

Clary Coin

Research rewards, with verifiable rules.

Technical White Paper | Version 0.1 | October 20, 2026

Research Rewards Contributors

Abstract

Clary Coin is an ERC-20 research-reward token, symbol **CLARY**, with 18 decimal places. A scheduled competition accepts sealed submissions, requires public proof artifacts, and awards a fixed prize to the best eligible verified result. All research families share one issuance schedule and a lifetime allocation ceiling of 1,000,000 CLARY.

The ledger and competition contracts run on **Base Sepolia**, an Ethereum layer-2 test network. Mathematical checking runs separately: an Internet Computer Protocol (ICP) application subnet executes a fixed Rust proof checker compiled to WebAssembly. After a supported Lean proof passes, the canister requests a threshold ECDSA signature. The Base verifier authenticates that signature and confirms publication of the committed artifact.

Current scope. This is a testnet implementation, not a new proof-of-stake chain. Research does not replace Ethereum consensus. The Base contract verifies an external attestation, not a zero-knowledge proof of Lean execution. General research participation has not opened; no public research reward had been claimed at this paper's October 20, 2026 snapshot. Testnet balances have no promised cash value or mainnet conversion.

1. Purpose and scope

Finding a better algorithm may require substantial search, but a complete formal proof can be checked without repeating that search. Clary uses this distinction to connect a deterministic research objective to token rewards. It pays for an eligible improvement to the recorded frontier; it does not determine historical authorship, world novelty, or priority outside the protocol.

The first selected track concerns the discrete Fourier transform (DFT). Its fixed theorem establishes an all-positive-length algorithm in an exact complex-arithmetic model with operation count $O(n * \log(n)^{(1 - \kappa)})$. A larger exact rational κ is better. The challenge includes its algorithmic model and operation accounting, not merely a proposed numerical value. The target and provenance are pinned to an [upstream challenge revision](#).

A theoretical rate improvement is eligible. It does not by itself establish floating-point stability, practical constants, a useful crossover size, bit complexity, or faster execution than existing FFT software. Those require separate implementations and measurements. Integer multiplication is a possible later track, conditional on rigorous verification of its full intended theorem.

2. System architecture

Responsibilities and boundaries

Component	Responsibility	Boundary
Participant tools	Research, compile, export, locally check, commit, and submit.	Run on the participant's computer; model access and signing require explicit limits.
ICP checker	Check fresh proof declarations against a fixed prior library and request a signature on success.	Replicated execution under ICP's consensus and key infrastructure.
Base verifier	Authenticate the bound score and artifact; require complete artifact publication.	Does not independently execute Lean or prove ICP state.
CLARY core	Record rounds, select the winner, establish claims, issue and transfer tokens.	Fixed monetary configuration; operator schedules future competitions.

Base orders and executes the token transactions as an Ethereum layer 2. Ethereum's proof-of-stake infrastructure underlies its settlement, but Base has additional sequencing, rollup, and operational assumptions. Fast inclusion and finalized history are different stages; see [Base's transaction finality specification](#). Clary neither creates its own validator set nor pays research rewards to Ethereum validators.

The ICP checker is a canister, an application deployed on an existing subnet. Honest replicas deterministically execute its update messages and agree on state. The observed application subnet had 34 nodes on October 20, 2026; that is a dated observation, not a permanent node count or a separate Clary staking network. ICP's [execution documentation](#) describes this replicated WebAssembly model.

After checking, the canister requests the production `key_1 secp256k1` threshold ECDSA service. Key shares are managed by ICP's signing infrastructure; Clary's operator does not hold a conventional private signing key. The application subnet and the threshold-signing service are distinct responsibilities. The returned signature is verified on Base with the admitted Ethereum authority address. This relies on [ICP's chain-key security assumptions](#).

3. What the checker establishes

Lean source is elaborated into explicit proof terms. A checker validates their types and dependencies under the permitted logical rules; it need not trust the AI or the tactic that produced them. See the [Lean type-system reference](#). Clary's Rust implementation uses a pinned external checking kernel derived from `nanoda_lib`, with a separate fixed DFT admission policy. It is not the Lean interactive editor running inside Ethereum.

A fixed library, then fresh work

Rechecking the entire mathematical library on every submission is expensive. The selected profile admits an exact prior library after independent native validation of original proof bodies, transformed declarations, and retained dependencies. Previously proved theorems may then be represented by fixed opaque headers. The candidate's entry theorem and the declarations needed for its new result remain outside that admitted base.

