Mikron Technical Paper

v1.0, November 2018, Adam Rotaru

Contents

Overview

Mikron is a blockchain-based system targeting community building, loyalty program and micro-donations within online communities. Its token is the Mikron digital currency, capable of fast and fee-less transactions. Mikron is a decentralized network, based on the novel block-lattice architecture, and it uses Delegated Proof-of-Stake consensus. It is based on the Nano project, with a few notable changes.

The Mikron network is designed to be practical for micro-transactions: frequent, small amount transfers. This is enabled by the fact that there is no transaction fee, and account chains can be updated asynchronously with low latency.

Mikron nodes are lean with low resource requirements, and as consensus is not based on proof-of-work, there is no need for power-hungry mining equipment and no waste of energy.

Differences to Nano

Mikron is based on the Nano digital currency, with fast and no-fee transactions. Nano (formerly named RaiBlocks), with its innovative block-lattice architecture, was chosen as the basis for Mikron based on several criteria: low fees, fast transactions, ability for micro-transactions, and clean implementation.

Mikron is based on Nano implementation, but it is a separate, unrelated network instance.

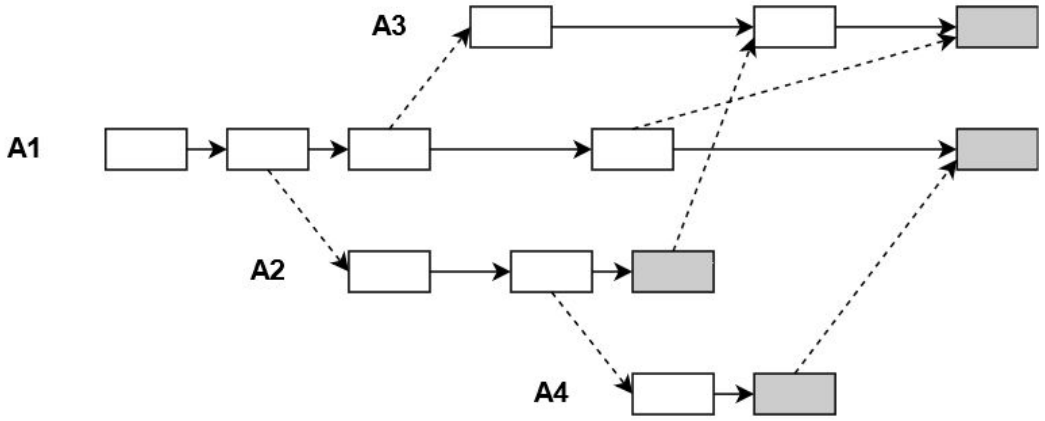
The most notable *differences to Nano* are listed here:

- Continuous generation. In Mikron the available supply is continuously growing (see later sections on manna creation and token economics).
- Different balance representation. Mikron uses 64-bit values for balance, which is less than in Nano. This representation size is more implementation-friendly (i.e., no extra libraries needed), while still offering a large enough range and resolution.
- Block creation times. In Mikron each block contains block creation times; this is missing in Nano. Hash-protected creation times ensures reliable history tracking.

Naturally, there are a host of other smaller, technical differences (account format starts with ‘mik_’ prefix, legacy block types are not supported, etc.).

Block-Lattice

Mikron has one blockchain for each account, controlled by the private key of the account, and each blockchain is replicated to all peers in the network. Transactions between accounts create connections between the account chains. This arrangement is called a block-lattice (a form of directed acyclic graph, DAG).



Distributed agreements like Proof-of-Work or Proof-of-Stake are unnecessary, since the account owner has authoritative control over transactions.

Wallets, Keys, Accounts

In the Mikron network, the *wallet* is a basic entity, enclosing a *seed*, a *public-private keypair*, and one or more *accounts*. The *seed* is a secret string, from which the keypair and the accounts can be generated deterministically. Therefore the seed should be handled as a secret. A wallet can be restored from the seed. A new wallet can be created any time (from a random seed).

