



Kairos Chain

The proof-native Layer 1 blockchain

VERSION 1.0 · OCTOBER 2026 · KAIROSCCHAIN.ORG

ABSTRACT

Blockchains ask every new participant to trust history: replay it, or accept someone else's snapshot of it. That cost grows with every block. Kairos Chain removes it. Each block carries a recursive proof attesting that the block is valid and that the proof before it was valid, so a single proof attests to the entire chain from genesis. Proof of work is retained only to order blocks; it cannot create value or validate an invalid state. Ownership uses zero-knowledge proofs of a secret instead of signatures, so no public key ever appears on chain. The proof system is built entirely on hash-based cryptography over binary fields, with no trusted setup. Kairos is an independent network forked from the open-source Parano1d protocol, launched with its own genesis, with no developer allocation in its block reward, and with a one-to-one migration of the \$KAIROS community token into native KAIROS at genesis.

1 The cost of history

A Bitcoin full node validates the present by re-executing every transaction since 2009. Ethereum clients replay or accept a checkpoint. Solana nodes join from a snapshot supplied by a peer. In each case, a newcomer either pays a cost that grows linearly with the age of the chain or delegates trust to someone else. We call this **replay debt**: the older a chain gets, the less practical independent verification becomes, and the more verification concentrates in a few operators.

Kairos inverts the dependency. Rather than history proving the present, the present proves history. A node joining at any height receives the live state, the latest proof, and a short suffix of recent headers, authenticates them, and becomes a fully independent verifier. Peers supply data, never authority.

2 Design overview

Kairos separates two questions every blockchain must answer. **Is this block valid?** is answered by a proof. **Which valid block comes next?** is answered by proof of work. The work of producing a block is divided accordingly:

1. The **wallet** proves private authorization: knowledge of the secret behind each input, bound to the transaction.
2. The **miner** proves the transaction logic, the exact state transition, and the validity of the previous block's proof, in one recursive proof.
3. The **miner** then freezes the block template and searches a 128-bit proof-of-work nonce.
4. The **network** verifies the proof and the work, then applies the writes.

Because the proof is completed before the nonce search, hashpower alone cannot produce a block. A miner on Kairos is a proving full node backed by hashpower.

3 Recursive history

Let T_h be the terminal proof at height h . The **HistoryStep** relation proves, inside one circuit, that the block at h is valid with respect to the post-state of $h-1$, and that T_{h-1} verifies:

$$T_h \vdash \text{Valid}(B_h, S_{h-1} \rightarrow S_h) \wedge \text{Verify}(T_{h-1})$$

By induction, verifying T_h establishes the validity of every block from genesis to h . The terminal's size and verification cost are constant in h : about 0.9–1.0 MB, verified in roughly one second on a laptop. Headers are kept outside the recursion and checked for cumulative work; a hard 18-block finality window bounds reorganizations.

4 The proof system

Every cryptographic object in Kairos, including addresses, authorization, transactions, state commitments, recursive history and proof of work, uses one binary proof stack.

4.1 Fields and hash

Execution traces are committed over the binary tower field $GF(2^{128})$; Fiat-Shamir challenges and recursive claims are drawn from $GF(2^{256})$. Multiplication uses the carry-less multiply instructions present in modern CPUs. The single hash is **Poseidon2b** over $GF(2^{128})$, width 4, S-box x^7 , 8 full and 58 partial rounds, separated by typed domain tags.

4.2 FROST-GKR

Block proofs are dominated by repeated Poseidon2b permutations and Merkle paths. FROST-GKR (Frobenius Reduction Over Shifted Tables) places every slot, round and lane of these computations on one Boolean hypercube, proves the x^7 relation in a single sumcheck (squaring is linear over a binary field), and links adjacent rounds through shifted tables. On a 59-permutation benchmark this reduces 472 product-chain sumchecks to 2, with a reported 10.7× faster prover and 51.7× smaller raw proof than layer-by-layer GKR.

4.3 Commitments and recursion

Relations are closed with batched sumcheck, zerocheck and lincheck, then opened through FRI-Binius / BaseFold polynomial commitments. There is no trusted setup and no elliptic-curve or pairing assumption anywhere in the stack. The verifier's Fiat-Shamir transcript is compiled into constraints so that the previous terminal can be verified inside the next one.

WORKLOAD	PROVE	VERIFY	TERMINAL
Small block, 63 payments	17.6 s	0.83 s	913 KB
Large block, 206 payments	42.6 s	2.02 s	981 KB

TABLE 1 · UPSTREAM MEASUREMENTS, INTEL I7-1365U, 12 THREADS

5 Live state

Kairos stores live outputs, not lifetime history. State is a sparse Merkle vector of 2^{24} to 2^{32} slots; each leaf records an amount, a monotonic creation identifier and an owner. Spending clears a slot, and new outputs reuse free slots with fresh identifiers, so a stale reference cannot spend a new occupant. When sustained occupancy reaches 75%, the domain doubles without moving any record. A chain's age therefore adds no storage or verification burden for a new node.