Each canister request authenticates the pinned base and policy, rejects changes to existing declarations and metadata, parses the complete bounded artifact, and checks every supported fresh declaration. The entry theorem must prove the unchanged DFT target at the exact submitted rational score. Only `propeft`, `Quot.sound`, and `Classical.choice` are permitted axioms. A new arbitrary axiom, substituted target, or unchecked candidate header is not an acceptable shortcut.

The original library admission remains a trust boundary. Base does not cryptographically verify that one-time native validation. Readers must be able to inspect and reproduce the pinned source, library reconstruction, and admission evidence. The checker, parser, exporter, compiler, and mathematical statement can also contain defects. Successful replay tests do not constitute a formal proof that this implementation is sound.

Bounded, not universal

The deployed DFT profile accepts artifacts up to 1 MiB, with additional decoded-size, expression, declaration, and inductive-group limits, including at most 512 fresh declarations. A correct Lean proof outside the supported profile may require a separately reviewed verifier version. A rejected artifact is not automatically a false theorem.

The EIP-712 message binds the checker identity, statement, schema, normalized numerator and denominator, artifact root, and byte length. Its domain binds the destination chain and verifier contract. Base checks canonical signature encoding, the exact identities, the configured key, and complete publication. This is authenticated distributed execution, not encryption of the mathematical proof and not a succinct proof that Ethereum independently verifies the computation. [EIP-712](#) defines structured signing; replay and recipient rules must additionally come from the application.

4. From a sealed result to a reward

1. **Announce a round.** The operator fixes the problem, verifier version, parent record, score direction, minimum improvement, block deadlines, and prize. There is at most one round per family per epoch and one active round per family.
2. **Prepare and commit.** The participant locally checks the artifact and submits a hash commitment during the commit window. It binds the chain, token contract, round, version, family, parent record, submitter, beneficiary, exact score, artifact identity and size, and a private random secret. A hash commitment hides the opening; it is not encrypted proof storage.
3. **Wait before disclosure.** The client requires the commitment and closed commit window in agreed finalized Base history before sending proof bytes to ICP or publishing them. A transaction receipt alone is insufficient. The EVM contract does not enforce this off-chain safety step.
4. **Check and publish.** ICP rechecks the supported fresh proof and signs its result. The participant publishes the exact artifact in ordered 8,192-byte chunks. A batcher combines up to 12 chunks per transaction; the registry marks publication complete only when the rolling content commitment matches.
5. **Reveal.** Before the reveal deadline, the opening and certificate are submitted. The core recomputes the commitment, checks eligibility against the fixed parent, and calls the registered verifier. Anyone may relay a correct opening; the caller cannot change its sealed beneficiary.

6. **Select and settle.** After the deadline, anyone can settle. The best eligible revealed score wins. Equal scores use the earlier canonical commitment index. The contract maintains the provisional best as reveals arrive, so settlement does not need an unbounded scan of all entrants.
7. **Claim and transfer.** Settlement creates one claim for the fixed prize. Anyone may execute the claim once, but the tokens are always minted to its recorded beneficiary. CLARY then uses ordinary ERC-20 transfers. The settlement caller receives no special research fee.

Copying a public opening cannot redirect its reward. Someone who copies an artifact after a safely finalized cutoff cannot create a new eligible commitment for that round. Independent identical submissions before the cutoff remain possible. Later research may reuse published work; the protocol does not claim exclusive ownership over mathematical ideas.

Artifact bytes are public transaction/log data, not just a promise to host a private file. Reconstructing them later requires access to historical chain data. The signature does not bind a beneficiary or round: the separate sealed commitment does. This separation permits relaying without making the relayer the reward owner.

5. Currency and allocation rules

The following values belong to the deployed rehearsal contract, not an editable website preference.

Deployed Base Sepolia monetary configuration

Parameter	Value
Initial supply / lifetime allocation ceiling	0 / 1,000,000 CLARY
Token precision	18 decimal places
Epoch origin / duration	Base block 48,428,152 / 43,200 blocks
Initial global epoch allowance	342.465753424657534246 CLARY
Maximum base prize per round	100 CLARY
Beta multiplier / minimum beta	1% / 2 completed epochs
Normal reward ramp	7 epochs
New-family warmup / recovery discount	3 epochs / 4 epochs, each followed by the ramp
Allowance era	1,460 epochs
Round notice / maximum scheduling distance	5,400 blocks / 2 epochs ahead
Verifier registration notice	At least 86,400 seconds

All families compete for the same epoch allowance. Let e be an epoch, $H = 1460$, and A_0 the initial allowance in integer token units. For $k = \text{floor}(e/H) < 256$, let $a = \text{floor}(A_0/2^k)$. The allowance is $A(e) = a - \text{floor}(a * (e \bmod H)/(2H))$; later eras have zero allowance. It declines gradually toward half within an era, then starts the next era at half its initial value. It is not Bitcoin's constant-per-era block subsidy.

