Whitepaper

Telegraph Protocol

The machine intelligence protocol for autonomous systems.

Ahmed Ali

info@telegraphprotocol.com

Telegraph Protocol

Version 1.0

4th May, 2026

Archived copy · cryp2nova.com

ABSTRACT

The Telegraph protocol defines a permissionless marketplace for verifiable artificial intelligence inference. The protocol separates the roles of Miners who supply computational intelligence from Validators who execute the Canonical Script assigned by the protocol to grade Miner responses via Stake-Weighted Median and Agents who consume inference outputs. Economic security is derived from a fixed-supply native protocol token, Machina, emitted on a strict Bitcoin-style halving schedule to Validators proportionally to real-world demand denominated in USDC. Instead of rewarding Miners with inflationary emissions for arbitrary GPU burn, they are compensated exclusively by a continuous TWAP (Time-Weighted Average Price) settlement of USDC into Machina – every paid signal creates an open-market buy, linking token value directly to consumption. This document provides a complete, formal specification of the Telegraph protocol.

MOTIVATION

Telegraph was born from a simple observation: the intelligence produced by open-source models, permissionless subnets, and global compute networks is growing exponentially, but it has no unified commercial layer. Builders on Bittensor and other protocols were creating extraordinary inference capabilities, yet there was no standardized way for autonomous agents, applications, or markets to discover, pay for, and consume that intelligence. Telegraph does not train AI models. It is a high-speed orchestration, payment, and signalization wrapper that takes inference from any provider – open-source, closed-source, or permissionless subnet – and turns it into a verified, payable signal.

We set out to build that layer – a lightweight, cross-chain protocol that acts as the commercial arm for the open intelligence economy. Telegraph does not compete with the permissionless subnets or the frontier AI labs. It captures their outputs, verifies them to a standard that machines and markets can trust, and makes them available through a single API and a set of on-chain Port contracts.

Telegraph is an experiment. A moonshot. The idea that autonomous agents will pay for verified inference as casually as servers pay for bandwidth may seem far-fetched today, but we believe it is inevitable. Every autonomous system – trading agents, prediction markets, risk engines, robotic fleets – needs a source of truth that is faster and more reliable than human consensus. Yet the current AI landscape offers little more than black-box models
with no audit trail.

Telegraph exists to fill that vacuum: a world where machines can request, verify, and act on intelligence without trusting a central authority, and where the builders of the best intelligence are paid directly for the value they create.

Archived copy · cryp2nova.com

We are not asking for permission to build this. We are building it because the problem is too important to leave unsolved, and because we believe that the right protocol, designed with the right incentives, can change how autonomous systems see the world.

At its core, Telegraph is a fully automated machine economy: agents create paid demand by requesting intelligence, the protocol routes those requests to miners who supply inference, validators verify the truth, and the entire loop settles on-chain without pause or human intervention. The interface – APIs, x402 payments, and Port Contracts – turns this into a trustless marketplace where machines buy answers and machines supply them.

Agents on Telegraph do not simply query an API; they broadcast open requests for intelligence to the network to power their internal logic and expand their capabilities. Miners compete to fill these orders with high-fidelity inference, creating a dynamic environment where machines constantly purchase intelligence to become smarter and more efficient.

For example, an autonomous agent planning a holiday receives a storm signal, reroutes the destination, purchases a logistics signal to book a rental car, and executes the entire chain through paid, verified inferences – without human intervention.

CRYP2NOVA

Archived copy · cryp2nova.com

1. PRELIMINARIES **7**
1.1. Notation 7
1.2. The Canonical Blockchain 7
1.3. The x402 Payment Primitive 7
2. THE MACHINA TOKEN **7**
2.1. Supply Parameters 7
2.2. Emission Allocation 8
3. ROLES AND STAKEHOLDERS **9**
3.1. Miners 9
3.2. Validators 9
3.3. Agents 10
3.4. Delegated Staking 10
4. INTENT AND WASM EXECUTION LAYER **10**
4.1. Intent Definition & Miner YAML Registration 10
4.2. Validation Scripts 12
4.3. Script Evolution and Catch-Rate Promotion 13
4.4. Protocol Boundaries 13
5. VALIDATOR ARCHITECTURE **14**
5.1. Validator Function & Stratified Portfolios 14
Participation Multiplier 14
Canonical Script Enforcement 14
Testing Cohort and Internal Audit Fee 14
The Validation Flow (The Autonomous Engine): 15
Miner Tournament and Routing 16
Intra-Epoch Spot Checks and Routing Revocation 17
5.2. Active Validator Set 17
5.3. Penalty Conditions 18
Category A – Liveness Failure 18
Category B - Ground Truth Fabrication / Fraud 18
Category C penalty (emission forfeiture) for consensus deviation 19
Testing Cohort Exemption 19
5.4. Ground Truth Integrity (zkTLS) 19
5.5. Fraud Detection 20
5.6. Miner Access and Credential Protection 20
Enclave Key Infrastructure 20
The Credential Store 21
Onboarding and Key Updates 21
Credential Decryption 21
Key Rotation 21

CRYP2NOVA

Archived copy · cryp2nova.com

Enclave Provisioning 22
Inference Flow 22
Security and Miner Control 22
Front-Running Prevention 22
Gossip-Time Enforcement 23
Clock Synchronisation 23
Post-Finalisation Delivery 23
Penalties for Deliberate Delay 23
Economic Rationality 24
6. CONSENSUS AND FINALITY 24
6.1. Leader Election 24
6.2. Leader Failover 25
6.3. BFT Finalization 25
Off-Chain BFT and BLS Aggregation 26
6.4. Network Partition Handling 26
6.5. Block Submission and Epoch Finality 26
Block Structure 26
Replay and Deterministic Audit 27
Precondition for Submission 27
Block Builder Selection 27
The Settlement Layer (Dumb Registry) 28
Verification and Execution 28
7. ECONOMIC MODEL AND TOKENOMICS 28
7.1. Protocol Treasury 28
7.2. Miner Compensation via TWAP Settlement 29
Settlement Parameters: 29
7.3. Validator Emission Distribution 30
7.4. Request Routing 30
7.5. Signal Pricing Mechanism 31
8. PAYMENT PIPELINE AND X402 INTEGRATION 32
8.1. Canonical Chain 32
8.2. Web3 Agent Payment Flow 32
8.2.1. Pre-Funded Agent Escrow (Funding Wallet) 32
8.3. Web2 Developer Onboarding 33
8.4. On-Chain Callbacks - Gas Escrow 33
8.5. WebSocket Pub/Sub Interface 34
8.6. ERC-8183 Agentic Commerce Integration 34
9. BOOTSTRAPPING & AUTONOMOUS ENGINE 35
9.1. Two-Phase Launch 35

CRYP2NOVA

Archived copy · cryp2nova.com

9.2. Genesis Intent Library and Foundation Miners 35
9.3. Intent Bounties & The Autonomous Engine (AE) 36
9.4. Signal Agora 36
9.5. Bootstrap Transition 37
10. GOVERNANCE 37
10.1. Activation Threshold 37
10.2. Governable Parameters 37
10.3. Proposal Process 38
11. SECURITY MODEL 38
11.1. Sybil Resistance 38
11.2. MEV Resistance 38
11.3. Anti-Spam Economics 38
12. GENESIS PARAMETERS 39



Archived copy · cryp2nova.com

1. PRELIMINARIES

1.1. Notation

The set of natural numbers is denoted by N. The closed interval [0,1] in real numbers is denoted R_[0,1]. Cryptographic hash function Keccak-256 is denoted H. Concatenation of byte strings is denoted with double vertical bars. The active validator set is denoted V. The cardinality of V is |V|. The set of all registered Intents is I. The set of all registered Miners is M.

1.2. The Canonical Blockchain

The Telegraph protocol is deployed on the Base network, an Ethereum Layer-2 rollup. All on-chain state, smart contracts, and native Machina tokens reside on Base. The choice of Base is motivated by native support for x402 payment semantics, integration with Coinbase AgentKit for frictionless Web2 onboarding, and low transaction costs suitable for micro-settlement.

1.3. The x402 Payment Primitive

An x402 payment is an HTTP 402 Payment Required response that includes a cryptographically signed demand for a specific amount of USDC to be transferred to a specified address on Base within a strict deadline. The client (Agent) must broadcast a compliant transaction to the Base network to satisfy the payment challenge and obtain the requested service.

2. THE MACHINA TOKEN

2.1. Supply Parameters

The native token of the Telegraph protocol is Machina. Its supply schedule is strictly fixed at genesis and is immutable.

Maximum Supply **S_max**: **21,000,000 Machina**
Genesis Daily Emission **E_0**: **7,200 Machina** per **24-hour epoch**
Halving Interval **H_halve**: **4 years** (approx. 1,461 epochs)
Pre-mine Allocation: **0**
Team / VC Allocation: **0**

Archived copy · cryp2nova.com

The emission at epoch t (where t equals 0 is the genesis epoch) is given by: E(t) = E_0 multiplied by 2 to the power of negative floor of (t divided by (365 times 4)).

$$E(t) = E_0 \times 2^{(-\lfloor t/1461 \rfloor)}$$

The protocol shall cease emission when the cumulative minted supply reaches **S_max**. Any epoch in which the remaining mintable supply is less than E(t) shall mint only the remainder, after which emission permanently halts.

2.2. Emission Allocation

The daily emission E(t) is distributed among three tiers according to fixed percentages, immutable at genesis.