The public and private keys are used for signing messages (blocks). The public and private keys are 32 byte long, typically shown as hexadecimal numbers.

Each wallet has a first account, but additional ones can be generated. The generation is deterministic, in the spirit of Hierarchically Deterministic wallets. Accounts from the same wallet are seemingly unrelated.

Account addresses are 32 byte numbers, they are exactly the same as the pubic key (i.e., not a hash of the public key). Account addresses are formatted with base58 encoding, with the '**mik_**' prefix, in 64 characters. A sample account address is:

```
mik_11ndz68qwn53uh1fwfbm6n9zemb4xeun6dd6oy1813tfap5361tq43ot7cma
```

Wallet seed and private key should be treated as secrets, as they allow controlling the wallet. Wallets should be protected by password also.

Balances

Amounts and balances in the Mikron network are represented as 64 bit unsigned binary numbers (unlike in Nano), with 10 decimal digits reserved as fractional. Thus the smallest non-zero amount is 0.0000000001 Mikron, while the largest amount is around 1.844 billion Mikrons.

The smallest fraction is also called one *Ant*. The range of possible Ant amounts is 0, 1, … , 18446744073709551615 (= 2^64-1).

Account-chain

An account-chain is a blockchain that belongs exclusively to one account. A blockchain has the usual general features of a blockchain, but in a somewhat simpler form due to the one account restrictions. Important features are:

- An account is a linear chain of blocks
- Every blockchain has a first block (a receive block from another chain, except the special genesis block)
- The currently last block in a chain is called *frontier* block
- All blocks belong to one account
- Each block consists of one operation, no multiple transactions per block
- Each block has a link to the previous block (except the first), by including its hash
- Each block is signed, with the private key of the account.

Transactions

Transactions are the transfer of an amount from a sender account to a receiver account.

In Mikron, transactions are:

- between two accounts,
- require acceptance by the receiver,
- are unconditional, and
- there is no transaction fee.

Block types

Currently there is a single block type used, called a *state* block, but other types may be added later. Note that the Nano network originally used several block types, which were later unified in the state block, but the original legacy types are also supported. This is not the case in the Mikron network, the legacy block are not used at all.

The state block is used for various purposes, and a logical type can be defined accordingly:

- *open genesis*: the very first block of the block-lattice, *all* blocks are derived from it. It has an initial balance, representative, and no links.
- *open receive*: the first block of a chain. It has a balance, a representative, and a link to a send block of another chain.
- *send*: a send to another account. It has a final balance (not the transaction amount), and a destination account.
- *receive*: a receive from another account. It has a final balance (not the transaction amount), and a link to a send block on another account, in the form of its hash.
- *change*: used to change the representative, all other fields are the same as in the previous block.

Common block contents

The important information content of blocks are described above. Fields present in all blocks are described here in more detail:

- Type. Currently only state block is used
- Owner account (32 bytes)
- Hash of the previous block (32 bytes; null for first blocks)
- Creation time (4 bytes). Its representation is standard unix time, minus the constant 1535760000. In other words, the number of seconds since Sept. 1, 2018 00:00 UTC.
- Representative account (32 bytes). The current representative of this account.
- Final balance (8 bytes). The balance of this account, right after present block. Format is unsigned int, in raw Ant units (described in detail later). For transactions, the amount is not explicitly stored in the block, but can be computed as the difference between balances of the send block and its previous block.
- Signature. The signature generated from the block data and the private key of the owner account.
- Work. Anti-spam proof-of-work.

In addition, a state block also contains:

- Link. The usage of this data fields depends on the logical subtype: for send it is the destination account, for receives it is the has of the matching send block. For other blocks it is unused.

Note that for practical reasons, the type, account, previous, representative, and balance fields are present in every block, this these can be obtained for frontier blocks promptly.

Inter-chain Transactions

Transactions are the only form of connection between two account-chains. A transaction consists of a send block and the receive block.

The *send block* is in the chain of the sender account, and contains the account of the receiver. The send is immediate and irreversible; the amount is immediately deducted from the sender's balance, and no longer available.