A round reserves its *undiscounted* base prize against both epoch and lifetime ceilings. Its actual prize is the base prize multiplied by the lower applicable global and family multiplier, rounded down in integer token units. A 100 CLARY base prize at 1% pays 1 CLARY but permanently reserves 100 CLARY. A numerically larger improvement does not increase the published fixed prize.

Beta begins at 1%. Normal promotion is one-way, requires the minimum beta period, and is announced for a future epoch. The seven-epoch ramp increases the multiplier to 100%. A new family remains at 1% until its first winning record is settled, then applies its warmup and ramp. Declared recovery begins its discount at actual reactivation, so a long pause cannot silently age the recovery period away. The lowest applicable multiplier determines the prize when the round opens.

Unused epochs, empty or canceled rounds, and discounts do not accumulate a later bonus. Old established claims remain payable; delayed claims may cluster token minting in wall-clock time without enlarging the reserved awards. New problems and verifier renewals never restart supply. These bounds limit issuance exposure; they cannot guarantee permanent research progress, profitability, demand, liquidity, or a stable market price.

6. Operator powers and continuity

The core has one owner address and supports two-step ownership transfer. The address may later be a multisignature wallet, but the present deployment has a single operator owner. There is no CLARY-holder voting system. Ownership renunciation is disabled to avoid abandoning required scheduling and recovery actions.

The owner can add families, register future verifier versions with notice, select future round conditions within immutable caps, announce normal mode, pause new configuration and commitments, and permanently disable a verifier version. Disabling permits cancellation of an unsettled round. A pause alone does not stop existing reveals, settlement, established claims, or ordinary token transfers.

The owner cannot edit constructor monetary settings, rewrite an existing family's statement or direction, change an opened round's prize and deadlines, mint through an unrestricted administrator function, confiscate balances, or revoke an already established claim. **However, verifier registration is a substantive trust power:** a malicious or defective newly admitted verifier could accept false claims within the issuance limits. A notice delay creates an observation window; it is not an automatic proof of soundness or a community veto.

The deployed ICP checker has an empty controller list, independently observed after the controlled proof trial. This prevents the developer from replacing its code through ordinary controller actions. ICP's own protocol and governance assumptions still apply; see [the canister trust model](#). The Base verifier's key and identities are immutable. Its manifest hash references admission evidence; it does not make Base validate that evidence.

Renewal deploys a separately identified checker or verifier and registers a new version for the same family. Balances, claims, best record, warmup history, and the global supply schedule remain on the existing ledger. A genuinely different mathematical objective needs a new family. Equivalent objectives under different labels require operator review even though they cannot bypass the shared monetary ceiling.

7. Security assumptions and remaining risks

- **Cross-network correctness.** A valid signature is meaningful only if ICP consensus, threshold signing, the frozen checker, and admitted library behave as intended. Base validators authenticate the attestation; they do not all run Lean. Compromise of that trust boundary can yield an authenticated false mathematical result.
- **Finality and endpoints.** Clients compare two fixed public RPC endpoints, use a common finalized anchor, and stop on disagreement or missing evidence. This is agreement between trusted services, not an independently verified light client. Consensus failures, misleading RPCs, or premature disclosure can break the copy-resistance assumption; no absolute rollback impossibility is claimed.
- **Availability and cost.** Anonymous callers can consume sponsor funds and finite job reservations, including through rejected work. One in-flight request, permanent attempt accounting, and a cycle floor bound admitted work but do not establish fair access or indefinite service.
- **Incentives.** Fixed prizes and winner-take-all rounds can encourage withholding, staged improvements, and strategic timing. Minimum improvements and shared budgets limit some consequences without eliminating those strategies. A valid submission may earn nothing if a better one wins.
- **Participant software.** A local research process, model service, runtime installer, or signing dependency may fail or be compromised. Pinned identities, finite budgets, a separately unlocked signer, and no automatic resend after an unknown broadcast reduce exposure; they do not replace independent review.
- **Privacy.** Addresses, transactions, proof artifacts, and disclosed results are public. Model, RPC, hosting, and verification services can observe requests or network metadata. Commitment secrets and wallet secrets must remain local until their intended use; the design does not provide anonymous networking or confidential computation.