Tier: Validators Allocation: 60 percent
Tier: Script Authors: 20 percent
Tier: Protocol Treasury: 20 percent

Miners receive zero Machina from protocol emissions. Miner compensation derives exclusively from USDC paid by Agents for inference requests *(Section 7.2)*. This separation is the central economic invariant of the protocol: the token supply rewards the infrastructure that verifies truth, while actual demand pays the intelligence providers. It eliminates the fake-work incentives present in emission-funded miner models and creates a permanent, mechanical bid for Machina on every paid signal.

The 20 percent Script Author tier is distributed via Hash-Math: the smart contract calculates the proportion of surviving (unslashed) Validator stake that utilised each specific WASM_Hash during consensus, and distributes the 20 percent emission pool strictly proportionally to the registrants of those hashes. Validators use the most accurate evaluation scripts for free to protect their emission share, while the protocol natively compensates the developers whose logic proves most essential to the network.

The current 60/20/20 split is the genesis allocation. As the protocol matures, governance may introduce additional tiers (e.g., a Use-Case Growth fund or a dedicated Gas & Maintenance allocation) by re-dividing the existing pools. Any change to the emission tier percentages requires a governance vote per Section 10 and is subject to the supermajority threshold.

If no WASM_Hash submissions exist in a given epoch, the full 20 percent Script Author allocation for that epoch is transferred to the Protocol Treasury, consistent with the protocol's policy of redirecting emissions rather than burning them, except where permanent slashing is required for cryptographic fraud *(Section 5.3, Category B)*.

Archived copy · cryp2nova.com

3. ROLES AND STAKEHOLDERS

The protocol defines three distinct, non-overlapping economic roles.

3.1. Miners

A Miner is a node that registers one or more Intents *(Section 4)* and serves AI inference requests. A Miner declares its supported Intents and API endpoint via a YAML configuration. Miners do not participate in validation, consensus, or scoring. They are compensated solely by the USDC to Machina TWAP settlement of successful inference requests. The protocol never mints Machina for Miners; every unit of Machina a Miner receives was first purchased from the open market using the Agent's USDC payment.

3.2. Validators

A Validator is a node that participates in the stake-weighted median pipeline. The elected Leader Validator scrapes live ground truth data, generates a zkTLS proof, and queries Miners. All Validators independently execute the Canonical Script (or, for Testing Cohort members, the assigned test script) against the locked payload to produce a Local_Score, participate in BFT finalization, and gossip signed receipts. The operational right to run a Validator is established by registering a Validator Profile on the canonical smart contract. Validators are selected by stake competition; the top N Validator Profiles by staked Machina constitute the Active Validator Set V, where N is the Validator Cap. At genesis, N equals 64. The cap may only be increased via governance (Section 10). Autonomous agents may also act as Validators by registering a profile, staking Machina, and running the validator binary.

To secure the network against server-side vulnerabilities, each Validator Profile is controlled by two distinct keys with strictly separated duties. The Manager Key is a high-security EVM address (cold wallet) that controls the Validator Profile and the staked Machina. This key remains strictly offline and is solely responsible for staking, unbonding, claiming rewards, and rotating the Operator Key. The Operator Key is a key pair residing on the active Validator's internet-facing server (hot wallet). The Manager Key executes an on-chain transaction to mathematically bind and authorise an Operator Key to its profile. The Operator Key's singular permission is to sign verifiable Signal Receipts and participate in BFT consensus. If an Operator Key is compromised, the attacker cannot access the staked Machina. The Manager Key may execute an instant on-chain rotation of the Operator Key to re-secure the node.

Archived copy · cryp2nova.com

3.3. Agents

An Agent is any client that submits inference requests and pays the required USDC fee via x402. Agents may be human-operated terminals or autonomous software.

3.4. Delegated Staking

Any address holding Machina may delegate its stake to an active Validator. The smart contract tracks the proportional shares of the resulting emission rewards according to a commission percentage specified at delegation time, representing the share of accrued emissions that the Validator retains before the remainder is claimable by the delegator. Delegators asynchronously claim (pull) their accrued Machina from the Port Contract to permanently prevent emission-distribution gas exhaustion. This mechanism allows Validators to aggregate stake to remain in the active set.

4. INTENT AND WASM EXECUTION LAYER

4.1. Intent Definition & Miner YAML Registration

An Intent is a uniquely identified, permissionlessly registered category of AI inference task. To ensure network resilience if the underlying Web2 APIs break, Intents utilise immutable semantic versioning (e.g., weather_check@v1.0). If an endpoint fails, organic USDC volume drops to zero, Validators economically abandon the broken version, and developers permissionlessly register v1.1 to recapture the routed demand.

To serve an Intent, a Miner must register via an on-chain transaction accompanied by a minimum stake bond of 100 Machina (genesis). Miner registration is permissionless. Any address that posts the required bond and a valid YAML configuration may register as a Miner for any Intent. There is no cap on the number of Miners; the bond and market forces naturally bound participation, as Miners who consistently rank poorly earn no Agent revenue and will eventually withdraw.

This transaction binds the Web2 API execution to the Web3 settlement layer and must contain:
Supported Intents: An array of Intent IDs that the Miner claims to serve.
Miner Fee Address: The definitive EVM address where the TWAP Settler will route all Machina payouts for fulfilled requests.
Minimum Signal Price (min_price_usdc): The Miner's declared floor price in USDC per signal for each supported Intent. The protocol minimum floor at genesis is 0.01 USDC. The Miner may declare a higher floor. This floor is immutable after registration; a Miner wishing to change their floor price must submit a new registration transaction.

Archived copy · cryp2nova.com

Credential Hash (credential_hash): The content-addressed hash of the Miner's encrypted API credential ciphertext, uploaded to the Credential Store via the Enclave Key Infrastructure (Section 5.6). This field binds the Miner's API access credential to their on-chain identity.
YAML Schema Hash and URI: A cryptographic hash and off-chain pointer to the Miner's unified declarative configuration.

The final price paid by an Agent is determined by applying the protocol demand multiplier to this floor as defined in Section 7.5.

The Unified YAML schema dictates both off-chain routing and on-chain data transformation. The configuration must include an on-chain_output block that maps raw JSON API responses directly into deterministic EVM data slots using the following canonical on-chain data structure:

```C/C++
struct OnChainData {
    address[] addresses;
    uint256[] integers;
    string[] strings;
    bool[] bools;
}
```

The onchain_output block maps each raw API response field to a named slot within this struct, declaring the array type, index, field name, and source path (e.g., strings[0] equals the confidence score; bools[0] represents the synthetic media flag). Slot indexes are immutable after registration. A Miner may only add new indexes via a new registration transaction. Existing downstream contracts, depending on any declared slot index, are never broken by a Miner update. The Port Contract rejects any registration where the on-chain_output block fails to compile against the OnChainData struct definition. Validators read this block to autonomously pack the API response into the on-chain struct. At the same time, downstream Agent smart contracts query the same schema to decode the callback data trustlessly without requiring off-chain parsers.

Intent_ID is derived deterministically as follows:
Intent_ID = H(Registrant_Address || YAML_Schema_Hash || Block_Number)

Archived copy · cryp2nova.com

The Port Contract rejects any registration transaction whose computed Intent_ID already exists in the registry. Collision probability is negligible by construction.

Miner Deregistration: A Miner may submit a deregistration transaction at any epoch boundary. The 100 Machina bond is returned to the Miner's registered address after a 7-day unbonding period, during which the Miner remains eligible for tournament evaluation but receives no new Agent routing. Any pending TWAP settlement for fulfilled requests completes normally during the unbonding period.

4.2. Validation Scripts

A Validation Script is a scoring algorithm compiled to WASM that a Validator executes locally to evaluate a Miner's response against verified ground truth data.

At the start of every epoch, each active Intent has exactly one Canonical Script, the WASM binary that all Validators must execute when scoring Miner responses for that Intent. The Canonical Script is determined automatically by the protocol *(Section 4.3)*; Validators have no discretion to choose a different script.

A small, rotating Testing Cohort of Validators (genesis: 10 % of the active set, selected deterministically from the epoch block hash and the Intent ID) is required to execute an assigned non-canonical script for that epoch. Cohort members submit their Local_Score using the assigned script's WASM_Hash, exactly as in a normal round. Refusal to participate counts as a missed round *(Section 5.3, Category A)*.
Script Registration, the anti-spam bond, WASM sandbox constraints, and WASM_Hash declaration rules remain unchanged.
Script Registration: Any developer may publish a Validation Script by posting a flat anti-spam bond of exactly 10,000 Machina and submitting an on-chain transaction.
The transaction must contain:

1. **Intent_ID:** the specific Intent this script is registered for.
2. **WASM_Binary_Hash:** Cryptographic hash of the compiled WASM module.
3. **Whitelisted_URLs[]:** The external ground truth endpoints the script may access.

WASM Sandbox Constraints: Regardless of which script a Validator selects, the runtime enforces:

1. No filesystem access.
2. No access to private keys or host environment.
3. Network access is restricted to the declared Whitelisted_URLs[].
4. Execution is strictly isolated from the Validator node process.

WASM_Hash Declaration Accountability: Validators declare their WASM_Hash in each signed Local_Score submission. Category C penalty for consensus deviation creates the economic enforcement layer: a Validator declaring a high-quality script's hash while running inferior logic will produce Local_Scores that deviate from the median and

Archived copy · cryp2nova.com

accumulate Category C strikes. The financial penalty for misrepresenting script usage is therefore implicit in the consensus mechanism itself.