The *receive block* is in the chain of the receiver account, and contains the hash of the matching send block. Only after the creation of the receive block is the amount credited to the receiver, thes the transaction is *settled*. Creating the receive block for pending transfers happens automatically, right away if the recipient's wallet is online, or later, when it is started (first incoming pending transfers are noticed, then checks are performed, then the receive block is created and broadcast). Creating the receive block can happen much later than the send, and in the transitional period the amount is called *pending*, the transaction *unsettled*. If the recipient does not create the receive block, it may occur that a sent amount remains pending indefinitely (the amount is lost).

Signing

Each block is cryptographically signed to prove its creator. The signature is derived from the contents of the block, the private key of the signee (owner account), and a signing algorithm. A modified ED25519 elliptic curve algorithm with Blake2b hashing is used (same as in Nano).

Hashing

A hashing method is defined to obtain a hash for each block, from the contents of the block. Hashes are used to ensure the integrity of a block, and for indexing blocks for retrieval. A hash is stored in 32 bytes. The hashing algorithm used is Blake2b.

The hash of a state block is computed from the following fields: account, previous, creation time, representative, balance and link.

Anti-spam Proof of Work

Mikron does not use Proof-of-Work for consensus, the way it is done Bitcoin and many other cryptocurrencies. However, PoW is used for preventing spam -- the malicious flooding of the network with bogus blocks. Each block has to contain a small-difficulty proof of work in order to be included. The difficulty threshold of the work is constant and relatively small, it takes about a few seconds to generate it for a new block. This way flooding becomes very costly.

Genesis

All blocks in the block-lattice are connected in a tree structure, its root being the *genesis block*: the very first block to which all transactions can be traced back. The genesis block is the first block of the special *genesis account*.

The genesis block was generated on Oct. 31, 2018. It contains the following important parameters:

```
account: mik_1yumkjq6xscowmmz6sca79sycoup1tpqabnc3zdqgpcd1756xgo1k53z7yeg
balance: 3000000000000000000    (300M)
time:    2592000                (Oct 1, 2018)
hash:    1B79F5821B6F6034AFA93527071F05009B727DC36B0370959DEC5D9DD0EB1199
```

Manna

The Mikron network contains *continuous generation*, a novel feature for a block-lattice architecture. New *'manna'* coins are continuously being created at a fixed rate.

Due to continuous generation the Mikron economics is inflationary. Manna is created in the genesis account, at the fixed rate of 82,000 Mikron per day (around 30 million per year). Manna is materialized only when there is a transaction on the account (send or receive), at that time the account balance is increased with the manna amount computed since the time of the last crediting.

Manna rules are included in the node software, with cooperating nodes consenting about them. The fact that all account balances are traced back to the genesis account makes it impossible to alter these rules by non-cooperating nodes.

Consensus

The benefit of the simple account blockchain setup is that intra-chain consensus is trivial. However, the intra-chain consensus adds a new dimension.

The following mechanisms exists in Mikron:

- Intra-chain: as one account controls the blocks in a single account-chain, block ordering and transaction are no problems.
- Inter-chain confirmation: Block from other account-chain are confirmed by other nodes. Here the account representatives play a role.
- Voting. Voting is used in the rare case of *forks*, conflicting block candidates following the same previous block. Votes of representatives are weighted by the total sum of represented accounts.

See the Nano documentation for more details on the consensus algorithm.

Representatives

For certain operations the node managing an account-chain must be online and accessible. This is an impractical restriction for many short-period nodes: for example, when sending a transaction to a recipient, the sender node would need to be online until the recipient accepts the transfer (which may happen only days later).

To ease these constraints, the concept of *representative* or *delegate* has been introduced. An account can name another account as representative, typically an account that is known to be online most of the time. The delegate node can verify blocks and participate in votings on the behalf of the represented account.

The representative account can be changed by a state block (of logical type *change*).

The Mikron wallet uses a set of default representatives for new accounts. Representation is a free service offered by the Mikron team for any user. However, the representative of an account, or the default representative for future accounts can be changed at any time, to any account. Importantly, the representative should be accessible continuously, for the proper working of the represented accounts. Using a diverse number of representatives helps to increase the decentralization of the Mikron network.