8. Deployment evidence and current status

This is a dated implementation snapshot, not a live dashboard. Core deployments and DFT registration were checked against two RPC-reported finalized histories. The following public identifiers let readers distinguish this CLARY deployment from unrelated or historical test tokens.

October 20, 2026 testnet identities

Component	Identifier
Network	Base Sepolia * chain ID 84532
CLARY core	0x964CC00CD423FaE7BFaE1f86A7D40c04DA4D6065
DFT verifier	0x02e872F14B7a4Db2c279754651595AEfCcEd92c1
Publication registry	0x18d4dB1D3271491B71461e19da0A0fc1B2dA63bC
Publication batcher	0x6AA589028F816Fcc67d327326f8a71b0f4700cD7
ICP checker	sj2rm-xaaaa-aaaar-qciga-cai
Wasm SHA-256	852610810df8e449b979adc221fe5e60da22f9450cb3a8ae54d06a871ab3d8fa
DFT version interval	October 21, 2026, 21:22:52 UTC through January 19, 2027, 21:22:52 UTC

The measured public ICP trial checked a **known** 475,720-byte artifact for $\kappa = 6327/5000000$ and returned a locally authenticated signature in 13.124 seconds. It reported 16,185,202,414 checker instructions. The observed cycle-balance decrease was 77,423,771,767 cycles, including intervening calls, ingress, storage, and time; it is not an isolated CPU bill. The trial ran before controller removal. Subsequent independent certified observations confirmed the same module and empty controllers.

The proof fixture comes from [revision 4263d8f of exact-dft-lean](#), observed and independently replayed in October 2026. This is compatibility evidence, not a new discovery or an assertion of the globally latest frontier. One sample is not a latency percentile or a worst-case resource guarantee.

Local end-to-end trials verified mathematical checking, attestations, discounted and normal rewards, transfers, and rejection controls. The source-to-artifact path took approximately five minutes in a separate local trial. The complete automatic worker also ran with a fixed offline model response and real proof processing. These observations do *not* establish public reward payment, useful live AI research, or compatibility on every platform.

Still pending at this snapshot: the public beta-to-normal submission, claim, and transfer rehearsal; general participant admission; bounded live Codex integration; final source/runtime publication; and a reviewed sponsored operating window. The tested research runtime targets macOS on Apple silicon. Windows and Linux packages are not validated. A browser wallet does not require this research runtime.

9. Operation and future versions

The initial checker is sponsor-funded: no participant verification fee, automatic reward deduction, or automatic conversion of CLARY into ICP cycles is implemented. Participants' computers, AI subscriptions, and Base gas are separate costs. A finite pool is not permanent cost recovery. The deployed policy permits one in-flight job and up to 4,096 lifetime job and signature reservations; those counters do not mean that 4,096 jobs are funded. Unknown outcomes retain reservations, and extra funds do not reset exhausted counters.

A later paid service must enforce payment at its verification boundary, not only in a website that direct callers can bypass. It would require a separately admitted route and published terms. Existing token balances and established claims need not move to introduce a new verifier version; the frozen free checker itself cannot simply be edited into a paid checker.

Public release proceeds through actual finalized reward and transfer evidence, reviewed capacity and monitoring, a reproducible participant package, explicit runtime admission, and a sanitized source snapshot. Live model use must have finite time, request, and transaction budgets. Broader operating-system support and independent security review remain work to complete. Mainnet issuance, market liquidity, listing, and any treatment of test balances require separate decisions; none is implied by this testnet paper.

10. Primary references

External references explain the underlying systems; they do not endorse Clary Coin. Deployment claims above are observations of the identified implementation. Full reproducibility additionally requires the exact checker source, library admission evidence, and build inputs; their reviewed public release is a remaining milestone.

1. [Base: Transaction Finality](#) - inclusion, Ethereum batch posting, and finality stages.
2. [ICP: Execution Layer](#) - deterministic replicated Wasm execution and query/update distinctions.
3. [ICP: Chain-key Cryptography](#) - threshold signatures, key derivation, and signing infrastructure.
4. [ICP: Trust in Canisters](#) - controllers, immutability, and network trust.
5. [EIP-712](#) - typed structured hashing and signing.
6. [Lean Language Reference: The Type System](#) - proof terms and independent checking.
7. [nanoda_lib](#) - upstream external Rust type checker; Clary uses a pinned derivative and policy.
8. [Pinned challenge targets and pinned DFT proof fixture](#) - mathematical scope and known-result provenance.
9. [DFT version registration transaction](#) - public registration, not a research reward.