4.3. Script Evolution and Catch-Rate Promotion

Script evolution is fully automated. No validator vote, human bounty, or off-chain coordination is required.

The protocol tracks the testing-round scores of every registered non-canonical script, comparing them with the Canonical Script's scores on the same Miner responses. After a script has been tested for at least T epochs (genesis: 3), it is automatically promoted to Canonical status at the next epoch boundary if it achieves the Catch-Rate condition:

- Across the identical set of Miner responses, the test script assigned a score at least delta_promote (genesis: 0.10) below the Canonical Script's score for at least one Miner response on which the Canonical Script scored above 0.70 – indicating the test script identifies underperforming Miners that the Canonical Script is overscoring.
- The test script's median score remained within the stake-weighted median band (|score – global median| ≤ delta_c), proving it is not broken.
- When promotion occurs, the previous Canonical Script becomes a regular registered script and may be tested again in future epochs. The new Canonical Script is recorded on-chain in the epoch block; all Validators must switch to it for the subsequent epoch *(Section 5.3, Category C)*.

Script registrants may deactivate their script and withdraw their 10,000 Machina bond at any time.

4.4. Protocol Boundaries

To preserve deterministic consensus and stateless scalability, the Telegraph protocol enforces a strict execution boundary between network operations and client operations.

The Flow (Protocol-Native): A Flow is an atomic, single-step pipeline executed entirely by the Telegraph network. It consists of receiving an x402-funded Intent, probabilistically routing it to a Miner, executing the selected Validation Script, and gossiping the finalised Signal Receipt. The protocol's purview ends immediately upon delivery of the verifiable Signal.

The Workflow (Client-Native): A Workflow is a dynamic, multi-step orchestration requiring logic, state, and memory. Workflows are executed exclusively by the Agent's native architecture and may chain multiple Telegraph Flows together using asynchronous x402 payment sessions.

Archived copy · cryp2nova.com

By maintaining this boundary, the protocol operates as a highly scalable, stateless signal layer.

5. VALIDATOR ARCHITECTURE

5.1. Validator Function & Stratified Portfolios

A Validator v in V is a scoring and consensus node. Because a Validator cannot physically execute validation for thousands of concurrent Intents, node software automatically allocates compute into a Stratified Portfolio:

1. Volume Tier (90 percent of compute): Intents with the highest trailing USDC volume.
2. Discovery Tier (10 percent of compute): A deterministic sample of Intents computed as follows: DiscoverySet = all Intents registered within the preceding 30 epochs, plus all Intents with a non-zero active USDC Bounty balance in the Port Contract. From this set, the node selects a subset bounded at 10 percent of the Validator's total active Intent count, ordered deterministically by H(Intent_ID || EpochNumber) to ensure consistent selection across all Validators without coordination.

Participation Multiplier

To unlock eligibility for the 60 percent Validator emissions in the Volume Tier, a Validator must not accumulate failed Discovery Tier rounds beyond the participation threshold defined in Section 12 in the epoch. A Discovery Tier round is considered failed if the Validator does not sign the receipt for that round. If the Validator successfully signs fewer than 95 percent of its assigned Discovery Tier rounds in a single epoch, the Validator forfeits all Volume Tier emissions for that epoch only.

Canonical Script Enforcement

For every Intent in its portfolio, a Validator must execute the Canonical Script declared in the on-chain registry for that epoch. A Validator that submits a Local_Score produced by any other script, and is not a member of the Testing Cohort for that Intent, incurs a Category C penalty *(Section 5.3)*.

Testing Cohort and Internal Audit Fee

For each active Intent, a Testing Cohort is selected deterministically at the start of the epoch:

$$\text{CohortSize} = \lceil 0.10 \times |V| \rceil$$
$$\text{CohortStartIndex} = H(\text{BlockHash}_{(e-1)} \,\|\, \text{Intent_ID}) \bmod |V|$$

Archived copy · cryp2nova.com

The Cohort consists of the Validators at indices CohortStartIndex through CohortStartIndex + CohortSize − 1 (wrapping around). Every Validator can verify the selection
independently.

Cohort members are required to execute a specific non-canonical script for that Intent, chosen deterministically by a similar hash-based rotation through the on-chain script registry.
Refusal to participate is treated as a missed round (Category A).

To incentivise participation, the Protocol Treasury allocates a fixed Internal Audit Fee for each epoch – genesis: an amount equal to 2 % of the epoch's total Validator emission pool, split equally among all Cohort members across all Intents. Validators, therefore, earn extra Machina for serving in the Cohort, transforming testing from a burden into a sought-after reward.

If no non-canonical scripts are registered for a given Intent in a given epoch, the Testing Cohort for that Intent is dissolved for that epoch only. Cohort members revert to standard Canonical Script execution and are not penalised for the absence of challenger scripts.

The Validation Flow (The Autonomous Engine):

The validation process does not merely grade historical data; it actively generates the intelligence feed. For every Intent in a Validator's portfolio, the protocol executes the following:

1. Scrape & Query: The elected Leader scrapes live ground truth from whitelisted URLs, generates a cryptographic zkTLS Proof for the live-Signal path, and queries every registered Miner for that Intent in parallel. Miner responses are authenticated via the Enclave Key Infrastructure (Section 5.6); no per-miner zkTLS proofs are generated for this tournament.
2. Signal Generation: The Miner_Response is returned to the Leader. This response IS the Signal. The Leader packages the locked payload [Ground_Truth, Miner_Response, zkTLS_Proof].
3. Delivery: The Leader gossips the payload to the mesh. After BFT finalisation, the elected Leader (or any Validator that submits the on-chain settlement) pushes the canonical Miner_Response Signal to all WebSocket subscribers whose filters match the Intent and whose minimum confidence threshold is met by the Canonical_Score. An Agent receives the Signal only after it has been cryptographically verified and finalised.
4. Mesh Verification & Scoring: The 63 non-leader Validators receive the gossip payload, mathematically verify the zkTLS proof locally for free, and execute the

Archived copy · cryp2nova.com

Canonical Script (or, for Testing Cohort members, the assigned test script) against the locked inputs to produce a Local_Score in [0,1].

Miner Tournament and Routing

For each Intent, the Leader queries all registered Miners in parallel during the epoch evaluation. Every queried Miner receives the same prompt and must return a response to remain eligible for agent traffic in that epoch. Answering this evaluation query is a condition of participation, not a paid service. The cost of querying all registered Miners during the epoch tournament is borne by the Protocol Treasury at the protocol minimum floor price per query, consistent with the validation query economics described in Section 7.1. Miners accept evaluation queries as a condition of network participation and are compensated via Agent traffic routing following the leaderboard publication. Miners are compensated exclusively when an Agent purchases their signal (Section 7.2).

A Miner that fails to return a response to a tournament or spot check query within the Leader's collection window receives a score of zero for that round and is excluded from the routing pool for the remainder of the epoch. Repeated non-response across epochs does not slash the bond; market forces handle exclusion as zero-scoring Miners earn no Agent revenue.

The elected Leader computes the Canonical_Score for each Miner using the stake-weighted median of the Local_Scores from the Validator mesh. The resulting leaderboard determines Miner routing for the remainder of the epoch. To prevent a single point of failure, paid Agent requests are not routed exclusively to the top Miner. Instead, the protocol distributes Agent traffic probabilistically among the highest-ranked Miners (e.g., the top Miner receives the predominant share, the second-ranked Miner receives a smaller fraction, and the third-ranked Miner receives a minimal trailing fraction). The exact distribution weights are governable via Section 10.

Until the first tournament of an epoch completes and a new leaderboard is published, the routing table from the previous epoch remains in effect. At genesis, before any tournament has been run, paid requests are routed uniformly among all registered Miners for that Intent, ensuring that every Miner receives an equal chance to earn initial Agent traffic until the first leaderboard is established.

The number of Miners per Intent is naturally bounded by the 100-Machina registration bond and by market forces: Miners who consistently rank poorly earn no Agent USDC and will eventually withdraw. If an Intent experiences extraordinary growth that threatens the Leader’s ability to query all Miners within the epoch window, the protocol may enforce a maximum Miner count per Intent, adjustable via governance.

Archived copy · cryp2nova.com

Intra-Epoch Spot Checks and Routing Revocation

To prevent a top-ranked Miner from degrading its answers mid-epoch, the protocol mandates unpaid, continuous spot checks.

Spot checks are triggered deterministically from on-chain data, outside the Leader's control. At the genesis frequency parameter of 10, a spot check is triggered whenever H (Latest_L2_BlockHash ‖ Intent_ID) modulo 10 equals zero – approximately once every 20 seconds at Base's 2-second block time. When triggered, the elected Leader must immediately initiate a spot check for that Intent.

The Leader queries the active Miners using the same evaluation script and stake-weighted median as the epoch-opening tournament. Spot checks are free for the network; answering them is a mandatory condition of receiving Agent traffic.

If any spot-checked Miner produces a Canonical_Score that falls below its original tournament score by more than the protocol-defined threshold (genesis: 20 percent relative drop), a BFT finalisation occurs. Instead of a standard receipt, the Leader gossips a Routing_Revocation payload containing the failed scores and the tau mesh signatures.

