PROTOCOL RESEARCH v2.1 mainnet · H279370

Parano1d v2.1. Native Finance

Backed tokens, a launchpad, permanent DEX liquidity and an NFT marketplace. Every operation settles inside the recursive block proof.

Ignotus Nemo 04 Nov 2026 7 min read

ABSTRACT

Parano1d v2.1 represents token supply, redeemable holder backing, curve positions, DEX reserves, NFT ownership and listings as authenticated live State. The Native proof class checks their exact transition rules together with ordinary payments. Fixed-supply tokens carry at least 0.001 NOID of backing per whole token. The standard launch curve allocates 60% of supply to trading and 40% to permanent DEX liquidity, with atomic graduation. Sequential trades respect each policy's limits, and NFT purchases couple payment and delivery in one transition. The wallet reads current financial state from its own node. Live rights remain in authenticated State as recent block bodies follow their retention schedule.

Native finance in the block proof

Parano1d v2.1 runs backed tokens, a launchpad, a permanent-liquidity DEX and an NFT marketplace inside the State transition checked by the recursive block proof. Token quantities, holder reserves, market balances, collection counters and listing terms are live protocol objects. Their exact inputs and outputs enter the same atomic transition as NOID payments.

The v2.1 profile is active from mainnet height **279370**. Its Native proof class checks the financial rules directly. A token creation deposits its backing, establishes its fixed supply and initializes its market together. A swap updates the market and pays its recipient together. An NFT purchase transfers the item and pays its seller together.

The wallet exposes these operations through the Tokens and NFT workspaces. Current balances and market state come from the user's own node. The sections below describe the running protocol's asset rules, trading mechanics and block resource limits.

Fixed supply and redeemable backing

A fungible token has six decimal places and a fixed initial supply of at least 100,000 whole tokens. Creation deposits at least **0.001 NOID per whole token** into its holder reserve. Creating 100,000 tokens therefore requires at least 100 NOID of backing, plus the network fee.

Holder backing and trading liquidity occupy separate State objects. The backing reserve serves token redemption. Curve cash and DEX reserves serve trading. Every outstanding token counts toward the backing obligation, including tokens held by the curve or pool.

Burning tokens redeems their proportional share of the holder reserve in the same operation. For reserve R, outstanding supply S and a burn of q base units:

```
payout = floor(R × q / S)
remaining reserve = R − payout
remaining supply = S − q
```

Integer rounding leaves any fractional remainder in the reserve. Backing per remaining token is nondecreasing. The wallet quotes the redemption and network fee separately, so the user sees the net NOID outcome before submitting the burn.

The creation rules fix supply and reserve custody. The available creator operations preserve that fixed supply and the holders' redemption rule.

The launch curve and automatic graduation

The standard launch allocates **60% of supply to the curve** and **40% to DEX liquidity**. Let S_0 be the initial supply in token base units and q the quantity sold by the curve. Its required NOID cash is:

$$C(q) = \operatorname{ceil}\left(\frac{3S_0q + 5q^2}{900S_0}\right).$$

Cash is measured in microNOID. Both assets have six decimal places. The marginal-price reference starts at **1/300 NOID per token** and reaches **1/100 NOID per token** at graduation. Accepted trades use exact integer differences in the cash function.

The creator can include an initial purchase in the creation transaction. That operation deposits the full holder backing, places the purchase cash into market custody and delivers the purchased tokens atomically. The initial purchase can cover any amount through the full curve allocation.

Standard launch allocations. Holder backing uses the minimum reserve requirement. Network fees are quoted separately.

Token supply	Holder backing	Curve tokens	DEX tokens	DEX NOID at graduation
100,000	100 NOID	60,000	40,000	400 NOID
1,000,000	1,000 NOID	600,000	400,000	4,000 NOID
5,000,000	5,000 NOID	3,000,000	2,000,000	20,000 NOID

For a one-million-token launch, a creator buying the entire curve allocation supplies 1,000 NOID of minimum holder backing and 4,000 NOID of curve cash, plus the network fee. The creator receives 600,000 tokens. The operation establishes a DEX pool containing the remaining 400,000 tokens and 4,000 NOID.

Reaching the curve allocation creates the permanent pool automatically in the same transition. The token retains its asset identity and holder balances. A buy that crosses the graduation point can spend its remaining budget against the new pool. Unusable dust returns to the buyer.

Permanent liquidity and exact swap rules

The initial DEX reserve ratio matches the curve's final marginal price. Its pool shares are permanently locked. Trading fees accrue inside the pool, and swaps update its reserves under the protocol's integer arithmetic.

