepaper-df0c8d36f93bee60ab9a3c7565faece5d37fa615f78a20c7b0e8075686f1ee53.pdf

Raiden Network (na). Raiden Network documentation v0.2.0. Retrieved October 20, 2017, from http://raiden-network.readthedocs.io/en/stable/index.html

Savu L. (2012). Signcryption Scheme Based on Schnorr Digital Signature. Retrieved October 20, 2017, from https://arxiv.org/pdf/1202.1663.pdf

Wood G. (na). Polkadot: Cision For a Heterogeneous Multi-Chain Framework. Retrieved October 20, 2017, from https://github.com/w3f/polkadot-white-paper/raw/master/PolkaDotPaper.pdf

Wood G. (2014). Ethereum: a Secure Decentralised Generalised Transaction Ledger. Retrieved October 20, 2017, from http://gavwood.com/paper.pdf