Upon receiving and verifying a valid Routing_Revocation, all Validators instantly update their local in-memory routing tables to drop the offending Miner. The Miner receives no Agent traffic from that block onward, and its routing share is redistributed proportionally among the remaining eligible Miners. The revocation is immutably recorded during the standard epoch block submission (Section 6.5).

The validation round, therefore, produces two simultaneous outputs: a verified Signal for instantaneous consumption by subscribing Agents and a canonical leaderboard score update for the evaluated Miner. This continuous loop IS the Autonomous Engine.

5.2. Active Validator Set

The active validator set V is determined by stake competition. At each epoch boundary, the top N Validator Profiles by staked Machina become the active validators for the following epoch. V_epoch equals the set of validators where the rank of their stake is less than or equal to N.

Each Validator Profile is controlled by a Manager Key (cold wallet) and an authorised Operator Key (hot wallet) as defined in Section 3.2. The Operator Key is the only key permitted to sign receipts and participate in consensus. The Manager Key has no signing rights during normal operation and remains offline except for staking, unbonding, reward claiming, and Operator Key rotation.

Archived copy · cryp2nova.com

Unbonding Period: When a Validator Profile is outbid or voluntarily exits, its staked Machina is frozen for 21 days. This period exists solely to allow a penalty for fraud committed during active tenure. Unbonding begins at the block of ejection or exit.

5.3. Penalty Conditions

Category A – Liveness Failure

Definition: A Validator that fails to sign the required receipts for the Intents in its portfolio *(Section 5.1)* for a continuous, unbroken period of 7 days is considered Liveness-Ejected. A single successful signature for any required round resets the
continuous counter.

Additionally, a Validator that misses more than 50 percent of the rounds required by its portfolio within any rolling 7-day window is also Liveness-Ejected, regardless of whether the absences are continuous.

During the offline period, the Validator does not earn any Validator emissions for the missed rounds. The emissions that would have been allocated to that Validator are redistributed equally among all other active Validators in the same epoch. After ejection, the 21-day
unbonding period begins. There is no stake burn for a liveness ejection.

Category B - Ground Truth Fabrication / Fraud

Definition: Elected leader Validator submits a locked payload containing a Ground_Truth or Miner_Response that fails zkTLS proof verification, meaning the data was not retrieved unmodified from the declared whitelisted endpoint. Because score disagreement among validators is expected and resolved by the median, score disagreement alone is never evidence of fraud.
Fraud is defined exclusively as cryptographic proof of data tampering or fabrication.
Slash Amount: 20 percent of the total staked Machina.
Destination: The slashed stake is permanently transferred to the Protocol Treasury, where it is held to fund the Miner Grant Programme and long-term infrastructure maintenance. The protocol performs zero token burns.
Blacklist: The associated Validator Profile is permanently revoked.

Both the Manager Key and the compromised Operator Key are
permanently banned at the protocol level. Re-entry using either
address is never permitted.

Archived copy · cryp2nova.com

Category C penalty (emission forfeiture) for consensus deviation

Definition: A Validator submits a Local_Score that deviates from the canonical stake-weighted median by more than the consensus deviation threshold delta_c.
Formally: |Local_Score - Canonical_Score| > delta_c.
Penalty: The Validator forfeits the round's Validator emission share for that specific Intent. The forfeited amount is redistributed equally among the other Validators who participated in the round and submitted a Local_Score within delta_c of the median.
Strike: 1 per Flow in which the deviation occurs.
Ejection Trigger: 5 strikes within any rolling 30-day window.
Ejection Result: Forced removal from V; 21-day unbonding begins. Re-entry permitted after correcting script configuration. A Validator that fails to switch to a newly promoted Canonical Script at the next epoch boundary will produce Local_Scores that systematically deviate from the median, incurring Category C penalties automatically without requiring a separate enforcement mechanism.
Rationale: Because validators all execute the same Canonical Script (or an assigned test script), minor score variations are expected due to node-level timing and platform-specific WASM execution differences, and are legitimate. However, persistent deviation beyond delta_c indicates the Validator is running obsolete, broken, or zero-compute scripts. The penalty enforces strict economic adherence to the most accurate grading logic available in the registry without ever touching the Validator's staked principal.

Testing Cohort Exemption

A Validator participating in the Testing Cohort for a specific Intent is exempt from the Category C deviation penalty for that Intent, provided its Local_Score falls within delta_c of the Cohort-only median (computed exclusively from the scores submitted by fellow Cohort members for that Intent). This exemption protects honest testers from being penalised for running a script that produces accurate but different scores. A Cohort member whose score deviates from the Cohort-only median by more than delta_c is subject to the standard Category C penalty, preventing lazy nodes from outputting random numbers.

5.4. Ground Truth Integrity (zkTLS)

To permanently eliminate the "Blind Trust" vector where a Leader could fabricate ground truth data to rig a Miner's score, the protocol mandates zkTLS (Zero-Knowledge Transport Layer Security) at the scraping layer.

During the execution of a flow, the Leader must generate a zkTLS Proof covering:

1. The TLS session handshake with the script's declared Whitelisted_URL.
2. The SHA-256 hash of the inbound Ground_Truth payload from the Web2 endpoint.

Archived copy · cryp2nova.com

3. The TLS session handshake and payload hash of the Miner_Response are selected as the canonical Signal for that epoch. During the epoch tournament, Validator mesh scoring of non-selected Miner responses relies on cryptographic commitment and stake-weighted consensus rather than individual zkTLS proofs, as specified in Section 5.1.

The Leader gossips this proof alongside the raw data. All 63 non-leader Validators independently verify the cryptographic validity of the zkTLS proof. Because verification requires zero external API calls, the mesh achieves verified ground truth with zero additional OpEx. An invalid or missing zkTLS proof results in the payload being rejected by the mesh. The rejected round counts as a missed broadcast for the Leader's continuous offline window (Section 5.3, Category A).

5.5. Fraud Detection

Leader fraud is detected strictly via zkTLS proof failure or cryptographic signature failure, not via score comparison. Because validators execute the script assigned by the protocol, score disagreement is expected only within narrow bounds and resolved entirely by the median *(Section 6.3)*.

Score disagreement alone is never evidence of fraud. If a Validator's local score diverges, they are economically penalized via Category C emission reduction, not blacklisted for fraud. This isolates objective cryptographic failures (Cat B) from subjective logic failures (Cat C).

Each Signal Receipt continues to include the optional Transparency Attachment field as defined previously, allowing Miners to voluntarily append reasoning traces and source citations to their responses. The protocol does not verify the accuracy of Transparency Attachments; verification of reasoning is the responsibility of the consuming Agent.

5.6. Miner Access and Credential Protection

Validators query Miners over direct TLS, authenticated with a dedicated, limited-scope API key that the Miner creates exclusively for Telegraph. The Miner never modifies their application code, API gateway, or firewall.

Enclave Key Infrastructure

The active Validator Enclaves collectively maintain a threshold encryption scheme. The public key is published on-chain at every epoch boundary; the corresponding private-key shares exist only inside the hardware-protected memory of the individual active enclaves.

Archived copy · cryp2nova.com

No single enclave ever holds the complete private key. Credentials are encrypted under the threshold public key and can only be decrypted by a quorum of enclaves.

The Credential Store

Miner ciphertexts are stored in a distributed, enclave-managed key-value store called the Credential Store. Each active enclave maintains a local replica. New entries are propagated across the enclave network using the same **libp2p** gossip mesh that carries Signal Receipts (Section 6). The chain records only the Merkle root of the entire store, updated at each rotation, allowing any enclave to verify the integrity of its local copy.

Onboarding and Key Updates

A Miner onboards by:

1. Creating a Telegraph-specific API key in their existing provider dashboard, with a low spend cap and rate limits.
2. Using the Telegraph web interface to fetch the current enclave public key, encrypting the API key client-side, and uploading the resulting ciphertext to the Credential Store via an attested channel. The upload is accepted only if it carries a signature from the Miner's registered address.
3. Including the content-addressed hash of the ciphertext in their on-chain registration under the field `credential_hash` *(Section 4.1)*. This transaction is signed by the Miner, binding the credential to their identity and stake.

The plaintext API key never leaves the Miner's browser. To update a credential, the Miner submits a signed on-chain transaction containing the new `credential_hash`. The Credential Store accepts an upload only if it is accompanied by a signature from the Miner's registered address, proving ownership of the on-chain registration.

Credential Decryption

At request time, the Validator's local Enclave coordinates with a quorum of other active enclaves to decrypt the Miner's ciphertext using their threshold shares. The resulting API key is supplied exclusively to the requesting Validator for a single TLS session and discarded immediately after use.

Key Rotation

Whenever the active Validator set changes, the enclaves automatically generate a new set of threshold shares, re-encrypt all stored credentials under the new public key, and publish the updated public key and Merkle root on-chain. The previous shares are permanently discarded, rendering the old key material useless:

1. Generate a new asymmetric key pair.

Archived copy · cryp2nova.com

2. Decrypt every entry in the Credential Store with the old private key, re-encrypt with the new public key, and store the updated ciphertexts.
3. Compute the new Merkle root of the Credential Store and submit it on-chain together with the new public key.
4. Permanently discard the old private key.

Former Validators excluded from the rotation cannot decrypt any credentials in the store after the rotation completes.

Enclave Provisioning

A new Validator that enters the active set receives a threshold share of the current private key through remote attestation with an existing active enclave. The share is sealed into the new enclave's hardware memory and is never accessible to the Validator operator.