Transactions are built from fixed 323-byte **Tx8x2** pages of eight inputs and two outputs. A **PagedSpend** joins up to 128 pages into one atomic transaction with one identifier, one fee and one authorization capsule. A sender can export a portable receipt, a Merkle inclusion path checkable against the canonical header, which remains verifiable after the block body is pruned.

6 Ownership without signatures

A Kairos address is the Poseidon2b image of a 256-bit secret. To spend, the wallet produces a freshly randomized, witness-hiding zero-knowledge proof of knowledge of that secret, bound to the logical transaction identifier and input owner. No signature and no public key is ever published. The authorization proof is independent of the state root; the miner proves that the inputs are live.

A wallet may derive its master secret from a photograph: the image is decoded locally to canonical pixels and hashed with BLAKE3 in derive-key mode. Changing a single pixel yields a different wallet. This is a usability feature, not a privacy one: Kairos is transparent, and addresses and amounts are public during consensus.

7 Contracts

A live output may commit to a program, spending authorities, recipients, height conditions and two persistent 64-bit counters. The integer core has sixteen instructions with checked arithmetic and no loops, timers or cross-contract calls, which keeps each call cheap to prove. Templates cover refundable payments, timelocked vaults, delegated allowances, per-period budgets, recurring payments and tranche releases. The Pruv language compiles readable programs to the core and verifies declared postconditions with an SMT solver.

8 Consensus parameters

Blocks target a 30-second mean with ASERT difficulty (6-block reference epoch, 180-second half-life). The proof-of-work hash is a domain-separated Poseidon2b sponge over sixteen header elements including a 128-bit nonce; a block is valid when its digest is strictly below the target. Fork choice follows the greatest cumulative work,

ties broken by the smaller tip hash. Blocks come in two proof classes, Small (63 pages) and Large (206 pages), both verified by every node.

9 Economics

Native KAIROS has six decimals. After genesis, all new KAIROS is created by block rewards, and **100% of every block reward goes to the miner**: Kairos removes the upstream protocol's developer allocation entirely. The reward starts at 16 KAIROS per block and steps down every 1,051,200 blocks (one year at the target block time) to a permanent tail of 1 KAIROS per block:

16 → 11.30 → 8 → 5.65 → 4 → 2.83 → 2 → 1.41 → 1 KAIROS (tail)

The eight declining years issue 53,810,928 KAIROS; the tail adds 1,051,200 KAIROS per year thereafter, keeping block production paid indefinitely. Transaction fees are 5,000 μ KAIROS plus 100 per input and 700 per output, plus a state-growth fee per net-new slot that scales from 1 $\times$ to 8 $\times$ with occupancy. The state-growth component is burned, so filling the state has a real cost while consolidation remains cheap.

10 The \$KAIROS token and genesis migration

The Kairos community forms before its network does. \$KAIROS launches as a Solana token on pump.fun. Before mainnet, a snapshot time is announced in advance and a migration portal opens in which holders bind their Solana wallet to a k1 Kairos address. Bound balances are written into the genesis block as migration allocations, one \$KAIROS to one KAIROS, and the genesis allocation file is published so that anyone can check it. The token exists to carry the community into the chain; it does not represent equity, revenue or any claim beyond that migration.

11 Security and limitations

The upstream protocol publishes an executable soundness certificate assessing the design at NIST post-quantum Category 1, conditional on stated premises: Poseidon2b

modeled as an ideal random oracle, proximity-gap conjectures for the codes used, and an idealized compiler. It is an accounting of known failure events, not a machine-checked proof, and the implementation has not been independently audited. Further stated limitations:

- Header validation and state transfer still grow with chain length and the live set, respectively.
- Proving costs roughly 16–18 seconds per block on a laptop, regardless of how many transactions it contains.
- Throughput figures are capacity calculations, not measured sustained throughput.
- The ledger is transparent; it is not a privacy chain.

12 Provenance

Kairos Chain is an independent fork of Parano1d, an open-source proof-native Layer 1 designed by Ignatus Nemo and released under the Apache-2.0 license. Its binary-field and sumcheck foundations derive from binius64 (Irreducible) and hekate-math (Oumuamua Labs), and its Ligerito commitment from bolt-rs (Bain Capital Crypto, Succinct). Kairos keeps every original copyright and NOTICE attribution. It is not affiliated with Parano1d, and KAIROS is unrelated to the NOID coin.

References

1. Parano1d. Proof-native Layer 1 ordered by proof of work. github.com/proof-native/parano1d (Apache-2.0).
2. Ignatus Nemo. FROST-GKR: Frobenius Reduction Over Shifted Tables. github.com/proof-native/frost-gkr.
3. Irreducible / Binius developers. binius64. github.com/binius-zk/binius64.
4. Grassi et al. Poseidon2 and binary-field Poseidon variants.
5. Bitcoin ABC. ASERT difficulty adjustment algorithm (aserti3-2d).
6. Zeilberger, Chen, Fisch. BaseFold: efficient field-agnostic polynomial commitments.