$NES Utility

PayForQuery Transactions

To publish data on Nesa developers can submit PayForQuery transactions. A PayForQuery transaction consists of the evolving request, the evolving size, the identity of the sender of the data to be made available for, the namespace, and a signature. Each PayForQuery transaction is split into two parts: the evolution which includes the data to be made available along with the namespace, and the executable payment transaction which includes a commitment to the data.

Both the evolution and executable payment transactions are put into the block within the appropriate namespace. The block data is extended using erasure coding and then Merkelized into a data root commitment included in the block header.

Fee Market Overview

Nesa uses a standard gas-price prioritized mempool. This means that transactions with higher fees will be prioritized by validators. Fees are comprised of a flat fee per transaction and then a variable fee based on the size of each evolve in the transaction.

Network Parameters

$NES holders - not just stakers - can propose and vote on governance proposals to change a subset of network parameters. Nesa will publicly list both the changeable and non-changeable parameters and their values of the system to vote on.

Community Pool

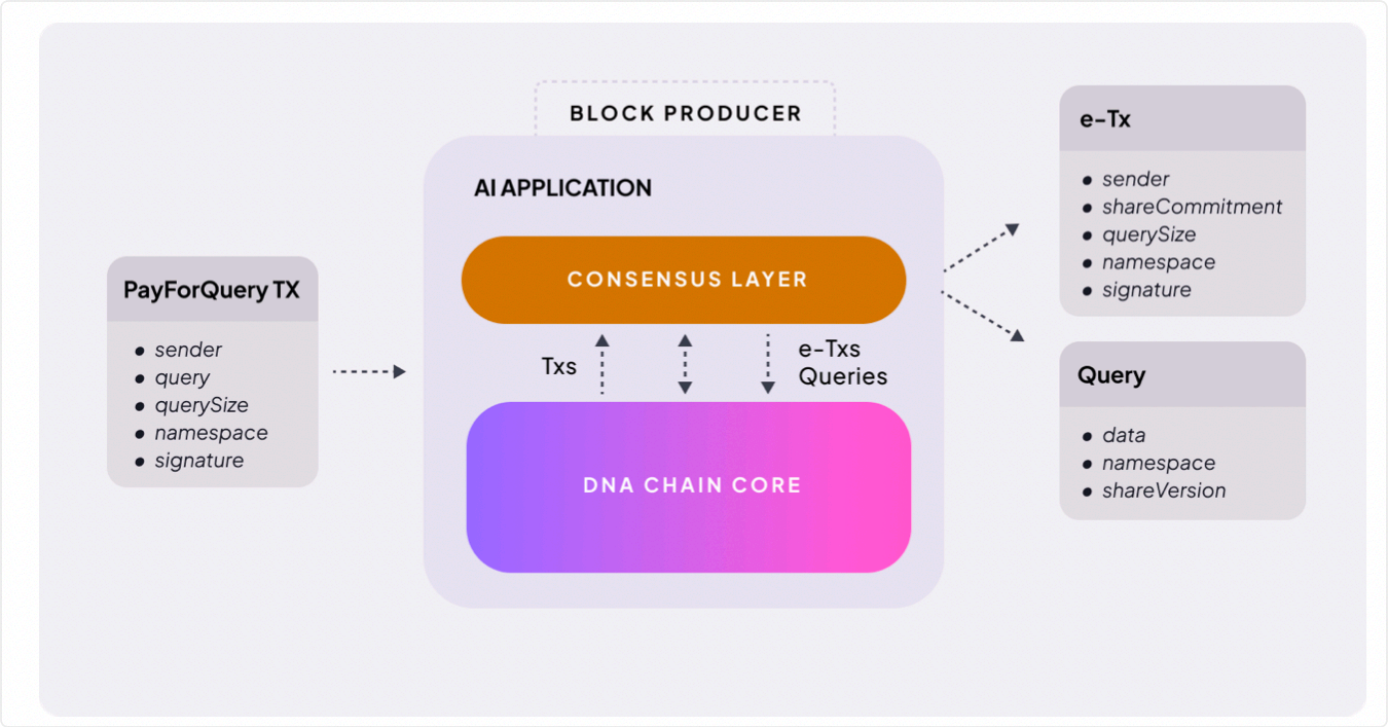
Starting at Genesis, Nesa's community pool receives 2% of all $NES Layer block rewards. $NES stakers may vote to fund ecosystem initiatives.

---

queryStream

queryStream interfaces with Nesa's crypto-economic framework backed by its Proof-of-Stake consensus protocol. By utilizing Nesa's Consensus Layer, computations for AI model inference are transparent and subject to scrutiny by the network of validators and their delegators through an efficient attestation mechanism.

In scenarios where dishonest activities are suspected, such as data withholding by validators, the system is designed to recognize any malfeasance if 2/3 of Nesa validators are implicated, prompting actions such as slashing to maintain network integrity. This modular verification layer provides strong crypto-economic security



Block production for a PayForQuery transaction. When initiated, a namespace and query details are established, and the transaction is signed. It is then passed through the system as it is shared with the Consensus Layer for verification and NESA Core where it is prepped for settlement.

assurances, raising the standard for trust in decentralized AI model evolution.

queryStream enables developers to tap into Nesa's scalable name space-specific storage, which may start at a few hundred kb per inference query on the Mainnet Beta, with potential increases manageable through on-chain governance. The actual storage provision is based on the complexity and size of the query and underlying model and supports the potential for tens of thousands of model parameters to be processed per second due to the efficient use of erasure coding and Merkle tree data structures.

The communication between Nesa's verification layer and its settlement layer (or Ethereum's settlement layer given that Nesa is interoperable) is facilitated by a peer-to-peer (P2P) network which includes a queryStream Relayer. The Relayer streams the query data from the verification layer to the settlement layer (or to the Ethereum network), where dedicated smart contracts formally encode and execute the query directives received from Nesa.

Every PayForQuery transaction submitted to Nesa is validated with a Byzantine Fault Tolerant (BFT) signature mechanism. Any AI model data included in Nesa can be independently verified on Ethereum using queryStream, with validation results backed by cryptographic proofs. On settlement, the AI model inference proofs and PayForQuery transaction data are published to Nesa.

For ZK rollups employing queryStream, data inclusion must be verified before considering any proof as valid. This can be done by incorporating the inclusion proof within the ZK proof itself for submission to Nesa's settlement layer, or another blockchain network like Ethereum.

Token Fee Mechanism

We summarize these high-level insights into the tokenomics of the platform which describes the token-based relationship between participants in the NES ecosystem. On Nesa, miners are required to stake a portion of $NES tokens. This requirement serves as a safeguard, ensuring that miners have a vested interest in the honest and efficient processing of AI queries.

This aspect of our system draws parallels to the concept of oracles in blockchain, yet with a unique focus on AI model execution.

Nesa's token fee mechanism introduces a flexible economic model: the more tokens committed, the larger the pool of miners, hence enhancing security with more proofs attached in the queryStream. Conversely, spending fewer tokens results in a smaller miner pool, indicating a trade-off between cost and security.

Previous
Overview of $NES

Next
The First Application on Nesa: DNA X

Last updated 1 month ago