Inference Flow

1. The Validator's local Enclave verifies that the Validator's Operator Key is in the Active Validator Set.
2. The Enclave fetches the Miner's ciphertext from its local Credential Store replica using the on-chain credential hash.
3. The Enclave decrypts the ciphertext and supplies the API key for a single request.
4. The Validator connects directly to the Miner over TLS, presents the key in the Authorization header, and generates a zkTLS proof (Section 5.4).
5. The key is discarded from memory immediately after the request completes.

Security and Miner Control

The dedicated Telegraph key carries minimal spend by design. A Validator who extracts the key gains access to a limited-spend API while risking permanent slashing under Category B. The Miner retains full control through their provider dashboard and may revoke or rotate the key independently at any time.

Front-Running Prevention

A malicious Validator who controls their own node software can always attempt to delay the dissemination of a Miner response to exploit it financially before the paying Agent or the subscribing users. The protocol cannot cryptographically prevent this act without requiring changes to the Miner's API, and it cannot reliably detect the Validator's trading activity outside the protocol.

Archived copy · cryp2nova.com

Instead, the Telegraph makes front-running economically irrational by eliminating the time window in which it could be profitable. This mechanism applies to both the live Signal path *(Section 5.1)* and the post-finalisation subscriber delivery.

Gossip-Time Enforcement

The zkTLS proof generated by the Leader (Section 5.4) contains a timestamped attestation of the exact moment at which the Miner response was received. When the Leader gossips the locked payload to the libp2p mesh, every receiving Validator records the local arrival time.

The protocol-defined tolerance for gossip delay is dynamically set to the 95th-percentile round-trip time of the active libp2p mesh over the preceding epoch, with a hard floor of 300 ms and a hard ceiling of 2 seconds. If the median arrival time across the mesh exceeds the zkTLS timestamp by more than this tolerance, the payload is rejected as stale. A rejected payload counts as a missed round for the Leader (Section 5.3, Category A).

This forces the Leader to broadcast the Miner response almost instantly after the TLS handshake completes – within a window that is too short for the Leader to submit a meaningful front-running trade and await its confirmation.

Clock Synchronisation

To prevent false rejections due to clock drift, all active Telegraph nodes must synchronise their system clocks to a standardised NTP server cluster maintained by the protocol. The tolerance window accounts for the maximal allowable NTP drift, ensuring that honest Validators are never penalised because of minor time discrepancies.

Post-Finalisation Delivery

After BFT finalisation, the Signal must be pushed to all active WebSocket subscribers and the public Dashboard (Section 8.5) without intentional delay.

Penalties for Deliberate Delay

A Leader or delivering Validator who deliberately withholds a payload to trade ahead of the Agent or subscribers is committing a protocol violation.

Gossip-latency violations are detected cryptographically and enforced automatically by the mesh (stale payload rejection, Category A missed round). Post-finalisation delivery delays, however, occur off-chain and cannot be verified programmatically. Claims of deliberate post-finalisation withholding must be submitted to the Validator Governance

Archived copy · cryp2nova.com

process (Section 10). During the bootstrap phase before governance activation (Section 10.1), claims of deliberate post-finalisation withholding are adjudicated by the Telegraph Operating Company as the designated interim authority, with all decisions recorded on-chain and subject to community review.

A permissionless vote by the active Validator set determines whether the evidence demonstrates a systemic attempt to exploit signal data. If the vote confirms the violation, the Protocol imposes the following penalties on the Validator:

- Forfeiture of all Validator emissions for the current epoch and the following 6 epochs (7 epochs total).
- Immediate Liveness-Ejection under Category A, regardless of the continuous-offline timer.
- All forfeited emissions are redistributed equally among the other active Validators in each affected epoch.

No stake is burned. The Validator may re-enter the active set after the forfeiture period and a new registration, but the offence is permanently recorded on-chain.

Economic Rationality

A Validator considering front-running must weigh a one-time gain from a trade executed against a sub-second Signal advantage against the complete loss of all emissions for a full 7-epoch period and immediate ejection from the active set. Because the gossip window is reduced to network-propagation speed and the penalty is catastrophic, the act is a guaranteed long-term loss for any rational economic actor.

This design eliminates the profitable front-running window without modifying the Miner's API, and without requiring the protocol to monitor external trading activity.

6. CONSENSUS AND FINALITY

6.1. Leader Election

For each active Intent iota in each epoch e, exactly one Validator is elected Leader. Election is deterministic and requires no communication

LeaderIndex = H(BlockHash_(e-1)

Concatenated with iota concatenated with e) modulo |V|. Where BlockHash_(e-1) is the finalized hash of the last block of the previous epoch. Every validator computes the identical result.

Archived copy · cryp2nova.com

6.2. Leader Failover

If the elected leader fails to broadcast a signed receipt within 90 seconds, the next validator in the deterministic sequence (wrapping around) is promoted. The original leader's round is considered missed, and they do not earn for that round.

6.3. BFT Finalization

The receipt must collect signatures from at least tau equals 43 of the 64 active validators to be considered Finalized. The threshold tau provides Byzantine Fault Tolerance for up to 21 malicious or offline validators.
Finalization proceeds as follows:

1. Leader broadcasts a signed receipt to the libp2p mesh.
2. Each Validator computes its Local_Score, then gossips a cryptographic commitment (hash of the score concatenated with a random nonce) to the mesh.
3. Once at least tau commitments are collected, each Validator reveals its plaintext Local_Score and nonce.
4. The Leader verifies each revealed score against its commitment and collects the valid ones.
5. If at least tau valid Local_Scores are revealed within 90 seconds, finalization proceeds.

The commit phase closes after 45 seconds or when tau commitments are received, whichever comes first. The reveal phase opens immediately and closes after a further 45 seconds. If at least tau valid Local_Scores are successfully revealed within this total 90-second window, finalization proceeds.

This commit-reveal sequence prevents a Validator from waiting to observe other scores before submitting its own. Any Validator that fails to reveal a score matching its commitment forfeits the round's emission share for that Intent (Section 5.3, Category C).

Score Aggregation: The elected Leader collects all signed Local_Scores and computes the canonical score using a Stake-Weighted Median.
Let S_total = sum of staked Machina across all responding Validators. Local_Scores are sorted in ascending order.
Canonical_Score = the Local_Score value at which cumulative Validator stake first exceeds (S_total / 2).

Archived copy · cryp2nova.com

Off-Chain BFT and BLS Aggregation

Telegraph decouples network consensus from L2 smart-contract execution entirely. The stake-weighted median, penalty calculations, and leaderboard updates are all performed off-chain inside the Validator node binaries.

Once the mesh reaches a tau consensus on an epoch's results, the validators use BLS (Boneh-Lynn-Shacham) signature aggregation.
The tau individual validator signatures are cryptographically compressed into a single Aggregate Signature. The off-chain payload – containing the Epoch_State_Root and the Aggregate_Signature – is handed to the Block Builder.

If fewer than tau signatures are obtained, the epoch fails for that Intent. The leader's round is considered missed, and they do not earn for that round. The next validator in sequence is promoted for a retry.

Bootstrap Exception: Until the Active Validator Set reaches 64 nodes, tau is dynamically defined as $\lceil 0.67 \times |V| \rceil$, with a minimum hard floor of tau = 1. The threshold locks to 43 once |V| reaches 64.

6.4. Network Partition Handling

If three consecutive rounds for the same Intent fail to reach tau signatures, the consensus threshold temporarily reduces to tau_reduced equals 33 (simple majority). Receipts finalized under the reduced threshold are permanently flagged. Reduced_Consensus equals true. Agents may independently decide whether to act on such signals. The threshold automatically restores to tau when a round successfully obtains 43 signatures.

6.5. Block Submission and Epoch Finality

At the end of every epoch, the finalised Signal Receipts for all active Intents are assembled into a single, verifiable epoch block and committed on-chain. The block is the canonical record of the epoch's work and triggers the distribution of all protocol rewards and penalties.

Block Structure

An epoch block contains, for each active Intent:

- Canonical_Score (the stake-weighted median),
- The ordered list of signed Local_Scores and their corresponding Validator identities,
- The WASM_Hash used by each signing Validator,

Archived copy · cryp2nova.com

- The hash of the finalised Signal Receipt (which itself contains the Ground_Truth, Miner_Response, and zkTLS_Proof), together with the epoch number, a merkle root of all Intent data, and the Block Builder's signature.
- For each Intent where Catch-Rate Promotion conditions were met during the epoch *(Section 4.3),* the Epoch_State_Root Merkle tree includes the promoted script's WASM_Binary_Hash as the new Canonical Script for that Intent, effective from the next epoch. All Validators compute the promotion deterministically from the same on-chain Testing Cohort scores and agree on the same state root without coordination.

The WASM_Hash declared by each signing Validator is included as a leaf in the Epoch_State_Root Merkle tree, enabling the Port Contract to verify Hash-Math distributions against the committed state without requiring individual receipts.

Replay and Deterministic Audit

Because WASM execution is strictly deterministic, the protocol provides a mathematically provable audit trail. The canonical block on Base stores only receipt hashes and Merkle roots to minimise gas costs; the full locked payloads (Ground_Truth, Miner_Response, zkTLS_Proof) are archived by the Validator mesh.