For input reserve *x*, output reserve *y* and gross input *g*:

```
fee = ceil(30 × g / 10000)
effective input = g − fee
output = floor(y × effective input / (x + effective input))
next input reserve = x + g
next output reserve = y − output
```

The swap fee is **0.3% of gross input**, rounded up to a base unit. Every accepted swap preserves positive reserves and a nondecreasing reserve product. The network fee is a separate charge. A crossing buy pays the DEX fee on its DEX portion. Curve execution uses the curve cash function.

The permanent lock applies to liquidity shares. Actual pool balances and market prices follow accepted trades and accumulated fees.

Several trades in one block

A trade policy fixes its asset, recipient, gross input, minimum output, maximum network fee and inclusion-height range. Exact funding references identify the resources available to that trade.

For the selected set, the relation derives priorities from the immutable parent and each policy's principal funding input. It orders those priorities and chooses the next side according to the current price relative to the block's initial price. Buys receive preference at or below that reference. Sells receive preference above it.

Each trade executes against the previous trade's resulting curve position or reserves. The proof checks the full selected sequence, intermediate prices, funding and final payouts. This permits several buys and sells in one block while checking each participant's individual limits.

Producers choose inclusion. The selected set and available funding identities can affect the result. Each accepted fill must satisfy its recipient, amount, minimum-output, fee and expiry constraints. An unfilled policy leaves its funding unspent.

Pending quotes use the local node's known candidate set. The confirmed block fixes the execution sequence and exact outcomes. The wallet retains pending operations and receipts across restarts.

Collections and atomic NFT purchases

Each collection has an issuance counter and a full 64-bit serial range. The creator can mint additional batches into the same collection. Every mint consumes and updates the counter. A burned serial remains issued.

The wallet supports standalone items, collections, batch minting, transfers, burns and marketplace listings. The simple Create NFT flow prepares a one-item collection and then mints its item through two separately reviewed transactions. The collection workflow supports up to 122 items in one mint transaction with one NOID funding input, subject to the actual shared resource budget.

A listing fixes the exact item, seller, asking price and expiry. It includes a **0.1 NOID listing credit**. A purchase consumes that listing, transfers the item to its buyer and pays the seller in one State transition. The seller receives the asking price and listing credit.

Cancellation returns the NFT and remaining credit after the network fee. Competing purchases consume the same listing incarnation, which makes their claims conflict on the exact live object.

The wallet and authenticated live State

F2 opens Tokens. My Tokens, Launchpad, DEX and Create provide balances, discovery, trading and issuance. **F3 opens NFT.** My NFTs, Marketplace, My Listings and Create provide collection and item operations.

The node supplies current token balances, curve positions, reserves, ownership and listing terms. Token and collection IDs bind their definitions to the exact funding anchor used at creation. NFT identity combines the collection ID and serial. Names and tickers are descriptive metadata. The full asset ID identifies the actual token.

Images, descriptions and optional HTTPS websites have committed metadata. The wallet checks retrieved manifests and image hashes against those commitments. Optional external data supplies media and historical views. Current financial state remains verifiable through the user's own node.

Authenticated State synchronization carries current live openings and their required definitions. Recent block bodies follow the retention schedule. Live rights remain represented in State, and exported receipts remain checkable against canonical headers. Wallet receipt exports preserve the records a holder chooses to retain.

One selected proof class per block

The authenticated bank provides Small, Large and Native classes. Native covers token, curve, DEX and NFT operations, and also admits ordinary NOID payments and ABI 3 calls within the shared budgets.

Current class ceilings. Native-operation resources share the selected block's page and input limits.

Class	Physical pages	Ordinary live inputs	ABI 3 calls
Small	63	504	63
Large	206	504	63
Native	63	504	63

The Native class additionally permits up to 48 trade policies, 64 shared funding inputs, 128 typed actions and four markets. These are simultaneous ceilings. For example, 32 policies with two funding inputs each consume all 64 shared funding inputs. Actual capacity depends on the complete funding and payout shape.

The node considers Small and Native automatically. The `--v2-large-blocks` option also permits Large. Template selection chooses the candidate with the greatest claimable fees and builds one proof for that candidate. Equal fees retain the earlier candidate, with Small considered first.

The native-finance specification gives the exact arithmetic and transition rules. The accompanying wallet guides describe the live Tokens and NFT interfaces. `getNativeProtocol` reports the installed capacities, and the native templates expose the operation types for application builders.

RESEARCH RECORD

Evidence and artifacts

The links below preserve the implementation, measurement or derivation behind this article.

PDF

Native Finance article (PDF) ↗

Architecture

Native-finance rules and exact arithmetic ↗

Analysis

Current proof classes and block resource limits ↗

Architecture

Wallet guides: tokens, NFTs and committed media ↗