Validation

A summary of the conditions for a valid new block are listed here. These conditions are checked at each new block, and only valid blocks are accepted into the block store.

- blocks needs to have a valid signature
- for existing accounts:
    - previous is filled, and previous block exists and is valid
    - creation time is later than of the previous block
- for send blocks:
    - link is set (to destination account)

    - balance delta is negative
- for receive (and open receive) blocks:
    - link is filled, and points to an existing and valid block
    - creation time is later than of the linked block
    - balance delta is equal to the balance delta of the linked send block
- blocks needs to have a valid work proof.

Mikron Network

The Mikron network is the set of cooperating nodes implementing Mikron transactions. It is a peer-to-peer, decentralized network, consisting of nodes, in agreement over the blocks that form a block-lattice. It handles transactions between nodes, with modest resource requirements.

P2P Network

The Mikron node is a P2P application, connecting to other nodes, and exchanging blocks with them.

Mikron uses a streamlined protocol, with small messages/blocks, based on UDP (as opposed to TCP). The default node listening port is 7042. Initial peer discovery uses some well-known addresses (node.mikron.io).

RPC

The Mikron node exposes an RPC (remote procedure call) interface, used for wallet, REST service, or other type of integration.

Beta Network

In addition to the main *live network*, there is a second network instance, called the *beta network*. Beta net is used for testing, with features identical to the live net, but different protocol identifiers, default port numbers, and network bootstrap nodes. A Mikron in the beta net has no real value.

Token Economics

Details of the Mikron Token economics is not in the focus of the present technical discussion.

The initial genesis amount (300 million Mikrons) has been created with the prepared and well-known genesis block. The supply model is inflationary, as continuous generation of 'manna' produces around 30 million Mikrons per year.

Limitations

The Mikron network can support simple transactions, but its was not designed for more complex, smart-contract capable transactions. No multi-account or multi-signature transactions are possible. Transaction comments are also currently missing (it is planned to be added later).

Even though there are a separate chain for each account, full nodes download and validate *all* blocks from all chains. As the network will grow larger and older, it will become more important to have 'light nodes' that do not download all the existing chains, only the ones that are interesting. Such an implementation is currently not available.

Software Suite

Mikron software is open source, available on GitHub.

The Mikron software suite includes the following components.

Node. The official Mikron Node software, written in C++. It is currently available for Windows and Linux.

Wallet. The official Mikron wallet is a Windows desktop application, developed in a JavaScript framework (electron). The wallet includes a background node instance, and connects to it through RPC.

REST API Service. A REST API layer built on top of the RPC interface of the node, used in partner site integration.

Partner Integration

In a Mikron-based loyalty or reward point system, the award points can ultimately flow into the Mikron network. Several different models are possible for the Partner-Mikron integration, especially regarding the storage of award credits within the Partner system or in the Mikron network.

In this section we present the concerns, and an analysis of different models.

Concerns for the Partner-Mikron integration are:

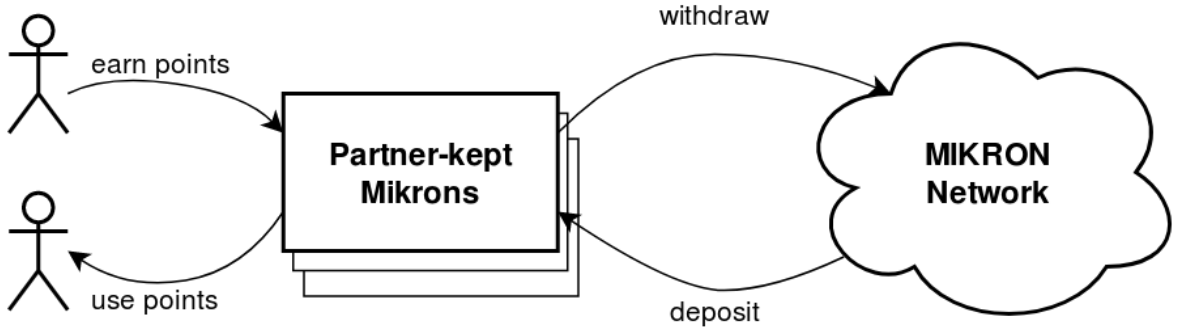
- Level of control / sense of ownership by the users over their points. Credit stored in user's wallet means more sense of ownership than a credit with a third party (although it also comes with more responsibility).
- Onboarding efforts of low-commitment users;
- Level of trust needed from users towards the Partner -- the spirit of cryptocurrencies is to minimize required trust;
- Easy of use of using the points within the Partner system;
- Easy of use of frequent, microdonation / user content reward usage;
- Ease of use of transfer of points between different Partners (where allowed);
- Complexity of implementation / integration efforts;
- Security aspects, especially regarding handling of private keys.
- Possibility of revoking reward points, if desired (e.g. forum moderation, excluded content).
- Business goals of the Partner, e.g. regarding point interchangeability.

Based on these, the following models are discussed.

Points in the Mikron network directly

While the Mikron network is capable of handling microtransactions, and integration is relatively easy, this is a plausible solution technically, and also innovative. However, it has two main drawbacks: each participating user has to perform some steps (set up a wallet), and usage of the points (or microtransactions) are not very practical. Therefore this model was not chosen in the first Mikron integrations.

Points in a Partner database, with withdrawal/deposit option



Awarded points are maintained in a traditional database by the Partner, with the option for the users to withdraw to (or deposit from) the Mikron network. This is a more conservative approach, as points are initially present in a traditional database. However, users can 'withdraw' their balance to the Mikron network if they wish, where they have complete control over it. Power users can control their Mikron, and transfer them as they wish, while low-commitment users can still participate in the reward program without special actions. In this model withdrawal from and deposit to a Partner is similar how it is usually done in a crypto exchange.

Hybrid model of partner storage up to a limit, with automatic transfer above

In this hybrid model reward Mikrons are stored with the Partner, but amounts in excess of a limit (e.g. 1000 Mikron) are automatically 'sent out' for users who prefer this (and provided a destination Mikron account). This flexible model is a good compromise between simplicity and option for more trusted setup.

Additionally, there is also the option for awarding points directly to the Mikron network, but having the Partner control the private key for these user accounts to simplify buying/giving. However, this option combines the drawbacks of technical integration overhead with the increased security risks, therefore this model is not recommended, and not discussed in more details here.

Glossary

Ant	The smallest raw unit of Mikron is called Ant. One Ant represents 0.0000000001 MIKRON, and 1 MIKRON is 10000000000 Ants.
block-lattice	The directed acyclic graph of blocks, organized in linear chains and inter-chani connections.
DPoS	Delegated Proof of Work, consensus algorithm.
frontier	The set of last blocks from all active blockchains.
manna	Name of the continuous coin generation feature, and of the generated amounts.
Nano	The digital currency that served as a starting point for Mikron.
P2P	Peer-to-peer.
PoS	Proof of Stake, consensus algorithm.
PoW	Proof of Work.
REST	Representational State Transfer. HTTP-based API style.
RPC	Remote Procedure Call.
TCP	Transfer Control Protocol, internet protocol.
UDP	User Datagram Protocol, an internet protocol without delivery guarantee. Mikron uses UDP packets for P2P communication.
wallet	Abstraction for securely storing digital currency. A Mikron wallet encloses one or more accounts, together with the associated key pairs. Also used to the software with UI implementing it.

References

J.-P. Aumasson, S. Neves, Z. Wilcox-O'Hearn, and C. Winnerlein, "Blake2: Simpler, smaller, fast as md5," 2012. [Online]. Available: https://blake2.net/blake2.pdf

Colin LeMahieu, "Nano: A Feeless Distributed Cryptocurrency Network" 2017. [Online]. Available: https://nano.org/en/whitepaper

S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system" 2008. [Online]. Available: http://bitcoin.org/bitcoin.pdf