Any third party may independently audit the network by fetching a historical payload, verifying its hash against the on-chain record, and re-executing the declared WASM script to prove that a Validator's Local_Score was honest. Furthermore, any party can apply the stake-weighted median formula to the on-chain array of Local_Scores to reconstruct the Canonical_Score without trusting the Leader.

Precondition for Submission

A block may only be submitted if, for every Intent included, the assembled receipts collectively contain valid signatures from at least tau distinct active Validators. This condition is already met when the validation rounds themselves finalise (Section 6.3), the Block Builder merely packages the previously finalised data.

Block Builder Selection

Exactly one active Validator is designated as the Block Builder for epoch e. Selection is deterministic and communication-free:

$\text{BlockBuilderIndex} = H(\text{BlockHash}_{(e-1)} \parallel \text{"block"} \parallel e) \bmod |V|.$

If the Block Builder fails to submit a valid block within 120 seconds after the epoch boundary, the next Validator in the deterministic sequence is promoted. The original Block Builder's failure to submit is recorded as a missed round for purposes of the

Archived copy · cryp2nova.com

continuous-offline counter and the rolling participation threshold defined in Section 5.3 Category A.

The Settlement Layer (Dumb Registry)

The Port Contract deployed on Base does not execute validation logic, loop through score arrays, or verify individual BFT receipts. It operates strictly as a lightweight state registry and financial settlement layer.

The elected Block Builder submits a single transaction to the Port Contract containing:

1. The Epoch_State_Root: A Merkle root of all finalised scores, miner balances, and updated Canonical Scripts.
2. The Aggregate_Signature: The BLS cryptographic proof that at least tau validators reached consensus off-chain.

Verification and Execution

The Port Contract executes a single BLS signature verification against the known public keys of the active Validator set. If the signature is valid, the contract mathematically guarantees that a supermajority of the network agreed on the submitted state.

The contract instantly accepts the Epoch_State_Root as absolute truth. It atomically:

- Updates the Leaderboard Registry,
- Distributes Validator emissions equally among all active Validators that met the epoch's participation requirements,
- Distributes Script Author rewards via Hash-Math,
- Records Category C strikes for the entire epoch as a single group update,
- Releases the Block Builder's gas reimbursement from the Gas Subsidy Pool.

The gas cost for this submission is permanently flat, regardless of how many Intents or Miners the network scales to.

This mechanism guarantees that every epoch's verified intelligence is immutably recorded, economic rewards flow without manual intervention, and no single node's failure can stall the network.

7. ECONOMIC MODEL AND TOKENOMICS

7.1. Protocol Treasury

The Protocol Treasury is a reserve account on the canonical Base smart contract. It is capitalised by two immutable streams:

- 20 percent of daily Machina emissions *(Section 2.2)*.

Archived copy · cryp2nova.com

- 2 percent protocol fee on every x402 payment (deducted atomically before TWAP settlement).

The Treasury serves five defined purposes.

- First, a fixed percentage of the Treasury balance is allocated to a Gas Subsidy Pool that autonomously converts reserves into native Base gas to reimburse Validators for submission costs, ensuring node operation is capital-neutral.
- Second, the Treasury funds the Autonomous Engine's bootstrap budget during the pre-organic-demand phase.
- Third, Treasury funds protocol development, security audits, and infrastructure scaling.
- Fourth, Treasury funds the Miner Grant Programme – a demand-driven initiative that issues grants to Miners who build and supply high-demand inference capabilities identified via the Signal Agora, attracting technical talent to the protocol based on real market signals rather than speculative subsidies.
- Fifth, the Treasury funds epoch tournament query costs, reimbursing the Leader Validator for the minimum floor price per Miner queried during each epoch evaluation, as defined in Section 5.1.

The exact allocation percentage to the Gas Subsidy Pool is governed by the parameters in Section 10.2. All other Treasury expenditures are subject to governance approval per Section 10.

7.2. Miner Compensation via TWAP Settlement

Miners are paid exclusively from fulfilled inference requests. The
The flow is as follows:

1. Agent pays Signal_Price_iota USDC via x402, where Signal_Price_iota is determined per Section 7.5 based on the Miner's declared floor and the current demand multiplier for that Intent. At genesis, with minimal demand, this is 0.01 USDC at the protocol minimum floor. As demand grows, the price increases automatically.
2. The protocol deducts a 2 percent fee to the Treasury.
3. The remaining 98 percent of Signal_Price_iota is credited to the TWAP Settler escrow.
4. Over the 24-hour epoch, the TWAP Settler continuously drips the accumulated USDC into the Machina/USDC Uniswap V3 pool on Base.
5. The resulting Machina is distributed to the Miner who fulfilled the request.

Settlement Parameters:

Drip Period: 24 hours (86,400 seconds).
Drip Size: Total_Epoch_USDC divided by 86,400.

Archived copy · cryp2nova.com

Jitter: Plus or minus 30 seconds, derived from H(BlockHash concatenated with e concatenated with DripsCompleted) modulo 61 minus 30.
Settlement Floor: 100 USDC. Below this threshold, USDC rolls over to the next epoch until the floor is reached.
Drip Executor: Any active Validator may submit drip transactions. Gas costs are deducted from the accumulated USDC escrow before the miner payout calculation. The executing Validator's exact Base gas cost is atomically reimbursed from the Gas Subsidy Pool *(Section 7.1)*. This continuous, jittered drip eliminates large, predictable swaps and renders MEV extraction unprofitable.

7.3. Validator Emission Distribution

All active Validators receive an equal base share of the total Validator emission pool for the epoch. The pool is divided equally among the Validators that meet the epoch's participation requirements *(Section 5.1)*.

If a Validator misses a round due to a liveness failure, they simply do not earn for that round; the unearned amount is redistributed equally among all other active Validators in the same epoch. Similarly, if a Validator incurs a Category C penalty on a specific Intent, the forfeited round reward is redistributed to the other participating Validators.

If total USDC volume across all Intents equals zero in a given epoch (V_net = 0), the Validator emission for that epoch is transferred to the Protocol Treasury and held for future distribution – never burned.

This design ensures that Validators compete solely by staking enough Machina to remain in the top-N active set. There are no APY races, no revenue-weighting based on individual Intent performance, and no advantage from delegator volume. Every active Validator earns the same base income per epoch, rewarding consistent, honest participation.

7.4. Request Routing

Requests for a given Intent are routed to Miners
probabilistically based on their current leaderboard scores:

P(route to Miner_m) = Score_m / Sigma(Score_j for all j in pool)

New Miner Grace Period: A newly registered Miner has no score and would receive zero routing. To break this circular deadlock, a pool of exactly 5 percent of the Intent's request volume is reserved for all Miners that are within their 7-day grace period. This 5 percent pool is divided equally among all currently grace-qualified Miners for that Intent. After a

Archived copy · cryp2nova.com

Miner's grace period ends, the Miner enters the standard probabilistic weighted routing, and the pool shrinks proportionally for the remaining grace-qualified Miners. If no Miners are currently within their grace period for a given Intent, this 5 percent allocation reverts to the standard probabilistic weighted routing pool.

7.5. Signal Pricing Mechanism

Signal prices are not fixed. They are determined by a two-layer model combining a Miner-declared floor with a protocol-enforced demand multiplier. This creates a market-driven intelligence equilibrium where high-demand signals command higher prices organically, without any centralised price-setting authority.

The Signal Price for Intent iota in epoch e is calculated as:

- Signal_Price_iota = min_price_usdc_iota × Demand_Multiplier(V_24h_iota)
- Where min_price_usdc_iota is the Miner's registered floor price for that Intent, and V_24h_iota is the rolling 24-hour request volume for that Intent, computed by the Node API aggregating both on-chain settlements and off-chain request logs identically to the Signal Agora data source. Price recalculates at the start of each epoch.

The Demand Multiplier schedule is as follows:

V_24h range	**Multiplier**
0 – 999 requests	1.0x (floor price, no uplift)
1,000 – 9,999 requests	1.5x
10,000 – 99,999 requests	2.5x
100,000 – 999,999 requests	5.0x
1,000,000+ requests	10.0x

The protocol minimum floor at genesis is 0.01 USDC. No Miner may register a floor price below this value. A Miner may declare any floor at or above the protocol minimum. The floor is immutable per registration and may only be changed by submitting a new registration transaction. During the 7-day Miner Grace Period, Signal_Price_iota is calculated identically using the Miner's declared floor and current demand multiplier.

The protocol fee of 2 percent is applied to Signal_Price_iota. The TWAP Settler receives 98 percent of Signal_Price_iota. All references to 0.01 USDC throughout this document refer to the protocol minimum floor price at genesis. As demand grows, actual prices paid by Agents will exceed this floor automatically.

Archived copy · cryp2nova.com

The Demand Multiplier brackets are governable via Section 10. The protocol minimum floor price is governable via Section 10. Both require a 43 of 64 supermajority vote and a 14-day timelock.

8. PAYMENT PIPELINE AND X402 INTEGRATION

8.1. Canonical Chain

All payments and smart contracts reside on Base. The following contracts are deployed:

- **Port Contract** Handles x402 payment verification and escrow.
- **TWAP Settle** Manages USDC accumulation and DEX drips.
- **Treasury** Receives protocol fees and slashed funds.
- **Leaderboard Registry** Stores finalised scores and Miner rankings.

8.2. Web3 Agent Payment Flow

- Agent sends inference request to Telegraph endpoint.
- Telegraph returns HTTP 402 Payment Required with: Amount Signal_Price_iota USDC (floor 0.01 USDC at genesis, demand-adjusted per Section 7.5), Recipient Port Contract address, Network Base, Deadline 30 seconds.
- The agent's wallet signs and broadcasts the USDC transaction.
- Telegraph verifies settlement on Base.
- Request is routed to a Miner via probabilistic weighted routing.
- Miner returns response.
- Validator generates signed Signal Receipt.
- The agent receives a response and receipt.

8.2.1. Pre-Funded Agent Escrow (Funding Wallet)

To simplify the payment flow and eliminate the need for per-request transaction signing, an Agent may deposit USDC into a pre-funded escrow account managed by the Port Contract. This escrow acts as a single Funding Wallet that covers all costs associated with a signal request:

1. The protocol deducts Signal_Price_iota USDC from the escrow for the miner's compensation (Section 7.2).
2. An additional gas margin of a fixed reserve is held to reimburse the executing Validator for on-chain submission costs. This reserve is calculated as Estimated_Submission_Gas multiplied by 2.0 using the current Base gas price, updated at each epoch boundary.

The purchase of Machina for the miner and the Validator's gas reimbursement are executed atomically as part of the same settlement logic, without requiring a

Archived copy · cryp2nova.com

separate transaction from the Agent. Agents may top up the escrow at any time and withdraw any remaining balance when no requests are in flight.

This mechanism provides the same Web2-like experience described in Section 8.3: the Agent holds USDC, and the protocol handles all conversion and settlement invisibly.

8.3. Web2 Developer Onboarding

Telegraph provides a three-layer stack requiring zero blockchain knowledge:

1. **Fiat Onramp:** Developer funds via Stripe fiat-to-USDC onramp; USDC deposited into an MPC wallet on Base.
2. **MPC Wallet:** Managed by Coinbase AgentKit; the developer manages no private keys.
3. **x402 Payment:** MPC wallet autonomously detects HTTP 402 challenge, signs the Signal_Price_iota transaction (minimum 0.01 USDC at genesis), and broadcasts it. The developer application receives inference response and cryptographic receipt.

This provides a pure Web2 developer experience on Web3 settlement rails.

8.4. On-Chain Callbacks - Gas Escrow

For permissionless applications (dApps) requiring native on-chain delivery of finalized Signals, the protocol implements a Funding Wallet mechanism to permanently insulate Validators from Base network gas volatility. A requesting dApp must pre-fund a dedicated Funding Wallet attached to the Port Contract with native Base gas tokens, alongside the standard USDC inference fee. When an on-chain inference request is initiated, the smart contract verifies that the Funding Wallet balance meets a minimum safety threshold calculated as follows: Minimum_Escrow = Estimated_Callback_Gas multiplied by Surge_Multiplier, where Surge_Multiplier = max(2.0, Current_Gas_Price divided by 30_day_avg_Gas_Price), with a floor of 2.0 and a ceiling of 5.0. The 30-day average gas price is computed by the Port Contract from an on-chain rolling average updated every epoch.

Upon BFT finalization, the elected leader Validator executes the on-chain callback transaction to deliver the Signal directly to the dApp's smart contract. The Port Contract autonomously deducts the exact transaction gas fee from the dApp's Funding Wallet to reimburse the executing Validator. If network gas prices spike beyond the wallet's balance prior to execution, the callback is deferred until fees normalize or the Funding Wallet is replenished. This architecture ensures that Validator capital at risk for third-party callback delivery is strictly zero. This Funding Wallet mechanism is utilized exclusively for the on-chain inference pipeline and is entirely distinct from the Protocol Treasury's gas subsidies utilized for baseline block submissions.

Archived copy · cryp2nova.com

8.5. WebSocket Pub/Sub Interface

The Telegraph node software provides a real-time WebSocket interface to support high-frequency automated workflows and enterprise telemetry. Agents and permissionless applications establish a persistent connection by declaring a JSON filter payload specifying Intent category, minimum confidence threshold, and status criteria. The connection itself is feeless.

Signal delivery occurs strictly after BFT finalisation. When a finalised Signal matches an active WebSocket subscription filter, the submitting Validator pushes the Signal to the subscriber over the persistent connection. The Validator appends a signed Delivery Log Entry containing: Agent_Address, Intent_ID, Finalised_Receipt_Hash, Epoch_Number, Timestamp, and Node_Signature.

At each epoch boundary, the Validator submits the compressed delivery log as a single batch settlement transaction to the Port Contract. The contract deducts Signal_Price_iota USDC from each Agent's pre-funded escrow per recorded delivery for the corresponding Intent, and credits the 98 percent Miner allocation to the TWAP Settler and the 2 percent protocol fee to the Treasury. The deduction per delivery reflects the Signal_Price_iota at the time of delivery as recorded in the signed Delivery Log Entry. Agents must maintain a minimum pre-funded escrow sufficient to cover at least 100 signal deliveries at the current Signal_Price_iota for their subscribed Intents, with a hard floor of 1.00 USDC regardless of price. If an Agent's escrow is insufficient to cover outstanding deliveries at settlement, the debt is recorded on-chain and WebSocket delivery is suspended until the Agent replenishes the escrow and clears the outstanding debt. No match implies no delivery and no payment.

8.6. ERC-8183 Agentic Commerce Integration

To serve as the foundational infrastructure for the autonomous economy, Telegraph operates as a fully compliant Provider under the ERC-8183 standard for trustless AI agent commerce.

While the x402 pipeline (Section 8.2) facilitates synchronous, micro-transactional inference, the ERC-8183 integration allows autonomous agents and dApps to programmatically hire the Telegraph network for asynchronous, escrow-backed execution without requiring custom integration logic.

The Port Contract natively implements the ERC-8183 AgenticCommerce interface, mapping Telegraph's validation mesh to a deterministic on-chain job state machine:

Archived copy · cryp2nova.com

1. Open & Funded (Job Creation): An Agent initiates an on-chain job specifying the desired Intent_ID, the required YAML input parameters, and escrows the total USDC budget. The job enters the Funded state.
2. Execution: The Telegraph Autonomous Engine detects the funded job. The elected Leader routes the request to the Miner mesh and executes the Canonical Script via the standard BFT pipeline (Section 5.1).
3. Submitted (Proof Delivery): Upon off-chain finalisation, the Block Builder's submission of the Epoch_State_Root and the Aggregate_Signature to Base (Section 6.5) acts as the cryptographic proof of execution. The Signal is delivered, and the job state transitions to Submitted.
4. Terminal (Payment Release): Because the protocol's BLS signature verification mathematically guarantees the accuracy of the intelligence, no secondary human or oracle evaluator is required. The successful block settlement automatically advances the ERC-8183 job to Terminal, releasing the escrowed USDC directly to the TWAP Settler and the Protocol Treasury according to the standard fee split.

This integration allows Agents to specify complex job parameters using the standard onchain_output YAML structure (Section 4.1), ensuring that downstream contracts can natively parse the finalized intelligence directly from the escrow release execution.

9. BOOTSTRAPPING & AUTONOMOUS ENGINE

9.1. Two-Phase Launch

Phase 1 - Testnet (Base Sepolia): The Telegraph Operating Company operates Node Zero (first Validator) and Agent Zero (first signal consumer) using testnet USDC. The purpose is to validate the complete protocol mechanics under sustained operation. No real capital is at risk.

Phase 2 - Mainnet (Base): The Telegraph Operating Company continues to operate Node Zero and Agent Zero as regular participants, earning Machina and paying USDC identically to any third party. No special privileges exist. To capitalise the Protocol Treasury and ensure the Autonomous Engine can execute Discovery Tier validation rounds before organic subscriber demand arrives, the Telegraph Operating Company deposits a one-time bootstrap amount of [X] USDC into the Port Contract at mainnet genesis, credited to the Protocol Treasury. This deposit is a non-reimbursable protocol seed; it creates no debt, token allocation, or governance privilege. The Treasury and TWAP Settler are seeded organically by the 2 percent protocol fee on Company queries.

9.2. Genesis Intent Library and Foundation Miners

At mainnet genesis, the circulating supply of Machina is zero, rendering the standard bond requirements temporarily unpayable. To solve this cold-start deadlock and provide

Archived copy · cryp2nova.com

immediate utility, the protocol includes a Genesis Intent Library of high-demand Intents registered directly in the genesis block state. These Intents and their corresponding Genesis Validation Scripts are registered directly in the genesis block state, bypassing the standard 10,000 Machina script bond requirement at the moment of genesis registration only. Subsequent updates, challenges, or re-registrations of these Intents follow the standard bond and registration process defined in Section 4. The Genesis Intent Library is determined exclusively by the Telegraph Operating Company and published in full at least 30 days prior to mainnet genesis for public review and community comment. The published list is final and immutable at genesis block zero. The Genesis Intent Library constitutes the entirety of the Company's curatorial authority over the protocol. All Intent registrations after genesis block zero are fully permissionless and require no Company involvement. The Telegraph Operating Company operates Foundation Miners for these Genesis Intents. These Foundation Miners are publicly labelled as Telegraph-operated on the Signal Agora. As external Miners register and achieve higher leaderboard scores, routing shifts organically via the probabilistic mechanism defined in Section 7.4; no manual cutover or centralised intervention is required.

9.3. Intent Bounties & The Autonomous Engine (AE)

The Autonomous Engine is the continuous execution of the validation loop defined in Section 5.1. It natively powers the live Intelligence Terminal and WebSocket feeds without simulated data or centralized bots.

To solve the "Cold Start" problem for newly registered Intents that possess zero organic Agent demand, developers or third parties may deposit USDC Intent Bounties directly into the Port Contract for a specific Intent. This bounty acts as synthetic demand, mathematically forcing the Intent into the Discovery Tier of the active Validator set. The Autonomous Engine immediately begins scraping, querying Miners, and grading responses to drain the bounty. This populates the public leaderboard with accurate Miner rankings before organic Agents arrive, bootstrapping the network flywheel permissionlessly.

9.4. Signal Agora

The Signal Agora is a public view that aggregates two data sources per Intent: on-chain Leaderboard Registry and settlement contract events, and off-chain Node API request logs including Intelligence Terminal queries, dashboard requests, and direct API calls. The Node API serves as the unified query endpoint. For each Intent the Signal Agora displays:

- Current top Miner score.
- Active Miner count.
- Total request volume (24h) (on-chain + off-chain combined).
- USDC volume settled (24h, 7d) (on-chain only).

Archived copy · cryp2nova.com

- Foundation Miner flag.
- Estimated daily earnings for the top Miner.
- This transparency serves as a Public Signal Agora, allowing independent developers to identify revenue-generating Intents and compete to displace Foundation Miners through superior performance.

9.5. Bootstrap Transition

The Company's Agent Zero query volume is automatically reduced when the following on-chain conditions are simultaneously satisfied:

- At least 10 distinct non-Company Agents have submitted valid inference requests in the preceding 7-day period.
- Organic USDC volume (7-day trailing) exceeds Company USDC volume (7-day trailing).
- Upon satisfaction of both conditions, Company query volume decreases by 10 percent per week until reaching zero. The Company thereafter participates solely as Node Zero validator. All conditions and reductions are verified and executed deterministically by the canonical smart contract.

10. GOVERNANCE

10.1. Activation Threshold

Governance is activated only when the active validator set reaches 43 validators. Until then, all parameters are frozen at genesis values. This prevents hostile governance during bootstrap.

10.2. Governable Parameters

The following parameters may be modified via on-chain governance proposals:
Validator cap N: Required Votes 43 of 64, Timelock 30 days
Protocol fee percent: Required Votes 43 of 64, Timelock 14 days
Penalty thresholds: Required Votes 43 of 64, Timelock 14 days
Minimum script bond: Required Votes 43 of 64, Timelock 14 days
Minimum intent bond: Required Votes 43 of 64, Timelock 14 days
Unbonding period: Required Votes 43 of 64, Timelock 30 days
BFT threshold tau: Required Votes 43 of 64, Timelock 30 days
Epoch duration: Required Votes 43 of 64, Timelock 30 days
Gas Subsidy Pool allocation percent: Required Votes 43 of 64, Timelock 14 days
Demand multiplier brackets: Required Votes 43 of 64, Timelock 14 days
Protocol minimum signal floor price: Required Votes 43 of 64, Timelock 14 days
Miner routing distribution weights: Required Votes 43 of 64, Timelock 14 days

Archived copy · cryp2nova.com

Permanently Immutable:
Maximum supply S_max equals 21,000,000 Machina.
Halving schedule (every 4 years).
Emission tier percentages (60 percent / 20 percent / 20 percent).
Zero pre-mine guarantee.

10.3. Proposal Process

- **Submission:** Any active Validator submits an on-chain proposal with target parameter, current value, proposed value, and an IPFS-linked rationale hash.
- **Discussion:** 7-day period for public community discussion; no votes counted.
- **Voting:** 7-day period; only active Validators may vote. Each Validator has one vote, independent of stake size. Passage requires at least 43 affirmative votes.
- **Execution:** If passed, the parameter-specific timelock begins. At expiry, the parameter updates atomically at the next epoch boundary.
- **Emergency Veto:** 54 of 64 validator signatures may cancel a proposal immediately. A vetoed proposal is subject to a 90-day cooldown before resubmission.

11. SECURITY MODEL

11.1. Sybil Resistance

Sybil resistance is purely economic, identical to Bitcoin and Bittensor. The cost to control a supermajority of the validator set is the market cost of acquiring the necessary Machina. Because the fixed supply of 21,000,000 Machina is subject to continuous TWAP buy pressure from organic usage, any large-scale acquisition drives up the token price during the acquisition itself, making the attack progressively more expensive.

11.2. MEV Resistance

The TWAP settler eliminates large, predictable swaps. With per-drip USDC value below Base gas costs and jittered timing, front-running is unprofitable by design.

11.3. Anti-Spam Economics

Fake Work / Self-Mining: An attacker spinning up a fake Miner to self-purchase signals must pay Signal_Price_iota per request, which is at minimum 0.01 USDC and increases automatically as the attacker's own fake demand drives up the multiplier bracket for that Intent. The protocol captures 2 percent of every payment as a fee. The attacker mathematically bleeds value at an accelerating rate as their attack volume grows, since

CRYP2NOVA

Archived copy · cryp2nova.com

their own requests escalate the demand multiplier and therefore the price they must pay per request. Self-purchase attacks become exponentially more expensive the larger they are.

Free Money / Ghost Town: Validators who go offline or fail to sign assigned rounds earn nothing for those rounds, with forfeited emissions redistributed to participating Validators. The Stratified Portfolio mandate (Section 5.1) ensures Validators automatically allocate 90 percent of compute to the highest demand Intents, aligning their participation with real market activity without requiring manual selection or exposing the network to fake-work loops.

12. GENESIS PARAMETERS

- S_max: 21,000,000 Machina
- E_0: 7,200 Machina per epoch
- Halving interval: 4 years
- N_genesis: 64
- tau (BFT threshold): 43 of 64
- tau_reduced: 33 of 64
- Unbonding period: 21 days
- Liveness ejection threshold (continuous): 7 days unbroken
- Liveness ejection threshold (rolling): 50 percent missed rounds in 7 days
- Consensus deviation penalty: emission forfeiture (redistributed)
- Fraud slash: 20 percent (transferred to Protocol Treasury; no burn)
- Failover timeout: 90 seconds
- Minimum script bond (Anti-Spam): 10,000 Machina
- Minimum intent bond (genesis): 100 Machina
- Script Evolution Mechanism: Autonomous Catch-Rate Promotion
- WASM update timelock: None (Immediate)
- Consensus deviation threshold delta_c: 0.15 scoring units
- Consensus deviation strike window: 5 strikes per 30 days
- Protocol fee: 2 percent
- TWAP settlement floor: 100 USDC
- Drip gas rebate: Exact Base gas cost reimbursed from Gas Subsidy Pool
- New Miner grace period: 7 days
- Grace routing share: 5 percent flat
- Emission split: 60 percent Validators / 20 percent Script Author / 20 percent Treasury
- Script Author distribution mechanism: Hash-Math proportional distribution
- Governance activation threshold: 43 active validators
- Governance discussion period: 7 days
- Governance voting period: 7 days
- Emergency veto threshold: 54 of 64

Archived copy · cryp2nova.com

- Proposal rejection cooldown: 30 days
- Veto cooldown: 90 days
- Protocol fee timelock: 14 days
- Cap change timelock: 30 days
- Bootstrap trigger - organic Agents: at least 10
- Bootstrap trigger - volume: Organic 7d greater than Company 7d
- Bootstrap reduction rate: 10 percent per week
- Gas Subsidy Pool allocation (genesis): 10 percent of Treasury balance
- Gas escrow minimum surge multiplier: 2.0
- Gas escrow maximum surge multiplier: 5.0
- WebSocket minimum subscription escrow: 1.00 USDC
- Intent_ID derivation: H(Registrant_Address || YAML_Schema_Hash || Block_Number)
- Genesis library pre-publication period: 30 days before mainnet genesis
- Validator portfolio mandate: 90 percent Volume Tier / 10 percent Discovery Tier
- Gas Subsidy Pool timelock: 14 days
- Protocol minimum signal floor price: 0.01 USDC
- Demand multiplier bracket 1 (0-999 req/24h): 1.0x
- Demand multiplier bracket 2 (1,000-9,999 req/24h): 1.5x
- Demand multiplier bracket 3 (10,000-99,999 req/24h): 2.5x
- Demand multiplier bracket 4 (100,000-999,999 req/24h): 5.0x
- Demand multiplier bracket 5 (1,000,000+ req/24h): 10.0x
- Demand multiplier timelock: 14 days
- Discovery Tier participation threshold: 95 percent of assigned rounds
- Signal floor price timelock: 14 days
- WebSocket minimum escrow floor: 1.00 USDC
- Canonical chain: Base
- Miner routing top-3 distribution (genesis): 70% / 20% / 10% of the non-grace routing pool.
- Miner routing weight timelock: 14 days
- Testing Cohort size: 10 percent of active set
- Catch-Rate threshold delta_promote: 0.10 scoring units
- Catch-Rate minimum test epochs T_promote: 3 epochs
- Internal Audit Fee: 2 percent of epoch Validator emission pool
- Spot check frequency parameter (genesis): 10
- Front-running forfeiture window: 7 epochs
- Gossip delay tolerance floor: 300 milliseconds
- Gossip delay tolerance ceiling: 2 seconds
- Routing revocation score drop threshold: 20 percent relative
- Epoch duration (genesis): 24 hours (86,400 seconds)

Archived copy · cryp2nova.com